Imperial College of Science, Technology and Medicine
Department of Physics
Astrophysics Group

Automated methods to find high-redshift quasars

Lena Lenz

Submitted in fulfilment of the requirements for the
degree of Doctor of Philosophy in Physics of
Imperial College London, December 2023

Statement of originality

With this statement I, Lena Lenz, declare that this PhD thesis is my own work and that, to the best of my knowledge, it contains no sources or resources other than the ones appropriately referenced and acknowledged.

Lena Lenz

London, December 16, 2023

Publications

L. Lenz, D. J. Mortlock, B. Leistedt, S. J. Warren, R. Barnett and P. C. Hewett. An automated method for finding the most distant quasars. Planned MNRAS submission.

© The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-Non Commercial 4.0 International Licence (CC BY-NC).

Under this licence, you may copy and redistribute the material in any medium or format. You may also create and distribute modified versions of the work. This is on the condition that: you credit the author and do not use it, or any derivative works, for a commercial purpose.

When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Where a work has been adapted, you should indicate that the work has been changed and describe those changes.

Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

‘Twinkle, twinkle, quasi-star
Biggest puzzle from afar
How unlike the other ones
Brighter than a billion suns
Twinkle, twinkle, quasi-star
How I wonder what you are.’

George Gamow, 1964

Acknowledgements

Thank you Daniel Mortlock for being a great supervisor and for always giving me detailed feedback.

Thank you Boris Leistedt for always giving me valuable advice.

Thank you Stewart Boogert for believing in me and pushing me to my best.

Thank you Johannes ‘sage mir, sage mir, sage mir’ Barton for being a great physics teacher.

Thank you to my mum and dad for always cheering for me and supporting me.

Thank you to my brother Gregor for always believing that I could achieve anything I set my mind to.

Thank you Nick, who is happy that I am finally contributing to society as a tax-paying citizen.

And finally, thank you to my dog Larry, who was ready for a break of walkies and cuddles at all times. I will forever miss you.

Dedication

For my mum and dad, Vera and Gerhard.

Abstract

Quasars are among the most luminous objects in the Universe and as such can be found to have existed within the first 700 million years after the Big Bang. This makes them useful for analyzing the epoch of reionisation and the black hole mass evolution. They can be identified by a distinct break in their spectral energy distribution (SED) around Lyα, which shifts into the near-infrared at $z \gtrsim 6.5$. However, due to their rarity and large contaminating populations, only a handful of quasars at $z > 7$ have been found in total so far. Current methods contain visual inspection of images from photometric surveys as part of the classification process. Manually examining a significant part of the candidates becomes infeasible for upcoming surveys such as the Large Synoptic Survey Telescope (LSST) and Euclid, which are expected to produce $\sim 10^{11} - 10^{12}$ bytes of data each night. In this work, we reduce the amount of required visual inspection by testing candidates against a model of a stationary point source that follows a quasar SED at $5.2 \lesssim z \lesssim 12$. Bayesian model comparison based on colour information is used to retrieve a quasar probability and the best fit SED. To automate this process, we implement a data processing pipeline that extracts relevant information from survey databases and photometric images. This information is then used to apply heuristic cuts in order to obtain high-redshift quasar candidates. To optimise the classification, the collected information is passed to decision tree and random forest machine learning classifiers. We also feed the raw image data to a convolutional neural network to compare performance. Our pipeline is implemented for the Sloan Digital Sky Survey (SDSS) and the United Kingdom Infrared Telescope Infrared Deep Sky Survey (UKIDSS), and can easily be extended to future surveys.

Acronyms

ADU Analog-to-digital unit

AGN Active galactic nucleus

ANN Artificial neural network

BBB Big blue bump

BL Lac BL Lacertae object

BLR Broad line region

BLRG Broad line radio galaxy

CAS Catalog Archive Server

CNN Convolutional neural network

DECaLS The Dark Energy Camera Legacy Survey

DT Decision tree

ETG Early type galaxy

FN False negatives

FP False positives

FPR False positive rate

FR-I Fanaroff-Riley Class 1

FR-II Fanaroff-Riley Class 2

FWHM Full width half maximum

GMM Gaussian mixture modelling

HSC-SSP Hyper Suprime-Cam Subaru Strategic Program

HWHM Half width half maximum

IGM Intergalactic medium

IR Infrared

Jy Jansky

KDE Kernel density estimation

MAP Maximum a posteriori

MCMC Markov chain Monte Carlo

MDI Mean decrease impurity

ML Machine learning

ML Maximum likelihood

NIR Near-infrared

NLL Negative log-likelihood

NLR Narrow line region

NLRG Narrow line radio galaxy

nmg nanomaggy

OVV Optically violent variable

Pan-STARRS The Panoramic Survey Telescope and Rapid Response System

PCA Principal component analysis

PSF Point spread function

QLF Quasar luminosity function

RBF Radial basis function

ReLU Rectified Linear Unit

RF Random forest

ROC AUC Area under the receiver operating characteristic curve

SAS Science Archive Server

SDSS Sloan Digital Sky Survey

SED Spectral energy distribution

SFG Star forming galaxy

SMBH Supermassive black hole

SSC Synchrotron self-Compton

SVM Support Vector Machine

TP True positives

TPR True positive rate

UKIDSS United Kingdom Infrared Telescope Infrared Deep Sky Survey

UKIDSS LAS United Kingdom Infrared Telescope Infrared Deep Sky Survey Large Area Survey

UV Ultraviolet

WFCAM Wide-field camera

WISE Wide-field Infrared Survey Explorer

WSA WFCAM Science Archive

Contents

Acknowledgements **vii**

Abstract **x**

Acronyms **xi**

1 Active galactic nuclei **1**

1.1 Subclasses of active galactic nuclei 3

1.1.1 Quasars 5

1.1.2 Seyfert galaxies 5

1.1.3 Radio galaxies 5

1.1.4 Blazars 5

1.2 Observational properties 6

1.2.1 Continuum and emission lines 6

1.2.2 Polarization and variability 6

1.3 Physical properties 7

1.3.1 Accretion disk 7

1.3.2 Thermal emission 7

1.3.3 High-energy emission 9

2 Quasars **11**

2.1 Spectral energy distribution 12

2.1.1 Big blue bump 13

2.1.2 X-ray 14

2.1.3 IR bump 15

2.1.4 Radio continuum 15

2.1.5 Broad line region 16

2.1.6 Narrow line region 16

2.2 Absorption lines due to intervening matter 16

2.2.1 Lyman-alpha absorption 17

2.2.2 Scattering cross section of neutral hydrogen 19

2.3 Quasars in the early universe 20

2.3.1 Cosmological probes 21

2.3.2 Quasar luminosity function 26

2.4 High redshift quasar surveys 28

2.4.1 Discovered quasars 28

2.4.2 Upcoming surveys 29

3 Finding high-redshift quasars **31**

3.1 Distinguishing features of high-redshift quasars from contaminants 32

3.1.1 Dark in the optical and red in NIR 32

3.1.2 Point-like objects 36

3.1.3 Stationary sources 37

3.1.4 Radio source 40

3.1.5 Variability 40

3.2 Data analysis mechanisms on selected sources 40

3.2.1 Colour cuts 41
3.2.2 Bayesian model comparison 42
3.2.3 Machine learning 43
3.3 Our approach 43

4 Bayesian inference **46**

4.1 Bayes' theorem 47
4.1.1 Intuition 47
4.1.2 Bayes' rule 48
4.1.3 Bayes' rule for i.i.d. events 51
4.1.4 Bayesian inference 52
4.1.5 Bayes factor 53
4.1.6 Bayesian model comparison 53
4.2 Examples 53
4.2.1 The likelihood in a coin flip 53
4.2.2 The marginal in a coin flip 55
4.2.3 The posterior probability of having Covid if tested positive 56
4.3 Prior 57
4.3.1 Subjective prior 57
4.3.2 Non-informative prior 59
4.3.3 Informative prior 62
4.3.4 Conjugate prior 62
4.4 Point estimates 62
4.4.1 Maximum Likelihood 63
4.4.2 Maximum a posteriori (MAP) 63
4.4.3 Chi-squared: Maximum likelihood using a Gaussian likelihood 64

5 Machine learning **65**

5.1 Overview 66

5.1.1 Machine learning is a subset of artificial intelligence 66

5.1.2 Building blocks of machine learning 69

5.1.3 Supervised learning 69

5.1.4 Classification 70

5.2 Machine learning vs Bayesian statistics 71

5.2.1 Machine learning: Parameter point estimate 71

5.2.2 Bayesian statistics: Parameter distribution 72

5.3 The difficulty of generalisation 73

5.3.1 Underfitting and overfitting 73

5.3.2 Bias and variance 75

5.4 Decision Trees 76

5.4.1 Leaf, node, root 76

5.4.2 Function representations 77

5.4.3 Node splitting functions 80

5.4.4 Advantages and disadvantages of decision trees 84

5.5 Random forests 84

5.5.1 Bootstrap aggregating 86

5.5.2 Feature bagging 87

5.5.3 Random forest feature importance measure 87

5.5.4 Advantages and disadvantages of random forests 89

5.6 Artificial neural networks 89

5.6.1 Forward propagation 90

5.6.2 Backpropagation 91

5.6.3 Activation functions of hidden layers 91
5.6.4 Output layer 95
5.6.5 Advantages and disadvantages of artificial neural networks 96
5.7 Convolutional neural network 97
5.7.1 Convolutional layer 98
5.7.2 Pooling layer 102
5.7.3 Fully connected layer 103
5.7.4 Prevention of overfitting 104
5.7.5 Advantages and disadvantages of CNNs 106
5.8 Difficulties of ML 106
5.8.1 A few scientific discoveries 107
5.8.2 Correlation vs causation 107
5.8.3 The importance of the training set 108

6 Data processing pipeline **110**

6.1 Step-by-step pipeline sequence 110
6.1.1 Downloading all available images 112
6.1.2 Cropping an image to a postage stamp 114
6.1.3 Plotting postage stamps 115
6.1.4 Estimate the background flux 118
6.1.5 Calculate the PSF flux of sources in a postage stamp 119
6.1.6 Calculate the quasar probability and find the best fit quasar SED 122
6.1.7 Reconstruct postage stamps 124
6.1.8 Calculate goodness of fit 124
6.1.9 Summarise candidate 127
6.2 Compatible surveys 127

6.2.1 SDSS . . . 127
6.2.2 UKIDSS . . . 131
6.3 Implemented point spread functions . . . 135
6.3.1 Gaussian . . . 137
6.3.2 Two-Gaussian . . . 137
6.3.3 Moffat . . . 138
6.3.4 Moffat-combined . . . 139
6.4 Magnitude calculations . . . 140
6.4.1 Pogson vs asinh magnitude . . . 142
6.4.2 SDSS asinh magnitude calculation . . . 143
6.4.3 UKIDSS Vega magnitude calculation . . . 144
6.4.4 Conversion from survey magnitudes to AB magnitudes . . . 146
6.5 Flux calculations . . . 147
6.5.1 The raw ADU flux . . . 148
6.5.2 Conversion from survey flux units to Jansky via AB magnitudes . . . 150
6.6 Summary . . . 152

7 Two-metric threshold classification 153

7.1 Metric 1: flux-level population modelling . . . 155
7.2 Metric 2: pixel-level goodness of fit . . . 157
7.2.1 Chi-squared . . . 158
7.2.2 Quasar model . . . 158
7.3 Simulations . . . 159
7.3.1 Quasars . . . 163
7.3.2 MLT dwarfs . . . 165
7.3.3 Early type galaxies . . . 166

7.4 Application to real quasar candidates . . . 168
7.4.1 Survey data . . . 171
7.4.2 Validation set . . . 172
7.4.3 Manual threshold selection . . . 172
7.5 Results . . . 173
7.5.1 Performance measure . . . 174
7.5.2 Classification results . . . 175
7.5.3 Discussion . . . 176
7.6 Conclusions . . . 182

8 Machine learning classification pipeline 184

8.1 Training data . . . 186
8.1.1 Training set 1 . . . 186
8.1.2 Training set 2 . . . 187
8.1.3 Purity of the training set . . . 187
8.2 Classifiers . . . 189
8.2.1 Feature sets . . . 189
8.2.2 Decision tree . . . 190
8.2.3 Random forest . . . 191
8.3 Results . . . 193
8.3.1 Performance measure . . . 193
8.3.2 Classification results . . . 194
8.3.3 Dimensionality reduction . . . 199
8.3.4 Discussion . . . 202
8.4 Conclusions . . . 209

9 CNN classification on raw image data **210**

9.1 Overview 210

9.2 Training dataset 212

9.2.1 Survey data 212

9.2.2 Training data 212

9.2.3 Pre-processing of the training set 212

9.3 Architecture 214

9.3.1 Input and output of the CNN 215

9.3.2 Standardisation 216

9.3.3 No pooling layers 216

9.3.4 Channels 216

9.3.5 Overfitting 217

9.4 Data augmentation 217

9.5 Results 220

9.5.1 Misclassifications 221

9.5.2 Limitations of this method 224

9.6 Conclusions 226

10 Conclusion **227**

10.1 Future Prospects 228

Bibliography **230**

A Visual classification labels **249**

B Bayesian inference vs frequentist statistics **272**

B.1 Frequentist statistics 272

B.1.1 Null hypothesis and p-values 273

B.2 Advantages and disadvantages of Bayesian and frequentist statistics 274

C Additional information on machine learning **276**

C.1 Loss functions and optimization techniques 276

C.1.1 Loss functions 276

C.1.2 Optimization techniques 277

C.2 Regularisation 280

C.2.1 Weight decay 280

C.2.2 Early stopping 281

C.2.3 Regularisation specific for neural networks 281

C.3 Dimensionality reduction 283

C.3.1 Curse of dimensionality 283

C.3.2 PCA: Principal Component Analysis 283

C.3.3 Non-linear PCA: Kernel trick 285

C.4 Backpropagation in artificial neural networks 286

List of Tables

2.1 The highest redshift quasars found to date 28

2.2 Comparison of past (SDSS, UKIDSS) and upcoming (LSST, Euclid, Roman) surveys 29

2.3 The predicted number of quasars detected in Euclid 29

3.1 Properties of MLT dwarfs, early type galaxies and AGN 32

5.1 Supervised vs unsupervised vs semi-supervised learning 70

6.1 Parameter values of our data processing pipeline 112

6.2 Stored information for each candidate source 113

6.3 Stored paths of the downloaded images 114

6.4 Stored paths of the postage stamps 114

6.5 Effective wavelengths and magnitude limits of SDSS 128

6.6 SDSS Data Releases with quasar discoveries 128

6.7 Filter ID for each UKIDSS passband 134

6.8 Information on variables that are needed for correcting the measured flux 145

6.9 AB offsets for SDSS and UKIDSS 146

7.1 Two metrics that test different quasar properties 153

7.2 Simulation results of simulated quasars, MLT dwarfs and ETGs 162

7.3 The thresholds in the residual chi-squared that give the largest F_β score for $\beta = 1, 2, 3$ 176

8.1 Feature sets A and B that are used for classification 190

8.2 The classification results (AUC score) on training set 1 195

8.3 The classification results (AUC score) on training set 2 ($P_q > 0.01$) 195

8.4 Tuned parameters of the RF 196

10.1 AUC scores of the RF and the CNN 228

A.1 Visual classification labels 249

List of Figures

1.1 Radio-loud AGN. Figure taken from Urry & Padovani (1995). 1

1.2 AGN unification 2

1.3 Radio visibility, variability and emission line width of AGN 3

1.4 Spectra of different subclasses of AGN 4

1.5 Blackbody SED for different absolute temperatures 8

1.6 Compton scattering 10

2.1 Rest-frame SED of quasars 12

2.2 Composite HST FOS rest-frame spectrum of 101 quasars with redshifts $z > 0.33$ . . 13

2.3 Illustration of the AGN's black hole, the accretion disk and the hot corona 14

2.4 Absorption and emission lines 17

2.5 Absorption at different energy levels 17

2.6 The Lyman series 18

2.7 The Lyman alpha forest 18

2.8 Spectrum of ULAS J1319+0959 of redshift $z = 6.13$ 19

2.9 Cross section of neutral hydrogen 19

2.10 Redshift vs lookback time, and redshift vs the age of the Universe 21

2.11 Redshift distribution of 2204 SDSS quasars 22

2.12 The damping wing 23

2.13 Black hole growth tracks for quasars at $z \geq 7$ 25

2.14 The quasar luminosity function at $z \sim 6$ (Jiang et al., 2016) 26

2.15 Expected number of quasars that can be found in upcoming surveys. 27

3.1 Spectral energy distribution of a quasar redshifted to $z = 6, 7, 8$ 33

3.2 The SEDs of M stars and LTY dwarfs 33

3.3 The relative flux of a modelled $z = 7$ quasar and a T8 dwarf 35

3.4 Colour tracks of quasars, MLT dwarfs and ETGs 36

3.5 Colour-colour diagram for $(\hat{Y} - \hat{J})$ vs. $(\hat{i} - \hat{Y})$ 37

3.6 Proper motion 38

3.7 Signal-to-noise ratio vs $W1 - W2$ colours of Gaia point sources 39

3.8 Redshift vs. proper motion of 355 moving quasars 39

3.9 The change in right ascension for quasar $4C39.25$ 40

3.10 Colour cuts to distinguish high-redshift quasars from contaminants 42

3.11 The process of classifying quasars 45

4.1 Binomial likelihood 54

5.1 A Venn diagram of artificial intelligence 67

5.2 Building blocks of machine learning 68

5.3 Underfitting vs overfitting for a given data set 73

5.4 Underfitting vs overfitting in terms of loss and complexity 74

5.5 An example of a decision tree 76

5.6 Advantages and disadvantages of decision trees 77

5.7 An illustration of how a decision tree splits up a data set. 78

5.8 An illustration of how inefficient a decision tree splits up a linearly separable data set. 79

5.9 Decision tree: A simple dataset that can be separated with one split. 81

5.10 An illustration of a random forest with majority voting 85

5.11 Advantages and disadvantages of random forests 86

5.12 An illustration of an artificial neural network. 90

5.13 Sigmoid activation function and its derivative 92

5.14 Tanh activation function and its derivative 93

5.15 ReLU activation function and its derivative 94

5.16 Leaky ReLU activation function and its derivative 95

5.17 Advantages and disadvantages of artificial neural networks 96

5.18 An example architecture of a convolutional neural network. 98

5.19 CNN: Convolution, bias and activation function 99

5.20 CNN: Convolution 100

5.21 CNN: Kernel examples 101

5.22 Max-pooling applied on a 4×4 array. 102

5.23 Average-pooling applied on a 4×4 array. 102

5.24 CNN: Flattening of image data for use in fully connected layer. 104

5.25 Data augmentation using elastic transformations 105

5.26 Advantages and disadvantages of convolutional neural networks 106

6.1 Flowchart of the data processing pipeline 111

6.2 Example postage stamps in SDSS and UKIDSS 116

6.3 Applying a circular mask on a postage stamp 117

6.4 Rotation of an image 117

6.5 Calculating the rotation angle of a postage stamp 118

6.6 The distribution of the pixel ADUs after sigma clipping has been applied 119

6.7 Convolution between a point source and a radially symmetric point spread function . 120

6.8 Example of a fitted source centre via the method of non-linear least squares 121

6.9 Reconstruction of nearby sources in the original postage stamp 123

6.10 The original image, residual and chi-squared pixel values of J092036.65+093407.9 . 125

6.11 Summarised candidate J092036.65+093407.9 126

6.12 SDSS camera arrangement 128

6.13 Illustration of SDSS' identifiers of run, camcol and field 129

6.14 Retrieve images corresponding to a given location from SDSS servers 130

6.15 Alignment of UKIDSS' WFCAM 132

6.16 Image identifiers of UKIDSS 132

6.17 Point spread functions for SDSS (Moffat-combined) and UKIDSS (Moffat-single) . . 136

6.18 Examples of three different point spread functions that have been normalized 137

6.19 Two Gaussians with different σ form a point spread function 138

6.20 Moffat vs Gaussian PSF for the same FWHM 139

6.21 Database magnitudes vs calculated magnitudes for SDSS 140

6.22 Database magnitudes vs calculated magnitudes for UKIDSS 141

6.23 Comparison between Pogson magnitudes and asinh magnitudes 144

7.1 A sequential process that sifts out contaminants from of a pool of candidates 154

7.2 Quasar ULAS J1120+0641 at redshift $z = 7.09$ 159

7.3 Candidate J000042.15+072229.9 has a probability of being an ETG of $P_{\mathrm{ETG}} = 1.00$. 160

7.4 A bright star in the vicinity of the source J000226.75−005604.5 160

7.5 A moving source J002318.32+000435.8 161

7.6 Distribution of flux in the J-band of our validation set 162

7.7 A set of simulated quasars with varying J-band flux 163

7.8 A set of simulated quasars with a nearby source varying in brightness and location . . 164

7.9 A set of simulated stationary MLT dwarfs 166

7.10 A set of simulated non-stationary MLT dwarfs 167

7.11 A set of simulated ETGs with varying J-band flux 168

7.12 A simulated source that has correct quasar template colours but has moved 169

7.13 A simulated faint quasar of magnitude $m_J = 28.90$ mag 170

7.14 Scatter plot of $\tilde{\chi}^2$ vs P_q based on the classification labels of the visual inspection . . . 171

7.15 The F_β score of the manual threshold classifier applied to the complete validation set 176

7.16 The F_β score of the manual threshold classifier applied to a subset with $P_q \geq 0.01$ . . 177

7.17 The F_β score of the manual threshold classifier applied to a subset with $P_q \geq 0.1$ . . . 177

7.18 This source J002101.25+004352.4 is one of a close pair 178

7.19 This source J091108.35+092732.0 is close to a diffraction spike 179

7.20 A quasar that was incorrectly rejected by the manual threshold classifier 179

7.21 Candidate J073749.93+261844.9 is the persistence image 180

7.22 A moving source that had originally been accepted via visual inspection 181

7.23 A source that had originally been rejected due to missing H- and K-band data 181

7.24 A possible quasar candidate 182

8.1 Flowchart of the data processing and the classification pipeline 185

8.2 Histogram of redshift estimates for a) training set 1, b) training set 2 ($P_q > 0.01$) . . . 186

8.3 Data processing issues of UKIDSS' WFCAM. 188

8.4 A Receiver Operating Characteristic (ROC) curve 194

8.5 ROC curve for an instance of the RF and the DT using training set 2 197

8.6 Confusion matrix for the DT threshold as in Fig. 8.5 197

8.7 Confusion matrix for the RF thresholds as in Fig. 8.5 198

8.8 Predicted redshifts of false positives and false negatives 198

8.9 The 10 most important features for feature set B and training set 2 201

8.10 A candidate that often gets incorrectly rejected by the RF 203

8.11 A source that often gets rejected by the RF 203

8.12 A candidate with cross-talk issues that gets accepted by ML classifiers 204

8.13 A candidate with cross-talk issues that gets accepted by the RF 204
8.14 A candidate that has been rejected by visual inspection is accepted by the RF 204
8.15 A candidate with a diffraction spike is incorrectly accepted by the DT and RF 205
8.16 A candidate consisting mostly of noise gets accepted by the RF 205
8.17 A moving source that incorrectly gets accepted by the RF 206
8.18 Candidate J003556.65+002931.9 is a moving source 207
8.19 Candidate J003648.23+001851.3 is a moving source 207
8.20 Candidate J003137.03+001232.8 is a moving source 208
8.21 Candidate J115132.21+075108.8 is a possible quasar candidate 208
8.22 Candidate J140823.78+075826.9 is a possible quasar candidate 209

9.1 Hierarchical architecture for the convolutional neural network 211
9.2 Architecture of the convolutional neural network for SDSS 218
9.3 Architecture of the convolutional neural network for UKIDSS 219
9.4 ROC curve and AUC score for the CNN 220
9.5 Confusion matrix of the CNN 221
9.6 Predicted redshifts of false positives and false negatives 222
9.7 An accepted candidate based on visual inspection is correctly rejected by the CNN . 223
9.8 An accepted candidate based on visual inspection is correctly rejected by the CNN . 223
9.9 An accepted candidate based on visual inspection is correctly rejected by the CNN . 223
9.10 An accepted candidate based on visual inspection is rejected by the CNN 224
9.11 A rejected candidate based on visual inspection is accepted by the CNN 224
9.12 A rejected candidate based on visual inspection is incorrectly accepted by the CNN . 225
9.13 A rejected candidate based on visual inspection is incorrectly accepted by the CNN . 225
9.14 A rejected candidate based on visual inspection is incorrectly accepted by the CNN . 225

A.1 Candidate J081827.26+053328.4 with classification A ia 253

A.2 Candidate J000226.75−005604.5 with classification IB u 254
A.3 Candidate J013035.08−002823.8 with classification A icr 255
A.4 Candidate J035115.82−002105.4 with classification A cr 256
A.5 Candidate J084119.73+272754.6 with classification ER 257
A.6 Candidate J084758.34+020504.1 with classification A fr 258
A.7 Candidate J002754.23−010736.9 with classification GR iprgr 259
A.8 Candidate J081959.57+064822.4 with classification IHRPRR ai 260
A.9 Candidate J124934.81+134759.9 with classification RPRHR irh 261
A.10 Candidate J083700.81+241219.2 with classification A jr 262
A.11 Candidate J115252.60+013045.7 with classification LR u 263
A.12 Candidate J112442.42+003517.8 with classification A mr 264
A.13 Candidate J035848.16+010102.7 with classification NOR i 265
A.14 Candidate J082207.64+002728.2 with classification A ipr 266
A.15 Candidate J132600.21+082335.1 with classification IA rsi 267
A.16 Candidate J154642.41−014756.6 with classification T u 268
A.17 Candidate J143135.16+102210.9 with classification A vr 269
A.18 Candidate J141902.93+001740.5 with classification IWR ai 270
A.19 Candidate J073749.93+261844.9 with classification zr 271

Chapter 1

Active galactic nuclei

Active galaxies contain supermassive black holes (SMBHs) in their centre and are actively accreting mass onto it, in comparison to inactive galaxies that do not accrete any mass. The bright central regions of active galaxies are called active galactic nuclei (AGN) (Lynden-Bell, 1969; Robson, 1996). AGN contain a SMBH of up to $M \sim 10^{10} M_{\odot}$ that is surrounded by a gas accretion disk (Soltan, 1982; Peterson, 1997), as is illustrated in Figure 1.1. Once all the mass in its vicinity has been used up, the accretion stops and the galaxy becomes inactive. Hence, the galaxies we see today may have been AGNs a billion years ago. In comparison, the SMBH of our inactive galaxy Sagittarius A* has a mass of $M \sim 10^6 M_{\odot}$.

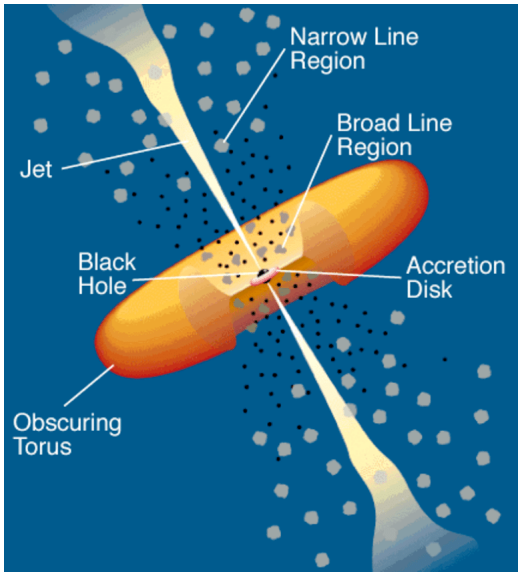


Figure 1.1: Radio-loud AGN. Figure taken from Urry & Padovani (1995).

Emission occurs across the entire electromagnetic spectrum over more than twenty orders of magnitude in frequency, having its peak in the UV (Padovani et al., 2017). Strong emission also occurs in the X-ray and radio, with a luminosity that can fluctuate every few hours. They further can have

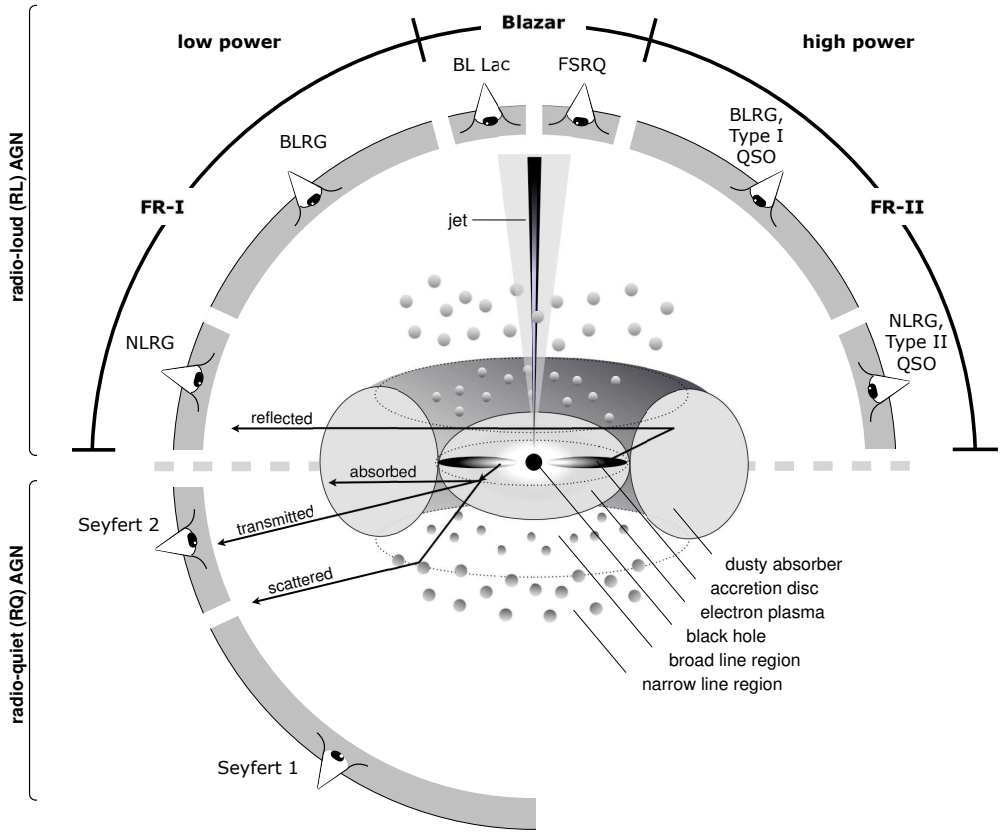


Figure 1.2: AGN unification. The distinction between different subclasses of AGN depends on the observer's viewing angle and on the presence or absence of a relativistic jet on the side of the observer. Figure taken from Beckmann & Shrader (2012).
Abbreviations: BL Lac - BL Lacertae object, - quasi-stellar object, BLRG - broad line radio galaxy, NLRG - narrow line radio galaxy, FR-I - Fanaroff-Riley Class 1, FR-II - Fanaroff-Riley Class 2.

winds and jets that are larger than the extent of their host galaxy. Different frequency bands capture different physical properties of AGN. The optical and UV emission is originating from the accretion disk (Urry & Padovani, 1995). It can be obscured by a dusty torus, which is traced by the infrared (IR). A hot corona is located above the accretion disk, which emits in X-rays. Non-thermal emission is captured in γ-rays and radio. In the vicinity of the SMBH are the broad line region clouds, while the narrow line region clouds are found further away and are less influenced by the central nucleus (Groves, 2007).

AGN is an umbrella term for the same phenomenon (Kembhavi & Narlikar, 1999), as can be seen in Figure 1.2. Examples of AGNs are quasars, radio galaxies, Seyfert galaxies and blazers. They mostly differ in their morphology or viewing angles towards the observer, which is illustrated in Figure 1.2. Depending on the viewing angle, a dusty torus can obscure the centre, which in turn results in a different spectral energy distribution (SED) than if the centre was visible (Koratkar & Blaes, 1999). The brightness in the radio is dependent on whether the observer is located on the side of a relativistic jet or not. This is illustrated in Figure 1.4, which shows the SED of different types of AGN, together with the SED of a normal galaxy in comparison.

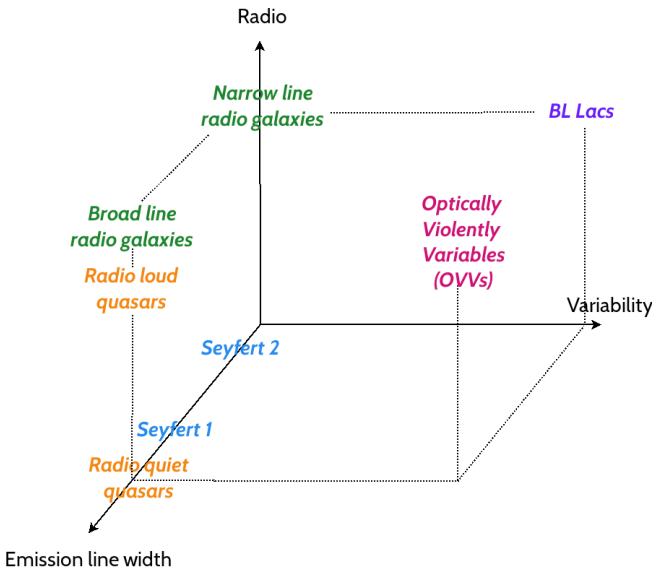


Figure 1.3: Different classes of AGN depending on their radio visibility, variability and emission line width. Figure closely adapted from (Krolik, 1999).

1.1 Subclasses of active galactic nuclei

Distinctions between subclasses of AGNs are made between variability, emission line width and visibility in the radio as is illustrated in Figure 1.3. We explain the subclasses of AGN below.

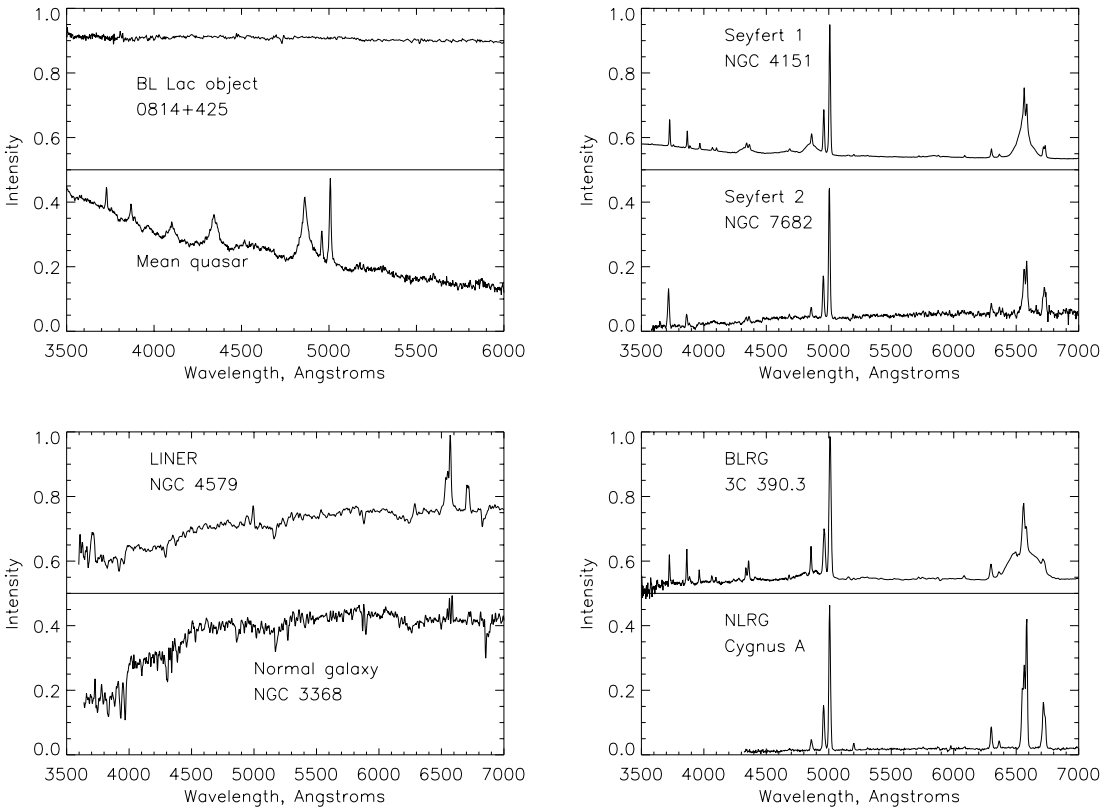


Figure 1.4: Spectral energy distribution of different subclasses of AGN, and that of a normal galaxy in comparison. Figure taken from Keel (2002a).

1.1.1 Quasars

Quasars, also called quasi-stellar radio sources, are the most luminous AGN with an absolute magnitude of $M_B < -21.5 + 5 \log h_0$ (Peterson, 1997), where $H_0 = 100\, h_0$ km s^{-1} Mpc^{-1} (Croton, 2013). Quasars have an unobscured view on the central nucleus . If the observer is positioned on the side of a relativistic jet, the quasar is radio-loud. Otherwise, the quasar is radio-quiet. A typical SED is shown in Figure 1.4. Quasars are explained in detail in Chapter 2.

1.1.2 Seyfert galaxies

Seyfert galaxies are less luminous compared to quasars, having an absolute magnitude of $M_B > -21.5 + 5 \log h_0$ (Peterson, 1997). They are often found in spiral galaxies. Due to their lower luminosity and lower redshift, the host galaxy is still visible. In Seyfert 1 galaxies, the view to the centre is again unobscured and hence both broad and narrow emission lines are visible in their SED. For Seyfert 2 galaxies, this view is obscured by the torus, which consequently hides the broad line region and only shows the narrow emission lines. This can be seen in the top right in Figure 1.4. While the narrow emission lines correspond to a low-density ionized gas with electron densities of $n_e \simeq 10^3 - 10^6$ cm^{-3}, the broad emission lines correspond to a high-density ionized gas with electron densities of $n_e \geq 10^9$ cm^{-3} (Beckmann & Shrader, 2012).

1.1.3 Radio galaxies

Radio galaxies are radio bright. Compared to Seyfert galaxies, radio galaxies are often found in elliptical galaxies (Peterson, 1997). Broad line radio galaxies (BLRG) and narrow line radio galaxies (NLRG) have analogous properties to Seyfert galaxies of type 1 and 2, but are radio-loud instead. The jets of radio galaxies are larger than those of radio-loud quasars (Barthel, 1989).

1.1.4 Blazars

Blazars include optically violent variables (OVVs) and BL Lacertae objects (BL Lacs). Here, the observer is in the line of sight of the jet, which makes blazars highly variable, highly polarized and radio-loud (Fan, 2002). OVVs have a polarisation fraction of $\gtrsim 2\%$ due to the synchrotron self-Compton (SSC) mechanism (Haardt & Maraschi, 1993; Peterson, 1997; Krolik, 1999). BL Lacs also exhibit a high polarisation fraction of $5 - 10\%$ (Hovatta et al., 2016). In comparison, most AGNs are weakly linearly polarized with a percentage of $\sim 0.5 - 2\%$ that is slightly above that of stars (Krolik, 1999). BL Lacs have a featureless continuum since the emission in the jet is strongly beamed (Xie

et al., 1993). This is shown in the BL Lac spectrum in the top left of Figure 1.4, where neither strong emission nor absorption lines are visible.

1.2 Observational properties

1.2.1 Continuum and emission lines

The majority of AGN show a broad-band continuum from IR to X-Rays (Krolik, 1999). They further have strong emission lines covering a wide range of ionization. One of the most prominent emission lines seen in their spectra are hydrogen lines from the Lyman and Balmer series, most notably Ly α. Further strong emission lines include C IV, Fe Kα and Mg II (Schmidt, 1969). Some of these are very broad that suggest velocities up to $\sim 10^4$ km s^{-1} (Matthews et al., 1964; Krolik, 1999). The widths of the emission lines can be used to constrain the radius of the nuclei to be approximately 1 pc to 100 pc (Peterson, 1997). The full-width-half-maximum (FWHM) of the Mg II line together with the quasar's luminosity can also be used to obtain the mass of the black hole (Vestergaard & Osmer, 2009). The continuum can trace AGN feedback which defines the effect an AGN has on its host galaxy (Elvis, 2000). This feedback constrains the maximum mass a galaxy can have (Di Matteo et al., 2005).

1.2.2 Polarization and variability

Most AGNs are weakly linearly polarized with a percentage of $\sim 0.5 - 2\%$ that is slightly above that of stars (Krolik, 1999). Only blazars are strongly polarized, which is due to the Synchrotron self-Compton mechanism, where photons are inverse-Compton scattered off the electrons that create the synchrotron radiation (Haardt & Maraschi, 1993; Peterson, 1997).

AGNs show variability in their luminosity that is typically 10% over a range of several years, but can have larger amplitudes over the course of a few hours as well. This depends on the luminosity of the AGN and the wavelength range. In general, the shorter the wavelength, the greater the amplitude of variability (Krolik, 1999). High variability implies that the emitting source must be smaller than a few light days. X-rays thus show rapid variations. The variability in the UV and optical is in phase, whereas the one in the IR is not (Peterson, 1997). This is explained by a difference in size and distance of the emitting regions between the UV/optical variability and the IR variability. The IR radiation is mostly due to the dust at large distances of $r \sim 1$ pc, whereas the emission in UV and the optical originate from the central compact source (Peterson, 1997). For the broad line regions, the highest ionization lines vary the most rapidly compared to the continuum variability as they are the lines closest in to continuum source.

1.3 Physical properties

1.3.1 Accretion disk

The fueling mechanism of AGNs is due to a deep gravitational potential well from their SMBH that converts potential energy from the infalling matter into kinetic energy (Robson, 1996; Elvis, 2000). This causes an accretion disk to form, which is a rotational support for the infalling matter (Shakura & Sunyaev, 1973). The disk in its simplest form is optically thick and geometrically thin (Kolykhalov & Sunyaev, 1984). Dissipative processes such as viscosity cause the matter to lose angular momentum that is transported outwards. A vast amount of energy is released in terms of heat or magnetic fields (Urry & Padovani, 1995). This is radiated away in the UV and soft X-ray wavelengths. In the vicinity of the black hole hard X-rays are produced. The accreted material can reach temperatures greater than that of hot stars with $T_{max} \propto L_{bol} M_{BH}^{-1/2}$ (Novikov & Thorne, 1973). The temperatrues range from 3×10^4 K to 3×10^5 K (Koratkar & Blaes, 1999).

The SMBH can be seen as having two phases - the first phase is the AGN phase, that is, when they are actively accreting. In this phase they gain most of their mass (Elvis, 2000). The second phase is the inactive phase, when the AGN has been shut down and no more accreting occurs. Most of the mass that forms the accretion disk is baryonic, and only $\sim 10\%$ is coming from dark matter (Peirani & de Freitas Pacheco, 2008; Soltan, 1982). The latter does not contribute to the luminosity of the quasar as no electromagnetic emission occurs when dark matter is accreted. A source of baryonic matter is given by gas and dust of the interstellar medium, and also by infalling stars. Stars need to be tidally disrupted in order to fuel the accretion disk (Peterson, 1997). The maximum radius a star can approach the black hole without being tidally disrupted is given by the Roche limit. In order for the gas of a star to become part of the accretion disk, this Roche limit has to be larger than the Schwarzschild radius of the black hole. This is because otherwise the star ends up increasing the mass of the black hole instead, which releases gravitational radiation instead of electromagnetic radiation. This gives (Peterson, 1997)

$$M_{BH} \lesssim 5 \times 10^8 \rho_*^{-1/2} M_\odot, \tag{1.1}$$

where ρ_* the density of the star. In general, the first objects that will be fully consumed are stars with low angular momentum (Lynden-Bell, 1969).

1.3.2 Thermal emission

Thermal emission in AGN can arise from blackbody radiation and free-free emission. The former is due to the accretion disk and the latter originates from the ionized hydrogen (HII or H^+) regions. Although free-free emission is usually non-thermal, it can be thermal as well. Here, the HII region is

in local thermodynamic equilibrium and hence the free electrons are as well.

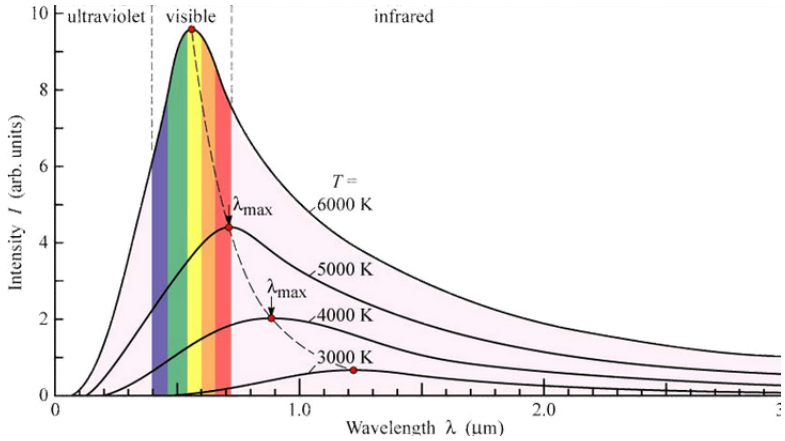


Figure 1.5: Blackbody SED for different absolute temperatures. As the temperature increases, the height of the peak decreases. Figure taken from Vagabond (2013).

1.3.2.1 Blackbody radiation

A blackbody emits thermal radiation which is isotropic, unpolarized and independent of its shape (Giancoli, 2000). A blackbody absorbs all radiation that falls onto it and therefore has a high optical depth. At the same time, it emits over all wavelengths. A peak is seen at a wavelength that corresponds to its thermal equilibrium temperature T_e. This wavelength can be calculated from Wien's displacement law,

$$\lambda_{max}\, T_e \approx 2.898^{-3}\mathrm{mK}. \tag{1.2}$$

Stars are examples of approximate blackbodies. Blackbody emission is given by Planck's law (Planck, 1991)

$$B_\nu(\nu, T_e) = \frac{2h\nu^3}{c^2}\,\frac{1}{\exp\left(\frac{h\nu}{k_B T_e}\right) - 1}, \tag{1.3}$$

where T_e the absolute temperature of the blackbody, k_B the Boltzmann constant, h the Planck constant, ν the frequency and c the speed of light. It has units of W m^{-2} sr^{-1} m^{-1}.

1.3.2.2 Radio emission

A UV photon from the central nucleus photoionizes a neutral hydrogen atom such that it loses its electron and turns into the positively charged HII. These HII regions then emit thermally in the radio via the free-free process arising from the interaction between the negatively charged electron and the positively charged HII.

1.3.3 High-energy emission

High-energy processes are non-thermal. This includes synchrotron radiation and Compton processes. The X-ray continuum of quasars can be seen in Figure 2.1. The energy flux of such high energy processes is given by a power-law,

$$F_\nu = C\,\nu^{-\Gamma+1} = C\,\nu^{-\alpha}, \tag{1.4}$$

where $\Gamma = \alpha + 1$ the photon index compared to the energy index α (Peterson, 1997).

1.3.3.1 Synchrotron radiation

The non-thermal ejection in the radio bands comes from the synchrotron mechanism arising from the presence of a magnetic field around the black hole. This magnetic field interacts with a relativistic electron (Robson, 1996). The frequency of emission of the relativistic electron that spirals within the magnetic field is given by

$$\nu_s = \frac{B\,e}{2\pi\,m}\,\gamma^2, \tag{1.5}$$

where γ the relativistic factor, B the magnetic field strength and e, m the charge and mass of the electron. Emission occurs in the radio wavelengths and is highly directed along the forward direction of the electron. The electron loses energy proportional to $(B \cdot E)^2$, where E the energy of the electron. Therefore, the greater the magnetic field strength and the greater the energy of the electron, the shorter the radiative lifetime of the electron.

1.3.3.2 Compton processes

Compton and Thomson scattering involve an interaction between a photon and an electron, where one of them is of high energy and the other of low-energy (Robson, 1996). This is illustrated in Figure 1.6. In Compton scattering, the photon is of high energy and the electron has low energy. The electron then gains energy from the incident photon. The photon changes its wavelength upon losing energy and scatters at an angle proportional to that energy loss. For Thomson (inverse Compton) scattering this process is reversed, where the incoming photon is of low energy and gains energy from the incoming electron. The relation for both processes is given by

$$\lambda_s - \lambda_i = 2\,\lambda_C\,\sin^2\left(\frac{\phi}{2}\right), \tag{1.6}$$

where λ_i and λ_s the wavelength of the incident and scattered photon, ϕ the angle at which the photon has been scattered and $\lambda_C = h/(m_e c)$ the Compton wavelength.

Emission due to Compton scattering acts like a cooling process for the photons and is present in the accretion disks of AGN. Photons arising from synchrotron radiation scatter electrons to higher

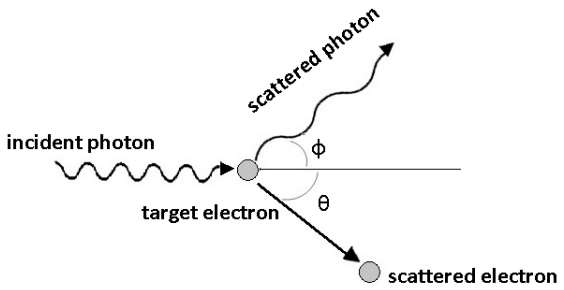


Figure 1.6: Compton scattering. An incoming photon of wavelength λ_i collides with a lower-energy stationary electron and scatters off at an angle ϕ. As it loses some of its energy to the electron, the scattered photon has a longer wavelength $\lambda_s > \lambda_i$.The electron gains energy from the incident photon, and scatters off at an angle θ. Figure taken from Venugopal & Bhagdikar (2012).

energies via Compton scattering. The maximum temperature of these electrons is $\sim 10^{12}$ K (Robson, 1996). At some point inverse Compton scattering kicks in that acts as a cooling mechanism for the electrons. This point occurs when electrons of high energies upscatter their own synchrotron photons. These high-energy photons can then easily escape. The spectrum of inverse Compton scattering has the same shape as that of synchrotron radiation, but is shifted to higher frequencies.

1.3.3.3 Fluorescence

Fluorescence defines a process where a photon is absorbed and re-emitted at a different energy. A high energy photon is absorbed to raise a low-energy electron to a much higher level. This causes an electron at a lower orbital to fall to the original low-energy level of the electron while re-emitting at that energy difference. For example, take an electron that is raised from energy level E_1 to level E_4 by absorbing an incoming high-energy X-ray photon. Since now the ground state is not occupied, an electron at energy E_3 falls to level E_1 while emitting a photon of energy $E_3 - E_1$. In AGN this happens due to an optically thick accretion disk in the vicinity of a black hole (Reynolds, 1999). The cooler regions of the disk get illuminated by hard X-rays from the accretion disk corona. The reprocessing of the primary X-ray continuum then causes emission lines to form.

Chapter 2

Quasars

Quasars are among the most luminous objects in the universe. Having luminosities of $10^{11} - 10^{13} L_{\odot}$ (Peterson, 1997), quasars can be detected at large distances of redshifts $z \gtrsim 6.5$ and are thus seen as they were in the first billion years after the Big Bang and beyond (Mortlock et al., 2011; Bañados et al., 2018; Yang et al., 2020; Wang et al., 2021). They act as background searchlights of the Universe, which let us trace the epoch of reionization and the black hole mass evolution (Weedman, 1986; Loeb & Barkana, 2001; McLure & Dunlop, 2004; Mortlock, 2016).

They can outshine their entire host galaxy, which gives them a point-like stellar appearance. At the same time, quasars are constrained to a small volume of ~ 2 AU no larger than the size of our solar system (Elvis, 2010; Lynden-Bell, 1969). They can have winds and jets that are larger than the extent of their host galaxy. Quasars are variable across their entire spectrum and on scales of a day to several years (Smith & Hoffleit, 1963; van den Bergh et al., 1973; Vanden Berk et al., 2004; Meusinger et al., 2011; MacLeod et al., 2012). This variability typically can affect the magnitude by $0.3 - 0.5$ mag (Peterson, 1997).

The first quasar, 3C 273, was discovered by Maarten Schmidt in 1963 (Schmidt, 1963). It has an absolute magnitude of $M = -26.7$, which corresponds to a luminosity of about $L \sim 10^{12} L_{\odot}$. It was first discovered in the radio and was later found to have a star-like optical source (Hazard et al., 1963). The term quasar actually is a short-hand form for *quasi-stellar-radio-source*. In fact, many of the initially discovered quasars were radio-loud quasars, even though they only make up $5 - 10\%$ of all quasars (Krolik, 1999). This might be partly due to the fact that AGN and star forming galaxies (SFGs) together make up the majority of sources that are bright in the radio, with SFGs still being fainter than AGN (Padovani et al., 2017). Quasars therefore do not have too many contaminants in that wavelength regime. In contrast, quasars that have been optically selected typically have radio to UV/optical emission ratio of 10^{-2} and thus are radio-quiet quasars. Contaminants in the optical are stars with a flux peak that coincides with the redshifted Lyman-alpha emission line.

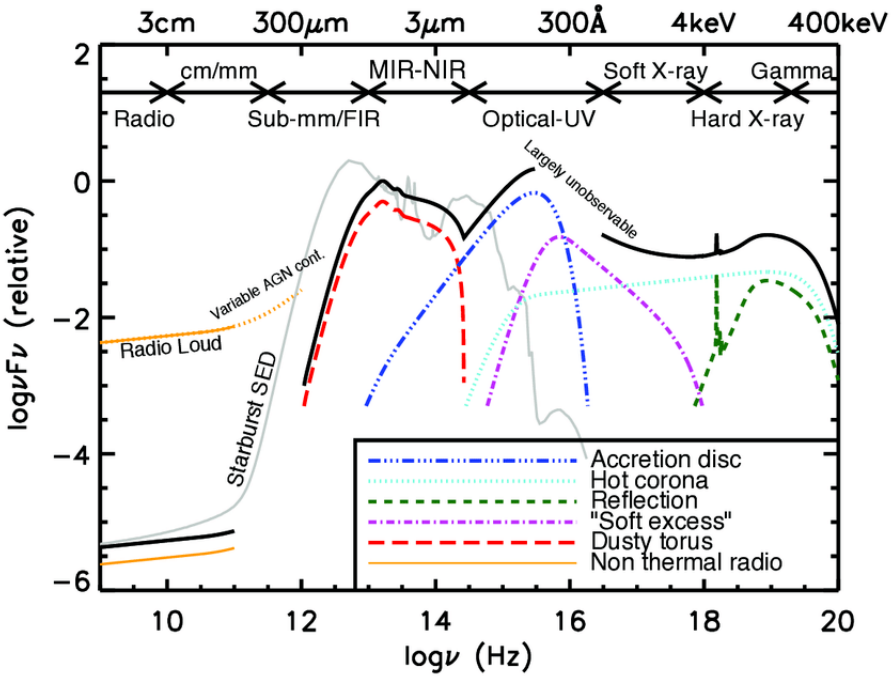


Figure 2.1: Rest-frame SED of quasars. The total SED is shown in black. Its components are shown in different dashed colour curves. Emission due to the accretion disk mostly occurs in the optical and UV. Emission in the IR arises from the dusty torus. Figure taken from Inoue et al. (2021).

2.1 Spectral energy distribution

The spectral energy distribution of quasars cannot be described via blackbody emission alone. This is due to the fact that not all of its radiation is of thermal origin. Only the UV and optical can be approximated by a ‘sum of black bodies’ due to their thermal origin (Elvis, 2000).

The quasar continuum is illustrated in Figure 2.1 and the rest-frame spectrum with emission lines can be seen in Figure 2.2. The thermal part of the continuum can be seen as the blue dashed line in Figure 2.1. Thermal emission in AGN can arise from blackbody radiation and free-free emission. The former is due to the accretion disk and the latter originates from the ionized hydrogen HII regions. The flux between the IR and X-rays is nearly constant, which indicates a high-energy source of radiation. High-energy processes are non-thermal. This includes synchrotron radiation and Compton processes.

There is a strong UV excess due to emission from the accretion disk. In addition there is some soft X-ray excess, which together forms the big blue bump (BBB). There is another bump in the infrared arising from the dusty torus. The radio emission depends on the radio-loudness of the quasar. The X-ray emission mostly arises from the magnetized corona above the disk and from Compton reflection (Koratkar & Blaes, 1999).

As a crude first approximation, the SED of the quasar continuum can be described by a power law

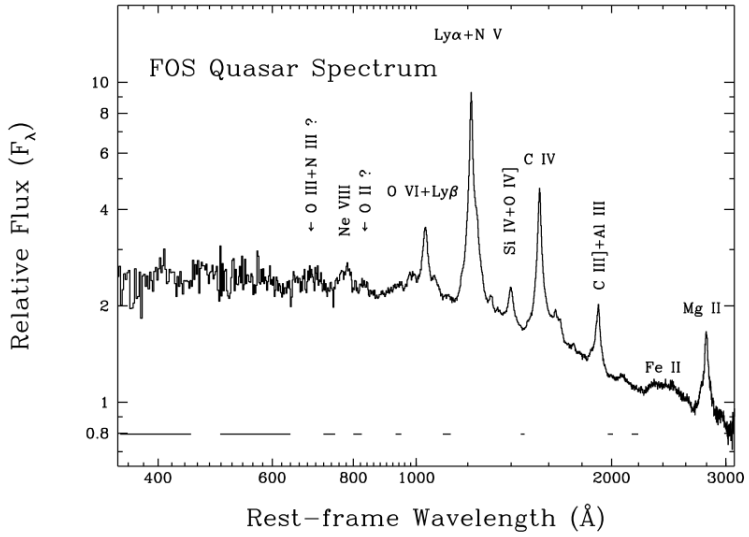


Figure 2.2: Composite HST FOS rest-frame spectrum of 101 quasars with redshifts $z > 0.33$. Figure taken from Zheng et al. (1997).

(Peterson, 1997). The specific flux F_ν per unit frequency interval is then given by

$$F_\nu = C\,\nu^{-\alpha}, \tag{2.1}$$

where ν the frequency, $\alpha \in \mathbb{R}$ the power-law index and $C \in \mathbb{R}^+$ a constant of proportionality. For AGNs, the power-law index is typically positive and in the range of $\alpha \in [0, 1]$, meaning that the specific flux F_ν decreases as the frequency ν increases. A single power law is typically a good approximation for the optical continuum. A more accurate prediction can be made by using several power-law fits over different wavelength ranges. The optical/UV continuum has a power-law index of $\alpha \simeq 0.32$ for radio-quiet quasars (Francis et al., 1991), and the UV slope has $\alpha \simeq 0.99$ (Zheng et al., 1997).

2.1.1 Big blue bump

The big blue bump describes the peak in the optical-UV and soft X-rays. The BBB can be seen in Figure 2.1 in the optical-UV and decreasing towards the soft X-rays. It is of thermal origin, due to the radiation from an optically thick accretion disk. It can reach temperatures of about $\sim 10^{5\pm1}$ K (Peterson, 1997).

2.1.2 X-ray

Quasars frequently have a much larger X-ray to optical flux ratio than any other astronomical object (Krolik, 1999). They can show high flux variabilities of a few hours in the X-rays, which implies that the region from where X-rays are emitted must be small (McHardy, 1989). X-ray emission arises from the up-scattering of photons from the accretion disk via inverse Compton scattering by free electrons in the corona (Peterson, 1997). UV-optical photons from cool regions of the optically thick accretion disk are inverse Compton scattered off relativistic electrons in the hot corona to heat up the denser layers. At the same time, they provide a cooling mechanism for the hot electrons within the corona (Haardt & Maraschi, 1993).

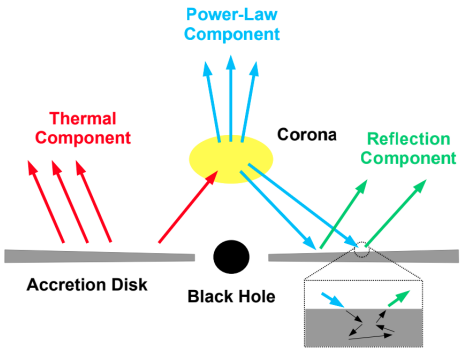


Figure 2.3: Illustration of the AGN's black hole in black, the accretion disk in grey and the hot corona in yellow. (red) The cool accretion disk emits thermal photons via blackbody radiation. (blue) The hot corona produces power-law X-ray emission via inverse Compton scattering from the incoming low-energy UV/optical photons from the accretion disk and the high-energy electrons in the hot corona. (green) The emitting high-energy photons from the corona are further reflected off the accretion disk. Figure taken from Bambi (2018).

Due to the non-thermal emission, the continuum forms a power-law in the X-rays. Compton reflection and fluorescence causes a reflection hump and Kα metal emission lines to form in the X-rays (Reynolds, 1999). Both features arise from fluorescence in the disk that is illuminated by X-rays. That is, X-ray photons from the hot corona get reflected by the accretion disk. This is illustrated in Figure 2.3. The effect depends on the inclination angle of the accretion disk. The Kα metal emission lines arise from the most abundant metals in the inner regions of the accretion disk. The strongest metal emission line is iron Fe Kα at $E_{Fe} = 6.4$ keV and it is highly broadened. The reflection hump forms at energies > 10 keV peaking around ~ 30 keV (Koratkar & Blaes, 1999; Nandra et al., 1991). The peak in the reflection mechanism at the Fe Kα emission line can be seen within the dashed green line in the hard X-rays in Figure 2.1.

2.1.3 IR bump

The bump at wavelengths from $\sim$ 1 μm to $\sim$ 100 μm is called the *IR bump* (Koratkar & Blaes, 1999). It is mostly due to dust emission that is reprocessing of the emission from the BBB. A local minimum can be seen at around $\sim$ 1 μm. This wavelength corresponds to the sublimation temperature $T_s \sim 1800 - 2000$ K of dust. The dust mostly consists of graphite due to its higher sublimation temperature to silicate. The sublimation radius $r_s \sim 1$ pc is the boundary between the hot thermal spectrum of the big blue bump and a cool thermal spectrum arising from the emission of dust grains (Peterson, 1997). Inside that radius the dust grains are destroyed by evaporation from the UV emission (Barvainis, 1987). The cooler dust grains outside r_s are responsible for emissions longwards of $\sim$ 2 μm. The temperatures go down to $\simeq 50 - 60$ K in the far-infrared (Wang et al., 2008). The dust absorbs some of the UV radiation and re-radiates isotropically in the infrared (Robson, 1996). The dust grains are semi-opaque and therefore not perfect blackbodies, causing emission in the Rayleigh-Jeans region.

2.1.4 Radio continuum

In contrast to the UV wavelengths, the radio emission is of non-thermal origin due to synchrotron radiation. The radio source can be extended or compact. An extended radio source often consists of two lobes that are optically thin. They often contain a relativistic jet on one side. Such AGN show a steep ($\alpha \sim -0.8$) radio spectrum (Robson, 1996; Kimball & Ivezic, 2007) and are radio-loud. They have a radio-to-optical flux emission of $F_{5GHz}/F_B \gtrsim 10$, where the B-magnitude has an effecive wavelength $\approx$ 4400Å (Schmidt & Green, 1983; Kellermann et al., 1989). A compact radio source is optically thick and can often not be resolved. It is radio-quiet and does not contain any jets. Radio-quiet quasars show a flat ($\alpha \sim 0$) radio spectrum (Padovani et al., 2017).

The first detected quasars were initially discovered in the radio (Matthews & Sandage, 1963; Sandage & Wyndham, 1965; Schmidt, 1969). Up to 20% of observed quasars are radio-loud. The actual number of radio-loud quasars is much smaller, going down to predictions of less than 1% (Wilson & Colbert, 1995; Kellermann et al., 1989; Krolik, 1999; Padovani, 2011; Padovani et al., 2017).

Radio-quiet quasars often still have some radio emission (Beckmann & Shrader, 2012). The ratio between absolute and relative radio power is given by $R \equiv \frac{f_{6\,\mathrm{cm}}}{f_{4400\,\text{Å}}}$, where $R \sim 10^2 - 10^5$ for radio-loud quasars and $R \lesssim 0.1 - 10$ for radio-quiet quasars (Laor & Behar, 2008). Some radio-quiet quasars have also been found to have transitioned to radio-loud quasars (Nyland et al., 2020). The source of radio emission in radio-quiet quasars is still up for debate (Herrera Ruiz et al., 2016). At least some of the emission arises from non-thermal activity from the nucleus, while other emission arises from star-forming processes within the host galaxy (Padovani et al., 2011).

2.1.5 Broad line region

Broad emission lines are lines that have been Doppler broadened (Peterson, 1997). As such one can infer the velocities of the components that have led to the emission. In the BLR, these correpond to velocities of $10^3 - 10^4$ km s^{-1} (Beckmann & Shrader, 2012). The broad line region (BLR) is close to the central nucleus, with a distance of $\sim$ 1 pc (Robson, 1996). The BLR is hence a useful tracer of the central source.

The rest-frame spectrum for the quasars of Sloan Digital Sky Survey (SDSS) is illustrated in Figure 2.2. One can see that the most prominent emission lines are the hydrogen emission lines Lyα ($2 \rightarrow 1$), Hα ($3 \rightarrow 2$) and Hβ ($4 \rightarrow 2$), the elements carbon C IV, C III, magnesium Mg II, the 'forbidden' oxygen O III line and iron Fe Kα (Weedman, 1986; Peterson, 1997). One can infer the redshift of a quasar from these strong characteristic emission lines. These emission line features stay nearly constant over a range of 7 orders of magnitude in luminosity and 13 Gyr of time, which would imply that the SMBH scales with luminosity (Koratkar & Blaes, 1999; Elvis, 2000).

One can compare the strengths of different emission lines via the equivalent width, which is defined as (Peterson, 1997)

$$W_\lambda = \int \frac{F_o(\lambda) - F_c(\lambda)}{F_c(\lambda)} d\lambda, \tag{2.2}$$

where $F_o(\lambda)$ the observed flux and $F_c(\lambda)$ the continuum flux at wavelength λ. It has units of length and therefore can be interpreted as a width.

2.1.6 Narrow line region

The narrow line region (NLR) is a region of clouds that is further away from the nucleus than the BLR, at a distance of about $\sim$ 10 pc to $\sim$ 1 kpc (Robson, 1996). Since they are further away from the nucleus than the BLR, their line widths are narrower, with velocities of a few hundred km s^{-1} (Beckmann & Shrader, 2012). The BLR and NLR can be separated by the sublimation radius of dust. One can have forbidden lines in the NLR. This is not the case for the BLR, since its high electron densities lead to the collisional suppression of forbidden lines (Peterson, 1997). An example of a narrow emission line is the twice-ionized oxygen [O III] (forbidden) λ5007.

2.2 Absorption lines due to intervening matter

Absorption lines occur when there is an absorbing material between the source and the observer. This is illustrated in Figure 2.4. The lines are located at discrete frequencies that allow to capture the molecular composition of the absorbing material. This stems from quantum mechanics, which states

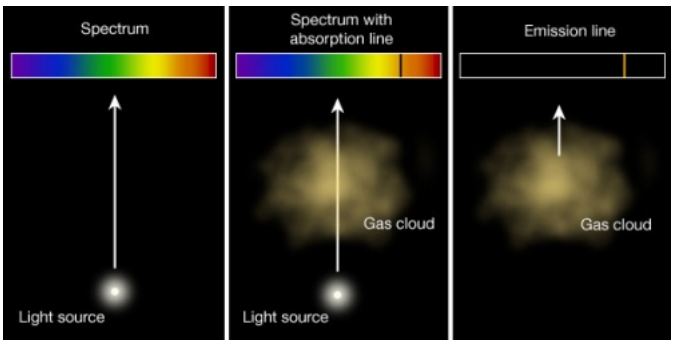


Figure 2.4: Absorption and emission lines. (left) A blackbody light source emits a continuous blackbody spectrum. (middle) A cool gas in front of a hot blackbody creates an absorption line. (right) A hot gas in front of a cool background emits an emission line. Figure taken from Strobel (2014).

that an atom or molecule can absorb a photon at given energy intervals. The allowed energies are those that correspond to the difference between two energy levels, as is shown in Figure 2.5. Note that the electron must be bound to an atom for emission lines to form. In contrast, free electrons can have continuum (non-quantized) energies, which cause the continuum emission (Robson, 1996).

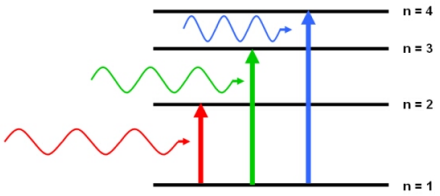


Figure 2.5: Photons can excite electrons into higher energy states. For that to happen the photon's energy must be equal to the difference in energy between two levels. The greater the energy difference, the higher the frequency of the photon. (red) The incoming photon promotes the electron from energy level 1 to level 2. The photon's energy equals $\Delta E_2 - \Delta E_1$. (green) The electron gets excited to level 3 from level 1. (blue) The incoming photon promotes the electron to level 4 from level 1. The energy of this photon will be greater than that of the photon in the red case. Figure taken from Swinburne Astronomy Online (2008).

The spectra of high-redshift quasars show many absorption lines. These probe the intergalactic medium (IGM) and baryonic matter, and give evidence to their existence (Rauch, 2000).

2.2.1 Lyman-alpha absorption

The Lyα hydrogen emission line at $\lambda_{\text{rest}} = 1215.67$ Å is the most distinct feature in the spectrum of quasars. It corresponds to an electron in an hydrogen atom that falls from energy level 2 to energy level 1. The Lyman series defines the emission line spectra that arise from a hydrogen atom whose electron falls from orbital $n \geq 2$ to orbital $n = 1$. This can be seen in Figure 2.6. The rest-frame wavelength of the Lyα emission line lies in the UV range. This ultraviolet light coming from the quasar then travels through space, while it gets redshifted on its way towards us. Every time the

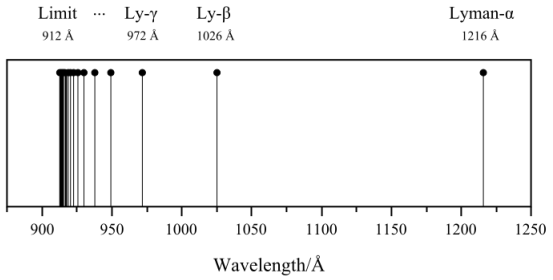


Figure 2.6: The Lyman series is the hydrogen spectral series where electrons fall towards the lowest energy level. It starts from $2 \rightarrow 1$ defined by α and goes up to the Lyman limit, which corresponds to the orbitals $\infty \rightarrow 1$. Figure taken from Wikipedia contributors.

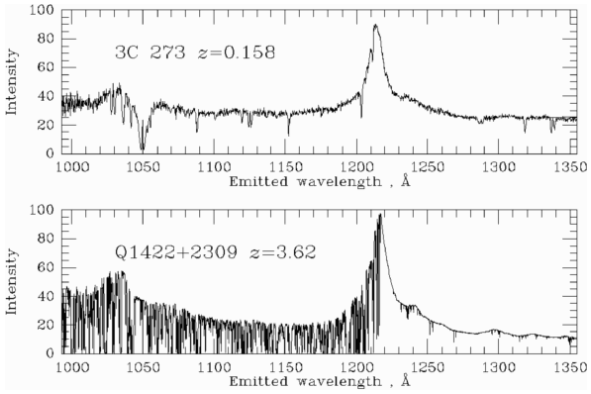


Figure 2.7: Spectra of (top) 3C 273 at $z = 0.158$ and (bottom) 1422+2309 at $z = 3.62$. Lyα absorption lines are discernible bluewards from the Ly α in the rest-frame of the quasar. The spectrum of the low-redshift quasar 3C 273 shows only a few absorption lines, while the higher redshift quasar 1422+2309 shows so many sharp absorption lines that there seems to be as much absorption as emission. Figure taken from Wright (2004); Keel (2002b).

light encounters a region of neutral hydrogen (HI), an absorption line is formed at the respective redshifted Lyα line. This suppresses its emission at that wavelength. The denser these regions, the more absorption lines occur. The wavelengths of these absorption lines can be calculated via

$$\lambda_{\text{obs}} = \lambda_{\text{rest}} \, (1 + z_{\text{abs}}), \tag{2.3}$$

where z_{abs} the redshift of the absorbing gas cloud, λ_{obs} the wavelength as seen from the observer and λ_{rest} the rest-frame wavelength. Since $z_{\text{obs}} \leq z_{\text{quasar}}$, one has $\lambda_{\text{obs}} \leq \lambda_{\text{quasar}}$ and thus the absorption occurs bluewards of the Lyα emission line. The spectrum then ranges from a Lyman-alpha forest with a few absorption lines towards a Gunn-Peterson trough, where almost everything is absorbed from the redshift of the quasar to the redshift of reionisation (Gunn & Peterson, 1965; Rauch, 1998; Haiman, 2002). This can be seen in Figure 2.7 for redshifts $z = 0.158$ and $z = 3.62$. Figure 2.8 illustrates that with the spectrum of a $z = 6.13$ quasar. Comparing these three spectra shows how the absorption increases with redshift.

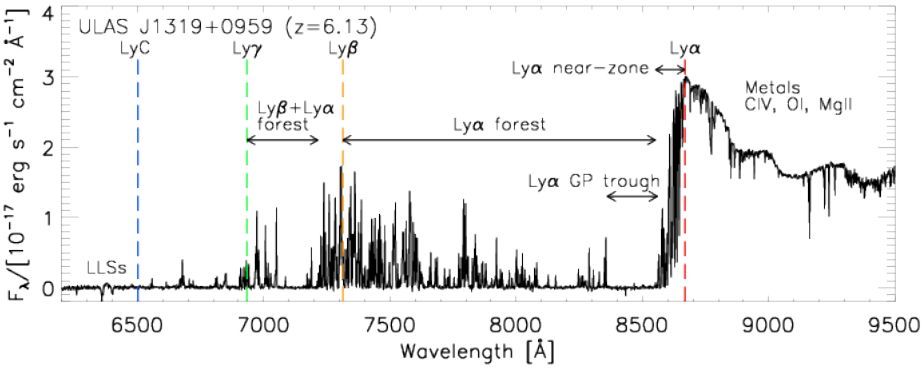


Figure 2.8: Spectrum of ULAS J1319+0959 of redshift $z = 6.13$. One can clearly see the redshifted Lyα emission line and the Lyα forest bluewards of it. Figure taken from Becker et al. (2015).

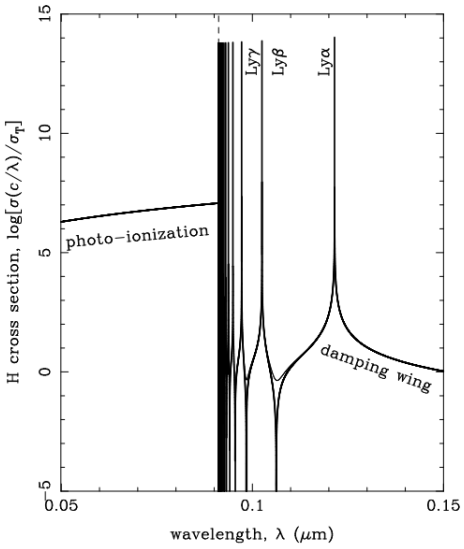


Figure 2.9: Cross section of neutral hydrogen. Figure taken from Mortlock (2016).

2.2.2 Scattering cross section of neutral hydrogen

The scattering cross section of neutral hydrogen is primarily given by Rayleigh scattering, Raman scattering and photoionization (Mortlock, 2016). This is illustrated in Figure 2.9.

2.2.2.1 Rayleigh scattering

Rayleigh scattering corresponds to elastic scattering, where energy is conserved. The incoming photon excites an electron to a higher energy virtual state and then drops back to the ground state causing a photon of the same energy to be emitted.

Rayleigh scattering explains the natural line broadening as a damped harmonic oscillator (Bach &

Lee, 2015),

$$\sigma(\nu) = \sigma_T \frac{\nu^4}{(\nu_0^2 - \nu^2)^2 + \gamma^2\nu^2}, \tag{2.4}$$

where ν the frequency of the incoming photon, ν_0 the resonance frequency, σ_T the Thomson cross section and γ the damping constant. The damping constant reflects the excited states plus the fine structure splitting (Rohrmann & Rueda, 2022). In the low frequency limit the cross section is proportional to ν^4. For $\nu \simeq \nu_0$, close to the resonance, this can be modeled by a Lorentzian function. In general, a Lorentzian profile is an excellent approximation of the cross section in the vicinity of the line centre, even though it is an overestimation at off-centre.

For damped Lyα systems the cross section is given by the Voigt profile (Rybicki & Lightman, 1979). It has a similar shape to the Lorentzian in the wings but is Gaussian near the peak.

2.2.2.2 Raman scattering

Raman scattering is inelastic, which means that the incoming photon experiences a shift in frequency. This implies that energy is either gained or lost. The incoming photon excites the electron to a higher energy states after which the electron falls to a lower energy state different to the ground state. The photon can excite vibrational and rotational modes of the atom. For example, an incoming photon of wavelength Lyβ ($1 \leftrightarrow 3$) causes an electron to be pushed from the ground state to $3s$, after which the electron falls into the $2s$ state and a Hα ($2 \leftrightarrow 3$) photon is emitted. This scattered Hα photon is of lower energy compared to the incoming Lyβ photon.

2.2.2.3 Photoionization

Photoionization is the interaction between a high-energy photon and an atom, in which the photon removes one or more electrons of that atom. In this case, a UV photon kicks out the electron of the hydrogen atom, which forms HII. The energy that is needed to ionize hydrogen is ≥ 13.6 eV, which in turn corresponds to UV wavelengths of ≤ 912 Å (Madau & Rees, 2000). The photoionization cross-section for neutral hydrogen increases as $\sim \lambda^3$ and has its maximum at 912 Å (Osterbrock, 2006). At larger wavelengths, the cross section is zero since those wavelengths correspond to energies lower than 13.6 eV.

2.3 Quasars in the early universe

Quasars are one of the most distant objects in the Universe. The higher the redshift of the quasar, the further into the distant past one can look. The measured spectra allow us to probe the evolution of the neutral hydrogen content and properties of the quasars themselves.

We would like to gain more information of the first billion years after the Big Bang ($z \gtrsim 6.5$) since this is a period that is rather unobserved. This is due to the fact that either sources at such high redshifts are too faint to be observed (e.g. galaxies), or they are sufficiently bright only over a small period of time (e.g. gamma ray bursts) (Mortlock et al., 2012). Quasars, however, have high luminosities over a long period of time, which allows them to be observed at significant look-back times. A plot of redshift vs. lookback time can be seen in Figure 2.10. The most distant quasars are further the most luminous ones, which can be seen in Figure 2.11. It should be noted that in a flux-limited survey, two sources with the same observed flux can be at different redshifts. Hence, at large distances one can only find the most luminous quasars, as the less luminous ones would fall below the observable flux limit.

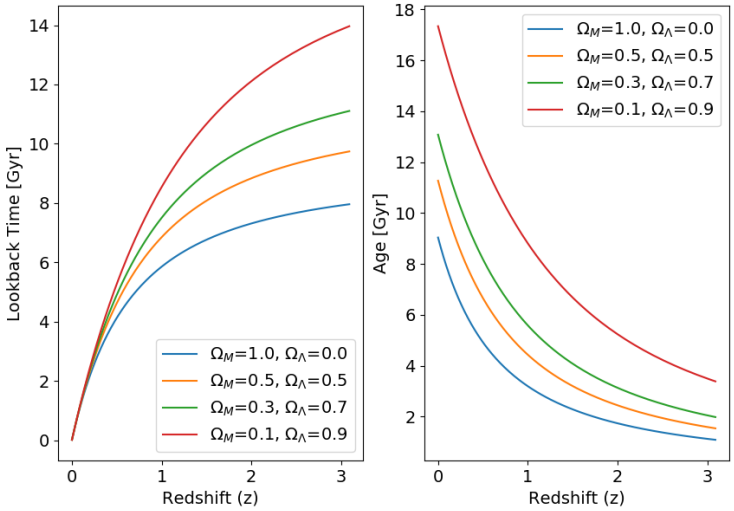


Figure 2.10: (left) Redshift vs lookback time, (right) redshift vs age of the Universe for different combinations of Ω_M and Ω_Λ. Our universe is predicted to have $\Omega_M \simeq 0.3, \Omega_\Lambda \simeq 0.7$, which corresponds to the green curve. Figure taken from Mihos (2018).

2.3.1 Cosmological probes

The spectra of high-redshift quasars can be used to identify the properties of the intervening matter due to arising absorption and emission lines. As such, they act like background searchlights of the universe (Weedman, 1986). Hence, high-redshift quasars allow us to deduce the epoch of reionisation and to put constraints on the early growth of SMBHs and baryonic matter (Fan et al., 2019).

2.3.1.1 Epoch of reionisation

As we have seen in Section 2.2, quasars can be used to trace properties of the IGM. Looking at the absorption bluewards of the Lyα emission line, one can constrain the reionization redshift.

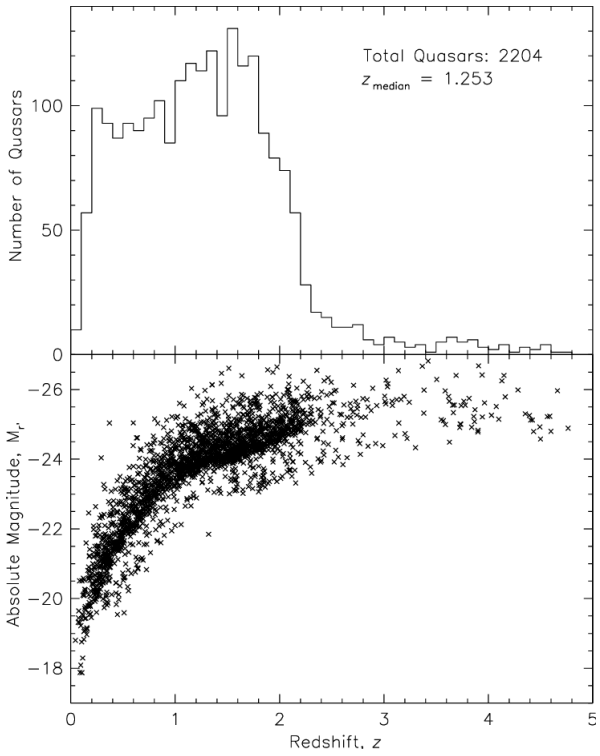


Figure 2.11: Redshift distribution of 2204 SDSS quasars. (top) redshift vs. number of quasars. (bottom) redshift vs. absolute magnitude. This shows that the most luminous quasars are also the most distant ones. Moreover, there is a peak in the quasar distribution at $z \sim 2$, after which it decreases drastically. Figure taken from Vanden Berk et al. (2001).

A technique for obtaining the neutral hydrogen fraction that is more sensitive to higher optical depths can be obtained by calculating the Lyα red damping wing and the ionized near-zone sizes (Strömgren sphere) around quasars (Miralda-Escudé, 1998; Shapiro & Giroux, 1987). This can be used for measuring neutral hydrogen fractions above 10^{-3} (Carilli et al., 2010).

The HII regions that surround quasars are also called Strömgen spheres. At redshift $z \sim 8$, these already can have radii of $\sim$ 10 Mpc, going up to $\sim$ 20 Mpc around $z \sim 6.5$; and growing to infinity at the end of the epoch of reionization at $z \sim 6$ (Alvarez & Abel, 2007). The size of the Strömgen sphere can be inferred by looking at the profile of the Lyman-alpha emission line (Cen & Haiman, 2000). The Lyman-alpha photons are less likely to be absorbed in HII regions compared to HI in the IGM, since HII has lost its electron and so cannot absorb an incoming photon.

At redshifts $z \lesssim 6$ the IGM was mostly ionized, showing a low neutral hydrogen fraction n_{HI}. The quasars discovered at $z = 7.54$ (Bañados et al., 2018) and $z = 7.085$ (Mortlock et al., 2011) showed a red damping wing implying that n_{HI} was still high at redshifts $z \gtrsim 7$. The red damping wing feature arises from the fact that the Lyα line is scattered off the observer's line-of-sight due to the diffuse neutral IGM (Madau & Rees, 2000).

Further, its optical depth is broadened by the natural width of the line and the velocity distribution of the atoms. The scattering cross section of the Ly α line depends on the natural width of the Ly

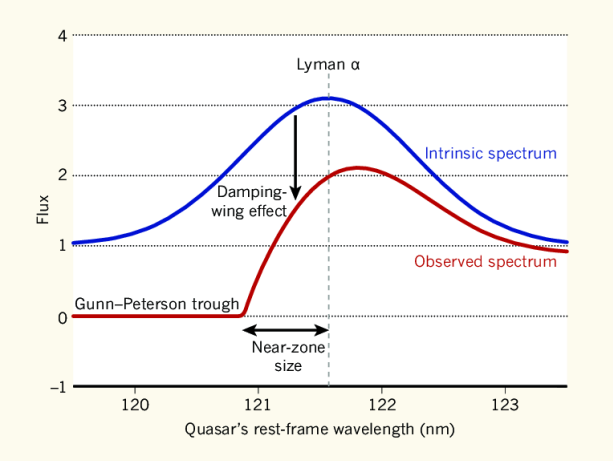


Figure 2.12: Damping wing as expected to show in a high-redshift quasar $z \gtrsim 7$ located within a neutral IGM. (blue) The intrinsic rest-frame spectrum of the quasar in a neutral IGM. (red) The spectrum as seen from the observer after neutral hydrogen absorption has occured. It takes a high neutral hydrogen fraction n_{HI} for the damping wing to be seen, compared to the Gunn-Peterson trough, which only needs a relatively small fraction of n_{HI}. The flux is transmitted within the HII ionized near-zone regions around the quasar. Figure taken from Willott (2011).

α resonance. This is illustrated in Figure 2.12. One can see the ionized near-zone of the quasar and the damping wing beyond this zone. Both HII regions and red damping wings are sensitive to high neutral hydrogen fractions n_{HI}, making these features typical for high-redshift quasars $z \gtrsim 7$ when the IGM still had a high neutral hydrogen fraction of $n_{HI} > 0.1$. This is in contrast to the Gunn-Peterson trough, which only needs a relatively small fraction of n_{HI} to be present in the observed spectrum. The greater the radius of the Strömgren sphere, the weaker the absorption in the damping wing redward of the Lyα line (Mesinger & Furlanetto, 2008). Gamma-ray burst afterglows are even more useful compared to quasars to analyse the red damping wing and therefore the neutral hydrogen fraction (Totani et al., 2014). This is because one can model their non-thermal synchrotron spectrum via a simple power law. In addition, due to their short lifetime, their radiation has little to no effect on the hydrogen ionization fraction. By fitting a neutral hydrogen absorption model on top of the power law to the observed spectrum, Totani et al. (2014) found that the end of the epoch of reionisation might not have yet been completed by $z \sim 6$.

The period of reionisation starts at $z \sim 20$ and ends around $z \sim 6$ (Becker et al., 2001; Robertson et al., 2010; Fan et al., 2006; Ciardi & Ferrara, 2005). During that era the first galaxies and quasars emerged , whose ionizing radiation caused reionization to start (Zaroubi, 2013). This indicates that the IGM was reionized inhomogeneously. Note that although the UV radiation from quasars is extremely powerful, quasars themselves are not responsible for the most part of reionization (Shapiro & Giroux, 1987). Further, the contribution of quasars becomes negligible for $z \gtrsim 6$ (Mitra et al., 2018). At $z = 6 - 6.6$, the quasar contribution to ionization is $\lesssim 7\%$ (Jiang et al., 2022). Instead, primary sources of reionization are early type dwarf galaxies and Pop III stars (Wise, 2012; Jeon et al., 2017). These sources emitted ultraviolet (UV) radiation, most notably the emission of Lyα. This high-energy

radiation was able to ionise the surrounding matter. These ionized bubbles grew until they overlapped (Arons & Wingert, 1972). Approximately 1 billion years after the Big Bang, the Universe was fully ionised again.

2.3.1.2 Initial black hole seed mass

Quasars trace the active black hole evolution from which one can infer the number of inactive black holes so as to get the full population. (Vestergaard & Osmer, 2009). Black holes also play a role in the evolution of their host galaxies (Di Matteo et al., 2005; Granato et al., 2004). In general, black holes form from the (partial) collapse of a gas cloud, supermassive star or a dense star cluster (Rees, 1984). SMBHs of mass $\sim 10^9 M_\odot$ were already visible after the first billion years of the Big Bang, which means they must have formed when the Universe was very young (Volonteri, 2010). At this early stage these seeds could have formed from the collapse of a primordial gas cloud, collisions in Pop III systems or dynamical processes in $\gtrsim 100 M_\odot$ stellar clusters (Alister Seguel et al., 2020). Supernovae explosions of massive stars that were more common in the early Universe lead to black hole remnants. These seeds then grow by accreting matter or by merging with other black holes, constrained only by their Eddington luminosity (Willott, 2011). It is not clear at what mass ranges the initial black hole seeds lie and how they were able to grow at such a quick rate (Fan et al., 2019).

The maximum luminosity a quasar can have for a given SMBH mass to not exceed the inward gravitational force such that $|F_{rad}| \leq |F_{grav}|$ is called the Eddington luminosity L_{Edd} (Peterson, 1997),

$$L_{Edd} = \frac{4\pi G\, c\, m_p}{\sigma_T} M_{\mathrm{BH}}, \tag{2.5}$$

where G the gravitational constant, c the speed of light, M_{BH} the mass of the quasar/black hole, m_p the proton mass and σ_T the Thomson cross section.

The Eddington accretion rate $\dot{M}_{Edd}$ is the maximum amount of accretion that is permitted to sustain the Eddington luminosity,

$$\dot{M}_{Edd} = \frac{L_{Edd}}{\eta c^2}, \tag{2.6}$$

where $\eta \in [0, 1]$ the efficiency of converting mass into energy with a typical value of $\eta \gtrsim 0.1$ (Yu & Tremaine, 2002; Davis & Laor, 2011). That is, η defines how much of the mass is released in the form of radiant energy.

The mass of a quasar's black hole at time t with initial seed mass $M_{seed} = M(t = 0)$ can be calculated via

$$M(t) = M(0)\, \exp\left(\frac{t}{t_S}\right), \tag{2.7}$$

where Eddington limited accretion rates is assumed (Volonteri, 2010; Eilers et al., 2020). The time t is measured from the birth of the quasar, and t_S is the Salpeter time that constitutes the typical

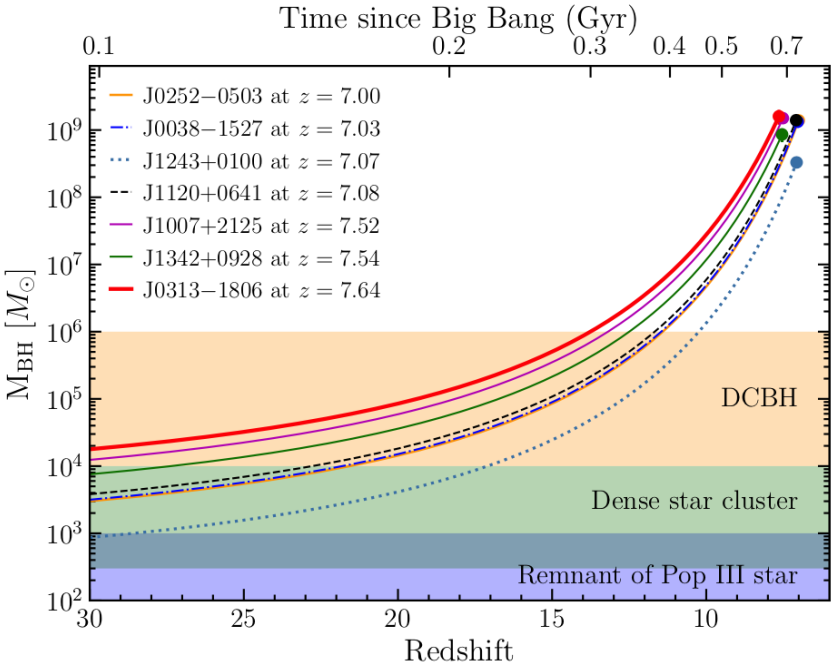


Figure 2.13: The black hole mass evolution for quasars at $z \geq 7$. The tracks are drawn for a radiative efficiency of 0.1. The seed masses that can be produced by Pop III stars, dense star clusters, and direct-collapse black holes (DCBH) are shown in the shaded regions. Figure taken from Wang et al. (2021).

timescale of black hole growth (Salpeter, 1964). It has a value of $t_S \sim 4.5 \times 10^7$ years for radiative efficiencies of $\eta \sim 0.1$ and accretion rates at the Eddington limit. This implies that it takes $t \approx \ln\left(10^7\right) \times 4.5 \times 10^7$ years $> 10^8$ years to grow a SMBH of mass $10^9 M_\odot$ that has an initial seed mass of $100 M_\odot$. However, such massive black holes are already visible at $z \sim 7$ ($\sim$ 700 million years after the Big Bang), which indicates that there must be a different explanation for their rapid growth.

One possibility is that the seeds are more massive than initially thought, with $M_{seed} \gtrsim 10^3 M_\odot$ (Lodato & Natarajan, 2006; Habouzit et al., 2017; Johnson & Haardt, 2016; Mezcua, 2017). Such high seeds of $M_{\rm seed} > 10^3 M_\odot$ at $z = 40$ are necessary to explain the growth of the SMBH observed at the most distant quasars so far (Bañados et al., 2018). The black hole growth tracks for quasars at $z \geq 7$ can be seen in Figure 2.13. At $z = 30$, we see what mass ranges are required and what mechanism could produce such masses. The minimum seed mass here corresponds to $M_{\rm seed} \sim 10^3 M_\odot$, produced either via dense star clusters or Pop III stars. The seeds cannot be observed directly since we cannot probe anything that has occured before the CMB.

Another theory states that quasars are accreting at low radiative efficiencies of $\eta \lesssim 0.01 - 0.001$ (Volonteri, 2010; Trakhtenbrot et al., 2017; Davies et al., 2019b). This would enable SMBHs to have episodes at super-Eddington accretion rates $\dot{M} > \dot{M}_{edd}$ (Pacucci et al., 2015; Collin et al., 2002).

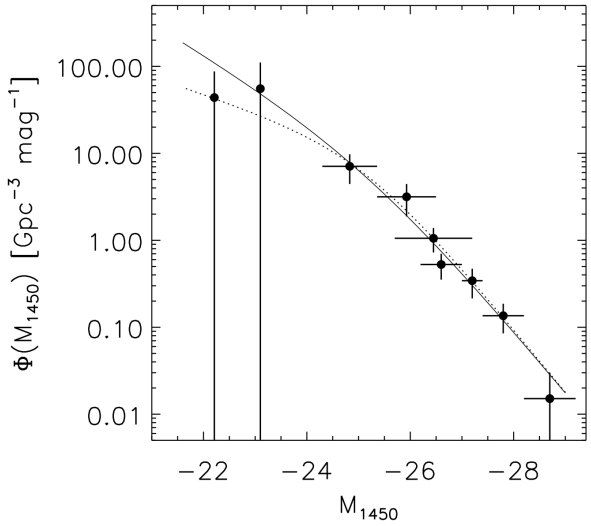


Figure 2.14: The solid line represents the quasar luminosity function at $z \sim 6$ (Jiang et al., 2016), which is fitted using the maximum likelihood method. The points represent the binned SDSS luminosity function, except for the two faintest points which are confirmed quasars (Willott et al., 2010; Kashikawa et al., 2015). The dotted line represents the QLF with a fixed slope $\alpha = -1.5$ (Willott et al., 2010).

2.3.2 Quasar luminosity function

To obtain an estimate of the number of quasars in the Universe, one first needs to know the number of quasars for a given absolute magnitude (or luminosity). This is given by the so-called quasar luminosity function (QLF).

The QLF for $z \sim 6$ is given by the double power-law fit (Jiang et al., 2016)

$$\Phi(M, z) = \frac{9.93 \times 10^{-0.7(z-6)}}{10^{0.4(-1.9+1)(M+25.2)} + 10^{0.4(-2.8+1)(M+25.2)}}, \tag{2.8}$$

where M the absolute magnitude and z the redshift. It has units of Gpc^{-3} mag^{-1}. The QLF is shown in Figure 2.14.

The redshift dependence of the QLF is formulated as

$$\Phi \propto 10^{kz}, \tag{2.9}$$

where k proportional to the steepness of the slope. Note that Jiang et al. (2016) used $k = -0.7$ for the redshift dependence in the stated QLF above. Wang et al. (2018a) find a redshift dependence of $k = -0.78 \pm 0.18$ for $z \sim 6 - 6.7$, which is illustrated on the left side of Figure 2.15 (Fan et al., 2019). Euclid Collaboration (2019) instead use $k = -0.72$ for $z \sim 6$, and up to $k = -0.92$ for $z \gtrsim 6$. This implies that the number of quasars that can be found in the observable universe declines drastically after redshift $z \sim 5 - 6$.

2.3.2.1 Observed numbers

The space density of quasars of absolute magnitude within $[M_1, M_2]$ as a function of redshift is then obtained by integrating the quasar luminosity function over that range,

$$n_Q(z) = \int_{M_1}^{M_2} \Phi(M)\, \mathrm{d}M. \tag{2.10}$$

A general approximation gives $n_Q \sim 20 h_0^3$ Gpc^{-3} (Blandford et al., 1990). Dead quasars are located at a distance of ~ 20 Mpc between each other, which resembles the distance between clusers of galaxies (Lynden-Bell, 1969).

At $J \sim 23$, there are $\sim 2.5 \times 10^{-2}$deg^{-2} quasars at redshifts $z = 7 - 9$, compared to ~ 3.6 deg^{-2} MLT dwarfs and $\sim 1.4 \times 10^3$ deg^{-2} ETGs (Euclid Collaboration, 2019). The quasar number density drops even further at higher redshifts: it is expected that there is only ~ 1 quasar at $z \sim 9$ with $M_{AB} < -26$ in the observable Universe (Fan et al., 2019).

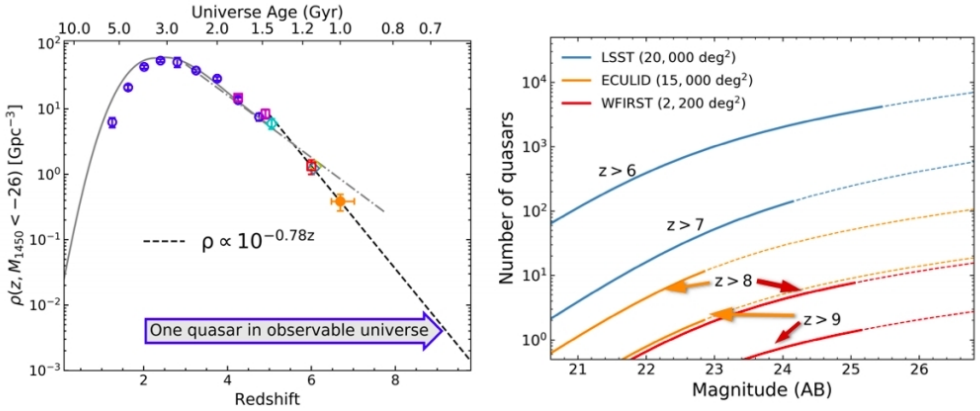


Figure 2.15: Left: The higher the redshift z, the less quasars can be observed due to a decreasing quasar density. The dotted line represents a redshift dependence of $k = -0.78$. It is assumed that there exists only ~ 1 HZQ with $z \gtrsim 9$ that has a billion solar masses. This will be examinable with the Nancy Grace Roman Space Telescope (formerly WFIRST) whose launch date is expected to be no later than May 2027 (Zellem et al., 2022). Right: Number of quasars that can be detected with surveys in the next decades of given magnitude limits. Figure taken from Fan et al. (2019).

Quasars are found to exist earlier than 1 billion years after the Big Bang (Mortlock et al., 2012). At $z \sim 6$ the density of quasars with an absolute magnitude of $M_{AB} \sim -26$ is given by < 1 Gpc^{-3} (Fan et al., 2019). This implies that there exist only a few hundred quasars at that redshift in the observable Universe. The number density drops even further at higher redshifts. There exists only one high-redshift quasar per ~ 200 deg^2 (Mortlock, 2014).

According to Wang et al. (Wang et al., 2018a), there will be only about 1 single quasar with a billion solar masses and magnitude $M < -26$ at a redshift of $z \gtrapprox 9$. This leads to the conclusion that the first quasars formed at a redshift of $z = 9 - 10$ (Fan et al., 2019). This is illustrated in Figure 2.15.

As quasars at high redshifts are extremely rare objects, only a handful of quasars have been discovered at redshift $z \gtrsim 7$. Hence, every quasar that is found is important. As the next decadal surveys arrive, observation of such a quasar will be possible with the Roman Space Telescope whose expected launch date is set in 2027 (Zellem et al., 2022).

2.4 High redshift quasar surveys

Table 2.2 gives a brief summary of past and upcoming surveys, together with their expected number of detected high-redshift quasars.

2.4.1 Discovered quasars

The redshift of discovered quasars ranges from $z = 0.056$ to $z = 7.54$. Over 500,000 quasars have been discovered so far, of which half of them are in the redshift range $z \sim 2.15 - 3.5$ (Pâris et al., 2018). This is in agreement with Figure 2.15. The highest redshift quasars that have been discovered so far can be seen in Table 2.1.

Quasar	Redshift	Survey	SMBH mass
J0313−1806	7.64	Pan-STARRS, DECaLS, UHS, VHS, WISE	$1.6^{+0.4}_{-0.4} \times 10^9 M_\odot$
J1342+0928	7.54	ALLWISE, UKIDSS LAS, DECaLS	$7.8^{+3.3}_{-1.9} \times 10^8 M_\odot$
J1007+2115	7.52	DECaLS, Pan-STARRS, UHS, WISE	$1.5^{+0.2}_{-0.2} \times 10^9 M_\odot$
J1120+0641	7.09	UKIDSS LAS, SDSS	$2.0^{+1.5}_{-0.7} \times 10^9 M_\odot$
J1243+0100	7.07	HSC	$3.3^{+2.0}_{-2.0} \times 10^8 M_\odot$
J0038˜1527	7.02	DECaLS, Pan-STARRS, WISE	$1.3^{+0.25}_{-0.25} \times 10^9 M_\odot$
J0252˜0503	7.00	DES, Pan-STARRS, WISE, UHS, VHS	$1.4^{+0.16}_{-0.16} \times 10^9 M_\odot$

Table 2.1: The highest redshift quasars found to date (Mortlock et al., 2011; Bañados et al., 2018; Matsuoka et al., 2019; Wang et al., 2018b; Yang et al., 2019, 2020; Wang et al., 2021).

The surveys that have contributed the most are the United Kingdom Infrared Telescope Infrared Deep Sky Survey Large Area Survey (UKIDSS LAS), Sloan Digital Sky Survey (SDSS), Hyper Suprime-Cam Subaru Strategic Program (HSC-SSP), The Dark Energy Camera Legacy Survey (DECaLS), The Panoramic Survey Telescope and Rapid Response System (Pan-STARRS) and the Wide-field Infrared

Survey Explorer (WISE). All of these have information in the optical and/ or the NIR wavelength bands.

Survey	SDSS DR12	UKIDSS (LAS)	LSST	Euclid	Roman
Covered area (deg^2)	14,555	4,000	20,000	15,000	> 1,000
Filter bands	u, g, r, i, z	Y, J, H, K	u, g, r, i, z, Y	VIS, Y, J, H	r, z, Y, J, H, F, W
Pixel scale (arcsec)	0.396	0.4	0.2	0.1	0.11
Depth (mag)	$z \sim 20.5$	$Y \sim 20.5$	$Y \sim 22 - 25$	$Y, J, H \sim 24.0$	$F129 \sim 28$
Expected Quasars	$z > 6.5$: ~ 10	$z > 6.5$: ~ 10	$z > 7$: ~ 100	$z > 8$: ~ 10	$z > 8$: ~ 10

Table 2.2: Comparison of past (SDSS, UKIDSS) and upcoming (LSST, Euclid, Roman) surveys (Fan et al., 2019; Euclid Collaboration, 2019; Tee et al., 2023). The predictions of the number of detected quasars stem from using $k = -0.78$ for the QLF at $z \gtrsim 6$ (Wang et al., 2018a).

2.4.2 Upcoming surveys

Table 2.2 and the right plot in Figure 2.15 show the expected number of quasars that might be found at the magnitude limits of the upcoming surveys LSST, Euclid and the Roman Space Telescope. LSST will be able to find quasars with $z \gtrsim 7$ and Euclid will further increase the redshift limit towards $z \gtrsim 8$. Finally, the Roman Space Telescope will have a total of 8 bands in a wavelength range of 0.48 – 2.30 μm with a magnitude limit of $F129 \sim 28$ and $F213 \sim 26.2$ (NASA, 2021). It therefore will be able to discover quasars at even higher redshifts, making it possible to probe redshifts $z \gtrsim 9$.

Redshift	Predicted quasars in Euclid
$7 < z < 7.5$	≥ 100
$7.5 < z < 8.0$	$\simeq 25$
$z \geq 8$	$\simeq 8$

Table 2.3: The predicted number of quasars that will be discovered by the Euclid survey in a given redshift range. Data taken from Euclid Collaboration (2019).

In 2024, Euclid will have its first data release. Euclid will have one visual band VIS and three near-infrared bands Y, J and H. This will allow the search for quasars at redshifts $z > 8$. The depth in the

visual band is $VIS = 24.5$ and the depth in the near-infrared bands is $YJH = 24.0$. It is predicted that Euclid will find of order of 100 quasars at redshift $z < 8$ and of order 10 at redshift $z > 8$ (Euclid Collaboration, 2019). This is summarized in Table 2.3. The goal is to have either LSST or Pan-STARRS covering the wide survey area. Since these two surveys have optical information as well (z band), they allow for the search of quasars $7 < z < 7.5$.

Chapter 3

Finding high-redshift quasars

Unfortunately, high-redshift quasars remain difficult to find as they are rare and can be confused with other astronomical sources. Contaminants of quasars at $z \gtrsim 6.5$ are low-redshift cool stars such as M stars and LTY dwarfs with a peak of their SED in the infrared (Warren & Hewett, 2002; Stern et al., 2007; Wang et al., 2016), and early type galaxies (ETGs) at $z = 1 - 2$ (Euclid Collaboration, 2019). Finding high-redshift quasars is hence a true needle-in-a-haystack problem. Fortunately, high-redshift quasars are observationally distinct from their more numerous contaminants. One wants to sift out as many contaminants to pin down possible quasars from initially available survey data. In order to do this, different techniques that test a range of physical properties have to be applied.

This often starts with photometric information in multiple bands from wide surveys in the optical and/or NIR (Sandage & Wyndham, 1965). This information is then used to try to reconstruct the SED of a source of interest. If measurements of a source are available over several epochs, then this gives additional information on the variability (Graham et al., 2014; Peters & Richards, 2015) of the source.

One can further make use of astrometry, which tries to accurately measure the position of stars and other objects. This involves correcting for several factors that can falsify the result such as atmospheric conditions and imperfections in telescopes. By observing a source over several epochs, one can then deduce its proper motion.

Finally, morphological analysis on image data gives rise to the general shape of the object. For instance, it can tell us whether the source is a point source or an extended object.

These methods can be used sequentially. The final pool of candidates should then only contain sources that resemble high-redshift quasars at $z \gtrsim 6.5$.

	Proper motion	*Compact*	*Variable*	*Radio source*
MLT dwarf	Yes	Yes	No	No
ETG	No	No	No	No
AGN	**No**	**Yes**	**Yes**	**Yes**

Table 3.1: Properties of MLT dwarfs, early type galaxies and AGN.

3.1 Distinguishing features of high-redshift quasars from contaminants

In the following section we explain the distinguishing features that high-redshift quasars $z \gtrsim 6.5$ exhibit compared to other astronomical objects such as stars and galaxies. This is summarised in Table 3.1.

High-redshift quasars have a distinctive spectral energy distribution that separates them from most other astronomical objects. However, as spectroscopy is time intensive, we usually have to work with photometric data instead. In the optical-near-infrared colour space the distinction between high-redshift quasars and contaminants is still visible but less clear.

Due to the large distances of high-redshift quasars, they appear to be stationary. Further, since they are able to outshine their host galaxy, they have a star-like appearance. Quasars can be variable from very short to very long timescales depending on the size of the emitting region. A small subset of quasars are radio sources as well. We will explain each of these items in detail below. Any combination of the above characteristics can be applied in the search for high-redshift quasars, but in most cases they must be used in a sequence which depends on the available data.

3.1.1 Dark in the optical and red in NIR

As explained in Section 2.2.1, the SED of high-redshift quasars shows a Gunn-Peterson trough. This trough is expected to only show for Lyα emitters. Further, as explained in Section 2.1, quasars emit both thermal and non-thermal radiation. Typically, stars are neither Lyα emitters nor do they emit non-thermal radiation.

Stars are approximate blackbodies and as such follow a (thermal) blackbody distribution. Blackbody radiation is explained in Section 1.3.2.1, and is illustrated in Figure 1.5. Their SED and peak is mainly determined by their absolute temperature. Cool stars have a peak in the (near-)infrared whereas hot stars have their peak in the UV (Warren & Hewett, 2002). At redshifts similar to high-redshift quasars, the peak of hot stars will have shifted into the NIR as well. However, current technology is not able to pick up such a faint signal at high redshifts. Therefore, the contaminants for high-redshift quasars are cool stars located in our galaxy. An example of such contaminants are M stars and LTY dwarfs with a

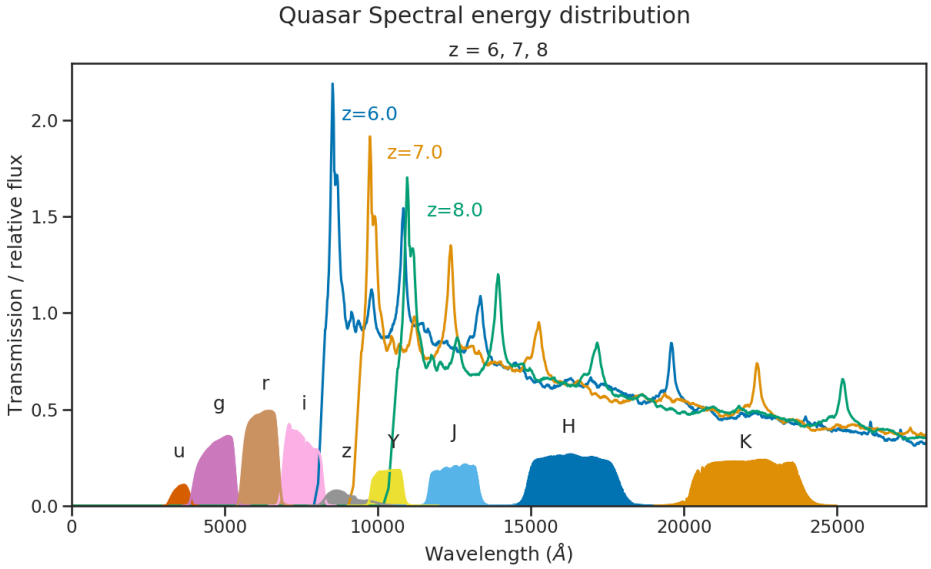


Figure 3.1: Spectral energy distribution of a quasar redshifted to $z = 6, 7, 8$ together with the transmission of the SDSS (u, g, r, i, z) and UKIDSS LAS (Y, J, H, K) filters. The higher the redshift of the quasar, the more the Lyα emission line shifts towards the infrared. Figure adapted from Mortlock (2014).

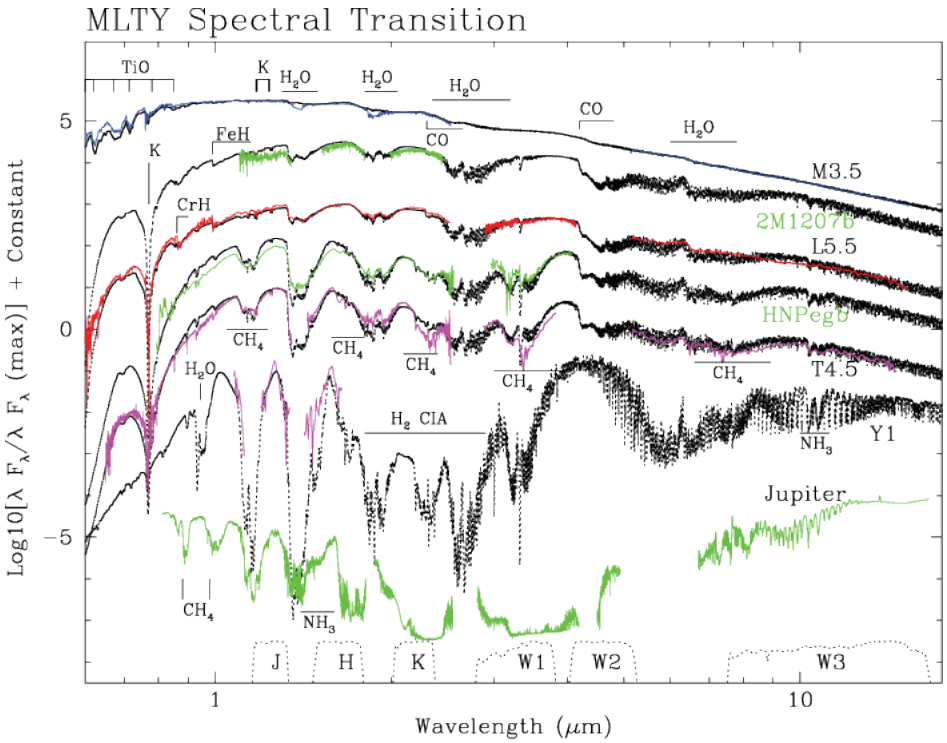


Figure 3.2: The SEDs of M stars and LTY dwarfs, together with WISE filters J, H, K, W_1, W_2, W_3. Figure taken from Allard (2014).

peak in the infrared. Their SEDs together with the WISE filter transmissions are shown in Figure 3.2. M stars are the hottest in that group with surface temperatures of $T \sim 2500$ K (Allard et al., 2003). M dwarfs have temperatures of $T \sim 1800$ K, and even colder T dwarfs have temperatures of $T \sim 1000$ K. Y dwarfs are the coolest with temperatures $T \sim 500$ K.

Figure 3.1 displays the quasar SED at redshifts $z = 6-8$ together with the band transmissions of SDSS (u, g, r, i, z) and UKIDSS (Y, J, H, K) in the optical and NIR. This shows that for different redshifts, the quasar will be brightest in the passband where the Lyα emission line falls into. The higher the redshift of the quasar, the longer the wavelength of the observed Lyα emission line. If we compare this to Figure 3.2, we can see that MLT dwarfs do not exhibit any such trough. However, in broad-band filters the distinction becomes less clear, which can be seen in Figure 3.3. This shows the SED of a $z = 7$ quasar and a T8 dwarf, together with the transmission filters of SDSS (i, z) and UKIDSS (Y, J, H). They both exhibit (near-)zero flux in the optical, and a large flux in the NIR. Even though their SED is clearly different, the flux in the broad-band filters can look similar.

At low signal-to-noise-ratios, the colours of early type galaxies at $z = 1 - 2$ can also resemble those of high-redshift quasars (Euclid Collaboration, 2019).

In order to separate these contaminants from high-redshift quasars, one needs to look at their colours in at least three filters (Weedman, 1986). This can be seen in Figure 3.4, in which Euclid Collaboration (2019) plot the colour tracks of quasars (green), MLT dwarfs (red) and ETGs (blue) using three filters each. The quasars are shown at $z \geq 7$, the MLT dwarfs for types between M0 to T8, and the ETGs are shown between redshift $z = 1$ and $z = 2$. Looking at the right plot, we can see that both MLT dwarfs and ETGs are somehow separable from high-redshift quasars but do have similar colours in $Y - J$ and $J - H$. optical $- Y$, shown in the left plot, can differentiate between high-redshift quasars at $z \geq 7$ and contaminants. This is due to the Gunn-Peterson trough, which gives a large difference in flux between the optical (zero flux) and the NIR bands for quasars at $z \geq 7$. These quasars would hence be expected below the plot, at optical $- Y > 3$. Quasars at $z < 7$, however, would have similar colours, as can be seen for $\hat{i} - \hat{Y}$ in the colour-colour diagram for quasars and MLT dwarfs in Figure 3.5. Hence, for quasars at redshifts $z \sim 6 - 7$, they can be discerned from contaminants by having photometric information in the three filter bands of i, Y and J.

This shows the main drawback of using colours as a selection technique: quasar colours vary a lot as a function of redshift due to the Lyman α emission line being shifted. Referring again to Figure 3.1, $z - i$ is expected to be very large for quasars at redshift $z \sim 6$, $Y - i$ should be very large for redshifts $z \sim 7$ and so on. At about redshift $z \gtrsim 6.5$ the Ly α line shifts outside of the optical into the infrared bands. Therefore, one usually separates quasar searches depending on the broadband filters that are available from a survey:

- $z \lesssim 6.5$: Optical surveys
- $z \gtrsim 6.5$: Near-IR surveys

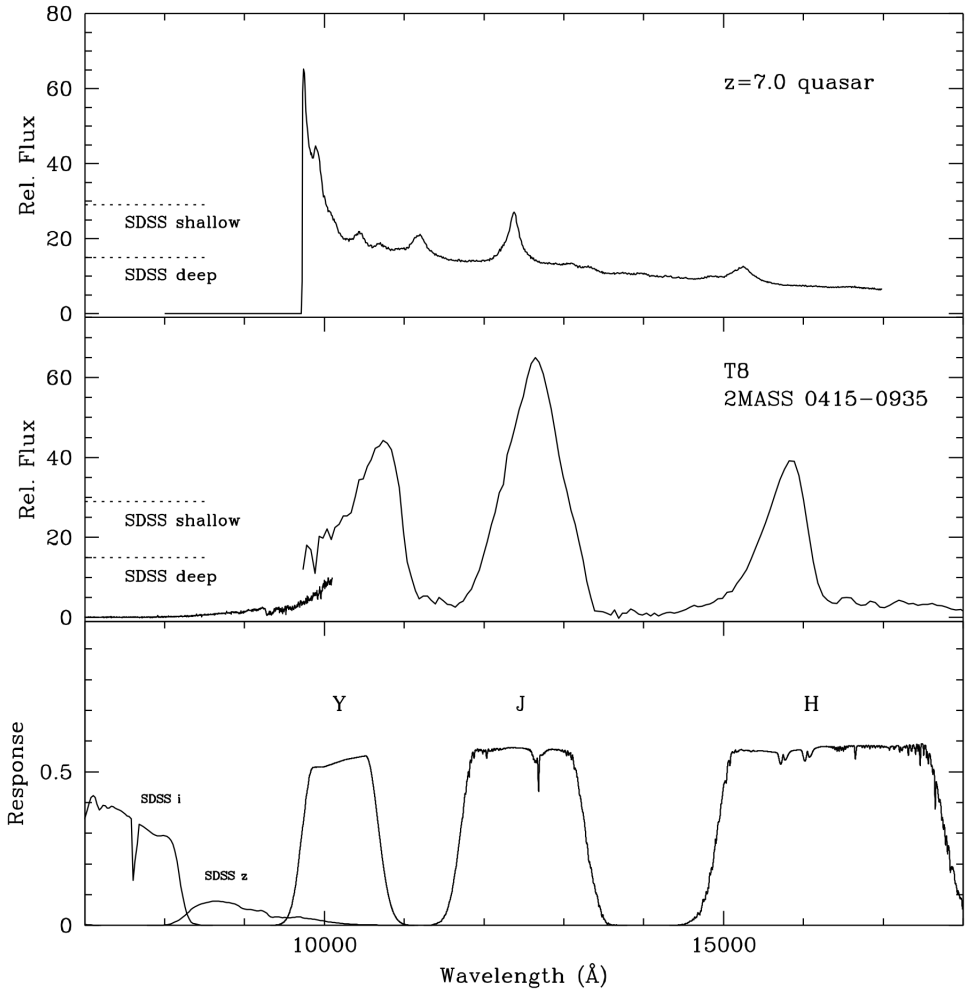
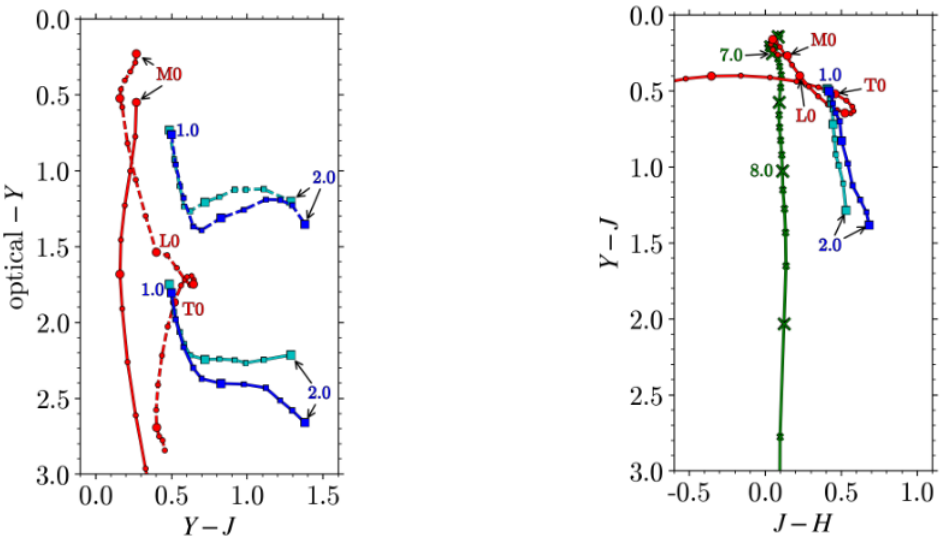


Figure 3.3: The first two rows show the relative flux of a modelled $z = 7$ quasar and a T8 dwarf. The third row shows the transmission bands for SDSS (i, z) and UKIDSS (Y, J, H). Figure taken from Chiu et al. (2007).



(a) Colours $Y - J$ and optical $- Y$.

(b) Colours $J - H$ and $Y - J$.

Figure 3.4: Colour tracks of quasars $7 \leq z \leq 9$ (green), MLT dwarfs (red) and ETGs (blue). Filters Y, J, H are taken from Euclid. The optical filter is plotted for both Euclid (solid) and ground-based telescopes (dashed). In the left panel, the quasar track starts at optical $- Y > 3$ and is therefore not visible. Figures taken from Euclid Collaboration (2019).

It is essential to detect both the absorption trough and the emission of quasars to differentiate them from contaminants, which is why both optical and NIR data is needed to classify them.

For quasars at a fixed redshift their colour remains relatively constant and enables to apply colour cuts on a given dataset. That is why it is necessary to initially define a redshift range in which one wants to search for quasars.

3.1.2 Point-like objects

As mentioned in Chapter 2, quasars are compact with a size of ~ 2 AU, and can outshine their host galaxies. This makes quasars look like point sources in the sky, which gives them a star-like (quasi-stellar) appearance (Peterson, 1997; Krolik, 1999).

The appearance of high-redshift quasars is different to that of galaxies such as ETG contaminants, which have a more extended (elliptical) morphology (e.g., Almeida et al. (2007)). However, faint ETGs at $J > 22$ may look very compact and hence can be mistaken for point sources (Euclid Collaboration, 2019), with a size of $R_{\text{eff}} \propto (1 + z)^{-1.48}$ that is dependent on redshift (van der Wel et al., 2014). Therefore, ETGs at $z = 2$ will look more compact than ETGs at $z = 1$.

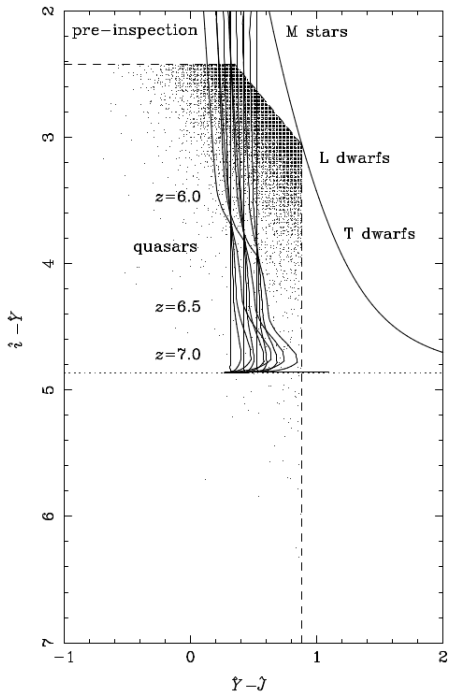


Figure 3.5: Colour-colour diagram for $(\hat{Y} - \hat{J})$ vs. $(\hat{i} - \hat{Y})$. Twelve quasar line models can be seen up to redshifts $z = 7$. The curve on the right represents the colour model of MLT dwarfs. Figure taken from Mortlock et al. (2012).

3.1.3 Stationary sources

Proper motion defines the angular velocity that an astronomical object exhibits relative to the isotropic expansion of the universe (or Hubble flow), as observed from the centre of mass of the Solar System (Lépine et al., 2003; Ghez et al., 1998; Eckart & Genzel, 1996). It is different to the parallax effect. For proper motion, the object intrinsically moves consistently in one direction, whereas for the parallax effect, the object *appears* to move back and forth between two positions. This is illustrated in Figure 3.6, which shows the angular proper motion μ and the respective peculiar velocity v_p at a distance d. The peculiar velocity consists of the tangential velocity v_t and the radial velocity v_r. The larger the distance, the smaller the proper motion for the same tangential velocity.

All extra-galactic sources have a non-zero peculiar velocity. Galaxies have peculiar velocities of $\sim$ 200 km/s (Carlberg et al., 2001; Karachentsev et al., 2006), sometimes up to $\lesssim$ 1000 km/s (Girardi et al., 1993; Sheth & Diaferio, 2001; Anand et al., 2019). Stars within galaxies have stellar velocities of $\sim$ 100 − 300 km/s (Rieke & Rieke, 1988; Walker et al., 2009; Anguiano et al., 2020). However, at very large distances d, this translates to a near-zero proper motion μ. Objects at high redshifts are so far away that no apparent proper motion is observable (and hence no peculiar velocities can be derived) (Hogg, 1999; Weinberg, 1972). Therefore, high-redshift quasars $z \gtrsim 6.5$ appear to be stationary, and hence any candidate that has moved in a survey can be discarded (Lang et al., 2009;

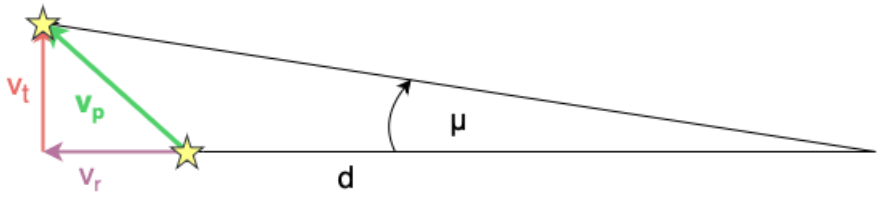


Figure 3.6: Proper motion μ, distance d and peculiar velocity v_p. The larger the distance, the smaller the proper motion for the same tangential velocity.

Richards et al., 2009; Heintz et al., 2018; Shu et al., 2019). Quasars therefore are ideal reference points as they are point-like objects and do not show any proper motion (Bachchan et al., 2016). The standard reference frame in astronomy is the International Celestial Reference System (ICRS) (Ma et al., 1998, 2009; Fey et al., 2015), which is defined by extragalactic radio sources such as quasars. Gaia (Gaia Collaboration et al., 2016) is the optical realisation of the ICRS (Bachchan et al., 2016).

Stars are too faint to be detected at such high redshifts. We can only detect local stars which are close enough that we do detect significant proper motions in them, especially near the Galactic Centre (Eckart & Genzel, 1996). They will be detected as moving sources in survey images that have been taken some time apart. Stars can have proper motions that exceed 0.5 arcsec / year (Lépine et al., 2003). Nearby galaxies can have proper motions of $\sim 0.2 - 2.0$ marcsec / year (Tinney et al., 1996; Piatek et al., 2003; Simon, 2018; Massari & Helmi, 2018). However, for ETGs at $z = 1 - 2$ this proper motion also becomes zero. Therefore, high-redshift quasars and ETGs do not show any proper motion, but local MLT dwarfs do. This method of sifting out contaminants based on their non-zero proper motion will become very useful for the upcoming LSST survey. LSST is a multiple epoch survey that will map the entire sky every few nights and as such enables proper motion studies.

Lang et al. (2009) use multi-epoch data from the SDSS Southern Stripe and apply an analysis on proper motion to successfully distinguish high-redshift quasars from brown dwarf contaminants. Heintz et al. (2018) classified quasars as objects having zero proper motion in Gaia DR2 with $18 < G < 20$ mag and high Galactic latitudes, achieving $\sim 75\%$ completeness. This is illustrated in Figure 3.7.

MacMillan (2005) has shown that quasars observed in the radio do have proper motions greater than 50 μarcsec per year due to source structure variations. This is because quasars can have temporal changes in emission structures which can have an effect on their apparent position. Figure 3.9 illustrates the proper motion of extragalactic radio sources that move > 50 μarcsec/year as a function of redshift.

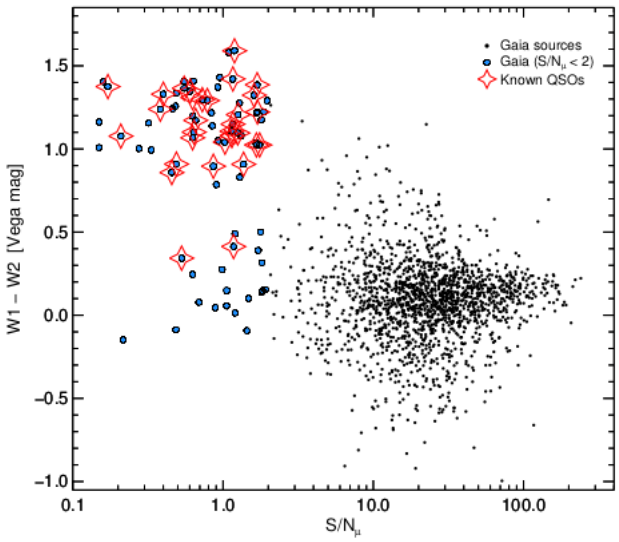


Figure 3.7: Signal-to-noise ratio vs $W1 - W2$ colours of Gaia point sources. (black) Gaia sources with magnitudes $18 < G < 20$ mag and proper motions. (blue) Gaia point sources with zero proper motions (have not been confirmed as quasars yet). (red) Confirmed quasars (Gaia point sources with zero proper motion cross-matched with SDSS-DR14Q (Pâris et al., 2018)). Figure taken from Heintz et al. (2018).

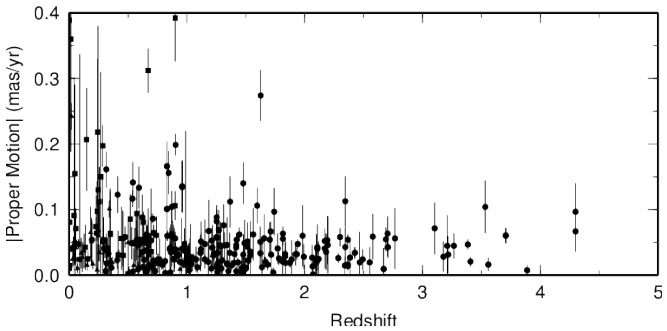


Figure 3.8: Redshift vs. proper motion of 355 moving quasars > 50 μarcsec/year observed in the radio. Figure taken from MacMillan (2005).

Figure 3.9 shows how the apparent position of quasar $4C39.25$ has changed over a period of 20 years as measured by VBLI. The radio source is thought to have changed its apparent position due to variations in structure (Fey et al., 1997). That is, the number and intensity of its compact components in the radio changed over that period.

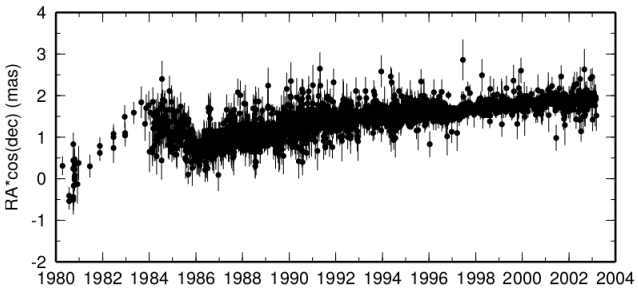


Figure 3.9: The change in right ascension for quasar 4*C*39.25. Figure taken from MacMillan (2005).

Since we do not observe our candidates in the radio, we do not have to take this into account. We observe in the optical and NIR, where we can assume zero proper motion.

3.1.4 Radio source

As explained in Section 2.1.4, up to 20% of quasars are bright radio sources due to relativistic jets. Compared to radio-loud quasars, neither MLT dwarfs (Williams, 2018) nor ETGs (Tantalo et al., 2010) exhibit such powerful radio emission. Therefore, by observing sources in the radio one can separate radio-loud quasars (and radio-quiet quasars with significant radio emission) from contaminants.

3.1.5 Variability

As explained in Section 1.2.2, quasars are variable with timescales up to several years, with the highest variability emerging from regions close to the central nucleus. Contaminants of high-redshift quasars are rather static in comparison. And when contaminants are variable, they do not show such drastic differences in magnitude. Variability in MLT dwarfs arises from star spots on the surface of the dwarf (Artigau, 2018). Due to their less extreme environment compared to quasars, this variability is very small. In order to detect a magnitude difference of > 0.1 mag, a spot coverage of $\gtrsim 50\%$ would be required (Goulding et al., 2012). Further, due to the smaller size of MLT dwarfs compared to quasars, the variability of MLT dwarfs is seen on smaller time scales from a few hours up to a several months (Artigau et al., 2009). Any variability in an ETG would be due to individual variable (old) stars within the galaxy, which becomes impossible to detect at $z = 1 - 2$.

3.2 Data analysis mechanisms on selected sources

Visual inspection is practical when one has a small pool of candidates. However, upcoming surveys such as Euclid and LSST will provide billions of sources, for which visual inspection becomes infea-

sible. We therefore need automated data analysis mechanisms that cut down the number of possible quasar candidates from large sample sets. This reduces the amount of required visual inspection.

The typical data analysis tools are the following:

- Heuristic cuts
- Bayesian statistics
- Machine learning

Heuristic approaches use educated guesses to decrease the sample sizes, where one trades accuracy for speed. Once the rules are set, the results are reproducible. The advantage of heuristic approaches is that they are quick and relatively easy to implement, whereas using statistical selection criteria are more time consuming.

Bayesian inference and machine learning are explained in detail in Chapters 4 and 5.

3.2.1 Colour cuts

A simple yet effective method is to apply criteria on colour and magnitudes, e.g. using heuristic linear cuts in colour space (Koo & Kron, 1982; Warren et al., 1994; Fan et al., 2001; Richards et al., 2002; Willott et al., 2007; Venemans et al., 2007, 2013; Bañados et al., 2016). Colour cuts are easy to reproduce and easy to interpret due to their lack of complexity. As described in Section 3.1.1, the specific cuts will depend on the redshift range you are interested in because of the wavelength shift of the Lyα emission line. For example, Venemans et al. (2013) used colour cuts in the optical and NIR to successfully detect quasars at $z > 6.5$. The conservative set of colour cuts were $Z - Y \geq 1.25$ AND $-0.5 < Y - J \leq 0.5$ AND ($Z > Z_{\mathrm{lim},3\sigma}$ OR $Z - Y > 0.75 \times (Y - J) + 1.45$) AND $-0.5 < Y - K < 1.3$ AND undetected in u, g, r, i, which is illustrated as the dash-dotted line in Figure 3.10. However, colour cuts can introduce biases in the selected data set (Wagenveld et al., 2022). Using broad colour cuts, the number of false positives can be high, especially over large observing areas. For example, at redshifts $z \simeq 6.5$, objects such as MLT dwarfs can look quite similar to quasars in colour space. The more conservatively one sets the colour cuts, the more quasars will potentially be missed. This is a tradeoff between completeness and efficiency. Further, colour cuts typically do not take into account the uncertainties of the measured photometry, and hence a source with a low signal-to-noise-ratio S/N can end up getting treated the same way as a high S/N source. In addition, there is no differentiation between sources closer to the edges of the cuts and the more interesting sources that lie exactly on defined quasar colour tracks (Mortlock et al., 2012). A probabilistic approach might hence be better suited.

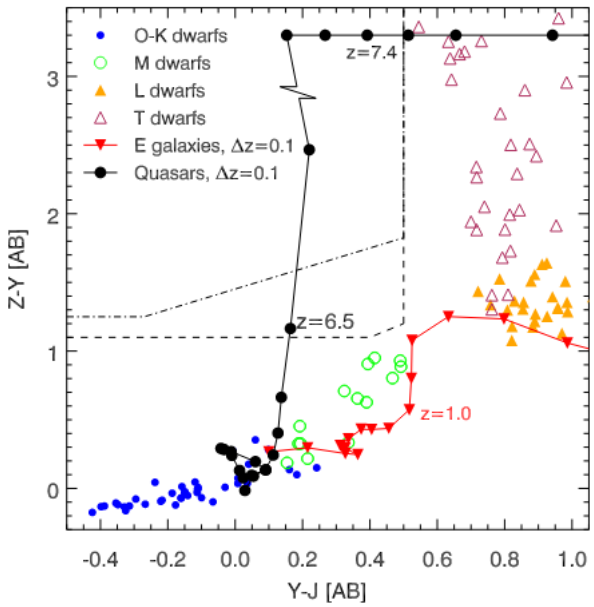


Figure 3.10: Colour cuts to distinguish high-redshift quasars from contaminants. While the dash-dotted line shows conservative colour cuts, the dashed line shows more extended cuts. Figure taken from Venemans et al. (2013).

3.2.2 Bayesian model comparison

The common idea behind Bayesian selection techniques for detecting high-redshift quasars is to have a prior that takes into account the fact that quasars are rare compared to stars. The likelihood then includes information about the brightness in the measured filters to account for the scatter in the quasar population. Bayesian inference has been proven to be a powerful tool for screening quasar candidates (Mortlock et al., 2012; Bailer-Jones & Smith, 2008; Richards et al., 2015; Pipien et al., 2018). It has led to the discovery of a $z \gtrsim 7$ quasar (Mortlock et al., 2011).

Given a set of population models, Bayesian model comparison retrieves a quasar probability based on which population model fits the observations best, without having to calculate the marginal distribution. Richards et al. (2004) use Bayes's rule based on four color measurements to determine the probability that a source is a quasar, using kernel density estimation (KDE) to obtain the likelihood. However, Richards et al. (2004) do not take into account measurement uncertainties. This is achieved by Mortlock et al. (2012), who incorporate photometric uncertainties in their Gaussian likelihood calculation. They further include a prior distribution for the population density. This is especially useful when the densities vary drastically between the populations, as is the case for quasars and contaminants. This technique has been applied in e.g. Kirkpatrick et al. (2011); Matsuoka et al. (2016, 2018); Euclid Collaboration (2019); Barnett et al. (2021), and is used in this work as well (see Section 7.1). Bovy et al. (2011) use extreme deconvolution to model the underlying likelihood as a sum of Gaussians, called Gaussian mixture modelling (GMM), which subsequently has also been applied in e.g. Bovy et al. (2012); Nanni et al. (2021).

Using a Gaussian likelihood further allows for the method of minimum chi-squared model fitting (Reed et al., 2017, 2019). Minimum chi-squared model fitting tests how well an observation fits to a pre-defined model rather than assigning probabilities. This is also used in this work, and is described in Section 7.2.

3.2.3 Machine learning

Novel machine learning techniques can improve the classification performance (Yèche et al., 2010; Doert & Errando, 2014; Carrasco et al., 2015; Hernitschek et al., 2016; Schindler et al., 2019; Kang et al., 2019; Bailer-Jones et al., 2019; Ma et al., 2019; Clarke et al., 2020; Wenzl et al., 2021). However, they come at the cost of acting as black boxes whose results can be difficult to impossible to interpret, and are typically not reproducible.

Moreover, machine learning has not been applied to quasars at redshifts $z \gtrsim 6.5$ yet, quite likely due to the small number of confirmed quasars at those redshifts. This is because machine learning needs large data sets to train on, which in that case is not given.

From a Bayesian point of view, the prior and therefore the obtained posterior point estimate are dependent on the training set. Bailer-Jones et al. (2008) are able to include the prior of quasars and their contaminant populations such that they do not have to adapt the training set. Wagenveld et al. (2022) applies methods from Mortlock et al. (2012) and Bovy et al. (2011) to obtain a GMM-likelihood using machine learning, while also including informed priors. Shin et al. (2022) combine the Bayesian information criterion (BIC) with deep learning to detect quasars at $z \sim 5$.

3.3 Our approach

We describe the steps that are taken to classify possible high-redshift quasars, as is illustrated in Figure 3.11. We then explain how we adapt the Bayesian model comparison and visual inspection steps (orange in Figure 3.11) from Mortlock et al. (2012), and how we augment and optimise them with machine learning (yellow). The set of candidates is taken from Mortlock et al. (2012), which are defined via their location in right ascension and declination.

The classification steps are the following:

1. Images: Retrieve all images of a candidate from a survey.
2. Catalogues: Query information about the source (such as photometry) from the database.
3. Colour selection: Broadly sift out contaminants by applying colour cuts.

4. Bayesian model comparison: Analyze if a source is a quasar candidate or a contaminant; rank it by their probability of being a quasar.
5. Visual inspection: Inspect the most probable quasar candidates.
6. Follow-up imaging: Retrieve more accurate photometry; feed it back into the model comparison.
7. Follow-up spectroscopy: Spectroscopy often enables the final classification of a candidate.

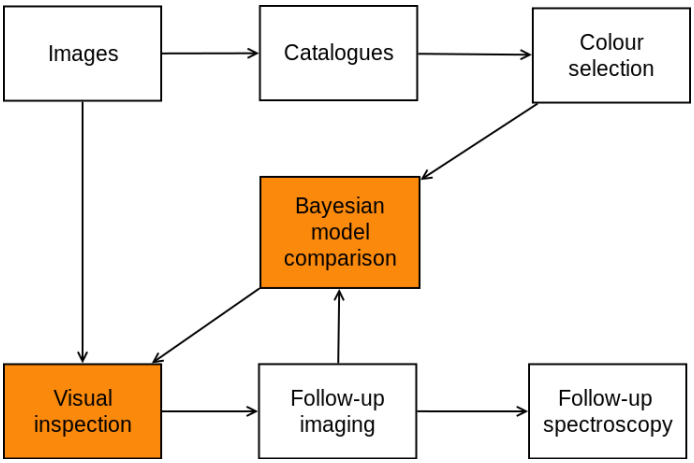
By only taking simple color cuts one is not able to reduce the amount of possible quasar candidates by enough to undertake any follow-up observations in a short period of time. Colour cuts are therefore used as the initial selection stage to broadly get rid of contaminants, while the next stage is to analyse these remaining candidates via statistical methods. By ranking the candidates based on their probability, the most promising candidates can then be visually inspected. Upon the arrival of new photometry from follow-up imaging, the probabilities are adapted. This refinement step is repeated iteratively until only a few hundred quasar candidates are left. Lastly, follow-up spectroscopy is needed to finally classify the remaining candidates.

In this project, we develop two metrics to help distinguish quasars from contaminants and therefore reduce the need of visual classification. First, we extend the Bayesian model comparison method described in Mortlock et al. (2012) and Euclid Collaboration (2019) to calculate a probability that the source is a quasar from its photometry. However, photometry alone does not test the morphology or stationarity of a source. We therefore also generate a model of a stationary point-source that follows the distinct SED of a quasar on the pixel level, which is tested via chi-squared on image cutouts. This now tests against moving sources, extended objects and sources whose SED is a poor fit to quasars.

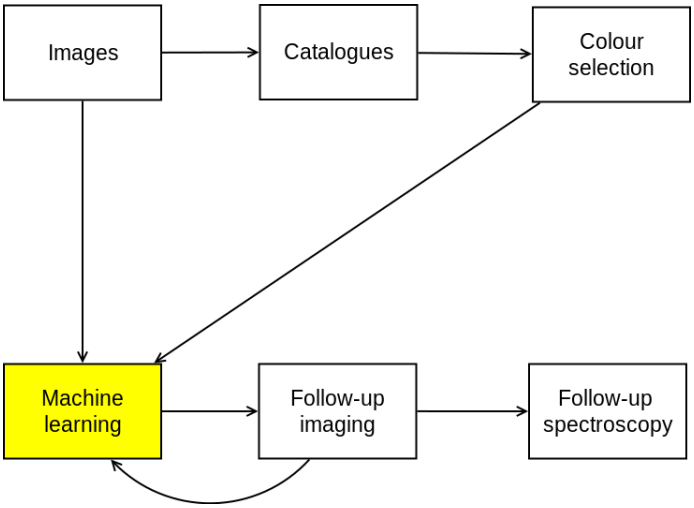
We further develop an automated data processing pipeline (Chapter 6), where the input consists of the nominal source position and the output is given by a summary of extracted features including calculated PSF photometry, chi-squared value and quasar probability. These features then serve for defining manual cuts in the feature space (Chapter 7), and/or as an input for the machine learning classifiers (Chapter 8) to further narrow down the pool of candidates.

We develop two different ML methods. One set of the ML classifiers consists of a decision tree and a random forest that are trained on the features from the data processing pipeline (Chapter 8). We also implement a a convolutional neural network that only considers the raw image data instead and does not make use of the results of our pipeline (Chapter 9).

This requires less human intervention, increases performance and therefore enables classification to occur much more efficiently. Some visual inspection is still needed to look at candidates whose final classification is different to the label that was given by Mortlock et al. (2012). If one were to run the classifiers on the full SDSS/UKIDSS dataset, one would only need visual inspection on a small subset to validate the classifiers. The final conclusion is presented in Chapter 10.



(a) After applying colour cuts, Bayesian model comparison is used to obtain ranked probabilities of the candidates. The most promising candidates are then visually inspected. Successful candidates undergo follow-up imaging and the updated photometry is fed into the Bayesian model again.



(b) In this project we exchange the *Bayesian model comparison* and *visual inspection* steps with machine learning. This will allow for an automatic way to sift out quasars from contaminants, enabling to observe far more candidates.

Figure 3.11: The process of classifying quasars. Starting from images of a given survey, we find sources in given catalogues. As a next step, colour selection methods are applied. This still leaves us with too many candidates. Before using follow-up spectroscopy, some additional selection stage has to be applied. (a) illustrates the method that has been used previously by (Mortlock et al., 2012), using Bayesian inference methods to rank the candidates by their probability before using any visual inspection. (b) shows the new approach that we want to take in this project: we want to augment these two steps with automatic machine learning techniques. Follow-up imaging can be used iteratively with both techniques to get rid of obvious contaminants.

Chapter 4

Bayesian inference

Bayes' rule is a powerful tool in statistics, and the number of papers that include Bayesian statistics increases drastically each year (Hackenberger, 2019). Thomas Bayes developed this new mathematical idea at the end of the 18^{th} century (Bayes & Price, 1763). It relates different conditional probabilities to describe the probability of occurrence of an event given some prior knowledge (Lambert, 2018). Laplace independently discovered Bayes' rule in 1774 (Laplace, 1841), which he refined later in 1785 (Laplace, 1912).

In Bayesian statistics, probabilities are expressions of subjective beliefs that allow to be updated upon arrival of new evidence. In contrast to that, classical/frequentist statistics is based on repetitions of an experiment, e.g., flipping a coin and counting how many times 'heads' and 'tails' were observed to obtain the probability. These repetitions are not necessary in Bayes' approach. In fact, no coin has to be flipped to arrive at a probability for 'heads' or 'tails'.

If we knew exactly that we had a fair coin, then neither frequentist statistics nor Bayesian inference would be needed, as we would know that the probability of 'heads' or 'tails' is 0.5. However, typically we do not know the exact underlying process and make approximations or ignore information we do not regard as important. This then requires a statistical approach to solve a problem. In addition, a set of measured data points almost always contain some uncertainty (Bolstad, 2017). Statistical inference takes into account these uncertainties in our measurements, and allows us to increase our knowledge based on the available data. A detailed discussion on Bayesian vs frequentist statistics can be found in appendix B.

4.1 Bayes' theorem

4.1.1 Intuition

An analogy to Bayes' rule is that of *cause* and *effect*, although there does not have to be a causal link in order for Bayes' rule to be applicable. The idea is that one can go from $P(effect|cause)$ to $P(cause|effect)$ and vice versa (Lambert, 2018). That is, one can arrive at a cause by observing its effect. The concrete formula is given by

$$P(cause|effect) = \frac{P(effect|cause)P(cause)}{P(effect)} \tag{4.1}$$

and is fully explained in Section 4.1.2.

4.1.1.1 Prior belief

The probability $P(cause)$ corresponds to our initial belief in the *cause*.

For example, let us say we are at a magician's show and the magician asks us to flip his or her coin. We might be right in assuming that this coin is not fair, and hence that the coin has a different probability to get 'heads' compared to 'tails'. The magician also tells us that he/she can make the coin to show 'heads' just by looking at it while we throw the coin. Our prior belief therefore might be that both sides of the coin are 'heads', and hence assume a probability of 1 that the coin comes up as 'heads'.

This belief is then tested against the observed data. E.g., we observe whether the coin shows up as 'heads' or 'tails' in the end. If it shows up as 'heads', we might feel confirmed in our initial belief. If it comes up as 'tails', however, we might revert our initial belief and assume that the coin is fair and that the magician did not manage to apply his/her trick.

4.1.1.2 How likely is that belief correct?

$P(effect|cause)$ is the probability of observing the *effect* given our initial belief in the *cause*. Going back to our example of flipping the coin for the magician and the coin ends up turning up as 'tails', then this goes against our initial belief that the coin is crooked in that both sides are 'heads'. In this case, $P(effect|cause)$ corresponds to $P(tails|both\ sides\ of\ coin\ are\ 'heads')$. As this is impossible, it would give a probability of zero based on our initial belief.

4.1.1.3 How likely is it that we observe this effect?

$P(\textit{effect})$ is the probability of an (observed) effect, given the set of possible causes. In our case, the effect corresponds to the coin showing up as 'tails'. Let us say that there are only three possible coins one can have: a fair coin, a coin where both sides are 'heads', and a coin where both sides are 'tails'. Independent of the magician, let us further assume that overall the probability of having a fair coin is 0.8, and that that of having an unfair coin is 0.1 in each case.

In the case of a fair coin where one side is 'heads' and the other one 'tails', the probability is $\frac{1}{2}$ that we receive 'tails' as a result. Now we need to multiply this by the probability that one has such a coin. Therefore, we get $\frac{1}{2} \cdot 0.8 = 0.4$. In the case of an unfair coin with both sides being 'heads', the probability is 0 that we receive 'tails' when the coin can only show 'heads'. The result for this case is therefore zero. In the case of an unfair coin with both sides being 'tails', the probability is 1 that we receive 'tails' when the coin can only show 'tails'. This gives a value of $1 \cdot 0.1 = 0.1$. In total, this gives a probability of receiving 'tails' of $0.4 + 0 + 0.1 = 0.5$. Therefore, the probability of having a coin that shows 'tails' is a half.

4.1.1.4 Given the observed event, how likely is the initial belief true?

$P(\textit{cause}|\textit{effect})$ asks how likely our initial belief (of the cause) is true given that we have observed the coin flip (the effect). Since $P(\textit{effect}|\textit{cause})$ is zero in our example, the probability of $P(\textit{cause}|\textit{effect})$ is zero as well. This means that there is zero chance that the coin consists of two 'heads' sides if we observe the coin to display 'tails'. Intuitively, this makes sense as well since it is impossible to have a coin that can only show 'heads' that then suddenly shows 'tails' instead.

4.1.2 Bayes' rule

Bayes' rule is given by (Bayes & Price, 1763; O'Hagan, 1994; Lambert, 2018)

$$p(\boldsymbol{\theta}|\mathbf{d}, M) = \frac{L(\mathbf{d}|\boldsymbol{\theta}, M)\pi(\boldsymbol{\theta}|M)}{m(\mathbf{d}|M)}, \tag{4.2}$$

where the parameters $\boldsymbol{\theta} = (\theta_1, .., \theta_K) \sim \pi(\boldsymbol{\theta}|M)$ are random variables and M our data model. The probability $L(\mathbf{d}|\boldsymbol{\theta}, M) = P(\mathbf{d}|\boldsymbol{\theta}, M)$ represents the so-called *Likelihood*. The density of the parameter random variables $\boldsymbol{\theta}$ is given by $\pi(\boldsymbol{\theta}|M) = P(\boldsymbol{\theta}|M)$ and is called the *Prior* distribution. The *Marginal* $m(\mathbf{d}|M) = P(\mathbf{d}|M)$ represents the probability of obtaining data $\mathbf{d}$ given the model M. Finally, the *Posterior* $p(\boldsymbol{\theta}|\mathbf{d}, M) = P(\boldsymbol{\theta}|\mathbf{d}, M)$ gives the probability distribution after our prior belief has been updated with new data.

Bayes' rule follows from the mathematical theorems of probability,

$$P(\boldsymbol{\theta}, \mathbf{d}) \quad = \quad P(\boldsymbol{\theta}|\mathbf{d})\, P(\mathbf{d}) = P(\mathbf{d}|\boldsymbol{\theta})\, P(\boldsymbol{\theta}). \tag{4.3}$$

After re-arranging the above via $\cdot \frac{1}{P(\mathbf{d})}$, we obtain Bayes' rule (4.2): $P(\boldsymbol{\theta}|\mathbf{d}) = \frac{P(\mathbf{d}|\boldsymbol{\theta})\, P(\boldsymbol{\theta})}{P(\mathbf{d})}$.

4.1.2.1 Likelihood

The likelihood $L(\mathbf{d}|\boldsymbol{\theta}, M)$ gives the probability of obtaining data $\mathbf{d}$ under the precondition that the parameters are given by $\boldsymbol{\theta}$. The data dependence in the posterior $p(\boldsymbol{\theta}|\mathbf{d}, M)$ only comes in via the likelihood (Gelman, 2014). Note that the likelihood itself is not a probability distribution (Lambert, 2018). Thus, the likelihood does not need to integrate or sum to 1. However, it is interpretable by taking the ratio of two different likelihood values (Etz, 2017).

4.1.2.2 Prior

The prior represents the knowledge of $\boldsymbol{\theta}$ *before* drawing a sample and stands for our belief before we have gathered new data. It is explained in detail in Section 4.3.

4.1.2.3 Marginal

The marginal $m(\mathbf{d}|M)$ acts as a normalisation for the posterior (Lambert, 2018). It is actually a probability distribution over the complete data space, but since we take the observed data as fixed in Bayesian inference, it rather represents a constant for the posterior. I.e., it is the probability distribution over the data space evaluated at the observed data.

In case of a discrete parameter space Θ, the marginal is given via the law of total probability by (Gelman, 2014)

$$m(\mathbf{d}|M) = \sum_{\boldsymbol{\theta}' \in \Theta} P(\mathbf{d}, \boldsymbol{\theta}'|M) = \sum_{\boldsymbol{\theta}' \in \Theta} L(\mathbf{d}|\boldsymbol{\theta}', M) \cdot \pi(\boldsymbol{\theta}'|M). \tag{4.4}$$

That is, one obtains the parameter independent marginal by removing all parameter dependence via summing over all individual parameter dependencies. As we can see, it entirely consists of the likelihood and the prior. So once the likelihood and prior are selected, the marginal is defined as well. In the continuous case, the summation is swapped with an integral,

$$m(\mathbf{d}|M) = \int_{\boldsymbol{\theta}' \in \Theta} d\boldsymbol{\theta}'\, P(\mathbf{d}, \boldsymbol{\theta}'|M) = \int_{\boldsymbol{\theta}' \in \Theta} d\boldsymbol{\theta}'\, L(\mathbf{d}|\boldsymbol{\theta}', M)\pi(\boldsymbol{\theta}'|M). \tag{4.5}$$

Unfortunately, the marginal can be computationally expensive to compute, especially when the parameter space is multidimensional and continuous (Lambert, 2018). A way to alleviate this problem is to use conjugate priors, as explained in Section 4.3.4, or to use sampling methods such as Markov chain Monte Carlo (MCMC). However, in parameter inference we are only interested in the relative dependence on the parameters and hence the marginal can often be neglected since it represents a constant. The general shape of the posterior and therefore the parameter dependencies stay the same, it is just scaled by a constant.

The law of total probability If we have events A and B, then the marginal $m(A)$ gives the probability distribution of event A irrespective of event B. This is in contrast to conditional probabilities, where $P(A|B)$ gives the probability of event A given that event B has occurred.

For events A and B, the probability of both events happening is written as $P(A \cap B)$. The probability that either event A or B occurs is written as $P(A \cup B)$. The probability that event B does not occur is written as $P(B^c)$. Independent of whether event B has occurred or not, we can calculate the probability for event A as the sum of $P(A \cap B)$ and $P(A \cap B^c)$ (Pishro-Nik, 2014),

$$P(A) = P(A \cap B) + P(A \cap B^c) \tag{4.6}$$

This is because B and B^c are disjoint, and combined B and B^c represent all possible outcomes of B. Therefore their probabilities must add up to 1 (Sivia & Skilling, 2006),

$$P(B) + P(B^c) = 1. \tag{4.7}$$

We also know that we can represent $P(A \cap B)$ as a weighted conditional probability $P(A|B)\,P(B)$ and hence equation (4.6) becomes

$$\begin{aligned} P(A) &= \sum_{b \in \{B, B^c\}} P(A|b)\,P(b) && (4.8)\\ &= P(A|B)\,P(B) + P(A|B^c)\,P(B^c) && (4.9) \end{aligned}$$

The marginal $m(A)$ can be seen as the average of the sum of conditional probabilities $P(A|b)$ for all possible outcomes of b. As we can see from this equation, the marginal represents the expectation value $\mathbb{E}_b[P(A|b)]$, or $\mathbb{E}_{\boldsymbol{\theta}'}[P(\mathbf{d}|\boldsymbol{\theta}')]$ for $A = \mathbf{d}, b = \boldsymbol{\theta}'$.

More generally, the law of total probability states that if we have a partition $B_1, \ldots, B_n$ of a space $\mathcal{S}$, then for any event $A \in \mathcal{S}$ it holds that (Pishro-Nik, 2014)

$$P(A) = \sum_{i=1}^{n} P(A \cap B_i) \tag{4.10}$$

$$= \sum_{i=1}^{n} P(A|B_i)\, P(B_i). \tag{4.11}$$

This looks just the same as the marginal equation of (4.4).

4.1.2.4 Posterior

The posterior $p(\boldsymbol{\theta}|\mathbf{d}, M)$ tells us about the probability of parameter values $\boldsymbol{\theta}$ given some observed data $\mathbf{d}$. While the prior represents the belief in the parameters before (prior to) seeing any data, the posterior represents the updated belief *after* (posterior to) the available data has been observed. After each data sample has been observed, the posterior is updated. It does so by combining the prior parameter belief with the information that the data sample contains about the parameters.

4.1.3 Bayes' rule for i.i.d. events

Let $X \sim f(\mathbf{d}, \boldsymbol{\theta}, M)$ be a random variable that depends on parameters $\boldsymbol{\theta}$ and on a model M. We construct a sample $\mathbf{d} = (d_1, .., d_N)$ of N observations of X, where the observations are independent of each other. This gives $P(d_i, d_j|\boldsymbol{\theta}, M) = P_i(d_i|\boldsymbol{\theta}, M) \cdot P_j(d_j|\boldsymbol{\theta}, M)$. Since the data $\mathbf{d}$ is sampled from $X \sim f$, the observations are identically distributed. This gives $P(d_i|\boldsymbol{\theta}, M) := P_i(d_i|\boldsymbol{\theta}, M) = P_j(d_i|\boldsymbol{\theta}, M), \forall i, j \leq N$. Hence, the data set $\mathbf{d}$ is i.i.d. (independent and identically distributed), and overall we receive $P(d_i, d_j|\boldsymbol{\theta}, M) = P(d_i|\boldsymbol{\theta}, M) \cdot P(d_j|\boldsymbol{\theta}, M)$.

Therefore, with an i.i.d. dataset $\mathbf{d}$ we can split up a likelihood of the full set of observations as a multiplication of likelihoods of each individual observation,

$$L(\mathbf{d}|\boldsymbol{\theta}, M) = \prod_{i=1}^{N} P(d_i|\boldsymbol{\theta}, M) \tag{4.12}$$

Taking all of this together with (4.2) and (4.5), one arrives at the posterior

$$p(\boldsymbol{\theta}|\mathbf{d}, M) = \frac{\left[\prod_{i=1}^{N} P(d_i|\boldsymbol{\theta}, M)\right] \cdot \pi(\boldsymbol{\theta}|M)}{\int_{-\infty}^{\infty} L(\mathbf{d}|\boldsymbol{\theta}', M)\, \pi(\boldsymbol{\theta}'|M)\, d\boldsymbol{\theta}'}. \tag{4.13}$$

Here, the posterior p represents the knowledge of $\boldsymbol{\theta}$ *after* drawing N samples of X. Equation (4.13) is a specific version of Bayes' rule (4.2) which assumes i.i.d. data and uses the law of total probability to normalise the denominator.

Instead of interpreting (4.13) as the posterior being updated after observing each new data sample, one can also interpret it as the prior being updated with each data sample. Let us say we had $N = 2$ i.i.d. data samples x_1 and x_2 such that $\mathbf{d} = \{x_1, x_2\}$. After observing sample x_1, the prior $\pi_0(\boldsymbol{\theta})$ is

updated to be the posterior $p(\boldsymbol{\theta}|x_1)$ such that $\pi_1(\boldsymbol{\theta}) = p(\boldsymbol{\theta}|x_1)$. This new prior is then used to obtain the posterior $p(\boldsymbol{\theta}|\{x_1, x_2\})$ of the combined data x_1 and the new sample x_2. Using equation (4.2), this can be written as

$$p(\boldsymbol{\theta}|\{x_1, x_2\}) = \frac{L(\{x_1, x_2\}|\boldsymbol{\theta})\,\pi_0(\boldsymbol{\theta})}{m(\{x_1, x_2\})} \tag{4.14}$$

$$\overset{[\{x_1,x_2\} i.i.d]}{=} \frac{L(x_2|\boldsymbol{\theta})\,L(x_1|\boldsymbol{\theta})\,\pi_0(\boldsymbol{\theta})}{m(x_2)\,m(x_1)} \tag{4.15}$$

$$= \frac{L(x_2|\boldsymbol{\theta})}{m(x_2)} \cdot \left[\frac{L(x_1|\boldsymbol{\theta})\,\pi_0(\boldsymbol{\theta})}{m(x_1)}\right] \tag{4.16}$$

$$= \frac{L(x_2|\boldsymbol{\theta})}{m(x_2)} \cdot p(\boldsymbol{\theta}|x_1) \tag{4.17}$$

$$\overset{[\pi_1(\boldsymbol{\theta}):=p(\boldsymbol{\theta}|x_1)]}{=} \frac{L(x_2|\boldsymbol{\theta})\,\pi_1(\boldsymbol{\theta})}{m(x_2)}. \tag{4.18}$$

This implies that we update our belief of the parameters with each new sample we see. Intuitively this makes sense as well. Let us say we assumed we had a fair coin that gives ‘heads’ or ‘tails’ with an equal probability of 0.5. We observe 15 coin flips which show ‘heads’ every time. We might then update our belief about the coin, and rather assume from now on that the coin is weighted such that it shows ‘heads’ more often.

4.1.4 Bayesian inference

Bayesian inference uses Bayes’ theorem (4.2) to update the posterior probability distribution p on parameters $\boldsymbol{\theta}$ based on the observed data $\mathbf{d}$ (Gelman, 2014). One can then make inferences about the outcomes of yet unobserved events given the observed data, including an estimate of uncertainty.

Firstly, one sets up a model by initialising the task with a probability model that includes all observable and unobservable variables. Next, after the joint probability distribution has been set up, one can use Bayes’ rule (4.2) to obtain the posterior probability distribution. From the posterior one can gain predictions of unobserved variables based on the observed data. Lastly, one evaluates the model by testing how well the probability model aligns with the data. That is, does it represent the data well? Do the resulting implications make sense? And how sensitive are the results to the model assumptions? One way to do this is via posterior predictive checks, where one compares the observed data to predicted samples and calculate a predictive accuracy (Gelman, 2007, 2014). Other methods include Bayesian cross-validation. For a more detailed analysis see Gelman (2007). One can then continuously iterate through these three steps until one is happy with the outcomes.

4.1.5 Bayes factor

If one wanted to test the hypothesis of two different priors or methods, one can do this with the so-called Bayes factor (Kass & Wasserman, 1996). The Bayes factor is the ratio of posterior odds to prior odds of the two hypothesis,

$$B = \frac{\int L_1(\boldsymbol{\theta}_1)\,\pi_1(\boldsymbol{\theta}_1) d\boldsymbol{\theta}_1}{\int L_2(\boldsymbol{\theta}_2)\,\pi_2(\boldsymbol{\theta}_2) d\boldsymbol{\theta}_2}. \tag{4.19}$$

For example, one could have parameters $\boldsymbol{\theta} = \{\alpha, \beta\}$ with hypothesis $\mathcal{H}_0 : \alpha = \alpha_0$ and an unrestricted hypothesis $\mathcal{H}_1 : \alpha \in \boldsymbol{\theta}$. Then we would have $\boldsymbol{\theta}_1 = \{\beta\}$ and $\boldsymbol{\theta}_2 = \{\alpha, \beta\}$, and two different likelihoods and priors (Jeffreys, 1983).

4.1.6 Bayesian model comparison

Bayesian model comparison enables us to test multiple hypotheses by comparing a set of statistical models to observed data (Burnham, 2002; Sivia & Skilling, 2006; Neath et al., 2017). Given two models M_1 and M_2, data D and additional background prior information I, the ratio of their posterior probabilities is

$$R = \frac{P\,(M_1|D, I)}{P\,(M_2|D, I)} = \frac{P\,(D|M_1, I)}{P\,(D|M_2, I)}\,\frac{P\,(M_1|I)}{P\,(M_2|I)}. \tag{4.20}$$

If $R \gg 1$, then model M_1 fits the data better than model M_2. The probability $P\,(M_i|I)$ is the prior probability that a source belongs to model M_i.

If we have a set of models $\{M_j\}$, the probability that a source is of type M_i given input data D is then

$$P\,(M_i|D, I) = \frac{P\,(D|M_i, I)\;P\,(M_i|I)}{\sum_j P\left(D|M_j, I\right)\;P\left(M_j|I\right)} \tag{4.21}$$

$$\overset{(4.5)}{=} \frac{P\,(M_i|I)\int P\,(D|M_i, \boldsymbol{\theta}_i, I)\;P\,(\boldsymbol{\theta}_i|M_i, I)\,d\boldsymbol{\theta}_i}{\sum_j P\left(M_j|I\right)\int P\left(D|M_j, \boldsymbol{\theta}_j, I\right)\;P\left(\boldsymbol{\theta}_j|M_j, I\right) d\boldsymbol{\theta}_j}. \tag{4.22}$$

4.2 Examples

We illustrate a few examples of likelihood, marginal and posterior.

4.2.1 The likelihood in a coin flip

Let us take the example of tossing a coin again, where $\boldsymbol{\theta} = \{\theta\}$ with θ being the probability of getting 'heads' from a coin flip. If we tossed the coin three times ($N = 3$), then the probability of observing

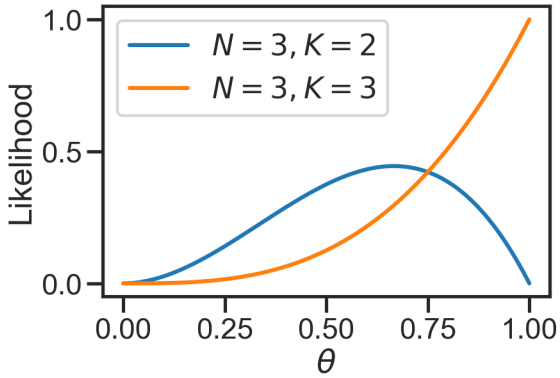


Figure 4.1: Binomial likelihood, where θ is the probability of 'heads' during a coin flip. N gives the number of observed coin flips and K defines the number of observed 'heads' during the N coin flips.

'heads' three times ($K = 3$) is equal to to the multiplication of the individual outcomes, $P(K = 3, n = 3|\boldsymbol{\theta}) = P(K = 1, N = 1|\boldsymbol{\theta}) \cdot P(K = 1, N = 1|\boldsymbol{\theta}) \cdot P(K = 1, N = 1|\boldsymbol{\theta}) = \theta^3$. For $\theta = \frac{1}{2}$ this would be $P(K = 3, n = 3|\boldsymbol{\theta}) = \left(\frac{1}{2}\right)^3$. There is only 1 possible way to have three 'heads' in three coin tosses, namely $\{H, H, H\}$, where H stands for 'heads' and T for 'tails'. If we threw the coin three times and had two 'heads' instead to get $P(K = 2, N = 3|\boldsymbol{\theta})$ then there would be 3 possible ways to toss the coin, $\{T, H, H\}$, $\{H, H, T\}$ and $\{H, T, H\}$. The resulting probability for $\theta = \frac{1}{2}$ would then be $P(K = 2, N = 3|\boldsymbol{\theta}) = 3 \cdot \left(\frac{1}{2}\right)^3$.

If we kept $\boldsymbol{\theta}$ fixed at $\theta = \frac{1}{2}$ representing a fair coin, and vary the outcomes of the coin tosses, then our likelihood above would represent a probability mass function. We would sum over the different probabilities of possible outcomes for a fixed $\theta = \frac{1}{2}$ and would get 1 as expected from a probability distribution. This is also what we usually expect in conditional probabilities $P(\mathbf{d}|\boldsymbol{\theta})$, where we hold the parameters of the condition $\boldsymbol{\theta}$ fixed and vary the data $\mathbf{d}$ (Etz, 2017).

However, this is not what we do for likelihoods in Bayesian statistics. This is why the likelihood is sometimes written as $L(\boldsymbol{\theta})$ to emphasize that it is a function of the condition $\boldsymbol{\theta}$. Instead of keeping the parameters fixed, we keep the data fixed, since the data $\mathbf{d}$ is what we observe. The parameters, on the other hand, are assumed to be a random quantity (Carlin, 2008). So if we again observed three coin tosses which showed 'heads' each time, then we would calculate $P(K = 3, N = 3|\boldsymbol{\theta})$ for each possible parameter value θ. This would then also include unfair coins for which $\theta \neq \frac{1}{2}$. We would not calculate e.g. the likelihood for showing 'heads' only twice during three coin tosses, since this is not what we observe.

Since $\boldsymbol{\theta} = \{\theta\}$ with θ the probability of getting 'heads', the probability for 'tails' is $1 - \theta$, which can be modelled via the Bernoulli distribution,

$$P(K = 1, N = 1|\boldsymbol{\theta}) = \theta, \tag{4.23}$$

$$P(K = 0, N = 1|\boldsymbol{\theta}) = 1 - \theta \tag{4.24}$$

$$\Rightarrow P(K = k, N = 1|\boldsymbol{\theta}) = \theta^k \cdot (1 - \theta)^{1-k} \tag{4.25}$$

For more than one coin flips $N > 1$, this becomes the binomial distribution,

$$P(K = k, N = n|\boldsymbol{\theta}) = \binom{n}{k} \theta^k \cdot (1-\theta)^{n-k}, \tag{4.26}$$

where $\binom{n}{k} = \frac{n!}{k!\,(n-k)!}$ the binomial factor that gives us the amount of possible arrangements of observing k 'heads' in n coin flips. So in the case of observing three heads ($K = 3$) during three coin flips ($N = 3$), our likelihood would be given by $P(K = 3, N = 3|\boldsymbol{\theta}) = \binom{3}{3}\theta^3 \cdot (1-\theta)^{3-3} = \theta^3$, which is a function of the parameter $\boldsymbol{\theta} = \{\theta\}$ as expected. Similarly, if we observed two 'heads' ($K = 2$) in three coin flips ($N = 3$) then our likelihood is given by $P(K = 2, N = 3|\boldsymbol{\theta}) = \binom{3}{2}\theta^2 \cdot (1-\theta)^{3-2} = 3 \cdot \theta^2 \cdot (1-\theta)$. In the case of $\theta = \frac{1}{2}$ this then gives $P(K = 2, N = 3|\boldsymbol{\theta}) = 3 \cdot \left(\frac{1}{2}\right)^3$. Note that $\binom{3}{2} = 3$ gives three different ways to have 2 'heads' in 3 coin flips as explained further above. This is illustrated in Figure 4.1, which shows the binomial likelihood for $N = 3$ and two different K, $K = 2$ and $K = 3$. The likelihood is a function of θ, the probability of receiving 'heads' during a coin flip. If we look again at $\theta = \frac{1}{2}$, we can see that the likelihood is larger for observing $K = 2$ two 'heads' compared to three 'heads', which is consistent with our obtained result. $\theta = 1$ means that one only can receive 'heads' during coin flips, and therefore the likelihood is 1 when we observe 'heads' all three times, and 0 when we observe 'heads' not during all of the coin flips. Similarly, $\theta = 0$ means that we can only observe 'tails', and hence we receive a likelihood of 0 if we observe even one 'heads' during the coin flips.

4.2.2 The marginal in a coin flip

If we take our coin flipping example again and assume that one can either have a fair coin with 'heads' on one side and 'tails' on the other, or an unfair coin with either only 'heads' on both sides or only 'tails' on both sides. This gives a parameter space of $\Theta = \{0, 0.5, 1\}$. The marginal is then obtained via equation (4.4) by summing over the dependence on θ,

$$\begin{aligned} m(\mathbf{d}) &= \sum_{\theta \in \{0,0.5,1\}} P(\mathbf{d}|\theta) \cdot P(\theta) && (4.27) \\ &= P(\mathbf{d}|\theta = 0) \cdot P(\theta = 0) + P(\mathbf{d}|\theta = 0.5) \cdot P(\theta = 0.5) + P(\mathbf{d}|\theta = 1) \cdot P(\theta = 1). && (4.28) \end{aligned}$$

If θ could take any value between 0 and 1 as in Figure 4.1, then we would obtain via equation (4.5) a marginal of

$$\begin{aligned} m(\mathbf{d}) &= \int_{\theta \in [0,1]} d\theta\, P(\mathbf{d}|\theta) \cdot P(\theta) && (4.29) \\ &= \int_0^1 d\theta\, P(\mathbf{d}|\theta) \cdot P(\theta). && (4.30) \end{aligned}$$

In both the discrete and continuous parameter case, the likelihood $P(\mathbf{d}|\theta)$ would be given by the binomial distribution.

4.2.3 The posterior probability of having Covid if tested positive

Let us say patient A takes a Covid test. This test has an accuracy of 99%. If patient A gets a positive test result, one might therefore assume that the probability of the patient having Covid is 0.99. But that is not necessarily correct. What we have not included is the prevalence of Covid in the population. If every single citizen had Covid, then the 0.99 probability of patient A having Covid after a positive test result would be correct. However, if the prevalence is less than 100%, then the estimated probability is wrong.

In this case, the prevalence of Covid in the population is the prior information. It is an essential part in the calculation. The prior is given as the probability that any citizen has Covid, $P(\text{has Covid})$. The probability that we have a positive test result is given by $P(\text{positive})$, and the probability of having a negative test result is $P(\text{negative})$. We want to find the probability $P(\text{has Covid}|\text{positive})$ that the patient has Covid given a positive test result. We use Bayes' rule (4.2) where $\boldsymbol{\theta}$ = has Covid and $\mathbf{d}$ = positive,

$$P(\text{has Covid}|\text{positive}) = \frac{P(\text{positive}|\text{has Covid})\, P(\text{has Covid})}{P(\text{positive})}. \tag{4.31}$$

One can test positive both when one has Covid and one has not Covid, and hence the marginal $P(\text{positive})$ is split up such that $P(\text{positive}) = P(\text{positive}|\text{has Covid})\, P(\text{has Covid}) +$
$P(\text{positive}|\text{has not Covid})\, P(\text{has not Covid})$ and thus

$$P(\text{has Covid}|\text{positive}) = \frac{P(\text{positive}|\text{has Covid})\, P(\text{has Covid})}{P(\text{positive}|\text{has Covid})\, P(\text{has Covid}) + P(\text{positive}|\text{has not Covid})\, P(\text{has not Covid})}. \tag{4.32}$$

Let us say we had a prevalence of 0.1% such that $P(\text{has Covid}) = 0.001$. This means that the probability of not having Covid is $P(\text{has not Covid}) = 1 - 0.001 = 0.999$. The sensitivity of a test gives the probability $P(\text{positive}|\text{has Covid})$ of having a positive test if the person has Covid. The specificity of a test tells us the probability $P(\text{negative}|\text{has not Covid})$ that a person tests negative if the person does not have Covid. Let us say we had a sensitivity of $P(\text{positive}|\text{has Covid}) = 0.99$ and a specificity of $P(\text{negative}|\text{has not Covid}) = 0.95$. Plugging these into equation (4.32) gives us

$$P(\text{has Covid}|\text{positive}) = \frac{0.99 \cdot 0.001}{0.99 \cdot 0.001 + (1 - 0.95) \cdot 0.999} \tag{4.33}$$

$$= 0.019. \tag{4.34}$$

This gives a probability of less than 2% that one has Covid if the test result is positive, given a prevalence in the population of 0.1%, a sensitivity of 99% and a specificity of 95%. This is much lower than one might assume, but actually reflects reality.

4.3 Prior

The prior probability distribution $\pi(\boldsymbol{\theta}|M)$ represents a weighting of each parameter value. The probability of each parameter value is multiplied with the likelihood of a specific result given that parameter value (Lambert, 2018). This prior is often chosen subjectively by one's own personal belief, and hence the choice of prior can be controversial. However, the subjectivity is given in an explicit way rather than in implicit assumptions, which makes it more transparent. A prior can give equal weight to each parameter value, which makes it a non-informative prior. It can also set different weights to different parameter values as long as it still represents a probability distribution, which is called an informative prior. To help with the prior configuration one can make the prior follow certain well-known distributions where one only has to choose the parameters of the distribution.

Depending on the prior distribution, the posterior will look different. However, upon receiving a large amount of data samples, the prior distribution will have less and less of an effect on the posterior. In this section we will be going through different types of priors.

4.3.1 Subjective prior

If we take the example of a coin again, if we have a fair coin we assume that the probability of given 'heads' and 'tails' is 0.5 for both. But how do we know that? It is because many independent coin flipping experiments have been made and the probability turns out to be 0.5 for both 'heads' and 'tails'. We can therefore make an informed prior decision. Another example is that of a doctor who checks a patient for a certain disease. Before examining the patient, the doctor will assign a prior probability of how likely he or she thinks the person has that disease. The doctor will make that assumption based on their experience as a doctor. This experience is personal and limited to the patients the doctor has seen in his or her lifetime.

However, sometimes we do not have any experiment available at all. How would we then decide what the prior distribution should be? This is where *subjective probability* comes in (Berger, 1985). It is essentially a personal belief without having the backup of neither a relative frequency of past events nor probability theory (Siegel & Wagner, 2022). This occurs at events that have not happened before and hence no past experience or theoretical prediction is available. The important thing to note here is that the assigned believed probabilities have to be consistent, e.g. $P(A) + P(\neg A) = 1$, and the prior has to be a probability distribution (either discrete or continuous). Unfortunately, us humans are very bad at estimating small probabilities (Tversky & Kahneman, 1974). A person untrained in the area of the task will likely overestimate his or her confidence in the subjective probabilities and therefore constrain the parameters too tightly.

An example of subjective probability would be the task to predict the stock value of a new technology.

One has no past experience in that technology since it is brand-new. There are also no probability equations that are applicable in this case. Instead of blindly guessing, one might want to ask an expert in the field about his/her opinion to increase the accuracy of the probability. There are people which are against subjective priors as they do not like the arbitrariness of the functional form choice (Kass & Wasserman, 1996). However, no probability is known per se but is model dependent (Sivia & Skilling, 2006). That is, if we have two models M_1 and M_2, then $P(\boldsymbol{\theta}|M_1) \neq P(\boldsymbol{\theta}|M_2)$, $P(\mathbf{d}|\boldsymbol{\theta}, M_1) \neq P(\mathbf{d}|\boldsymbol{\theta}, M_2)$ and $P(\boldsymbol{\theta}|\mathbf{d}, M_1) \neq P(\boldsymbol{\theta}|\mathbf{d}, M_2)$.

Jeffreys (1983) wanted to make the selection of priors a convention. One was allowed to have an initial belief, but to hold no opinion on whether that belief was true yet. If one hypothesis was as likely as another, then the prior probabilities should be equal. One could only choose a specific hypothesis from definite reason. If there are many equally good hypothesis, they should be clearly stated before a consistent decision is made (Jeffreys, 1955). In comparison, a person using a subjective prior would not have to defend why one hypothesis is better than or equal to another. Subsequently, a range of suggestions were developed on how to select prior distributions (Kass & Wasserman, 1996).

4.3.1.1 How to create a subjective prior

In order to define a prior probability distribution, there are a few possible solutions (Berger, 1985). One could make a histogram of subjective beliefs over certain intervals and create a smooth probability distribution from that. One could also use relative likelihoods based on a few anchor points. For example, one might think the most likely parameter value is at point x_1 and the least likely at x_2. If one has a few other likely/unlikely points relative to these most likely/unlikely ones, one can sketch the prior probability distribution from that. This is easiest when one has been given with a finite parameter space.

Lastly, a common approach is to assume that the prior is of a certain functional form that aligns with the prior beliefs and then one only has to define the parameters of that function. An example would be a prior that is Gaussian distributed. In this case, one would only have to define the mean and the variance. However, it can be quite difficult to subjectively decide which parameter values to choose for that distribution. A useful tool are fractiles, where one chooses points in the parameter space and defines subjective probabilities that a random variable that follows the prior is smaller or equal to this point. This acts as a guide to find the most appropriate parameters for the functional form of the prior distribution. If the fractiles differ too much from the chosen function, one should use a different prior instead.

For multi-dimensional prior parameter spaces, it is difficult to imagine the probability distribution. In those cases it can be best to use the last option of assuming a functional form and just subjectively estimating the parameters. Alternatively, if the prior parameters are independent, one can use any of the methods mentioned above and just multiply the resulting probabilities with each other.

4.3.2 Non-informative prior

In case we do not have any prior knowledge then non-informative priors can be used (Berger, 1985). Non-informative priors give equal probability to each parameter value, and hence do not favour one parameter value over the other. Exceptions to this are discussed in the last paragraph.

Non-informative priors are especially useful when we cannot favour one parameter value over another and hence cannot build a subjective prior. Laplace (1995) called this the 'principle of insufficient reason'. It is further useful when we want to "let the data speak for itself" via the likelihood (Lambert, 2018). Thus, it should not have a big influence on the probability outcomes of an experiment (Box & Tiao, 1973). However, even though non-informative priors do not prefer one value over another, all priors contain some level of information. Fisher criticised the use of non-informative priors as he argued that knowing versus not knowing that mutually exclusive events are of equal probability represent two very different states of knowledge and should not be treated the same (Syversveen, 1998). Another example of when to use non-informative priors is when we have a small sample size compared to the number of parameters (Kass & Wasserman, 1996). (Bayes & Price, 1763) was the first person to use a non-informative flat prior for the binomial distribution.

As an example of a non-informative prior, if we have N possible parameter values, then each parameter value would be assigned a probability of $\frac{1}{N}$. Another example would be the uniform distribution given over an interval $[a_i, b_i]$ such that $\pi(\theta_i) \sim U(a_i, b_i)$,

$$\pi(\theta_i|M) = \begin{cases} \frac{1}{b_i - a_i}, & a_i \leq \theta_i \leq b_i, i = 1, .., K \\ 0, & \text{otherwise} \end{cases} \tag{4.35}$$

where $\{a_i\}_{i=1,..,K}$ and $\{b_i\}_{i=1,..,K}$ are the lower and upper bounds of the parameters $\{\theta_i\}_{i=1,..K}$ respectively. For instance, if we know that we have a fair coin, then the probability for 'heads' and 'tails' is the same and we use a uniform prior. Let us say we hide the coin under one of five cups and ask someone else to guess under which cup it is. Without any further knowledge, one would take a uniform prior with an equal probability of $\frac{1}{5}$ for each cup. This also makes the multiplication with the likelihood much easier to calculate. In such cases of constant prior the shape of the posterior is determined by the likelihood, not by the prior, and therefore has a peak at the same value as the likelihood (Lambert, 2018),

$$p(\boldsymbol{\theta}|\mathbf{d}, M) \propto L(\mathbf{d}|\boldsymbol{\theta}, M). \tag{4.36}$$

An issue can arise when we apply a transformation to the parameter space of our non-informative prior. Let us take the example of a finite parameters space with N possible values again and let us define the parameters by $\boldsymbol{\theta}$. If we now made a parameters transformation of $\boldsymbol{\xi} = \exp(\boldsymbol{\theta})$, which

translated to $\boldsymbol{\theta} = \log(\boldsymbol{\xi})$, then our prior would change to

$$\pi(\boldsymbol{\xi}) = \left|\frac{d\boldsymbol{\theta}}{d\boldsymbol{\xi}}\right| \pi(\boldsymbol{\theta}) = \frac{1}{\boldsymbol{\xi}} \pi\left[\log(\boldsymbol{\xi})\right]. \tag{4.37}$$

This re-parameterised prior is not constant anymore, and hence does not give equal probability to each parameter value. One could adapt the prior $\pi(\boldsymbol{\theta})$ to multiply the constant probability with $\boldsymbol{\xi}$, which would then make prior $\pi(\boldsymbol{\xi})$ constant. However, then $\pi(\boldsymbol{\theta})$ would not be constant anymore. Hence, one cannot have both priors $\pi(\boldsymbol{\theta})$ and $\pi(\boldsymbol{\xi})$ be constant, that is, give equal weight to each parameter. This means that invariance is not kept under transformation of this constant prior.

4.3.2.1 Improper priors

A difficulty arises when the parameter space is infinite. In that case the integral over a constant prior would not equal to 1 and hence the prior would not be a probability distribution. This is called an *improper prior*. An improper prior can also be a problem for calculating the marginal probability. However, an improper prior can still lead to a proper posterior (Syversveen, 1998). Hence, an improper prior does not always represent as an issue.

4.3.2.2 Jeffreys prior

Finally, Jeffreys (1983) gave the initiative to find non-informative priors that are invariant under transformation that can be used per convention. Jeffreys (1946) defined a non-informative prior invariant under transformation. It is proportional to the determinant of the square root of the Fisher matrix $\mathbb{I}$ of the prior parameters $\boldsymbol{\theta}$,

$$\pi(\boldsymbol{\theta}) \propto \det \sqrt{\mathbb{I}(\boldsymbol{\theta})}, \tag{4.38}$$

where the Fisher matrix has to be positive definite. The Fisher matrix $\mathbb{I}(\boldsymbol{\theta})$ of size $m \times m$ is defined as

$$\mathbb{I}(\boldsymbol{\theta}) = -\mathbb{E}_{\boldsymbol{\theta}}\left[\frac{\partial^2}{\partial\theta_i \partial\theta_j} \log L(\mathbf{d}|\boldsymbol{\theta})\right], \qquad 1 \leq i, j \leq m, \tag{4.39}$$

where $L(\mathbf{d}|\boldsymbol{\theta})$ the likelihood. The Fisher information tells us how much information our observed data gives us about our model parameters. It is a forecast on the precision of the model parameters before we have done the full observations Wittman (Wittman). One can then compare different models and predict which would give the most accurate results. Fisher forecasts are commonly used in astronomy, for example for the planning of astronomical surveys, and for finding new cosmological probes (Bhandari et al., 2021).

The Fisher matrix is essentially the negative of the Hessian matrix applied on the log-likelihood with respect to the prior parameters, which results in a matrix of second derivatives of the log likelihood

with respect to the prior parameters. In the one-dimensional case the Jeffrey prior can be written more simply as $\pi(\boldsymbol{\theta}) \propto \sqrt{\mathbb{I}(\boldsymbol{\theta})}$.

The Jeffrey prior $\pi(\boldsymbol{\theta})$ is invariant under parameter transformation $\boldsymbol{\xi} = f(\boldsymbol{\theta})$. This is because

$$\mathbb{I}(\boldsymbol{\theta}) = \mathbb{I}[f(\boldsymbol{\theta})] \left(\frac{\partial f(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)^2 \tag{4.40}$$

and hence in the one-dimensional case

$$\pi(\boldsymbol{\theta}) \propto \sqrt{\mathbb{I}(\boldsymbol{\theta})} = \sqrt{\mathbb{I}[f(\boldsymbol{\theta})]} \left|\frac{\partial f(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right| = \pi[f(\boldsymbol{\theta})] \left|\frac{\partial f(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|, \tag{4.41}$$

which is the transformation formula. More generally, it holds that

$$\pi(\boldsymbol{\theta}) = \pi(\boldsymbol{\xi}) \left|\det \frac{d\boldsymbol{\xi}}{d\boldsymbol{\theta}}\right|. \tag{4.42}$$

For the normal distribution $\mathcal{N}(\mu, \sigma^2)$ it is recommended to apply Jeffrey's prior to both μ and σ independently. This is because depending on whether we use only σ or both μ and σ as the parameters, we get the same amount of degrees of freedom in the posterior which is not correct. If we take μ as fixed, then Jeffrey's prior for σ as the only parameter is given by $\pi(\sigma) \propto 1/\sigma$. For the two-dimensional case where both μ and σ are variable, however, Jeffreys prior is given by

$$\begin{aligned}
\pi(\mu, \sigma) &\propto \det \sqrt{\mathbb{I}(\mu, \sigma)} && (4.43)\\
&= \det \sqrt{-\mathbb{E}\begin{bmatrix} \frac{\partial^2 \log L}{\partial \mu^2} & \frac{\partial^2 \log L}{\partial \mu \partial \sigma} \\ \frac{\partial^2 \log L}{\partial \sigma \partial \mu} & \frac{\partial^2 \log L}{\partial \sigma^2} \end{bmatrix}} && (4.44)\\
&= \det \sqrt{-\mathbb{E}\begin{bmatrix} -\frac{1}{\sigma^2} & -2\frac{(\mathbf{d}-\mu)}{\sigma^3} \\ -2\frac{(\mathbf{d}-\mu)}{\sigma^3} & -3\frac{(\mathbf{d}-\mu)^2}{\sigma^4} + \frac{1}{\sigma^2} \end{bmatrix}} && (4.45)\\
&= \det \sqrt{\begin{bmatrix} \frac{1}{\sigma^2} & 0 \\ 0 & \frac{2}{\sigma^2} \end{bmatrix}} && (4.46)\\
&\propto \frac{1}{\sigma^2}, && (4.47)
\end{aligned}$$

where the log-likelihood is given by $\log L(\mathbf{d}|\boldsymbol{\theta}) \propto -\log \sigma - \frac{(\mathbf{d}-\mu)^2}{2\sigma^2}$. This is an improper prior. If we treat μ and σ as independent, we get back $\pi(\mu, \sigma) = \pi(\mu) \cdot \pi(\sigma) \propto 1/\sigma$ as wanted, which gives us the correct number of degrees of freedom.

4.3.3 Informative prior

If we have an existing set of observed data or have analysed the data at some time beforehand, we can create an informative prior (Lambert, 2018). This is where the parameter probabilities differ and thus are not constant. It makes sense to use an informative prior in such cases. Firstly, this prevents the usage of subjective beliefs one might have about the problem at hand. Let us say we had tossed a coin $10,000$ times and it came up 'heads' $1/3$ of the time and 'tails' $2/3$ of the time. This seems to be an un-fair coin and goes against one might subjectively assume about the coin. That is, one might have otherwise assumed that the coin is fair as one did not have any reason to believe so otherwise. Based on the observed data we therefore would assign a prior probability of $1/3$ for 'heads' and $2/3$ for 'tails'.

4.3.4 Conjugate prior

As the posterior can be very computationally intensive to compute, especially the marginal within that computation, one might choose a prior distribution that is of the same functional form as the likelihood. These are called *conjugate priors* (Raiffa & Schlaifer, 1961). For example, if the likelihood is Gaussian distributed, one could choose another Gaussian for the prior probability distribution as to make the posterior Gaussian distributed as well. Of course, one should only use the same family of distribution for the prior if it makes sense. If we knew that the parameters were uniformly distributed, then using a Gaussian instead would falsify the results (if the dataset is small).

Thankfully, today there are methods that can circumvent or ease the burden of computational costs (Hackenberger, 2019). An example is MCMC, which is a method that, instead of directly sampling from the complex posterior, uses a reversible Markov chain with an equilibrium distribution that matches the posterior.

4.4 Point estimates

Even though Bayes' rule gives us a probability distribution for the posterior, sometimes a point estimate can be useful or necessary. This is especially the case when the posterior is too time consuming to compute. This can be made easier by applying approximations such as taking a uniform prior and/or a Gaussian likelihood.

Let again $\mathbf{d} = (d_1, .., d_N) \in X$ be a sample of data of an independent random variable X, $\boldsymbol{\theta} = (\theta_1, .., \theta_K)$ a realisation of random variables $\boldsymbol{\theta}$ and M the representing model. It follows from Bayes' rule (4.2) that (Sivia & Skilling, 2006)

$$Posterior \propto Likelihood \times Prior \tag{4.48}$$

which can be written as

$$p(\boldsymbol{\theta}|\mathbf{d}, M) \propto L(\mathbf{d}|\boldsymbol{\theta}, M)\, \pi(\boldsymbol{\theta}|M) \tag{4.49}$$

4.4.1 Maximum Likelihood

The value of highest probability in the likelihood L for a parameter θ_i can be obtained via

$$\frac{\partial L}{\partial \theta_i}\Big|_{\boldsymbol{\theta}_{ML}} = 0, \forall i \in \{1, .., K\} \tag{4.50}$$

with $\boldsymbol{\theta}_{ML}$ being the vector of parameter values with highest probability - the maximum likelihood vector. This is called the *maximum likelihood* ML estimate. Note that we could find multiple maxima with this technique, or even a constant region or interval.

If we assume our data to be i.i.d., the maximum likelihood estimate is given by

$$\boldsymbol{\theta}_{ML} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, L\,(\mathbf{d}|\boldsymbol{\theta}) \tag{4.51}$$

$$\overset{[i.i.d.]}{=} \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \prod_{i}^{m} L\,(d_i|\boldsymbol{\theta})\,, \tag{4.52}$$

where m the number of samples.

It follows from equation (4.36) that if we assume a uniform prior, then we can get the position of the maximum posterior by maximising the likelihood. In that case, if we maximise the likelihood, we maximise the posterior.

4.4.2 Maximum a posteriori (MAP)

Often, we do not have a uniform prior, or we generally cannot find the complete posterior $p(\boldsymbol{\theta}|\mathbf{d})$. In such cases one might be interested in finding the maximum of the posterior instead, which is called the maximum a posteriori (MAP) estimate. It is similar to the maximum likelihood estimation, but in this case we apply it on the posterior distribution,

$$\frac{\partial p}{\partial \theta_i}\Big|_{\boldsymbol{\theta}_{MAP}} = 0, \forall i \in \{1, .., K\}, \tag{4.53}$$

where $\boldsymbol{\theta}_{MAP}$ is the vector of parameter values with highest probability.

It can also be written via the following

$$\boldsymbol{\theta}_{MAP} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, P(\boldsymbol{\theta}|\mathbf{d}) \overset{(4.2)}{=} \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, [P(\mathbf{d}|\boldsymbol{\theta}) \cdot P(\boldsymbol{\theta})]\,, \tag{4.54}$$

where we removed $P(\mathbf{d})$ from the last equation as it is independent of $\boldsymbol{\theta}$.

4.4.3 Chi-squared: Maximum likelihood using a Gaussian likelihood

Take the likelihood to be Gaussian distributed, $L \sim \mathcal{N}(\mu, \sigma)$. Hence, the individual probabilities of the data $\{d_i\}$ are given by

$$P(d_i|\boldsymbol{\theta}, M) = \frac{1}{\sigma_i \sqrt{2\pi}} \exp\Big(- \frac{(F_i - d_i)^2}{2\sigma_i^2}\Big), \tag{4.55}$$

where $\mathbf{F}$ is the ideal noiseless data.

Combining equations (4.12) and (4.55) one obtains for the Likelihood

$$L(\mathbf{d}|\boldsymbol{\theta}, M) \propto e^{-\frac{1}{2}\sum_{k=1}^{N}\left(\frac{F_k - d_k}{\sigma_k}\right)^2} = e^{-\chi^2/2}, \tag{4.56}$$

with

$$\chi^2 = \sum_{k=1}^{N}\Big(\frac{F_k - d_k}{\sigma_k}\Big)^2. \tag{4.57}$$

Equation (4.57) is called the *chi-Squared* value. It is a measure of how much your data differs from the predicted value.

If the prior is taken to be uniform, equations (4.36) and (4.56) lead to

$$p(\boldsymbol{\theta}|\mathbf{d}, M) \propto e^{-\chi^2/2}. \tag{4.58}$$

The exponential function is a monotonically increasing function and so is the logarithm. Moreover, the logarithm varies more slowly with x compared to the exponential function. One usually takes the logarithm of the posterior distribution as it is easier to calculate. Due to the monotonicity the maximum of the posterior will be at the same location as the maximum of the logarithm of the posterior. Hence, taking the logarithm of equation (4.58) gives

$$\log p = \log_e[p(\boldsymbol{\theta}|\mathbf{d}, M)] = const - \frac{\chi^2}{2}. \tag{4.59}$$

The maximum posterior occurs where the chi-squared χ^2 is at its minimum. Therefore, if we assume a uniform prior and a Gaussian likelihood, we can obtain the maximum parameter value by minimising the chi-squared.

Chapter 5

Machine learning

The sheer volume of Petabytes of data flow in astronomy requires an automated tool to analyse the incoming data. This tool should be able to find patterns in high-dimensional data while being simple to use (Ball & Brunner, 2010). Machine learning (ML) can give all that, and does so in a short amount of time.

Machine learning consists of algorithms that can identify relationships between data points and use these to predict the output of new data sets (Murphy, 2012). An example would be to distinguish between different classes of objects. As Thomas Mitchell defines it (Mitchell, 1997): "A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E." For instance, experience is a set of the observations in an astronomical survey, the task is to classify stars and galaxies within those observations, and the performance measure is the accuracy.

The learning module of the algorithm consists of representation, evaluation and optimization (Domingos, 2012). Representation includes the question of which information to use as the input data. The input data for each sample is composed of a vector of features. A feature is a numerical or categorical variable that describes that data point. Evaluation defines the metric or scoring function that can differentiate between good and bad classifiers. It gives the performance on a given task. Optimization corresponds to finding the best parameters of the classifier that minimises the scoring function. This step makes sure that the algorithm improves upon experience. We want the classifier to perform well not only on the data set it has been trained on, but also on any data set it has not seen yet. Hence, we do not necessarily want to find the minimum of the cost function, but we want to be close to it. The final classifier then outputs the predicted class for a given feature vector.

In astronomy, most of the data is inherently noisy and often ground truths are not known. A ground truth is the accurate label of the corresponding input that is known. A bit of label uncertainty and input noise is actually useful for the generalization of the classifier. However, some data is partly

incomplete and not all machine learning algorithms can cope with that. In the ideal case, we want an algorithm that can take uncertainties in measurements and in labels into account, and can correct for incomplete data.

ML is a widely used tool in astronomy, especially for classifying astronomical objects. A brief summary is given in Section 3.2.3. It has been used to separate galaxies from stars and to analyse galaxy morphology. Several methods have been used to classify quasars at redshifts $0 < z < 6$ (Peng et al., 2013; Gao et al., 2008; Jin et al., 2019; Abraham & Philip, 2010). Artificial neural networks (ANNs) (Yèche et al., 2010; Claeskens et al., 2006; Tuccillo et al., 2015; Carballo et al., 2008), decision trees (DTs) (White et al., 2000; Suchkov et al., 2006; Zhao & Zhang, 2008; Meshcheryakov et al., 2018), principal component analysis (PCA) (Yip et al., 2004; Davies et al., 2018; Pâris et al., 2011; Suzuki, 2006), support vector machines (SVMs) (Gao et al., 2008), kd-Trees (Gao et al., 2008), probabilistic principal surfaces and the negative entropy clustering (D'Abrusco et al., 2009), kernel density estimation (KDE) (Richards et al., 2009; Myers et al., 2008), and convolutional neural networks (CNNs) (Pasquet-Itam & Pasquet, 2018a; Guo & Martini, 2019a; Parks et al., 2018a) all have been used to classify AGN and/or to determine their redshifts.

5.1 Overview

We give an overview of machine learning, its building blocks and different types of learning.

5.1.1 Machine learning is a subset of artificial intelligence

Artificial intelligence is an umbrella term that encompasses machine learning, representation learning and deep learning. A Venn diagram of artificial intelligence and its subsets is displayed in Figure 5.1. In this work, we use a decision tree (see Section 5.4) and a random forest (see Section 5.5), which fall into the category of machine learning but not into its subsets. We also experiment with a convolutional neural network (see Section 5.7), which is a part of deep learning.

5.1.1.1 Artificial intelligence

Artificial intelligence encompasses both machine learning and deep learning. It essentially includes any algorithm that mimics human intelligence. Other than machine learning and deep learning, this can come in the form of rule-based systems where the algorithm is given a set of rules to deduce the result.

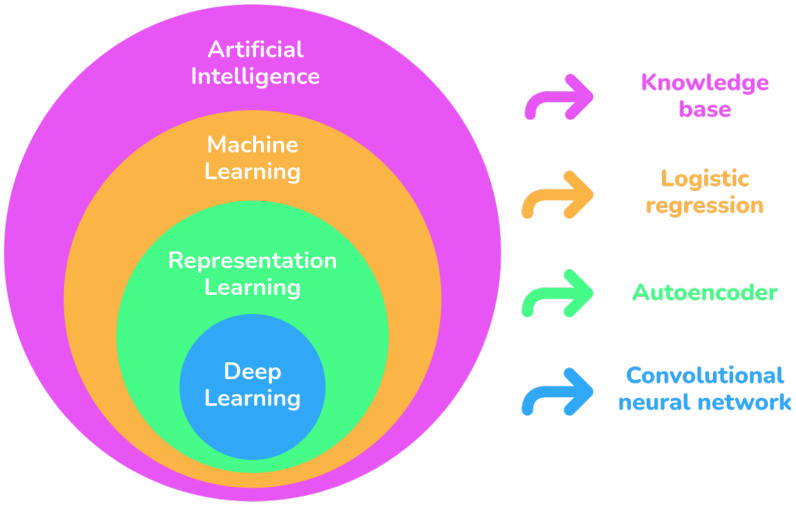


Figure 5.1: A Venn diagram with artificial intelligence as the overarching term. An example for each subset is given. Deep learning is a subset of representation learning, representation learning is a subset of machine learning, and machine learning is a subset of artificial intelligence.

5.1.1.2 Machine learning

Machine learning allows algorithms to increase their performance on a task based on new information, without having to explicitly program the algorithm. An example of machine learning is a decision tree, which is explained in Section 5.4. In machine learning, the input features can be hand-designed features or can be discovered by the algorithm itself.

5.1.1.3 Representation learning

Representation learning is about training an algorithm to find a representation of the raw input data that is effective for learning (Bengio et al., 2012; LeCun et al., 2015). This can be as part of a classifier that learns features on its own, or as an algorithm that only deals with finding a good representation of the data. For example, if the input is an image, one wants to downscale the input to a lower dimension which removes noise and represents the data without loss of information. Using this representation as the input data to a machine learning classifier then will result in a better performance than if the original input data had been used. This process extracts the most important information from the input data while downscaling it to lower dimension. This allows the architecture of the final classifier to be less complex which in turn makes for more efficient training.

5.1.1.4 Deep learning

Deep learning is machine learning based on layers of neural networks. Here, the algorithm trains itself to learn data representations of different levels of abstractions (LeCun et al., 2015). This is achieved

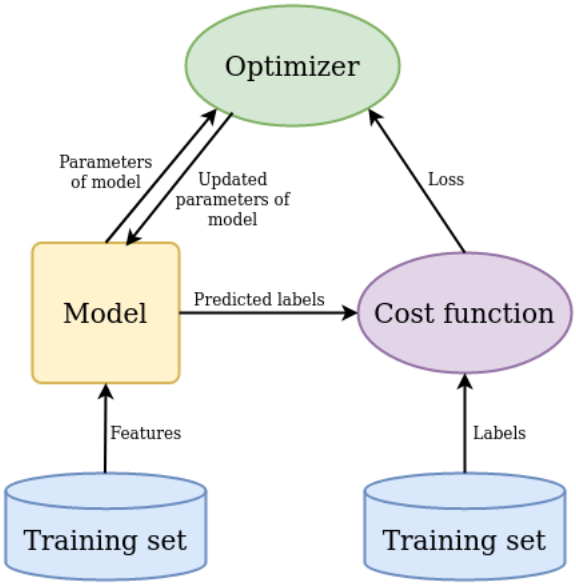


Figure 5.2: Building blocks of machine learning.

by a network of a hierarchy of layers (Cartwright, 2015). The idea is to mimic neurons of the human brain. Examples of this are convolutional neural networks and deep neural networks. Examples of applications are object detection and speech recognition (Krizhevsky et al., 2017; Russakovsky et al., 2014; Szegedy et al., 2014; LeCun et al., 2015).

For instance, in an image of a person, the first neural layer might detect edges, corners and contours. Going into deeper layers one finds more complex features such as the nose, eyes and ears, and at even deeper layers one encounters more abstract features that are difficult to interpret but are vital to a good classification/regression result.

In deep learning, features can be distributed across layers of multiple scales (Hinton et al., 1986). For example, one neuron can be attributed to many concepts, and one concept can be represented by many different neurons. Whereas local representation is a one-to-one mapping, distributed representation is a many-to-many mapping.

For instance, we could have separate neurons for colour (‘blue’, ‘green’) and shapes (’rectangle’, ’circle’). In case of a green rectangle, both the neuron that is associated with ‘green’ and the one that is associated with a ‘rectangle’ are fired. In a local representation, we would instead have a separate neuron that is associated with a ‘green rectangle’. We can therefore see that the distributed representation is much more memory efficient, especially at higher dimensions. It is also robust to unseen or outlier data. For example, if we had a red ellipse, then the local representation would not be able to deal with that. The distributed distribution, however, could represent a red ellipse by firing the ‘red’, ‘circle’ and ‘rectangle’ neuron all together.

5.1.2 Building blocks of machine learning

In order to build an ML classifier, one needs four ingredients (Goodfellow et al., 2016):

- a dataset
- a cost function
- an optimization procedure
- a model

The input data can be anything from a set of hand-picked features to raw images. The model is the classifier we want to use, such as a random forest or a convolutional neural network.

The cost function measures how well our classifier performs on the training set. It assesses how much the empirical data distribution differs from our model distribution. The lower its value, the better the performance of the classifier. Weights for each class can be included to penalize an incorrect classification of an important class more than of a less important one. This is useful when dealing with an imbalanced dataset: It gives a large penalty for incorrectly classifying a candidate of the minority class, which will result in a large value for the cost function.

The optimization procedure then updates and improves the parameters of our model such that the cost function is minimized. It continues to update the parameters until a stopping criteria has been reached or convergence has been achieved.

This is illustrated in Figure 5.2. First, the features of the training set are fed to a model. In this case we have a classification problem. The model then predicts the labels belonging to the input samples, and feeds those to the cost function. The cost function compares the predicted labels to the actual labels of the training data and computes a loss value. This loss is then fed to the optimizer. The optimizer uses this loss together with the parameters of the model in order to update those parameters. Finally, it sends the updated parameters back to the model, and the cycle begins again and continues until a stopping criteria has been fulfilled.

5.1.3 Supervised learning

In this project, we use supervised learning. There other main classes of learning techniques are unsupervised and semi-supervised learning. They are summarised in Table 5.1.

Learning technique	Input	Function to learn
Supervised	$\{(x, y)\}$	$f : x \mapsto y$
Unsupervised	$\{x\}, K$	$f : x \mapsto k \in \{1, \ldots, K\}$
Semi-supervised	$\{(x_l, k_l)\}_{l \leq n}, \{x_m\}_{m>n}$	$f : x \mapsto k \in \{1, \ldots, K\}$

Table 5.1: Supervised vs unsupervised vs semi-supervised learning. Supervised techniques take both the data x and their output y, whereas unsupervised learning takes in the data x and the number of clusters K. Supervised learning then learns to map an input x into output y, whereas unsupervised learning clusters input x into one of the K groups. An example of semi-supervised learning shown here is to have n labelled examples, where sample x_l belonging to class $k_l \in \{1, \ldots, K\}$. The group representations are then updated via unsupervised samples $\{x_m\}$, where $m > n$. An example would be to input images of known objects such as galaxies, stars and quasars. There are especially few known high redshift quasars. The data with known labels define prototypes for each class. Unlabelled data can then refine these prototypes.

Supervised learning techniques use data points together with their corresponding class labels to train the classifier. We are interested in finding a function f that maps input x into an output y. The input in supervised learning therefore are pairs of x and y data points so that the ML algorithm can find function f that maps these points,

$$f : \mathbb{R}^n \longrightarrow \mathbb{R}^m$$
$$x \longmapsto y$$

In contrast, unsupervised techniques are only fed the data points without their labels. And semi-supervised learning algorithms use a mixture of labelled and unlabelled data as their input.

5.1.4 Classification

The goal of a classification task is to predict which class a certain input belongs to. Typically one has at least 2 classes to distinguish the input data from. An example of a classification task could be to predict the shape of the input data, where the classes are 'circular' and 'rectangular'. The machine learning classifier cannot understand 'circular' or 'rectangular'. Instead, the classes are encoded into integer numbers such that class 'circular' is represented by the class label 0, and class 'rectangular' is represented by the class label 1. Other examples of a classification task are

- Quasars: 'contaminant' (0) or 'quasar' (1)
- COVID test: 'negative' (0) or 'positive' (1)
- Weather: 'sunny' (0), 'cloudy' (1), 'rainy' (2) or 'snowy' (3)

- Dog breed: 'Spaniel' (0), 'Golden Retriever' (1), 'Australian Shepherd' (2)

In a supervised classification problem, the input per sample consists of a feature vector plus the class label. The output dimension depends on the machine learning classifier. One can have as many outputs as there are classes. In a binary classification problem, one can stick to one output instead. In general, the output is the probability that the input sample is of a certain class. Keep in mind that this probability is entirely dependent on the training data and hence is less accurate than e.g. a probability distribution obtained via Bayesian inference (see Section 5.2). A (user-defined) threshold then decides what class one should attribute the input sample to.

In a multi-class problem of n classes and m inputs, the dimension of one's input labels y is $m \times n$. If k_i represents the class label of sample $i \leq m$, then the labels y are given by

$$\text{For} \quad i = 0, \ldots, m-1;\ j = 0, \ldots, n-1;\ k_i \in \{0, \ldots, n-1\}: \tag{5.1}$$

$$y_{ij} = \begin{cases} 1, & k_i = j, \\ 0, & k_i \neq j. \end{cases} \tag{5.2}$$

Each sample i has an $1 \times n$ label vector where each element is 0 except for the one which corresponds to the class of that sample. This element then has a value of 1.

5.2 Machine learning vs Bayesian statistics

In machine learning, the classifier finds point estimates. But in Bayesian statistics, one infers whole probability distributions.

5.2.1 Machine learning: Parameter point estimate

Using point estimates means that we predict only a single value of a parameter, the one which we find to be most likely / the best estimate. We do not take into account uncertainties or other parameter values. If our data is given by $X_{\text{train}} = \{x_j\}_{j=1,\ldots,m}$, then the point estimate $\hat{\boldsymbol{\theta}}$ is given as a function g of these data points,

$$\hat{\boldsymbol{\theta}}_k = g(x_1, \ldots, x_k), k \leq m. \tag{5.3}$$

Examples of point estimates are the maximum likelihood estimate (Section 4.4.1) and the maximum a posteriori estimate (Section 4.4.2). Typically, in machine learning we want to minimise a loss function. We therefore rewrite the maximisation of a function as a minimisation task instead.

The standard assumption in ML is that the data is i.i.d. (Goodfellow et al., 2016), although this is not necessarily the case in real-life applications. Relaxing these assumptions can lead to more accurate

results, with the downside of being more computationally intensive (Dundar et al., 2007). Here, we also take the i.i.d. assumption. Our training and test sets are both drawn from a data generating distribution $p_{\text{data}}(x)$ such that $X_{\text{train}} \sim p_{\text{data}}$ and $X_{\text{test}} \sim p_{\text{data}}$. This data distribution p_{data} is unknown. Using our machine learning algorithm, we estimate $p_{\text{data}}(x)$ with our model distribution $p_{\text{model}}(x, \boldsymbol{\theta})$. This model distribution depends on x and also on the parameters $\boldsymbol{\theta}$ of our model. Ideally, we would like to have $p_{\text{model}} \sim p_{\text{data}}$.

5.2.1.1 Negative log likelihood

Let us look at the maximum likelihood estimate from Section 4.4.1. Firstly, as multiplications of many small values can lead to underflows, and as maximising the likelihood gives back the same $\boldsymbol{\theta}$ than if we maximised the log likelihood, we take the logarithm of (4.51) to obtain

$$\boldsymbol{\theta}_{ML} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \sum_{i}^{m} \log p_{\text{model}}\left(x_i|\boldsymbol{\theta}\right). \tag{5.4}$$

The middle term looks similar to the expectation value, and hence we can write

$$\boldsymbol{\theta}_{ML} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, \mathbb{E}_{X\sim\hat{p}_{\text{data}}} \log p_{\text{model}}\left(X|\boldsymbol{\theta}\right) \tag{5.5}$$

Here we introduced the empirical data distribution $\hat{p}_{\text{data}}$ because we do not know the true data distribution.

Transforming the maximum likelihood estimate into a minimisation problem gives rise to the minimisation of the negative log likelihood (NLL) loss function,

$$\boldsymbol{\theta}_{NLL} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} -\mathbb{E}_{X\sim\hat{p}_{\text{data}}} \log p_{\text{model}}\left(X|\boldsymbol{\theta}\right). \tag{5.6}$$

In supervised learning, our model distribution becomes a conditional likelihood of label y given input x and parameters $\boldsymbol{\theta}$,

$$\boldsymbol{\theta}_{\text{NLL}}^{\text{Supervised}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} -\mathbb{E}_{X\sim\hat{p}_{\text{data}}} \log p_{\text{model}}\left(Y|X,\boldsymbol{\theta}\right) = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} -\sum_{i}^{m} \log p_{\text{model}}\left(y_i|x_i,\boldsymbol{\theta}\right). \tag{5.7}$$

5.2.2 Bayesian statistics: Parameter distribution

Compared to single point estimates, in Bayesian statistics we obtain a whole probability distribution $p(\boldsymbol{\theta}|X)$ over our parameters. This is much more powerful than just finding the maximum of a function. We can still obtain the most likely parameters, but we can additionally obtain the error of

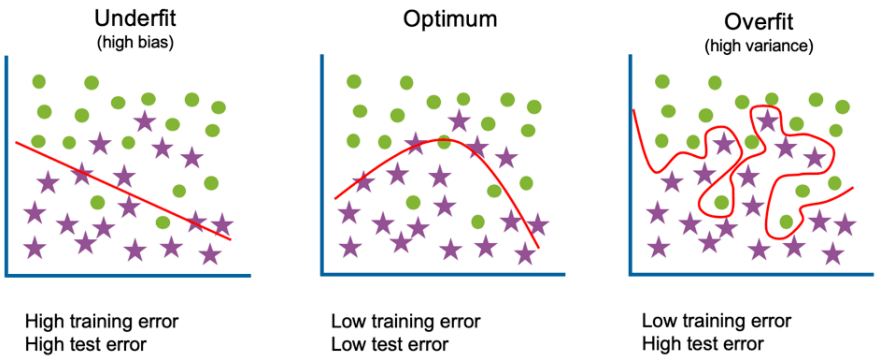


Figure 5.3: An illustration of underfitting vs overfitting. Figure taken from IBM Cloud Education (2021).

this parameter estimate as well. The point estimates above (ML, MAP) are actually derived from Bayesian principles. A detailed analysis of Bayesian statistics is discussed in Chapter 4, with derived point estimates being listed in Section 4.4.

5.3 The difficulty of generalisation

A classifier that performs well on both the training and any unseen test set is said to 'generalise' well (LeCun et al., 2015). One trains the classifier on the training set, but tests the classifier on the unseen test set. The question is how one can influence the results of the test set when the classifier is only aware of the training set during training. The short answer is that based on the loss from both the training and the test set, we adjust the training of the algorithm.

5.3.1 Underfitting and overfitting

Our goal is to

1. Reduce the training error
2. Reduce the difference between the training error and the test error

If 1. is violated, we have a large training error. This is called *underfitting*. If 2. is violated, the test error is much larger than the training error. This is called *overfitting*. We therefore want to find the sweet spot between underfitting and overfitting.

Overfitting and underfitting are illustrated in Figure 5.3. The plot shows two classes (green dot, purple star) that we would like to separate by a function. On the left, we assume a linear model. This model

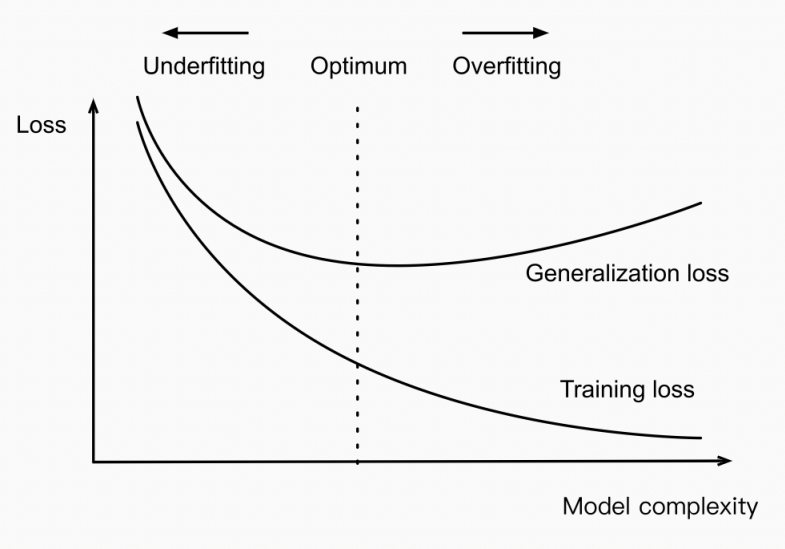


Figure 5.4: An illustration of underfitting vs overfitting. Figure taken from Zhang et al. (2021).

does not fit the data well. Our training error will be large which means we are underfitting the data. In the middle, we can see the optimum model. It fits the training data well, but not too well to not be generalisable to a new data set anymore. Both training and test error are small. This is the model we should pick for this data set. On the right, we can see a very complex model. It fits the training data perfectly, but because of this over-complexity it fails to fit the test set. This model has a low training error, but a high test error.

This is further exemplified in Figure 5.4. The x-axis represents the complexity of the model. For instance, a linear model has a low complexity, whereas a 100-order polynomial has a large complexity. The y-axis represents the loss which is explained in more detail in Section C.1.1. Essentially, a large loss means that the data is not fit well by our model. Ideally, we would have a loss close to zero. As we can see, the training loss decreases with increasing complexity as expected. For instance, a high-order polynomial will always fit the data better compared to a linear model. The generalisation loss, however, initially decreases, hits a minimum at the optimal model and then starts to increase again. While the generalisation loss initially decreases, we are underfitting the data (see left plot in Figure 5.3). After we have reached the model that fits the test data best, the complexity of the model starts to increase. The more the generalisation loss increases, the less generalisable the model becomes (right plot in Figure 5.3).

This is similar to the difference between maximum likelihood and maximum a posteriori estimates. Whereas the former only includes the likelihood, the latter considers the prior distribution as well. A prior could penalize a high-order polynomial such that one does not choose the most complex model and hence finds a good balance between low training error and low test error. This penalisation does not exist in the likelihood, and hence one would choose the most complex model as that fits the training data best.

5.3.2 Bias and variance

Bias and variance are important measures in supervised machine learning (see Figure 5.3). Bias represents the difference in predicted outcome and actual outcome. It can be seen as an incorrect assumption of the classifier. Variance represents the difference in outcome for similar input samples. If too many unnecessary details are modelled by the classifier, then the variance will be large.

5.3.2.1 Bias

Bias is the difference between the expected function value $\hat{f}(x)$ (point estimate) and the true function $f(x)$,

$$\mathcal{B}(x) = \mathbb{E}\left[\hat{f}(x)\right] - f(x). \tag{5.8}$$

An estimator is unbiased if $\mathbb{E}\left[\hat{f}(x)\right] = f(x)$. In terms of parameters $\boldsymbol{\theta}$, one has a point estimate $\hat{f}(x) = \hat{\boldsymbol{\theta}}$ and a true parameter value $f(x) = \boldsymbol{\theta}$.

Generally, the bias decreases with complexity. In the case of underfitting, we have a high bias, as our model is not complex enough to properly differentiate between the two classes. Overfitting has a low bias, as it perfectly fits the training data.

5.3.2.2 Variance

Variance is defined by how much we expect the classifier to vary given a new dataset,

$$\mathcal{V}(x) = \mathbb{E}\left[\mathbb{E}\left[\hat{f}(x)\right] - \hat{f}(x)\right]. \tag{5.9}$$

Generally, the variance increases with complexity. While we expect the variance to be low for underfitting, it will be large for models that overfit. For example, the classifier may model the noise of the training data. This noise will be different for each sample, and hence will result in a high variance.

At the optimum between underfitting and overfitting, both bias and variance are low. This is the so-called 'bias-variance tradeoff'. This is in alignment with Occam's razor approach, which states that one should choose the simplest model that explains the data well (Hamilton & Turnbull, 1853). Note that the 'bias-variance tradeoff' is not necessarily applicable in deep learning, for which the difference between the training and test error is often very small (Zhang et al., 2016).

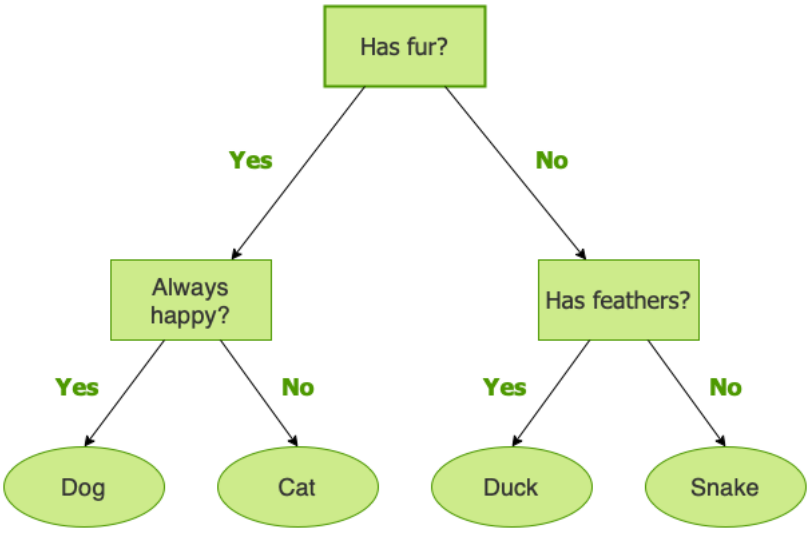


Figure 5.5: An example of a decision tree. The root node is at the top, and the leaf nodes are in oval at the bottom. It has a fixed depth of 2.

5.4 Decision Trees

A decision tree is illustrated in Figure 5.5. It reminds of an upside down tree. We start at the root at the top and go down towards the leaves. One of the reasons for its popularity is that it is easy to visualise and hence easy to interpret. This is unlike many other machine learning algorithms which act like black boxes due to their increased complexity. It can be a useful tool to understand the training set better. E.g., what feature is used as the root node - and does it makes sense/is it actually an important feature? Their advantages and disadvantages are shown in Figure 5.6. This will become clearer throughout this section and will be discussed at the end.

5.4.1 Leaf, node, root

A decision tree consists of nodes, branches and leaves (Quinlan, 1986). The root is the starting point of the tree, and is the parent node of every other node. There can only be one root node in each decision tree. In a binary decision tree, each node can have at most 2 child nodes. Nodes are connected via branches. Essentially, a node represents a variable that questions a certain condition, and the branches represent the possible outcomes of that variable. The endpoints of the tree are marked by the leaves. These represent the final classification label. To sum up, we start at the top, walk through the tree and end up at one of the leaves - the final destination.

Let us look again at the decision tree in Figure 5.5. We assume that each sample of our input data belongs to one of the classes 'Dog', 'Cat', 'Duck' or 'Snake'. The depth of the tree has to be picked beforehand. In this case, we have a depth of 2 which means we have to split 2 nodes to get to a leaf (classification label). The root node checks whether the input sample has a fur or not. The possible

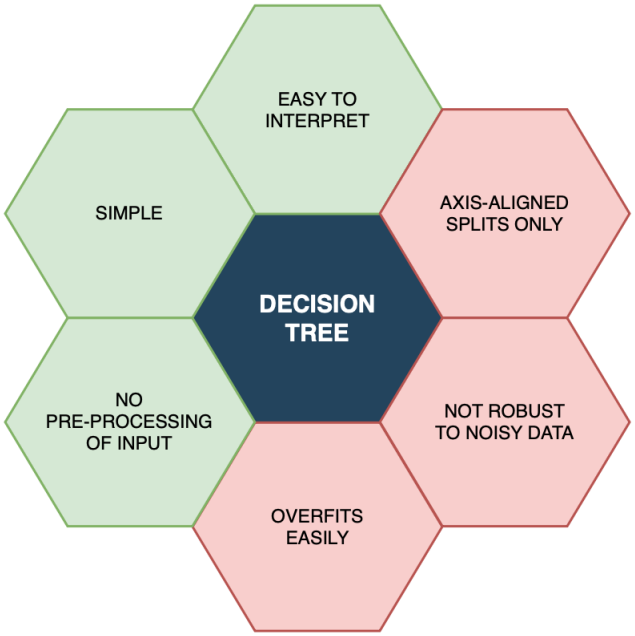


Figure 5.6: Advantages (green) and disadvantages (red) of decision trees.

outcomes are 'Yes' and 'No'. This is an example of a categorical variable, with a discrete set of outcomes. For a continuous variable, one selects discrete non-overlapping sets of possible outcomes. For example, one could split up a variable $x \in \mathbb{R}$ into $x < 1$, $1 \leq x < 2$ and $x > 2$.

If the answer to 'Has fur?' is 'Yes', we follow the left-side branch. If the input sample is not furry, then we follow the right-side branch which represents the answer 'No'. The left subtree then has a node that asks if the sample is always happy. If yes, then we follow the left-hand side towards the leaf 'Dog'. If not, then it is classified as a cat via the leaf node 'Cat'. Similarly, the right subtree contains the node 'Has feathers?' and depending on the outcome ('Yes' or 'No') we arrive at the leaf 'Duck' or 'Snake'.

5.4.2 Function representations

The setup of a decision tree enforces axis-aligned splits. This limits the range of functions it can represent, although it can approximate any function if using a deep enough tree. This is illustrated in Figure 5.7. In this example, the root node corresponds to $\boxed{x_1 < s_1?}$, which splits up the data into two subtrees. This is marked as the root split in the plot. The left subtree consists of samples where $x_1 < s_1$, and the right subtree of samples where $x_1 \geq s_1$. Next, we split up the left subtree via the node $\boxed{x_2 < s_2?}$. Within this subtree, this splits up the data perfectly into purple polygons and yellow stars. The purple polygon correspoonds to the left leaf, and the yellow stars to the right leaf. This subtree is finished. Now let's look at the right subtree of the root node. Here, we have a node that splits up the data as $\boxed{x_2 < s_3?}$. Similarly to before, the left subtree of this subtree consists of samples with $x_2 < s_3$, and the right subtree of samples with $x_2 \geq s_3$. As we can see in the plot, the samples with $x_2 \geq s_3$

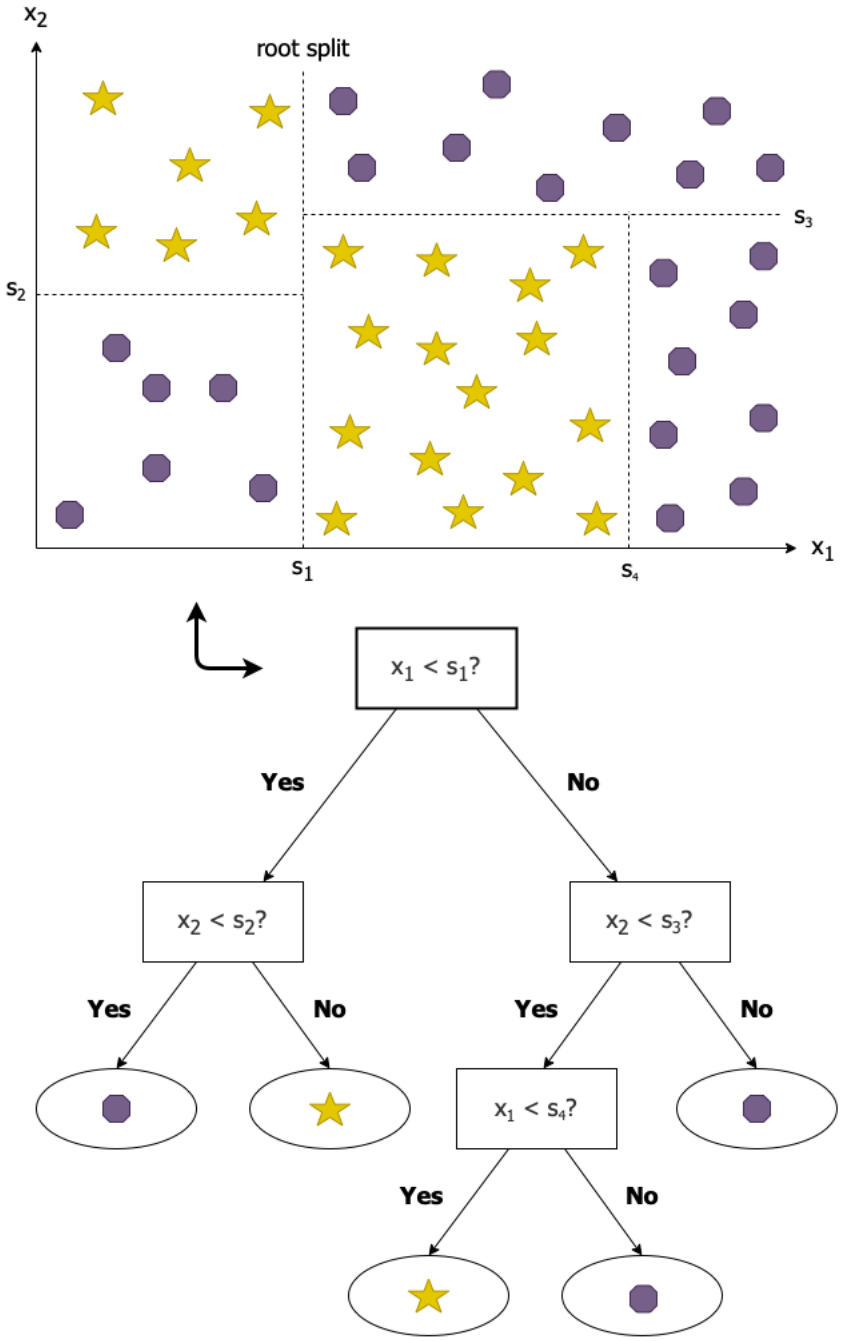


Figure 5.7: An illustration of how a decision tree splits up a data set.

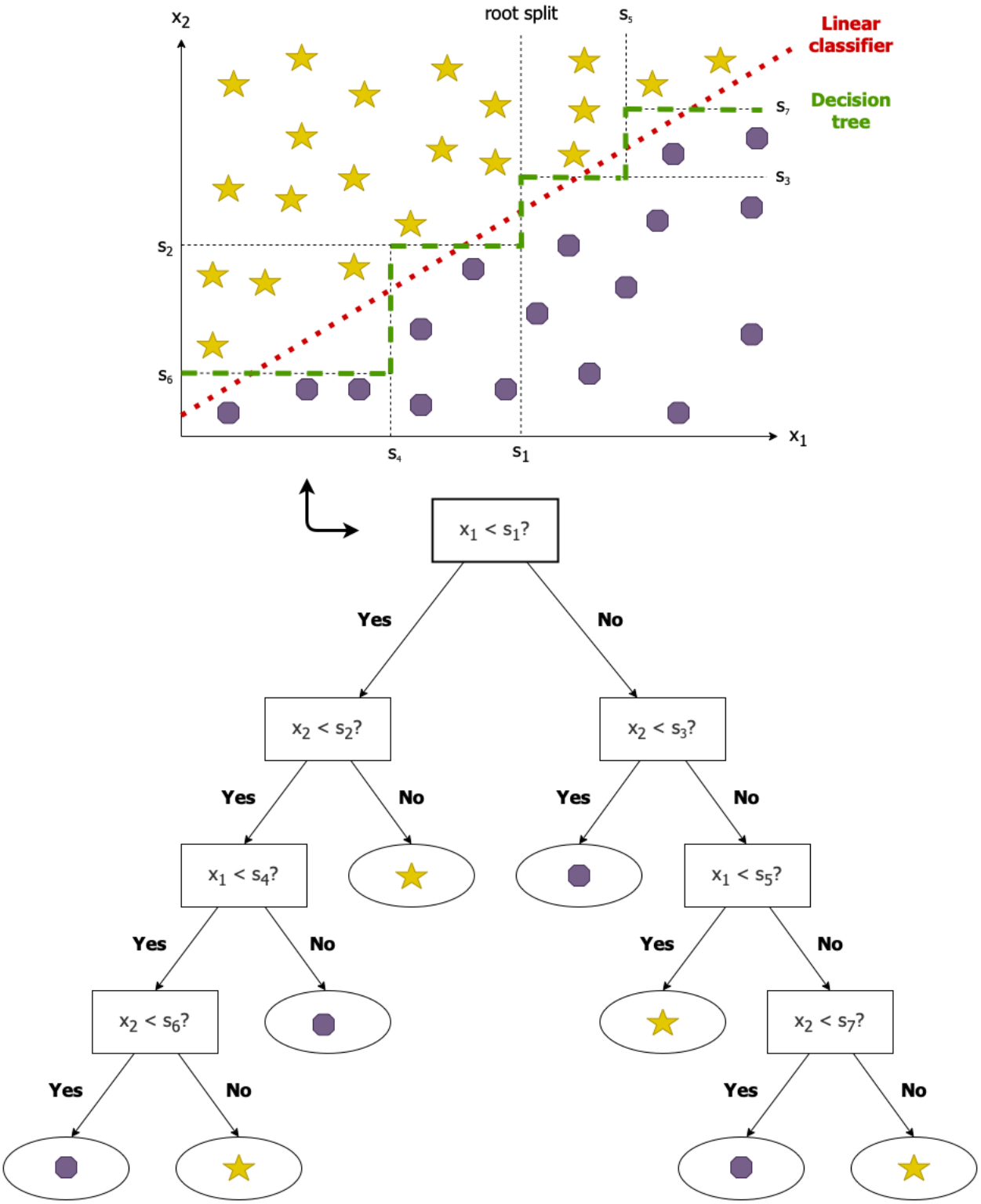


Figure 5.8: An illustration of how inefficient a decision tree splits up a linearly separable data set.

only consist of purple polygons, and hence we can put a leaf there. Samples with $x_2 < s_3$, however, have to be split up again in order to differentiate between the two classes. This is done with a node that splits up the data into groups $x_1 < s_4$ and $x_1 \geq s_4$. This splits up the data perfectly again, and we can write down the leaves yellow star for the former group and purple polygon for the latter group. This tree is now finished.

One disadvantage of an axis-aligned split is that a decision tree handles linearly separable data very inefficiently compared to e.g. logistic regression, as can be seen in Figure 5.8. Using logistic regression, we can quickly find a linear function that perfectly separates the data set. The result of logistic regression is drawn in red. The decision tree, however, is incapable of finding this linear decision boundary. It can only make splits aligned with feature axes x_1 and x_2. A linear separation therefore looks more like a zig-zag pattern in decision trees, as is shown in green. The deeper the tree and the more samples present, the more the decision boundary will resemble a linear line. One could also instead try to re-parameterise the features such that the two classes can be easily split along the axis. However, this requires a deeper understanding of the input data set. Plus, the beauty of decision trees is that one does not have to apply any pre-processing steps (i.e., normalisation) to the data and so this advantage would be lost.

5.4.3 Node splitting functions

In this section, we discuss how a decision tree decides which combination of feature and splitting value is used at what depth of the tree. The root node typically is the most important feature as it can split up the data best. But how do we know what feature to use as the root node, and at value to split it? A decision tree will, at each level, split up each feature at each possible value and select the feature (and splitting value) that splits up the data best. This is calculated using either the Information gain or the Gini impurity.

In general, the Gini impurity is the preferred node splitting function as it does not contain any logarithm. This makes it slightly quicker to compute, though the difference is negligible. Raileanu & Stoffel (2004) test how often the Gini impurity and the information gain disagree in a split, and found this frequency to be on average less than 2%.

5.4.3.1 Information Gain

Information gain is essentially the expected reduction in entropy due to a feature split. Entropy H measures the homogeneity of our dataset X (Mitchell, 1997), and is defined as

$$H(X) = \sum_{i=1}^{N_c} -p_i \log_2 p_i \,, \tag{5.10}$$

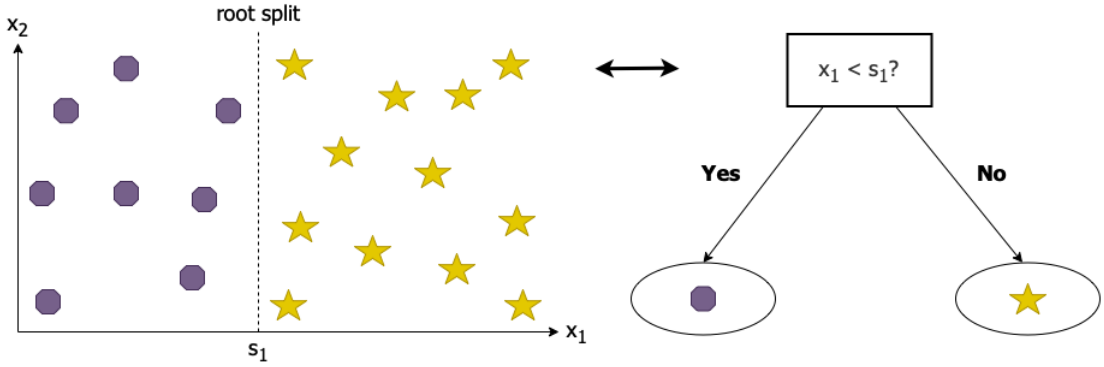


Figure 5.9: A simple dataset that can be separated with one split.

where N_c the number of classes and p_i the fraction of samples in the split belonging to class i. For $N_c = 2$, entropy ranges between 0 and 1. It is measured in units of bits. Given a feature and a splitting value, the entropy tells us how pure or impure the divided data is. A perfect decision tree would split up the data perfectly, and hence each subset that emerges from the split should have samples of only one class each. That is,

$$\forall \text{subtree} \;\; !\exists j \leq N_c : p_j = 1 \text{ and } p_{i \neq j} = 0, \; i = 1, \ldots, N_c, \tag{5.11}$$

where a subtree is defined as the subtree that emerges from a split in a feature. We know that $\log_2(1) = 0$, which leads to an entropy of zero. It also follows from l'Hôpital's rule (de L'Hospital, 1696) that $0 \cdot \log_2(0) = 0$. Hence, our aim is to minimize the entropy H as this either maximises or minimizes the class proportions $\{p_i\}_i$.

Let us say we had a sample as in Figure 5.9. Before we climb down the tree, we calculate the entropy of the sample before the split,

$$\begin{aligned} H(\text{before split}) &= -\frac{8}{20} \log_2\left(\frac{8}{20}\right) \text{ bits} - \frac{12}{20} \log_2\left(\frac{12}{20}\right) \text{ bits} \\ &= 0.529 \text{ bits} + 0.442 \text{ bits} \\ &= 0.971 \text{ bits} \end{aligned} \tag{5.12}$$

This is the entropy of the overall dataset.

Here, we have a dataset that can be easily separated with feature x_1 at value s_1. This splits up our dataset in one subtree of 8 purple polygons, and another subtree of 12 yellow stars. Let us define purple polygons as class 1, and yellow stars as class 2. We first calculate the entropy of the left branch,

$$\begin{aligned} H(\text{left subtree}) &= -p_1 \log_2 p_1 \text{ bits} - p_2 \log_2 p_1 2 \text{ bits} \\ &= -1 \log_2(1) \text{ bits} - 0 \log_2(0) \text{ bits} \end{aligned} \tag{5.13}$$

$$= \quad 0 \text{ bits.}$$

Similarly, the right branch also gives an entropy of zero, $H(\text{right subtree}) = 0$. The calculation is the same, with the values of the class distributions swapped. The overall entropy of the split is then given by the weighted average

$$\begin{aligned} H(\text{split}) &= \frac{\#\text{s of left subtree}}{\#\text{s before split}} H(\text{left subtree}) + \frac{\#\text{s of right subtree}}{\#\text{s before split}} H(\text{right subtree}) \\ &= \frac{8}{8+12} \times 0 \text{ bits} + \frac{12}{8+12} \times 0 \text{ bits} \\ &= 0 \text{ bits}, \end{aligned} \tag{5.14}$$

where #s stands for the number of samples.

The information gain is given as the difference in entropy between before and after the split,

$$\begin{aligned} G(\text{split}) &= H(\text{before split}) - H(\text{split}) \\ &= 0.971 \text{ bits} - 0 \text{ bits} \\ &= 0.971 \text{ bits} \end{aligned} \tag{5.15}$$

Our aim is to choose a feature and dividing value that maximises the information gain.

More formally, information gain for data X and feature A with split value a is defined as (Mitchell, 1997)

$$G(X, A) = H(X) - \sum_{v \in \text{Node}(A)} \frac{|X_v|}{|X|} H(X_v), \tag{5.16}$$

where Node(A) the set of possible outcomes for the current node that splits feature A, and X_v defined as the number of samples where feature A has a value of v (categorical feature), or feature A is in the range given by v (continuous feature). If we have m samples in our dataset and A is a continuous feature, there are $m - 1$ possible splits one can make. For large datasets, this can be quite a large number of splits to go through, and one can instead use selected percentiles of the distribution of values of feature A.

Let's take another example, this time our decision tree of Figure 5.7. The entropy of the complete dataset X, before splitting the root node, is given by

$$\begin{aligned} H(X) &= -\frac{23}{44} \log_2\left(\frac{23}{44}\right) \text{ bits} - \frac{21}{44} \log_2\left(\frac{21}{44}\right) \text{ bits} \\ &= 0.999 \text{ bits}, \end{aligned} \tag{5.17}$$

where we have 23 purple polygons and 21 yellow stars in our sample, so a total of 44 samples. Splitting feature x_1 at the root node at value s_1, we receive two subtrees. The information gain of this

split is then given by

$$
\begin{aligned}
G(X, x_1) &= H(X) - \frac{|X_{x_1<s_1}|}{|X|} H(X_{x_1<s_1}) - \frac{|X_{x_1\geq s_1}|}{|X|} H(X_{x_1\geq s_1}) \\
&= = H(X) - \frac{12}{44} H(X_{x_1<s_1}) - \frac{32}{44} H(X_{x_1\geq s_1}) \\
&= H(X) - \frac{12}{44}\left[-\frac{6}{12}\log_2\left(\frac{6}{12}\right) - \frac{6}{12}\log_2\left(\frac{6}{12}\right)\right] \text{ bits} \\
&\quad -\frac{32}{44}\left[-\frac{17}{32}\log_2\left(\frac{17}{32}\right) - \frac{15}{32}\log_2\left(\frac{15}{32}\right)\right] \text{ bits} \\
&= 0.999 \text{ bits} - 0.273 \text{ bits} - 0.725 \text{ bits} \\
&= 0.001 \text{ bits}
\end{aligned}
\tag{5.18}
$$

This is a low information gain, but that is because there is no feature that perfectly splits the data at the root level. Next, let's look at the node that uses x_2 with split value s_2, which is located in the left subtree of the root node. Again, let's calculate the information gain

$$
\begin{aligned}
G(X^{\text{left}}, x_2) &= H(X^{\text{left}}) - \frac{|X^{\text{left}}_{x_2<s_2}|}{|X^{\text{left}}|} H(X^{\text{left}}_{x_2<s_2}) - \frac{|X^{\text{left}}_{x_2\geq s_2}|}{|X^{\text{left}}|} H(X^{\text{left}}_{x_2\geq s_2}) \\
&= 0.001 \text{ bits} - \frac{6}{12}\left[-\frac{6}{6}\log_2\left(\frac{6}{6}\right) - 0\right] \text{ bits} \\
&\quad -\frac{6}{12}\left[-0 - \frac{6}{6}\log_2\left(\frac{6}{6}\right)\right] \text{ bits} \\
&= 0.001 \text{ bits} - 0 \text{ bits} - 0 \text{ bits} \\
&= 0.001 \text{ bits}
\end{aligned}
\tag{5.19}
$$

Here, there is no reduction in information gain, as the node perfectly splits up the data. Had we tried out a different splitting value that didn't separate the two classes, then the entropy would have been non-zero. Hence, the information gain would have decreased and therefore the decision tree would not have selected that split.

5.4.3.2 Gini impurity

Gini impurity is a measure of how likely it is that a new sample that follows the class distribution caused by the split is misclassified by the decision tree (Sany Laber & Murtinho, 2018). Gini impurity is defined as

$$
I = \sum_{i=1}^{N_c}\left(p_i \sum_{k\neq i}^{N_c} p_k\right) = \sum_{i=1}^{N_c} p_i\,(1 - p_i)\,, \tag{5.20}
$$

where again N_c the number of classes in the dataset and p_i the probability of a sample belonging to class i. Therefore, p_i is the probability of correctly assigning label i to a new sample of that class, and $1 - p_i$ the probability of misclassifying it. Again, its minimum is reached when the subtree consists

of only one class j such that $p_j = 1$ and $p_{i \neq j} = 0$. While entropy lies in the range $[0, 1]$ for $N_c = 2$, the Gini impurity lies in the range $[0, 0.5]$ for $N_c = 2$. In that case, the minima lie at $p_i = 0$ and $p_i = 1$, and the maximum lies at $p_i = 0.5$.

5.4.4 Advantages and disadvantages of decision trees

Advantages and disadvantages of decision trees can be seen in Figure 5.6. Decision trees are simple and easy to visualise. This makes them easy to interpret. When plotting a decision tree, one can immediately see whether the feature that was chosen as the root node makes sense or not. One can then adjust the training set accordingly, and can quickly detect and remove redundant features. Another advantage of decision trees is that they do not require any data preparation, hence neither feature scaling nor centering has to be applied as a pre-processing step (Géron, 2017).

Decision trees can only make axis-aligned splits and are therefore less efficient if the data is linearly separable but not axis-aligned. Although one could try to use PCA to transform the data space such that it is axis-aligned, this would lose the major advantage of decision trees that one does not have to do any pre-pocessing of the data set. Another issue with decision trees is that they can easily overfit if they are built too deep (Ho, 1995). They can be very susceptible to noisy input features. This means that small changes in the input features can lead to completely different decision trees. Lastly, at each node, features are favoured that have many possible splits (Strobl et al., 2007). This is explained in more detail in Section 5.5.3. The reason for this is that a decision tree tries out every combination of feature and splitting value at each level which means that a variable which has many possible splits will more likely lead to a higher gain than one which does not offer many splits (such as a categorical variable with two options). Typically, this means the tree favours continuous variables and categorical variables with many class options.

5.5 Random forests

A random forest is an ensemble classifier consisting of a collection of decision trees (Ho, 1995; Baron, 2019). The analogy behind ensemble learners is the *wisdom of the crowd*: in general, a crowd will be more knowledgeable than a single person. In the case of a random forest, the crowd corresponds to the several decision trees.

The *random* in random forests represents the fact that at each node split, one looks at a random subset of all the features, whereas the decision tree would look at all of the features each time. Using a random subset of all features allows for diversity in the results and therefore gives a more robust model. In addition, each decision tree in the random forest gets trained on a random subset (with replacement) of the training data.

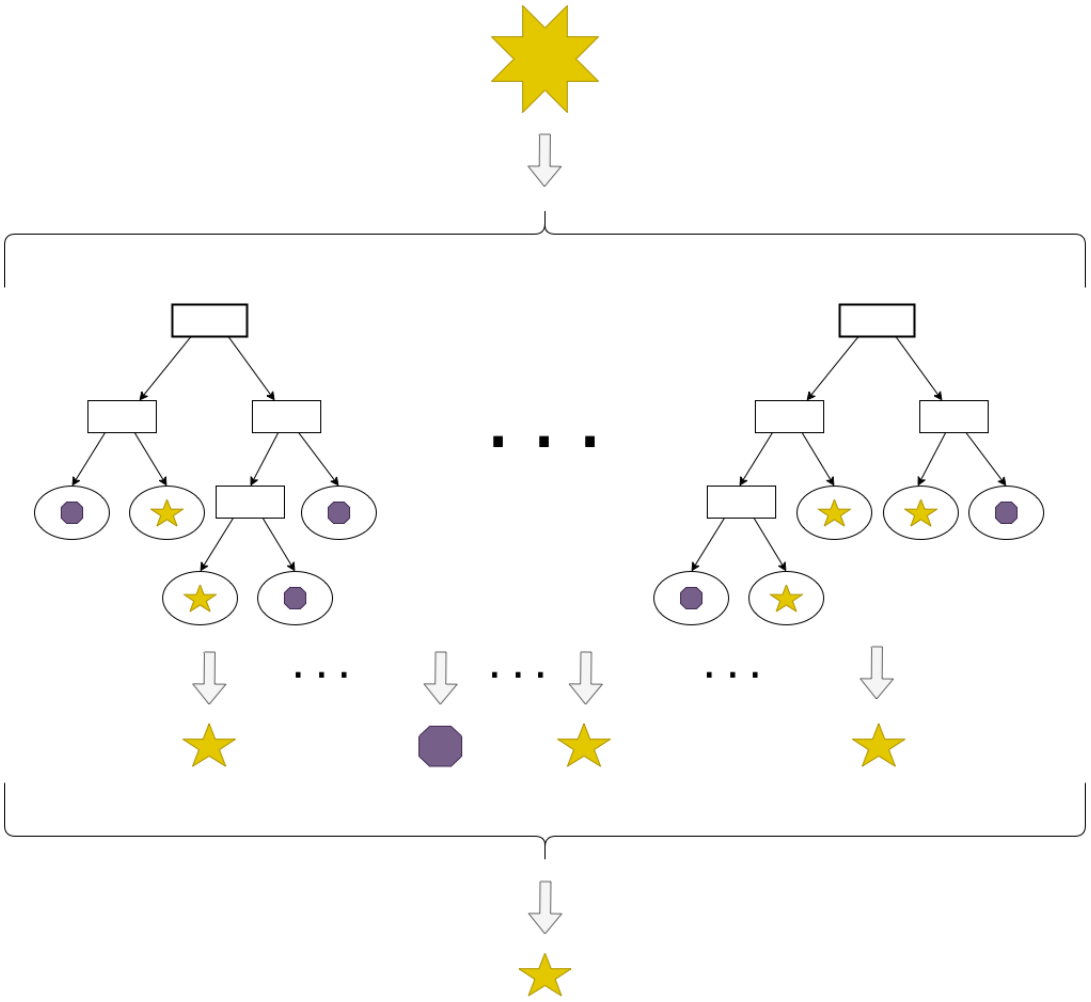


Figure 5.10: An illustration of a random forest with majority voting. The input at the top is a star. This star is fed to each individual random forest and the final classification corresponds to the class that the majority of decision trees have predicted.

There are different methods on how to decide what the final classification result should be. Typically, the final label for a new sample corresponds to the label that the majority of all the decision trees predict. This is called the *hard voting classifier*. This has the effect of reducing the variance (Géron, 2017). An example of majority voting is illustrated in Figure 5.10. The input consists of an 8-point star. Running it through the ensemble of decision trees, the majority of trees correctly classifies the input as a star, which is subsequently used as the final classification label of the random forest.

Another possibility is to use a *soft voting classifier*, whereby the outputs of each decision tree are added up and the one with the largest score is used as the final label. However, keep in mind the outputs of decision trees may be probabilities, but these are based on the input samples that tree has seen alone! Hence, one decision tree that has seen more samples of one class might be skewed towards it and this tree will then bias the overall results.

Advantages and disadvantages of random forests will be explained at the end of this section, but can be seen already in Figure 5.11.



Figure 5.11: Advantages (green) and disadvantages (red) of random forests.

5.5.1 Bootstrap aggregating

Training a classifier on a random subset of the training set with replacement is called *bootstrapping* or *bootstrap sampling*. For a classifier that contains multiple sub-classifiers, we have an individual bootstrapped data set for each sub-classifier. This is called *bootstrap aggregating* or *bagging*.

In a random forest we use bootstrap aggregating as the individual decision trees are trained on random subsets of the training set with replacement. It is one of the differences between training decision trees and random forests. Bootstrap aggregating decreases the variance without changing the bias. This is because it trains the individual trees on different subsets of the training set. The algorithm goes as follows: First, one randomly selects m data samples with replacement from the training set. Let's call this set X_b. That means one could have e.g. five copies of the same sample in X_b. This naturally means that there will be samples of the training set X that are untouched by the selected set X_b. One then trains one decision tree of the random forest on X_b. This is reiterated for each individual tree of the random forest, and hence each decision tree will be trained on a different set of samples. This helps to de-correlate the trees. A single decision tree is very sensitive to slight changes or noise in the input data. However, a collection of trees trained on different sets is not, as long as the individual trees are un-correlated. Finally, one can validate the results of the random forest on the difference set $X \backslash \{X_b\}_b$.

5.5.2 Feature bagging

The splitting function can either be the Information gain or the Gini impurity, as discussed in Section 5.4.3. For each node, the splitting function is calculated for a random subset of features. This is called *feature bagging*. It is different to the usual implementation of a decision tree, where all features are taken into account at each node. Typically, if we have N features, the subset at each node would consist of $\sqrt{N}$ samples. The reason for feature bagging is that it reduces the correlation between each tree and helps random forests to generalise well. Imagine if splitting the data by one specific feature has a very low Gini impurity / high information gain. If all trees had access to that feature, it would be the root node of every tree. As such, the individual trees would become correlated. Using a subset of features at each node decreases that correlation.

5.5.3 Random forest feature importance measure

Similarly to the decision tree, the default random forest feature importance measure is biased. Instead, one should use the permutation importance or the conditional feature importance measure explained below. Even though those methods are more computationally expensive, one will receive a more accurate picture of the most important features.

5.5.3.1 Variable importance measure

The original RF variable feature importance measure by Breiman (2001) is biased. Note that the variable importance measure is implemented by default in the RF package of the Python library `sklearn`. This measure is biased because the mean decrease in impurity is used which favours features that vary in their scale of measurement or their number of categories (Strobl et al., 2007). When calculating the best feature and threshold to use for a split in a decision tree, for every feature every threshold that splits up the data into different subsets is tested. The more often a feature is used to split the data, the more informative it should be. For example, a feature that can split the data such that on one side all samples are red and on the other side all samples are blue then the branches on both sides are pure and therefore have zero impurity (and zero entropy). This feature therefore will often be selected as a node. This will bias features that can split up the data into many different subsets compared to features that e.g. can only use one separating threshold. The chances are higher that the Gini impurity will be lower when you have more thresholds to try.

This problem is enhanced at large datasets, since then the amount of possible splits for continuous variables are increased, while the number of categories in categorical variables stays the same. It is therefore not recommended to use the built-in variable importance measure if the conditions above apply.

5.5.3.2 Permutation importance

Permutation importance is far less biased than the variable importance measure, especially when it is used with sampling without replacement (Strobl et al., 2007). The algorithm first records a baseline accuracy. Next, the values of one feature are permuted. That is, it changes the values of that feature for each sample. Finally, the importance of that feature is measured as the difference between the baseline accuracy and the drop in accuracy. This is repeated for every feature.

5.5.3.3 Drop-col importance

In the drop-col feature importance measure, a feature is dropped from the set in each iteration (Parr et al., 2018). The classifier is then re-trained and the performance is measured. If the feature that has been removed is important, one expects a drop in performance compared to the classifier in which all features are used. This will give a more accurate picture of the most important features, however it is very computationally expensive as it requires the re-training of the classifier for every dropped feature.

However, it does not work well when a subset of the features are duplicates or collinear. Let us say we used the same feature twice. That is, two features with the same information. In that case, the drop-col method will give low importance measures to both features. This is because if we drop one of the features, we still have the other feature with the same information content. A low importance measure would drop both features, when we actually only want to drop one of them. The feature itself might actually be quite important.

5.5.3.4 Conditional random forest importance

The importance measures above do not take conditional probabilities into account (Strobl et al., 2007). Take for example hospitals that treat very sick patients. They will have a higher mortality rate and more complications than hospitals that treat patients with mild symptoms. Mortality rate and the frequency of complications thus seem to be correlated,

$$P(m, c) = P(m|c)\, P(c), \tag{5.21}$$

where m the mortality rate and c the rate of complications. However, this correlation mostly disappears once one makes the probabilities conditional on the seriousness of the illness (Silber & Rosenbaum, 1997; Strobl et al., 2008). That is, we only compare the mortality rate among patients with the same severity in their illness. Thus, including the rate of complications c gives us only little additional information for predicting m and can be disregarded.

The solution to this is to use conditional random forests without replacement (without bootstrapping). The feature importance measurement is implemented by minimizing the p value of a conditional inference independence test in Hothorn et al. (2006); Strobl et al. (2007). This gets rid of biases that arise when a feature either varies in its scale of measurement or when the feature set consists both of numerical and categorical variables.

5.5.4 Advantages and disadvantages of random forests

Advantages and disadvantages of random forests are visualised in Figure 5.11. Advantages of random forests consist of the wisdom of the crowd which leads to better generalisation than decision trees due to small correlations between the individual decision trees. Similarly to decision trees, no pre-processing of the input data is necessary. In contrast to individual decision trees, random forests are robust to noisy data and are not prone to overfitting.

Compared to a decision tree, a random forest acts more like a black box and is thus more difficult to interpret. Even though the individual trees are easy to interpret, the collection of hundreds of them is much more difficult to visualise and to understand. It further has a few of the same issues as the individual decision trees, namely that the splits at each node are axis-aligned, and hence a linearly dependent data set is more easily solved via linear regression. Finally, as with decision trees, random forests are biased to continuous variables and variables with many categories.

5.6 Artificial neural networks

Artificial neural networks (ANN) try to mimic the networks of neurons in our brain in a simplistic way. Our brain contains approximately 10^{11} neurons (Mitchell, 1997). There, each neuron only fires if enough neurotransmitters have been accumulated in its cell body, otherwise it remains quiet (McCulloch & Pitts, 1990; Cross et al., 1995). This idea has been reproduced by ANNs, where the output of each neuron is applied to a threshold function. If the output of that neuron is below the threshold, the output is set to zero.

These threshold functions are not only there to mimic human neurons, they are also needed to represent non-linear output functions: ANNs are hierarchical layers of units of neurons that are connected by weights, as can be seen in Figure 5.12. The output of each layer of neurons is a linear combination of the incoming neuron outputs of the previous layer,

$$\mathbf{y} = \mathbf{X}\,\mathbf{w} + \mathbf{b}, \tag{5.22}$$

where the weights $\mathbf{w}$ of the connections represent the coefficients and $\mathbf{b}$ represent the biases. The bias

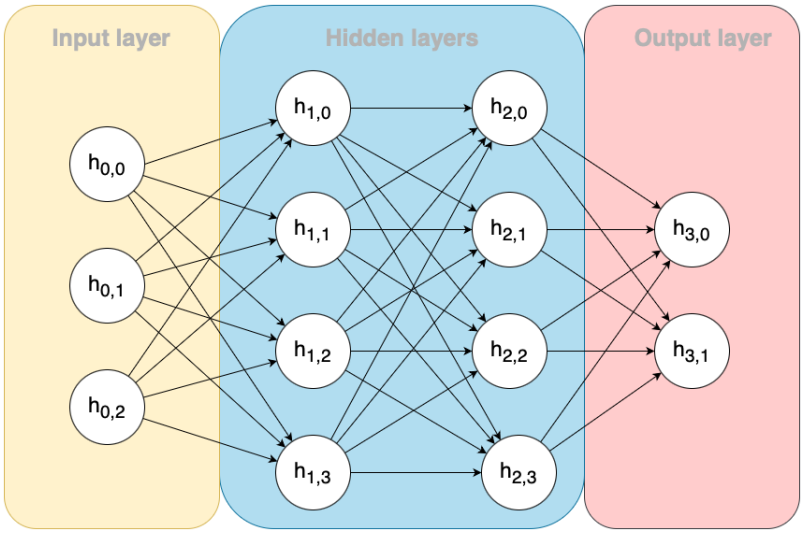


Figure 5.12: An illustration of an artificial neural network for a two-class problem. The first layer represents the input features. The last layer represents the probability (based on the seen training data) that the input is of class 1. The layers in between are hidden layers. The more hidden layers, the more complex the output function can be.

is a constant numerical value and is separate to the input. One can think of it as the intercept with the y-axis in a $2D$ linear equation $y = w \cdot x + b$. If we did not subsequently apply a threshold or non-linear function to this linear combination (equation (5.22)) of the input values, the neural network could only learn linear functions. However, running this linear combination through a non-linear function, such as that of a threshold, allows the ANN to represent non-linear functions as well (Goodfellow et al., 2016). These non-linear threshold functions are also called *activation functions*. This enables ANNs to learn complex functions, which makes them very powerful machine learning tools.

If we only had one layer of neurons in our network and used a logistic output function, the network would be equivalent to logistic regression (Cartwright, 2015). The more layers one adds to the network, the more complex the output function can be.

5.6.1 Forward propagation

Let us look again at Figure 5.12. The inputs of an ANN are given by the images h_0 of a source. The weights for each layer l are given by $\mathbf{W}_l$, and the biases are given by $\mathbf{b}_l$. Together they form the parameters of the network. These need to be updated in order for the ANN to learn the desired output function. After each layer, its output $\mathbf{a}_l$ is passed to an activation function $\mathbf{h}_l = v_l(\mathbf{a}_l)$. Let the network have a depth of D. Then, for each layer $l = 1, \ldots, D$, one calculates the following (Mitchell, 1997; Goodfellow et al., 2016):

$$\mathbf{a}_l = \mathbf{b}_l + \mathbf{W}_l \mathbf{h}_{l-1} \tag{5.23}$$

$$\mathbf{h}_l = v_l(\mathbf{a}_l) \tag{5.24}$$

In the last layer, the activation function v_D is a probability distribution. This is required so that we can classify the input sample.

5.6.2 Backpropagation

Backpropagation updates the weights of a network layer to improve the classification task (Rumelhart et al., 1986; LeCun et al., 2015). It does so based on the weights of the outputs of the previous layer. Specifically, it uses the gradient of the activation function $v_l'(\mathbf{a}_l)$ for each layer l in order to update the weights. While forward propagation calculates the output in a forward moving manner, the weights are updated backwards via backpropagation (Mehlig, 2019).

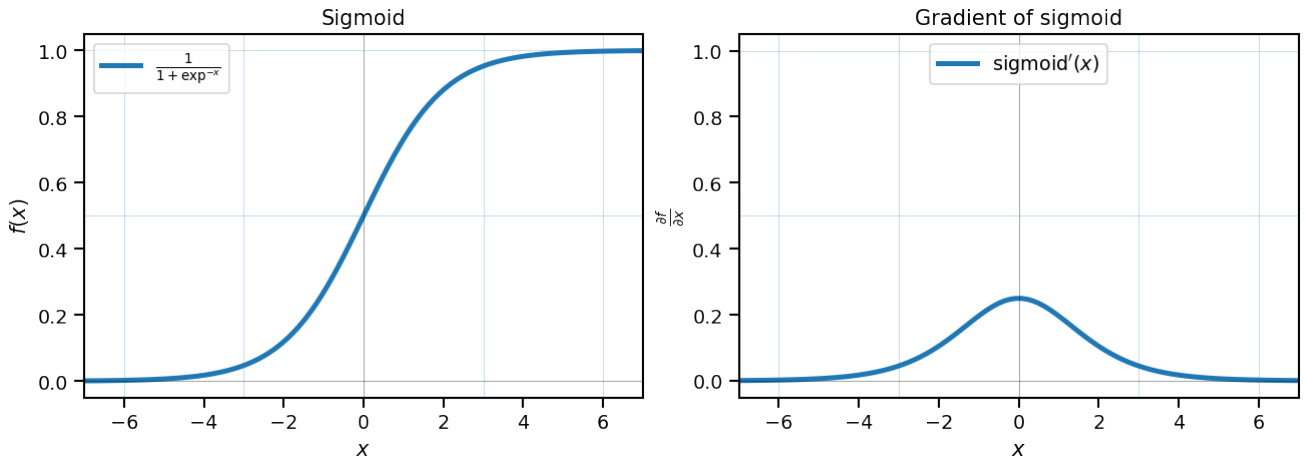
Backpropagation is a numerical efficient method (Goodfellow et al., 2016). It stores the values of a gradient that is used more than once. The gradients we want to retrieve are the partial derivatives of the cost function J w.r.t. the parameters of the CNN, which are the weights $\{\mathbf{W}_l\}_l$ and the bias terms $\{\mathbf{b}_l\}_l$ for each layer l. These gradients are then needed in the optimization algorithm explained in the next section in order to update the parameters of the CNN. The full implementation is shown in appendix C.4.

5.6.3 Activation functions of hidden layers

We give some examples of activation functions that are used in the hidden layers. They are different to the activation function of the final layer as in that case the output needs to be a probability so that we can get a predicted label to a given input (see Section 5.6.4).

Activation functions are important in order for the network to be able to learn non-linear functions (Goodfellow et al., 2016). What each of these activation functions have in common, is that they act as some kind of threshold. Therefore, below a certain value the output of activation functions is zero. One might think that a simple step function would be perfect for this.

However, as we have seen above, backpropagation uses the gradient of the activation function $v_l'(\mathbf{a}_l)$ for each layer l in order to update the weights. This means that above a certain threshold, we want the activation function to have a non-zero gradient. Otherwise, the weights are not updated and no learning occurs. This is the so-called *vanishing gradient problem*, where the majority of gradients are zero or close to zero such that the weights are not updated and hence no learning occurs. Coming back to the step function, it is firstly not differentiable at zero. Secondly, it has a zero gradient everywhere else. Hence, the idea of the step function as an activation function is discarded.



(a) Sigmoid activation function.

(b) Derivative of the sigmoid activation function.

Figure 5.13: Sigmoid/Logistic activation function and its derivative.

5.6.3.1 Sigmoid (Logistic)

The sigmoid or logistic function is defined as

$$\mathcal{S}(\mathbf{a}_l) = v_l(\mathbf{a}_l) = \frac{1}{1 + \exp(-\mathbf{a}_l)} \tag{5.25}$$

The sigmoid function has the shape of the letter 'S' and is shown in Figure 5.13a. It is essentially a soft step function. Compared to the actual step function, the logistic function is differentiable and monotonically increasing. Very negative values will be set to 0 as required by a threshold. Its maximum value is 1. We therefore do not propagate very large weights, where one neuron drastically outweighs the others. The sigmoid function is non-negative for all possible input values. This means that the output of the sigmoid function will always be of the same sign and does not allow for negative values, which can limit the capacity of the network.

Its derivative is given by

$$\begin{aligned} \frac{\partial \mathcal{S}}{\partial \mathbf{a}_l} &= \mathcal{S}(\mathbf{a}_l) \cdot [1 - \mathcal{S}(\mathbf{a}_l)] \\ &= \frac{1}{1 + \exp(-\mathbf{a}_l)} \cdot \left[1 - \frac{1}{1 + \exp(-\mathbf{a}_l)}\right] \\ &= \frac{\exp(-\mathbf{a}_l)}{[1 + \exp(-\mathbf{a}_l)]^2} \end{aligned} \tag{5.26}$$

and is illustrated in Figure 5.13b. The fact that the derivative can be stated as the function itself is computationally very useful for calculating gradients during backpropagation (Mitchell, 1997). Here, one can see the problem with the sigmoid function. The derivative is non-zero for only a very small range. The largest values occur in the interval $[-2, 2]$, reaching to smaller values within the intervals $[-4, -2)$ and $(2, 4]$. Otherwise, the derivative of the sigmoid function is mostly zero. The largest

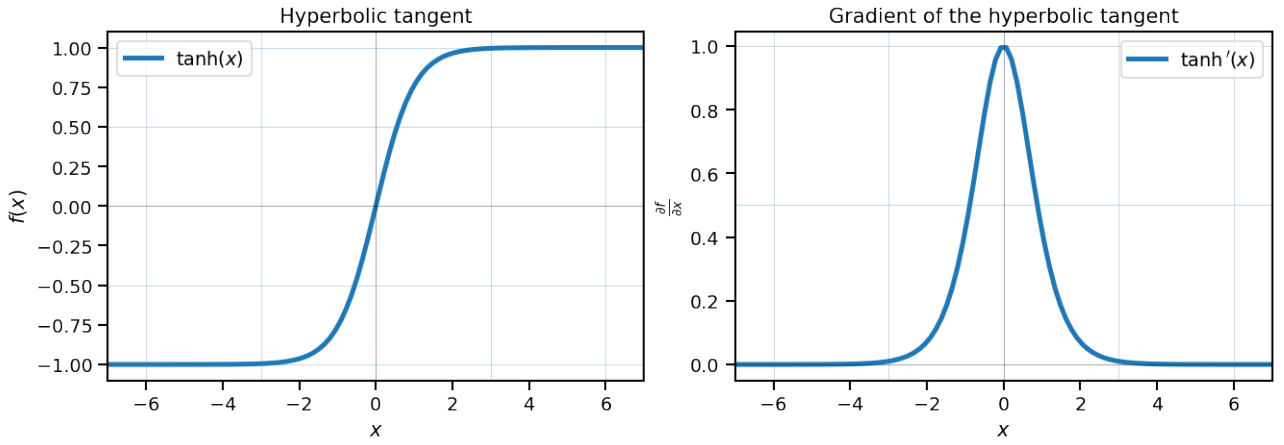
gradient value is just 0.25 at $x = 0$. As the gradient approaches zero, learning slows down drastically and stops when the derivative of the activation function is zero. This is a problem as it hinders the network from learning.

5.6.3.2 Hyperbolic tangent (Tanh)

The hyperbolic tangent is defined as the ratio between the hyperbolic sine and the hyperbolic cosine,

$$\tanh(\mathbf{a}_l) = v_l(\mathbf{a}_l) = \frac{\sinh(\mathbf{a}_l)}{\cosh(\mathbf{a}_l)} = \frac{e^{\mathbf{a}_l} - e^{-\mathbf{a}_l}}{e^{\mathbf{a}_l} + e^{-\mathbf{a}_l}}. \tag{5.27}$$

As we can see in Figure 5.14a, the hyperbolic tangent is also 'S'-shaped. However, its range is between -1 and 1, compared to the range between 0 and 1 for the sigmoid function. This allows for both negative and positive outputs. But this also means that it then does not act as a threshold function. The advantage of using tanh is that zero will be mapped to zero, which is not the case for



(a) Tanh activation function.

(b) Derivative of the tanh activation function.

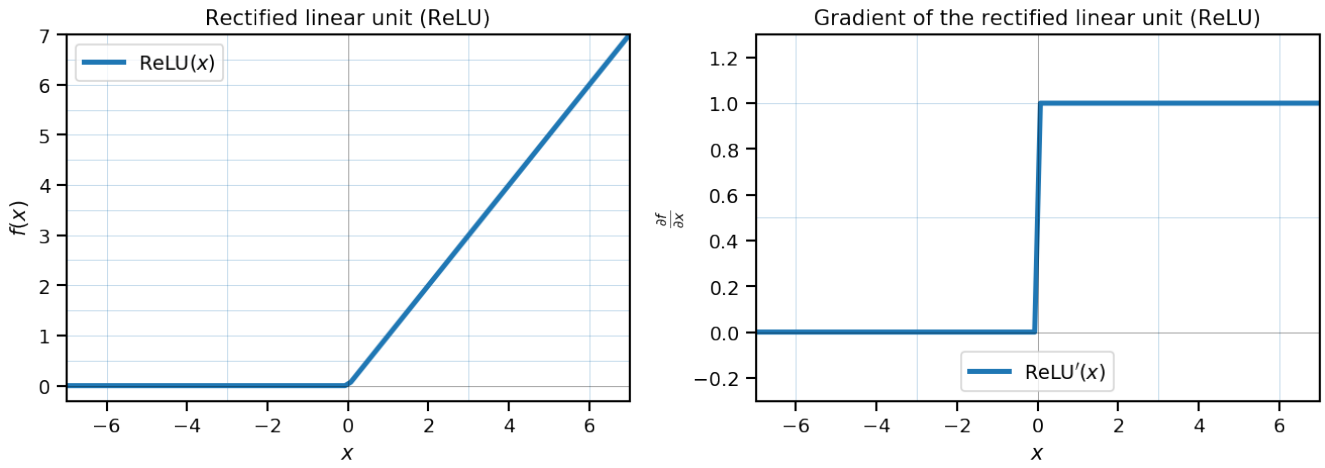
Figure 5.14: Hyperbolic tangent (tanh) activation function and its derivative.

the sigmoid function.

The derivative of the hyperbolic tangent is

$$\frac{\partial \tanh}{\partial \mathbf{a}_l} = 1 - \tanh^2(\mathbf{a}_l) = \operatorname{sech}^2(\mathbf{a}_l), \tag{5.28}$$

which is plotted in Figure 5.14b. Here, the gradient goes up to a value of 1 compared to maximum gradient of 0.25 for the sigmoid function. This allows for much faster learning. However, the x-range that gives nonzero gradient values is similar to the sigmoid function.



(a) ReLU activation function.

(b) Derivative of the ReLU activation function.

Figure 5.15: Rectified Linear Unit (ReLU) activation function and its derivative.

5.6.3.3 Rectified Linear Unit (ReLU)

The rectified linear unit (ReLU) activation function is shown in Figure 5.15a. It is defined as

$$\text{ReLU}\left(\mathbf{a}_l\right) = v_l(\mathbf{a}_l) = \begin{cases} 0 & \text{for } \mathbf{a}_l < 0 \\ \mathbf{a}_l & \text{for } \mathbf{a}_l \geq 0 \end{cases} \tag{5.29}$$

It is zero for $\mathbf{a}_l < 0$ and linear above that. This acts like a threshold that only fires neurons that are above zero. In comparison, the sigmoid function is small but non-zero for many negative x-values and hence one still has to take that neuron into account in the computation. This is not the case for ReLU which therefore makes it more computationally efficient (Krizhevsky et al., 2017). Though, it also means that any negative value immediately is set to zero, which removes the possibility to update the weights in that range. Another disadvantage is that for positive values the weights can become very large, but this effect can be overcome or lessened by regularisation techniques as explained in Section C.2.

The derivative of ReLU is given by

$$\frac{\partial \text{ReLU}}{\partial \mathbf{a}_l} = \begin{cases} 0 & \text{for } \mathbf{a}_l < 0 \\ 1 & \text{for } \mathbf{a}_l \geq 0 \end{cases} \tag{5.30}$$

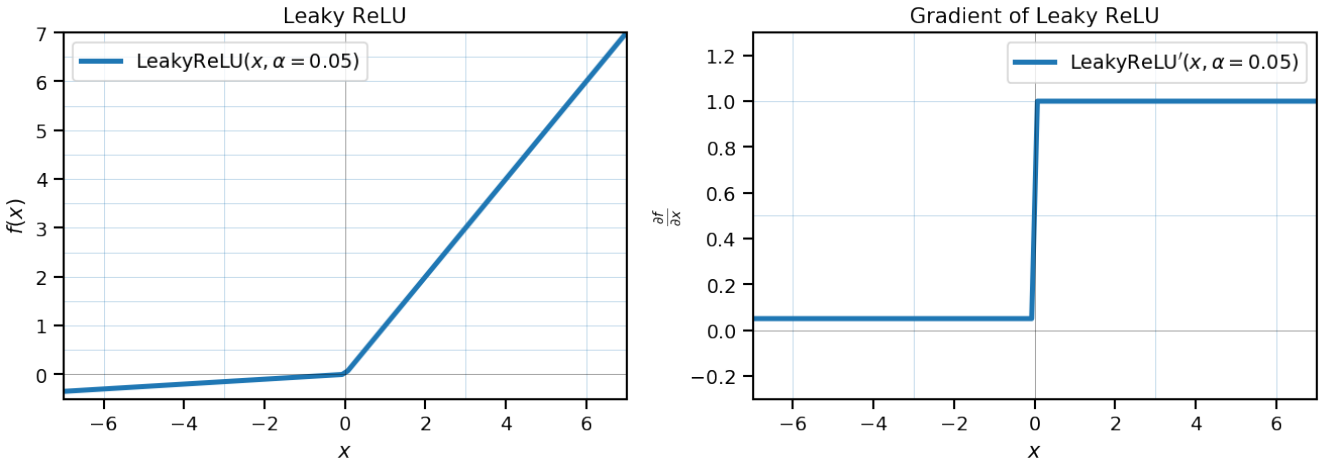
and can be seen in Figure 5.15b. Here, the gradient is zero for $\mathbf{a}_l < 0$ and is 1 otherwise. This means that for zero and positive values, the gradient is constant. It therefore does not 'prefer' specific values over another in that range, and learning is constant for zero and positive weights.

5.6.3.4 Leaky ReLU

The solution to zero gradients is Leaky ReLU. Instead of setting everything to zero for $x < 0$, one multiplies negative values by a small constant. This is shown in Figure 5.16a. Leaky ReLU is defined as

$$\text{LeakyReLU}(\mathbf{a}_l) = v_l(\mathbf{a}_l) = \begin{cases} \alpha \cdot \mathbf{a}_l & \text{for } x < 0 \\ \mathbf{a}_l & \text{for } x \geq 0 \end{cases}, \tag{5.31}$$

where $\alpha \in \mathbb{R}$ a small positive number. Typically, $\alpha = 0.01$ (Clevert et al., 2015). We do not want to set α to a large value since this function would then firstly not act as a threshold anymore, and secondly would give large weights to negative numbers. The more negative the number, the larger the output would be. Setting α to a very small value, however, allows the network to update weights for negative inputs while simultaneously acting like a soft threshold function.



(a) Leaky ReLU activation function.

(b) Derivative of the Leaky ReLU activation function.

Figure 5.16: Leaky ReLU activation function and its derivative.

The gradient is plotted in Figure 5.16b and is obtained to be

$$\frac{\partial \text{LeakyReLU}}{\partial \mathbf{a}_l} = v_l(\mathbf{a}_l) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases} \tag{5.32}$$

Similarly to ReLU, the gradient for zero and positive numbers is constant at 1. The difference lies in the gradient for negative numbers, which is non-zero in the case of LeakyReLU. Due to the non-zero gradient for negative numbers, we can update the weights of the network.

5.6.4 Output layer

In classification problems, one needs to use a probability function as the activation function in the final layer. This gives us a probability that the input sample belongs to a certain class. In neural

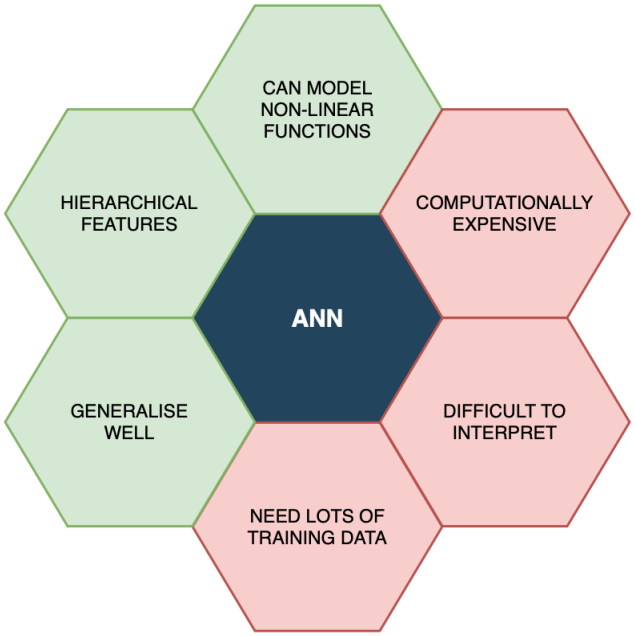


Figure 5.17: Advantages (green) and disadvantages (red) of artificial neural networks.

networks, there are as many output neurons as there are classes N_c. Therefore, each neuron i in the final layer represents the probability that the input belongs to class i. So in a binary classification problem the output layer consists of two neurons.

5.6.4.1 Softmax

The probability $P(y = i|x)$ that the input $\mathbf{h}_0$ belongs to class i, given that the output of the final layer is $\mathbf{a}_D \in \mathbb{R}^{N_c}$, is obtained as

$$P(y = i|\mathbf{a}_D) = \mathbf{h}_{D,i} = \frac{\exp(-\mathbf{a}_{D,i})}{\sum_{j=0}^{N_c-1} \exp\left(-\mathbf{a}_{D,j}\right)} \tag{5.33}$$

This is called the *softmax* function. For a two-class problem $N_c = 2$ this gives

$$P(y = i|\mathbf{a}_D) = \mathbf{h}_{D,i} = \frac{\exp(-\mathbf{a}_{D,i})}{\exp(-\mathbf{a}_{D,0}) + \exp(-\mathbf{a}_{D,1})} \tag{5.34}$$

5.6.5 Advantages and disadvantages of artificial neural networks

Advantages and disadvantes of artificial neural networks are illustrated in Figure 5.17. One of the advantages of ANNs is that they can learn non-linear functions. This is different to decision trees or random forests, which can only approximate non-linear functions. Further, the features of ANNs are built in a hierarchy, which means that the simplest features are in the first layers, and the more complex features are found in deeper layers. Depending on how complex the classification task is,

one can build shallow or deep networks. Another advantage is that ANNs with large S/N of parameter gradients during training generalise well, where the S/N measures how similar the gradients of the parameters are for different input samples (Liu et al., 2020). This is especially the case in networks with many layers.

One of the disadvantages of ANNs is that in a network of many neurons, the set of parameters is big. As each neuron in one layer is connected with each neuron in the previous layer, this quickly adds up to a significant amount of parameters and hence becomes computationally expensive. Further, apart from the first layers ANNs are difficult to interpret. This is because in deeper layers the features become so abstract that it is difficult to understand what they represent. Finally, ANNs need far more training data compared to e.g. random forests to obtain similar results. The deeper the network, the more training data is required. However, while most other machine learning classifiers stagnate after a certain amount of training data has been observed, ANNs continue to improve.

5.7 Convolutional neural network

Neural networks act on one-dimensional input data such as a list of numerical features. However, sometimes we would like to have higher dimensional input data. This is were convolutional neural networks (CNN) come into play. Originally proposed as 'neocognitron' as an unsupervised learning technique by Fukushima (1980); Fukushima & Miyake (1982), it today is mostly used for supervised tasks. The 'neocognitron' was updated and improved by LeCun et al. (1989) and Lecun et al. (1998) as the convolutional neural network we know today, and was given best-practices and guidelines by Simard et al. (2003). Typically, they are used when the input consists of image data, or image data in different colour channels (e.g. RGB), to classify the images (Cireşan et al., 2012; Russakovsky et al., 2014; Szegedy et al., 2014; LeCun et al., 2015; Krizhevsky et al., 2017). For example, the recognition of hand-written digits (LeCun et al., 1989) or words (Lang et al., 1990).

Similarly to neural networks, CNNs consist of several layers. In contrast to neural networks though, one can have a multitude of different kinds of layers. A fully connected layer represents a typical ANN layer and is just one of the possibilities. The two other main layers of CNNs are convolutional layers in combination with pooling layers which are explained below. The convolutional and pooling layers extract features with increasing complexity the deeper one goes. The fully connected layers then combine the different features and output class probabilities. The same rules apply about activation and output functions as they do for neural networks, described in Section 5.6.

An example of a CNN can be seen in Figure 5.18. Here, the task is to classify a digit between 0 to 9 based on an input image of a hand-written digit in the same range. As we can see, the first layers consist of a combination between convolution and pooling which down-scale the image size. One can have several different kernels at each layer in order to find different kinds of features at each level

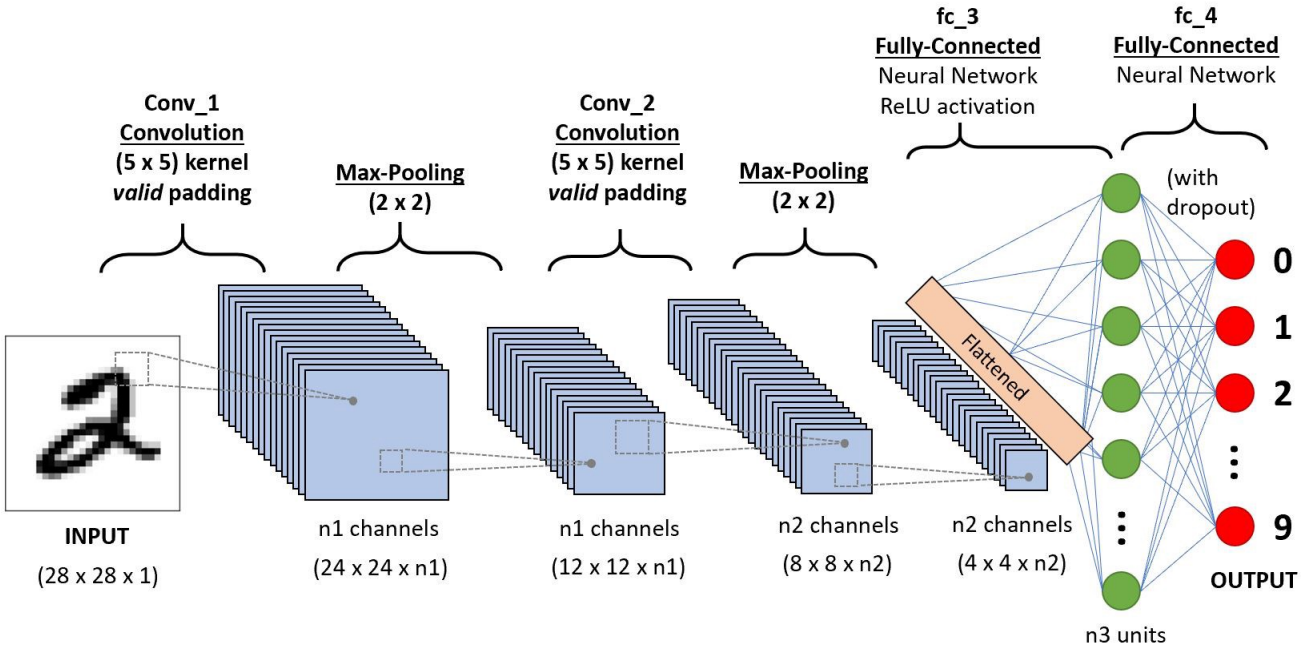


Figure 5.18: An example architecture of a convolutional neural network. Figure taken from Saha (2018). The first layers consist of convolutional layers in combination with pooling layers. The last layers are fully connected such that one can make a prediction about the class of the input image. Valid padding stands for zero padding.

of granularity. The last layers consist of fully-connected layers that lead to the final predicted class probabilities.

5.7.1 Convolutional layer

The most important layer for CNNs is the convolutional layer. A convolution is an operation between an image and a kernel, where the kernel typically is of smaller size than the input image. This has the effect of creating another set of images that are smaller than the input images.

One can think of a kernel as a search method to find a specific feature in an image. After the kernel is convolved with a subset of the image, one can infer whether the feature is present in that region or not. These kernels are convolved at multiple regions within an image, and hence it does not matter where the feature is located in the image, as the convolved kernel will detect it.

Let us define the kernel by K and our image by I. Let us further assume a $2D$ kernel of size $(d_f \times d_f)$. The convolution result at output pixel (i, j) is then defined as (Goodfellow et al., 2016)

$$(K * I)(i, j) = \sum_{m=0}^{d_f-1} \sum_{n=0}^{d_f-1} I(i+m, j+n)\, K(m, n) \tag{5.35}$$

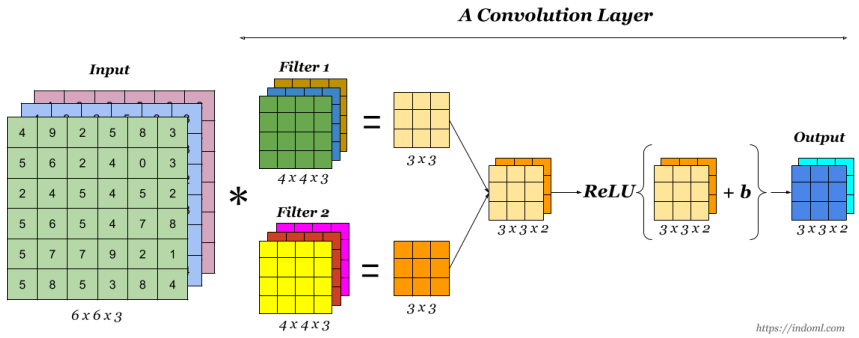


Figure 5.19: CNN: Convolution, bias and activation function. Two kernel filters are applied to a $3D$ image. A bias is added before running the convolved images through the ReLU activation function. Figure taken from Prijono (2018)

Each pixel in the convolved output $(K * I)$ represents local regions in the input image. The larger d_f, the larger those regions. If we had a 5×5 filter kernel, then each pixel of the convolved output consisted of a 5 px $\cdot$ 5 px = 25 px^2 area of the input image. The convolution essentially is a dot product between the kernel and a subpart of the input image of the same size as the kernel. The kernel is then moved horizontally and vertically to cover the whole image.

How many pixels the kernel filter is moved after each calculation is defined by the *stride* (Goodfellow et al., 2016). If one would move one pixel with each step then that would represent a stride of one. Sometimes one would like to have a certain output dimension but one cannot achieve that with the given input dimension. In this case on can *pad* the input matrix with a specified amount of pixels around it. There are different options for what values to use for these pixels. For example, one could pad it with zero or use the same pixel value as the one next to it. This can also be useful when we want to go over the regions in the corners more than once. Otherwise a kernel visits the pixels in the edges only once, whereas the pixels in the middle are visited several times.

Convolution is a linear operation and we therefore also apply activation functions in order to learn non-linear functions. We can also add a constant bias before running the output through the activation function. This is illustrated in Figure 5.19, where we have a $6 \times 6 \times 3$ input image and two separate kernels of size $4 \times 4 \times 3$. After the convolution between input image and filter we get two 3×3 images. Before applying the ReLU activation function on these, one adds a constant numerical bias.

5.7.1.1 Convolution as a down-sampling operation

As explained earlier, convolution is a linear down-sampling operation. The width w of layer $(l + 1)$ after the convolution is applied is given as

$$w^{(l+1)} = \frac{w^{(l)} - d_f^{(l)} + 2\,p^{(l)}}{s^{(l)}} + 1 \tag{5.36}$$

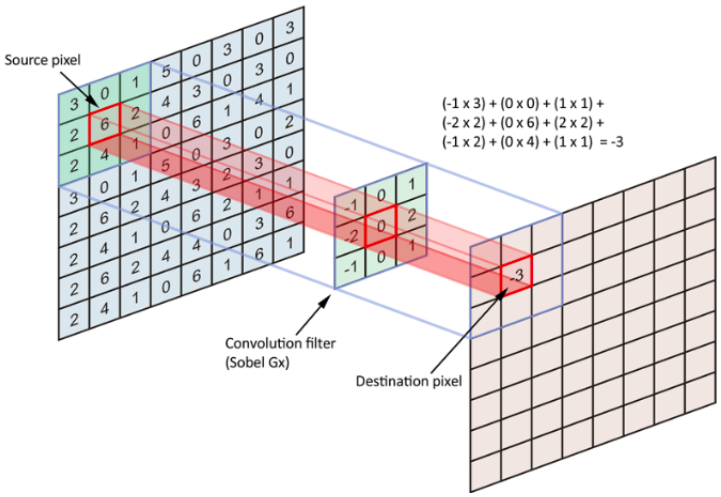


Figure 5.20: CNN: Convolution. Figure taken from Stewart (2019). The background shows the original pixel map, in the middle we can see the kernel and in the foreground one can see the convolution result of the top left part of the image convolved with the kernel Sobel filter.

where $p \in \mathbb{N}_0$ is the padding size and $s \in \mathbb{N}^+$ equivalent to the stride. Note that we can disregard the number of channels, as one sums over them during the convolution. The number of channels in the following layer are determined by the number of feature kernels.

Looking at the example in Figure 5.19, we have $w^{(1)} = 6$ and two kernels of size $d_f = 4$. We have a stride of $s = 1$ and zero padding. Without looking at the formula above, if you imagine hovering the kernel over the image and move one pixel at each step, one can apply the kernel exactly 3 times in each direction before hitting the end of the image. Therefore, the convolved image has a size of 3×3. We can confirm this with equation (5.36), which correctly gives $w^{(2)} = \frac{6-4+0}{1} + 1 = 3$.

If instead we had an input image of size 31×31, a 5×5 filter kernel, a stride of $s = 2$ and padding $p = 1$, then the output after a single convolution with this filter would be $w^{(2)} = \frac{31-5+2\cdot 1}{2} + 1 = 15$. This gives a 15×15 image, which is essentially half the size of the original image. Note that the biggest down-sampling effect is due to the stride. If we had taken a stride of $s = 1$ instead, then the output would have been 29×29, which is close to the original size.

What if we want to use a stride of 1 and no padding, but still down-sample the input image? This is where the pooling layer comes in, which is described next.

5.7.1.2 Examples of kernels

An example of a convolutional layer can be seen in Figure 5.20. Here, the kernel filter is defined as the horizontal Sobel filter that finds edges along the horizontal direction, which is the same as finding

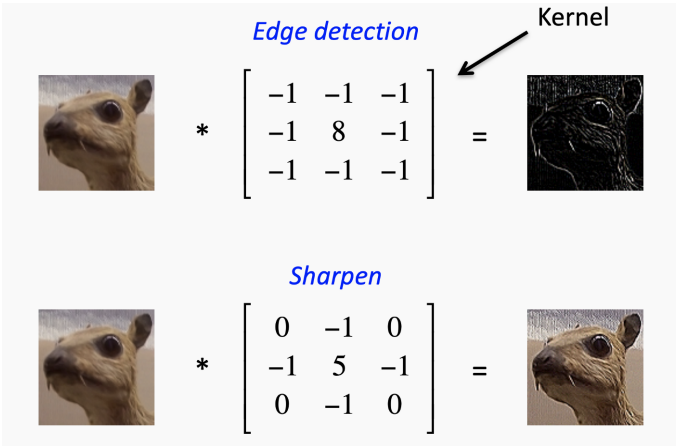


Figure 5.21: CNN: Kernel examples. Figure taken from Stewart (2019). The first column shows the original image, the second column shows the kernel filters and the third column shows the convolution of the image with the kernel. The first kernel detects edges in every direction. The second filter sharpens the image.

vertical edges. It is defined as (Sobel, 2014)

$$\text{Sobel } G_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \tag{5.37}$$

In this case this gives $-1 \cdot 3 + 0 + 1 \cdot 1 - 2 \cdot 2 + 0 + 2 \cdot 2 - 1 \cdot 2 + 0 + 1 \cdot 1 = -3$. Both a very negative and very positive value are indicators of detected edges. In this case this means that a vertical edge or change in horizontal direction has been detected.

Intuitively the Sobel filter makes sense as well. Imagine we had a 3×3 image I where the first two columns consisted of ones and the last one of zeros. Applying the Sobel G_x filter would give us $G_x * I = -1 \cdot 1 - 2 \cdot 1 - 1 \cdot 1 + 0 = -4$, and hence an edge has been detected. For comparison, if our image only consisted of ones then the convolved filter would result in 0 which is equivalent to no detected vertical edges which is correct in this case. Analogously to the horizontal filter, the vertical Sobel filter detects edges along the vertical direction, and can be interpreted as the transpose of the horizontal Sobel filter (Gonzalez, 2018).

Another set of examples of kernel filters is given in Figure 5.21. Here, we have another edge detection filter. In this case this filter tries to find both horizontal and vertical edges/ changes is brightness at the same time. The second filter acts to sharpen the image. In a CNN, we do not explicitly state the filters we want to use, but rather make the CNN learn valuable kernels itself. We do this with backpropagation which essentially works the same way as in fully connected layers.

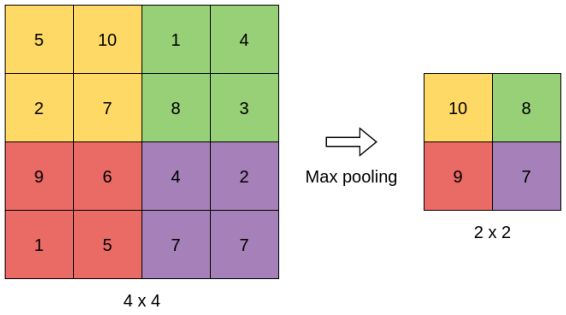


Figure 5.22: Max-pooling applied on a 4×4 array with a stride of 2 pixels and zero padding.

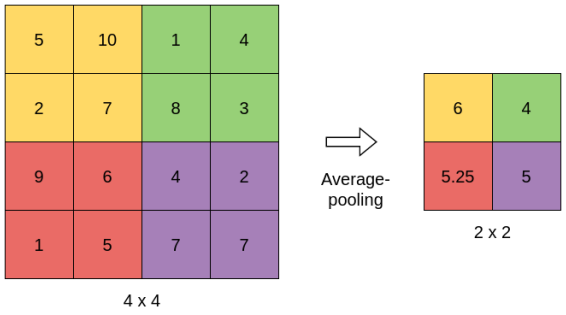


Figure 5.23: Average-pooling applied on a 4×4 array with a stride of 2 pixels and zero padding.

5.7.1.3 Feature complexity

Deeper convolutional layers measure a different level of granularity compared to the first layer (LeCun et al., 2015). The deeper one goes within a CNN, the more complex features one detects. The first layer usually measures the presence of edges and changes in brightness at various orientations. The next layers are able to look at more complex features that can combine and make sense of the edges that are detected and other edges that are not detected in the previous layer, from which one can discern parts of objects. Finally, the last layers can then combine the parts or patterns detected in the previous layers to form objects that represent the task at hand.

For example, if one wanted to classify dog pictures, then again the first layer(s) might find edges in various orientations, the next layer(s) can combine previous findings and might find features such as paws, legs, eyes and ears of a dog. The final layer(s) might find complete dogs with all the features needed that make up a dog.

5.7.2 Pooling layer

Pooling was brought to attention by Scherer et al. (2010) and Cireşan et al. (2011). It is a down-sampling operation that allows for positional variance over larger regions. Typically, a stride of more than 1 is used to effectively reduce the size of a feature image (LeCun et al., 2015). Reducing the complexity of the images helps to reduce computational cost and prevents overfitting. It further aids

in detecting important features. There are two different kinds of pooling: max-pooling and average-pooling (Goodfellow et al., 2016).

5.7.2.1 Max-pooling

As the name suggests, max-pooling takes the maximum pixel value within a location of an image of a given kernel size. An example of max-pooling is illustrated in Figure 5.22. Here, the kernel is of size 2×2 and is applied on an image of size 4×4 with stride $s = 2$ and zero padding. The maximum of the first 2×2 pixels (in yellow) is $\max\{5, 10, 2, 7\} = 10$. Then we walk two pixels to the right as per stride. The maximum in this 2×2 area (in green) is $\max\{1, 4, 8, 3\} = 8$, and so on.

Max-pooling aids in extracting the most important features. This is because significant features have large absolute values. We have seen this in the example of edge detection in Section 5.7.1.2. However, max-pooling takes the largest positive values and does not take into account the largest negative values. Therefore, depending on the sign in the feature kernels, max-pooling either misses out on bright features on top of a dark background or the other way around. However, this can be circumvented by applying the activation function before the max-pooling layer. Although ReLU and max-pooling commute, $MaxPool(ReLU(x)) = ReLU(MaxPool(x))$, and hence the problem persists (Goodfellow et al., 2016). This issue does not occur with average-pooling.

5.7.2.2 Average-pooling

In average-pooling, one calculates the average pixel value within a location of an image of a given kernel size. Average pooling is illustrated in Figure 5.23. Here, the first 2×2 area in yellow on the top left has an average of $(5 + 10 + 2 + 7)/4 = 6$ Walking two pixels to the right to the green area, we obtain an average of $(1 + 4 + 8 + 3)/4 = 4$. The red patch has an average of 5.25, and the purple on on the bottom right has an average of 5.

Average-pooling aids in extracting the average presence of features, and therefore gives more smooth results compared to max-pooling. Due to the average, it takes into account all features that are present in a given patch. However, it might miss or reduce the importance of the most significant features. On the other hand, max-pooling only concentrates on the most important features but might miss out on some more subtle features that might be important as well.

5.7.3 Fully connected layer

A fully connected layer works in the same way as a neural network, which is described in detail in Section 5.6. If the image output in the previous layer is of dimension $N \times N$, then flattening results

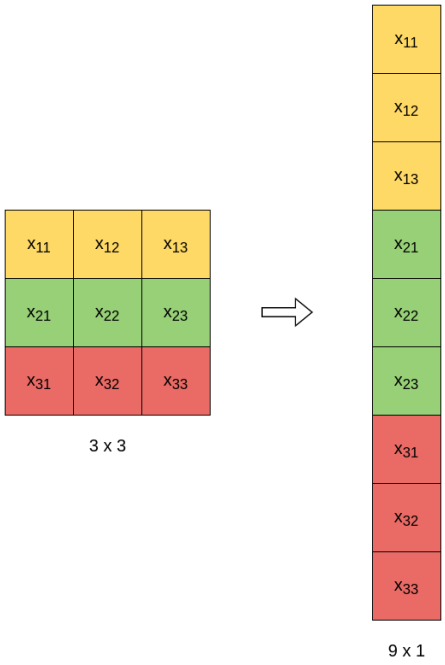


Figure 5.24: CNN: Flattening of a $2D$ output of a convolutional layer in order to use it as the $1D$ input to a fully connected layer.

in a $(N \cdot N) \times 1$ column array, as is illustrated in Figure 5.24. If N is large, then one has to deal with a significant amount of parameters in the fully connected layer. It is therefore useful to down-sample the images before feeding them to any fully connected layer.

5.7.4 Prevention of overfitting

In CNNs, one uses two main approaches to reduce overfitting,

- Data augmentation of the training set
- Dropout in fully connected layers

5.7.4.1 Data augmentation

Firstly, one augments the image data to increase the training set. This is to account for different orientations and locations an object might have within an image (Cireşan et al., 2012; Krizhevsky et al., 2017). For example, let us say we had face-on pictures of dogs and cats. The network then will have issues if the face of the animal is tilted or the gaze is side-ways. It is therefore crucial to have a data set that represents all the different poses and orientations the target object could have. Of course, we cannot easily artificially change the pose of an object, but changing an orientation is easy to do. For example, one can mirror the data, rotate the image, scale or translate the object and/or shear the

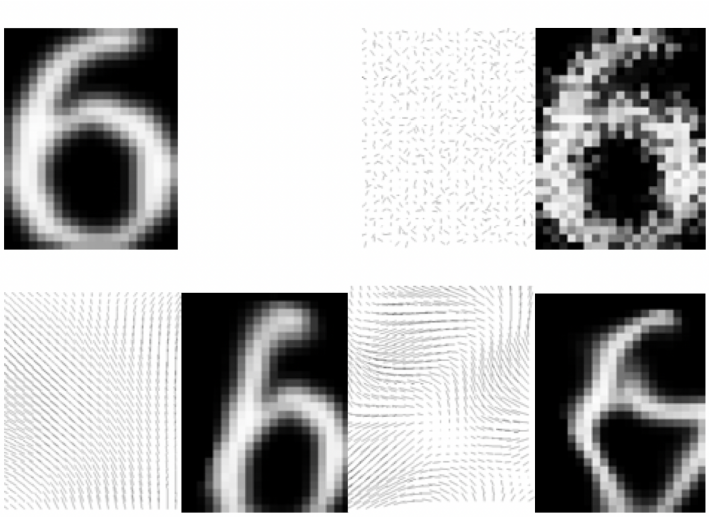


Figure 5.25: Data augmentation using elastic transformations. Figure taken from Simard et al. (2003). The top left shows the original image of the number six, while the others show distorted versions of the original using the displacement fields shown to their left.

image to show the object at different orientations and scales (Cireşan et al., 2011). These are called affine transformations.

Another set of transformations are elastic transformations based on displacement fields (Simard et al., 2003), which can be seen in Figure 5.25. The image in the top left is the original image. The other images show the distorted image after the displacement fields shown to their respective left have been applied to the original image. These have been created by a random displacement field of random displacements. These displacements were then convolved with a Gaussian of different standard deviation σ. If σ is close to zero pixels, then the displacements are essentially random, which can be seen in the top right. Generally, adding random noise to the data can make the classifier more robust to that. For larger values of σ, one arrives at smooth elastic deformations, which can be scaled by another parameter.

5.7.4.2 Dropout

As a second technique to prevent overfitting in CNNs, one uses dropout in the fully connected layers of the network as described in Section C.2.3.1 (Hinton et al., 2012). Typically, one uses a probability of 0.5 for each neuron that its weight is temporarily set to zero to enhance feature diversity and robustness. The neurons that are set to zero in that round then do not contribute to the classification. This essentially represents different architectures of networks at every step, which reduces overfitting (Goodfellow et al., 2013).



Figure 5.26: Advantages (green) and disadvantages (red) of convolutional neural networks.

5.7.5 Advantages and disadvantages of CNNs

Advantages and disadvantages of CNNs are similar to that of ANNs and are illustrated in Figure 5.26. The main usage of CNNs is on high-dimensional data such as images (Russakovsky et al., 2014; Szegedy et al., 2014). Imagine we used a neural network instead. Firstly, one would have to flatten the data which loses any spatial information that is important in recognizing features within the image. Secondly, one would need huge amounts of parameters, as every neuron in one layer is connected with every neuron of the previous layer. This effect would be especially drastic if the input consisted of large images. For CNNs, however, we only need to store the kernels themselves, which are usually far smaller than the image. The main cost lies in the fully connected layers, which are all connected to the complete flattened data of the previous multi-dimensional features. If one has many kernels in that layer before the fully connected layer then this can be very expensive.

5.8 Difficulties of ML

Machine learning thrives on problems that might be easy to solve for a human expert in a specific domain, but are time consuming and/or difficult to write down as a set of formal rules. Typically, a well trained machine learning algorithm can solve the same task more accurately and more efficiently than the human expert. However, artificial intelligence is limited by its non-interpretability, non-reproducibility and its non-generalisability. Non-interpretability means that they often act as black boxes where we know the input and decision output, but not the ‘reasoning’ behind the decision. Non-reproducibility means that if you were to implement the same algorithm and run it through the

same training set, you would get a different result each time. Non-generalisability is the idea that an AI can be excellent at one specific task but not at any other task.

5.8.1 A few scientific discoveries

So far, machine learning mostly re-discovers things we already know, but it has not really made any scientific discoveries yet. However, it can be essential in aiding scientific discoveries (Wang et al., 2023).

One of the first successes of AI was in 1997, when the IBM Deep Blue AI defeated the then chess world champion Garry Kasparov (Wang, Fu, Du, Gao, Huang, Liu, Chandak, Liu, Van Katwyk, Deac, Anandkumar, Bergen, Gomes, Ho, Kohli, Lasenby, Leskovec, Liu, Manrai, Marks, Ramsundar, Song, Sun, Tang, Veličković, Welling, Zhang, Coley, Bengio & Zitnik, ibm). However, this was not a discovery per se, as it calculated probabilities of success of every possible move and used the one that would lead to the most likely success.

A first snippet of scientific advancement was seen in 2016, when Google DeepMind's AlphaGo defeated the 18-time world champion Lee Sedol. It discovered new moves that not even the Go masters had known beforehand (Silver et al., 2016). It was initially trained on the games of human expert players. After AlphaGo, they developed a new deep learning AI called AlphaGo Zero that was not trained on human knowledge but rather only played against itself (Silver et al., 2017). It outperformed AlphaGo within just 3 days of training and won by 100 games to 0. This exposes the limits of our human understanding, and that leaving our knowledge out of it can actually make a machine become more 'creative' so to speak.

In 2020, DeepMind's AlphaFold won the Critical Assessment of protein Structure Prediction (CASP) (Jumper et al., 2021). This is an experiment that takes place every two years to find the best algorithms that come closest to predicting the 3D protein structure from their 1D amino acid sequence. They are tested on around 100 amino acid sequences, where the shapes of the proteins have been evaluated in the lab beforehand but not made public yet to allow for a blind test. Before AlphaFold no algorithm in 50 years of the existence of CASP had come close to the levels of precision of lab analysis. This will prove useful for drug discovery for different kinds of diseases and much more.

5.8.2 Correlation vs causation

A machine learning algorithm can easily find correlations, but these are not necessarily causations.

Messerli (2012) detected a linear correlation ($r = 0.791$, $P < 0.0001$) between chocolate consumption per year per capita of different countries and the number of its Nobel laureates per 10 million

inhabitants. Not surprisingly, Switzerland made the top spot in chocolate consumption. However, it also has the most Nobel laureates per 10 million people. Messerli (2012) state the following:

" Of course, a correlation between X and Y does not prove causation but indicates that either X influences Y, Y influences X, or X and Y are influenced by a common underlying mechanism. However, since chocolate consumption has been documented to improve cognitive function, it seems most likely that in a dose-dependent way, chocolate intake provides the abundant fertile ground needed for the sprouting of Nobel laureates. "

5.8.3 The importance of the training set

Even though machine learning algorithms on average perform better than people, they are only ever as good as its input data. Choosing a broad and representative training set can become more important to the result than the architecture of the algorithm itself.

5.8.3.1 Output class probabilities

The output of classification tasks consists of probabilities that a sample belongs to a certain class. Note that the probabilities in machine learning should be interpreted with a grain of salt, as they are based on the distribution of the training set. Often this distribution is not representative of the real world but is adapted to solve a specific task. For example, if one wants to find high-redshift quasars, one will put a larger quasar-to-contaminant ratio in the training set compared to the true ratio. This is so that the classifier can find relationships between the input features and quasars, which ultimately leads to a better classification. However, what it essentially does is to skew the prior distribution of quasars. Hence, one should not take the probabilistic output of a machine learning predictor at face value.

5.8.3.2 Uncertainties in the input data

Uncertainties in the input measurements can lead to wrong classifications in the input labels. The problem with sources in astronomy is that, unless an object has been spectroscopically observed, there is always an uncertainty in its class label. Supervised learning uses 'ground truths' as its input data. But what if the labels of the input are wrong without us knowing it? In the worst case, an algorithm gets trained on wrongly labelled input data and hence will misclassify yet unknown objects. This problem gets worse the more complex the algorithm is allowed to be. Overfitted models will have an especially bad effect on wrongly classified labels, as the model then tries to fit for outliers as well. On the other hand, if the algorithm is not overfitted and the wrong labels are distributed randomly throughout the feature space, then this should not be too big of a problem, given that the number of

wrong labels is much smaller than the number of correct labels. The correct labels will outweigh the bad ones. If, however, the wrong classified labels form clusters, the bad labels will outweigh the good ones and the decision boundary will adapt to this. In order to circumvent these effects, it is important not to overfit the model.

Chapter 6

Data processing pipeline

The data processing pipeline assesses image quality and extracts two quasar detection metrics and other information for a given source from broad-band photometric survey images. It fits point sources to our candidates of interest from which their fluxes and magnitudes are obtained.It further gets rid of any images that contain any survey data processing issues. The parameters used in different stages of the pipeline are listed in Table 6.1. These are different radii from the central source within which a specific quantity is calculated.

Currently, the pipeline works fully for SDSS and UKIDSS LAS. It will be straightforward to include upcoming surveys as the code has been written to be very modular. One only has to adapt the database access and optionally the form of the point spread function (PSF) specific to each survey.

Note that the input to the pipeline is a set of source positions. These will typically be taken from the database of one of the surveys. Since SDSS and UKIDSS have been running several years apart, a local star will not have the same position in both surveys. It will have moved, and thus will have a different source location in the other survey's database. Therefore, this pipeline is also able to detect if a source has moved between two (or more) surveys.

6.1 Step-by-step pipeline sequence

The input for this pipeline is a list of (RA, dec) locations. The output is a summary file that contains extracted and processed information summaries for each source (see Table 6.2). These encompass information on the photometry (fluxes and magnitude), background, residual chi-squared values from model checking and a quasar probability from Bayesian model comparison. This summary can then be used as input features for the classification task of finding high-redshift quasars.

A flowchart of our code structure is shown in Fig. 6.1. The pipeline automatically downloads all

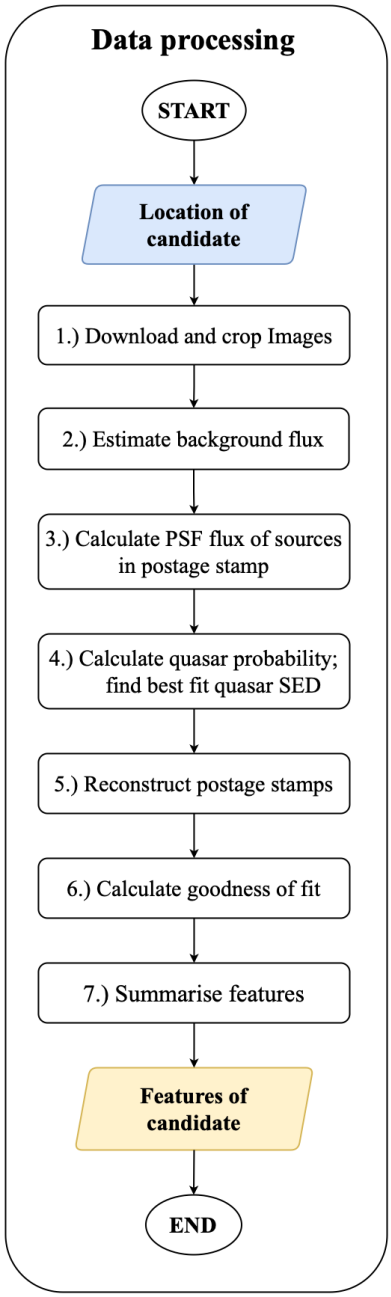


Figure 6.1: Flowchart of the data processing pipeline. For each available source image, the background pixel flux and the flux of the central source are calculated. These are used to obtain a quasar probability and a best fit quasar SED. The latter is used to reconstruct the postage stamps based on our quasar model, which are subsequently tested against the real data via chi-squared. The complete set of information of the input source can then be utilised to classify it as a quasar or contaminant.

Table 6.1: Parameter values of our data processing pipeline, where radii are given in arcseconds.

Parameter	Description	Default value
l	Size of each postage stamp ($l \times l$)	$34\farcs8$
l_{plot}	Plotted postage stamp size ($l_{\mathrm{plot}} \times l_{\mathrm{plot}}$)	$24\farcs4$
R_{clip}	Radius for sigma clipping	$8\farcs4$
α	Sigma clipping: pixel p is removed if $\lvert p\rvert > \lvert\mathrm{median} \pm \alpha\sigma\rvert$	3.0
R_{keep}	Radius within which no other source gets reconstructed	$1\farcs2$
R_{flux}	Radius for fitting the flux	$2\farcs6$
$R_{\tilde{\chi}^2}$	Radius for calculating $\tilde{\chi}^2_b$	$1\farcs2$

available images in each passband for given surveys, makes image cutouts, removes sources outside the centre and finally calculates the PSF flux of the central source together with the background pixel flux and signal-to-noise ratio. This allows us to retrieve photometry information even for candidates that are too faint to have been detected in a given survey. Thus, by applying forced photometry (i.e., measure the flux at a given location even if no source has been detected by that survey) on the raw pixel data of each survey in each band for each source, we retrieve a more accurate best-fit quasar SED template and quasar probability.

A list of locations of astronomical sources given by their right ascension α and declination δ serve as the input for the pipeline. The following subsections describe the steps that are applied to each input location to arrive at the summarized candidates. These summaries can then be used as input features for a classifier to detect if the source is a possible quasar candidate or not.

6.1.1 Downloading all available images

Given a location of right ascension and declination, the databases of a set of surveys are queried to obtain all available images that contain this location. Images are discarded if specific error flags are set that would indicate any data processing issues. These are specific to each survey. The relative directory structures for the downloaded images in SDSS and UKIDSS are shown in Table 6.3. The image identifiers in the path names are explained in Section 6.2.

Each database query takes approximately 1 second to complete. The image downloads take the longest time with approximately 5 seconds per image (compressed: $\approx$ 110 kB). We therefore do

Table 6.2: Stored information for each candidate as part of the last step in the data processing pipeline. The fluxes are stored in units of Jansky (Jy) and the magnitudes as AB magnitudes. The fluxes and magnitudes are averaged over n_r runs. The minimum and maximum chi-squared values are $\tilde{\chi}^2_{\min} = \min\{\chi^2_{b,n}\}_{b\in B, n\leq n_r}$ and $\tilde{\chi}^2_{\max} = \max\{\chi^2_{b,n}\}_{b\in B, n\leq n_r}$.

Stored	**Summary**
RA	right ascension (J2000)
dec	declination (J2000)
$\tilde{\chi}^2$	residual chi-squared (7.9)
$\{\tilde{\chi}^2_b\}_{b\in B}$	chi-squared for each passband (7.8)
$\tilde{\chi}^2_{\min}$	minimum chi-squared
$\tilde{\chi}^2_{\max}$	maximum chi-squared
$\tilde{\chi}^2_{\text{pos}}$	$\tilde{\chi}^2$ with fitted source centre
$\tilde{\chi}^2_{\text{pos,max}}$	$\tilde{\chi}^2_{\max}$ with fitted source centre
$\hat{z}$	estimated redshift
P_{q}	quasar probability (7.1)
$\{F_b\}_{b\in B}$	PSF flux
$\{\sigma_{F_b}\}_{b\in B}$	uncertainty in the PSF flux
$\{m_b\}_{b\in B}$	magnitude
$\{\sigma_{m_b}\}_{b\in B}$	uncertainty in the magnitude
$\{F^{\text{db}}_b\}_{b\in B}$	database flux
$\{\sigma_{F^{\text{db}}_b}\}_{b\in B}$	uncertainty in the database flux
$\{m^{\text{db}}_b\}_{b\in B}$	database magnitude
$\{\sigma_{m^{\text{db}}_b}\}_{b\in B}$	uncertainty in the database magnitude
$\{(\text{S/N})_b\}_{b\in B}$	signal-to-noise ratio
$\{\mu_{0,b}\}_{b\in B}$	mean background pixel flux
$\{\sigma_b\}_{b\in B}$	uncertainty in the background pixel flux

Survey	Stored image path
SDSS	Images/<rerun>/<run>/<camcol>/ frame-<passband>-<run(6)>-<camcol>-<field(4)>.fits.bz2
UKIDSS	Images/<mfid>/<extno>/<uniq>.fits.gz

Table 6.3: Stored paths of the downloaded images. The numbers in the brackets correspond to the amount of characters the field should contain, filling the unused characters up with zeros (e.g. $47 \overset{(6)}{\rightarrow}$ 000047).

not download an image if it already exists. Although if needed, we can set a flag in the pipeline that instructs the re-download of images.

6.1.2 Cropping an image to a postage stamp

We cut out the images to postage stamps of size $l \times l = 34\farcs8 \times 34\farcs8$ that are centred on the nominal source positions. The reason we use such a relatively large area is that we also want to reconstruct the flux of sources near the central source, since not all of the flux in the centre of the postage stamp necessarily comes from the central source. And we can only reconstruct a source with the PSF-fitted flux if we have the image data to fit it to.

We store both the original sized image and the smaller postage stamp. If one wants to use a larger postage stamp size then it is easy to do so without having to re-download the original image. Moreover, we do not want to load the full image into our workspace since we only need to analyse a small region around the source. If the postage stamp already exists, then it is loaded into the workspace. The paths for the postage stamp images in SDSS and UKIDSS is shown in Table 6.4.

Survey	Stored postage stamp path
SDSS	Postage_stamps/<rerun>/<run>/<camcol>/ frame-<passband>-<run(6)>-<camcol>-<field(4)><IAU>.fits.bz2
UKIDSS	Postage_stamps/<mfid>/<extno>/<uniq><IAU>.fits.gz

Table 6.4: Stored paths of the postage stamps. The numbers in the brackets correspond to the amount of characters the field should contain, filling the unused characters up with zeros (e.g. $47 \overset{(6)}{\rightarrow}$ 000047).

When the source is located near the edge of the original image, it can happen that at least one of the edges of the postage stamp lies outside the original image. In these cases, the data array is filled up with NaN values such that the overall shape of the postage stamp stays the same.

We mentioned that we do not download an image if it has some data processing issues around the source of interest. However, the survey's algorithms do not capture every processing issue all the time. We have noticed that in our dataset we have quite a few images that contain lines caused by

e.g. satellite trails, which have not been flagged up by the surveys. We therefore apply a Hough line transform (Galamhos et al., 1999) on every postage stamp in order to detect those. We use `probabilistic_hough_line`[1] from the Python library `skimage` to do that. If a line is detected, we discard that postage stamp.

6.1.3 Plotting postage stamps

We plot the postage stamps at a size of $l_{\mathrm{plot}} \times l_{\mathrm{plot}} = 24\farcs4 \times 24\farcs4$ each. This is useful for analysing re-occurring false positives / false negatives, either to update their classification label or to potentially see what types of source images a classifier might get wrong.

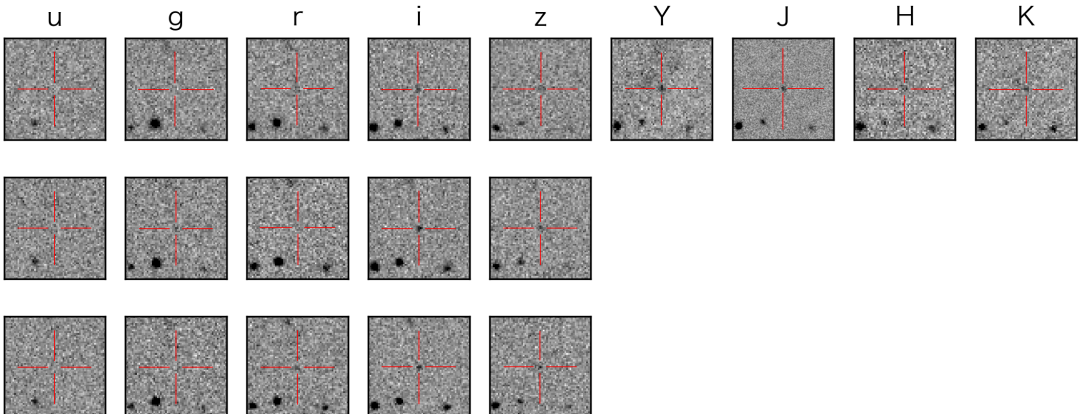
6.1.3.1 The orientations of postage stamps

The orientations of survey images can vary, even within the same survey. When plotting postage stamps, we want to compare the data between different passbands and surveys and therefore want them to be oriented in the same way. We rotate each postage stamp by an angle ω such that North points upwards and East towards the left. If required, we also mirror the postage stamp after applying the rotation. This is required for SDSS, where North points upwards and East towards the right, and so the image has to be mirrored horizontally. Examples of plotted postage stamps that have been aligned are depicted in Figure 6.2. This shows a point source (6.2a) and a quasar (6.2b) at redshift $z = 5.82$ (Fan et al., 2001) in the passbands of SDSS and UKIDSS.

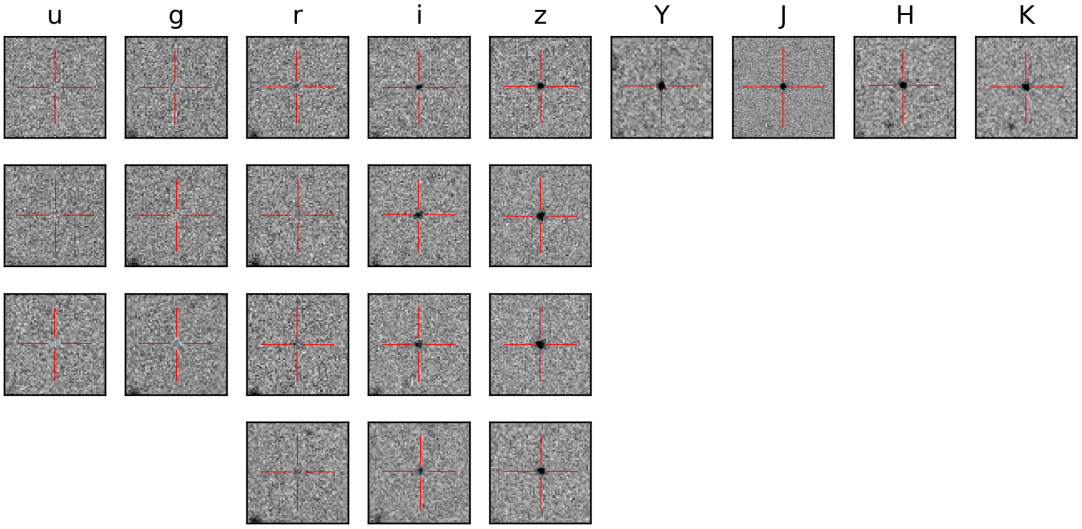
If we rotate the postage stamp, it gets interpolated which leads to a loss of information (if ω is not a multiple of 90 deg). The rotated image therefore is only used when we want to plot a postage stamp. All of our calculations are done on the original pixel data. To make our calculations independent of orientation, we apply a circular mask on the central source of our postage stamp. Only pixels that lie within that mask are taken into account for the calculations, whereas the remaining pixels are ignored. This can be seen in Figure 6.3.

If we rotate an image by an angle that is not a multiple of 90 deg, the resulting square image becomes larger. This is illustrated in Figure 6.4. Therefore, the stored image has to be larger than the part of the image that is plotted. If we want to use a square of side D, then the stored image has to be at least of size $\sqrt{D^2 + D^2} = \sqrt{2}\,D$ to allow for rotations at any angle (i.e. for 45 degree rotations). For our plotting size of $l_{\mathrm{plot}} = 24\farcs4$, the required postage stamp size is $34\farcs51$. We select a slightly larger postage stamp size of $34\farcs8$.

[1] `https://scikit-image.org/docs/dev/api/skimage.transform.html`



(a) Point source J092036.65+093407.9



(b) Quasar J083643.86+005453.2 at $z = 5.82$ (Fan et al., 2001)

Figure 6.2: Postage stamps for two sources, plotted in the passbands of SDSS and UKIDSS. The rows correspond to different runs/epochs.

6.1.3.2 Obtaining the rotation angle

In order to calculate the rotation angle ω , two locations p_1, p_2 are selected within the postage stamp,

$$p_1 = (\alpha, \delta) \tag{6.1}$$

$$p_{2,\mathrm{SDSS}} = (\alpha + \epsilon, \delta + \epsilon) \tag{6.2}$$

$$p_{2,\mathrm{UKIDSS}} = (\alpha - \epsilon, \delta + \epsilon), \tag{6.3}$$

where $\epsilon > 0$, α the right ascension and δ the declination. p_2 differs between SDSS and UKIDSS as the images are mirrored to each other.

Next, the pixel locations that correspond to the angular location are obtained via the Python library

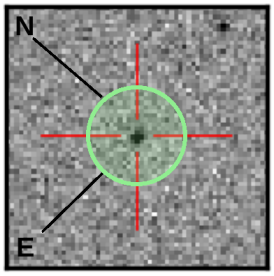


Figure 6.3: Applying a circular mask (green) on a postage stamp. Only the pixels within the mask are used for calculations. The circular mask is independent of the orientation of the postage stamp, and hence it does not matter how North(N) and East(E) are aligned.

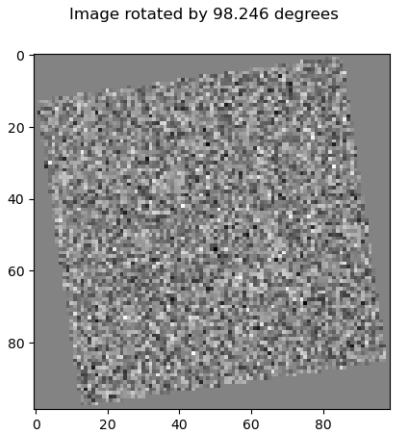


Figure 6.4: An image A is rotated by 98.25 degrees to arrive at image B. In order to capture the original image A in its rotated form, the resulting image B has to be of larger dimension compared to image A.

`WCS` [2]. Using the pixel locations, the angle between the two points is calculated,

$$\omega = \tan^{-1}\left(\frac{p_{1,x} - p_{2,x}}{p_{2,y} - p_{1,y}}\right). \tag{6.4}$$

This is illustrated in Figure 6.5. Using inverse tangent instead of e.g. inverse sine gives a slightly more

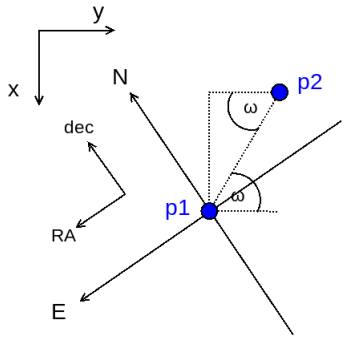


Figure 6.5: Illustration of how to calculate rotation angle ω for a CCD setup where, if North points upwards, East points towards the left.

accurate result as we do not need to calculate the hypotenuse $\overline{p_2 p_1}$. Depending on which quadrants p_1 and p_2 are located in, the image is rotated accordingly.

6.1.4 Estimate the background flux

We apply sigma clipping (Ratnatunga & Newell, 1984; Akhlaghi & Ichikawa, 2015) in order to get an estimate of the background pixel flux. The algorithm applies these two steps until convergence, where we use $a = 3$:

1. Calculate the median and the standard deviation σ of the underlying data subset
2. Remove all pixels with absolute analog-to-digital (ADU) values |ADU| > median $\pm\, a\,\sigma$
   The remaining pixel values form the new subset for the analysis.

After the algorithm has finished, one obtains the mean μ and standard deviation σ from the remaining pixel data. This is illustrated in Figure 6.6, which shows the histogram of the ADU values of the remaining pixels of a source after sigma clipping has been applied. This distribution looks Gaussian as expected, and hence the source pixels have successfully been removed and the remaining pixels represent the background. The algorithm is applied on a circular region of radius $R_{\text{clip}} = 8\farcs4$ around the centre of the postage stamp. We use this radius as it gave the best classification results. We need a big enough radius such that the source can be discerned from the background, and a small enough radius such that we receive local background pixel flux values. This works well except for when there

[2] `https://docs.astropy.org/en/stable/wcs/`

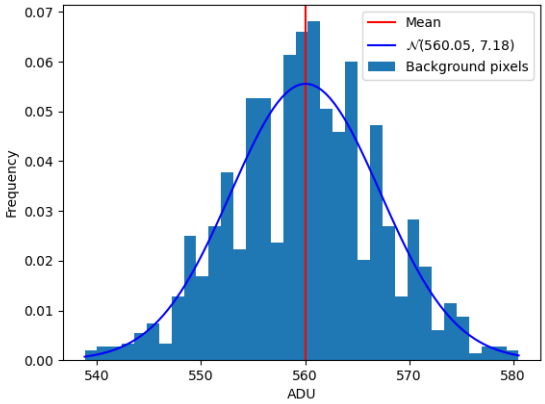


Figure 6.6: The distribution of the pixel ADUs after sigma clipping has been applied.

is a bright source that takes up a large part of the postage stamp. In this case we increase the area for the background calculation until 80% of the used area is identified as part of the background by the sigma clipping algorithm, i.e., until the majority of the contained pixels represents the background.

6.1.5 Calculate the PSF flux of sources in a postage stamp

We calculate the Point spread function (PSF) flux for every source within the postage stamp. This is because we do not want any source to interfere with the flux of the central candidate of interest. Hence, if we can subtract the PSF flux of any nearby source from the postage stamp, then the central source flux calculation will be more accurate.

6.1.5.1 Point spread functions

A point spread function shows a diffraction pattern resulting from capturing a point source with an optical system (Born, 1999).The convolution of the intrinsic surface brightness of a source and the PSF gives the resulting image as recorded by the apparatus [3]. This is illustrated in Figure 6.7.

The surface brightness μ of a point source at location (x_s, y_s) of flux $F \geq 0$ is given by

$$\mu(x, y) = \mu_0 + F\, S\left[\sqrt{(x - x_s)^2 + (y - y_s)^2}\right], \quad \forall x, y \in \mathbb{R}, \tag{6.5}$$

where μ_0 the background of the source and $S(r)$ the point spread function. Assuming that the system is isolated and that the apparatus only redistributes light (Lewis, 1926), photon conservation implies

[3]Since the optical lens does not converge the incoming light to an infinitely small point, the waves interfere and a diffraction pattern occurs. For a spherical aperture, an Airy disk is formed.

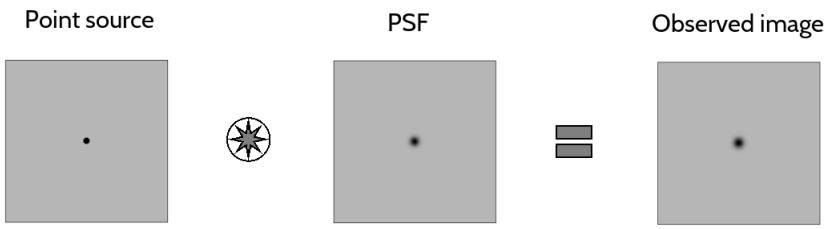


Figure 6.7: A point source (left) is convolved with a radially symmetric point spread function (middle) to arrive at the final blurred observed image (right).

that the integral of the sky subtracted surface brightness must equal the flux F,

$$\int_{-\infty}^{\infty} \mathrm{d}x \int_{-\infty}^{\infty} \mathrm{d}y \, [\mu(x, y) - \mu_0] = F, \tag{6.6}$$

which, transforming into radial coordinates, leads to

$$F \left[\int_0^{\infty} 2\pi r \, S(r) \, \mathrm{d}r \right] = F. \tag{6.7}$$

That is, we can multiply the integrated point spread function by any scalar factor F and get back F. From equation (6.7) it follows that the point spread function S must be normalized such that

$$\int_0^{\infty} 2\pi r \, S(r) \, \mathrm{d}r = 1. \tag{6.8}$$

Another requirement is that the surface brightness minus the background cannot be negative, $\mu(x, y) - \mu_0 \geq 0$. In order for that to hold, the PSF S must be non-negative as well, so that

$$S(r) \geq 0, \quad \forall r \in \mathbb{R}_0^+. \tag{6.9}$$

Combining (6.8) and (6.9) implies that S is a probability distribution over (x, y), which can be seen as a consequence of photon conservation.

Let us define a symmetric function f that is at maximum at radius $r_0 = 0$. The half width at half maximum (HWHM) is then given by $r_{1/2} \in \mathbb{R}_0^+$ such that $f(r_{1/2}) = f(r_0)/2$. The FWHM of f can then be calculated to be twice the HWHM,

$$FWHM = 2 \times HWHM, \tag{6.10}$$

which corresponds to the width of the function where its value is half of the maximum.

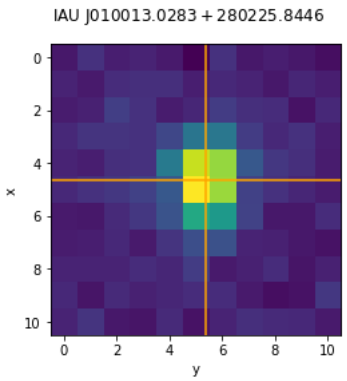


Figure 6.8: Example of a fitted source centre via the method of non-linear least squares, where the orange lines represent the fitted centre.

6.1.5.2 Using the PSF to obtain the flux

In order to fit for the flux of a source, we take a circular area of radius $R_{\text{flux}} = 2\farcs4$ around it. The flux is then obtained via non-linear least squares by fitting the observed pixel data to the generated pixel data of a point source with PSF S as a function of flux. Details on the flux and magnitude calculations are presented in Section 6.5 and 6.4. One could also think of it as the sum over the pixel fluxes within the postage stamp. This will give incorrect results when the source is not a point source.

In general, we separately fit for the following:

1. PSF flux: fit the flux of the central source
2. Source centre: fit the location of the central source
3. PSF flux + source centre: fit the flux and the location of the central source

So for the central source of interest we separately fit the PSF flux, the source centre and the PSF flux + source centre. In the first option we take the input location of the source (i.e. from one of the surveys' databases) as fixed and fit the PSF flux for that fixed position. This can give low PSF fluxes if e.g. the source has moved between two surveys or generally is not at the specified location.

That is why we add the second option, where we take the database flux of the source as fixed and fit for the position of the source. This should give a lower chi-squared value compared to the first option if the source has moved from the input location while having the same flux. Hence, fitting for the source centre helps us to find sources that have moved compared to e.g. a survey's database location. A fitted source centre of a postage stamp is shown in Figure 6.8.

In the third option, we fit for both the PSF flux of the central source and for its position. This should give the lowest chi-squared value of the three. If it is much lower than the first fit, then we know that

the source has moved from the input location. If it is much lower than the second, then we know that the database flux of that source is incorrect.

For all of the three options above it holds true that we fit the PSF flux of the sources that are nearby the central source - that is, for sources in the postage stamp at radius $r > R_{\rm keep}$. This is so that their flux does not interfere too much with the central chi-squared calculation within $R_{\rm keep}$.

6.1.6 Calculate the quasar probability and find the best fit quasar SED

We use the database photometry as the input for our calculations. Whenever no database photometry is available, we use the calculated PSF photometry instead. In the majority of cases, the database values are only absent when the source is very faint and so has not been detected by the survey.

The method to obtain the quasar probability and the best fit quasar SED and redshift is described in Section 7.1. We build on the codebase `Pq_server` from (Euclid Collaboration, 2019).

For the surface density of the high-redshift quasar population, Euclid Collaboration (2019) use the QLF of Jiang et al. (2016) and take k=-0.72. The surface density of quasars at $J \approx 22$ is much lower compared to that of the contaminants, with the density being of order $\sim 10^{-3}$ lower compared to that of MLT dwarfs and $\sim 10^{-5}$ to that of ETGs.
The template SEDs were obtained from a set of quasars at $0.1 \leq z \leq 4.0$ from SDSS DR7 (Schneider et al., 2010), cross-matched with NIR data from UKIDSS (Lawrence et al., 2007) and WISE (Wright et al., 2010). Using `synphot` (STScI Development Team, 2018), Euclid Collaboration (2019) then use the transmission filters of the surveys together with the template SEDs to obtain the colours and k-corrections at different redshifts.

Temple et al. (2021) show that dust reddening can significantly distort the template quasar colours from Euclid Collaboration (2019). These reddened colours of quasars at $7 < z < 8$ are similar to the predicted colours of ETGs. Therefore, such high-redshift quasars would likely be misclassified as ETGs in our pipeline. The same goes for high-redshift quasars with weak emission lines. This is slightly concerning since we are interested in the highest redshift quasars. In future, such models could be included in the SED templates of quasars.

For the MLT dwarf contaminants, Euclid Collaboration (2019) assume them to be members of the Galactic thin disk. The authors expect that Euclid will be able to observe some of the thick disk population at faint magnitudes, though they find that the largest population of contamination for high-redshift quasars are types $T2-T4$, which will only be visible to ~ 450 pc with $J < 24$. The individual number densities and spectral colours of the modelled MLT dwarf types $M0-T8$ can be seen in Table 2 in Euclid Collaboration (2019).
Euclid Collaboration (2019) derive the colours of the individual MLT dwarfs from measuring their respective median colours in the SpeX Prism Library Burgasser (2014). For types $M0 - M6$, the

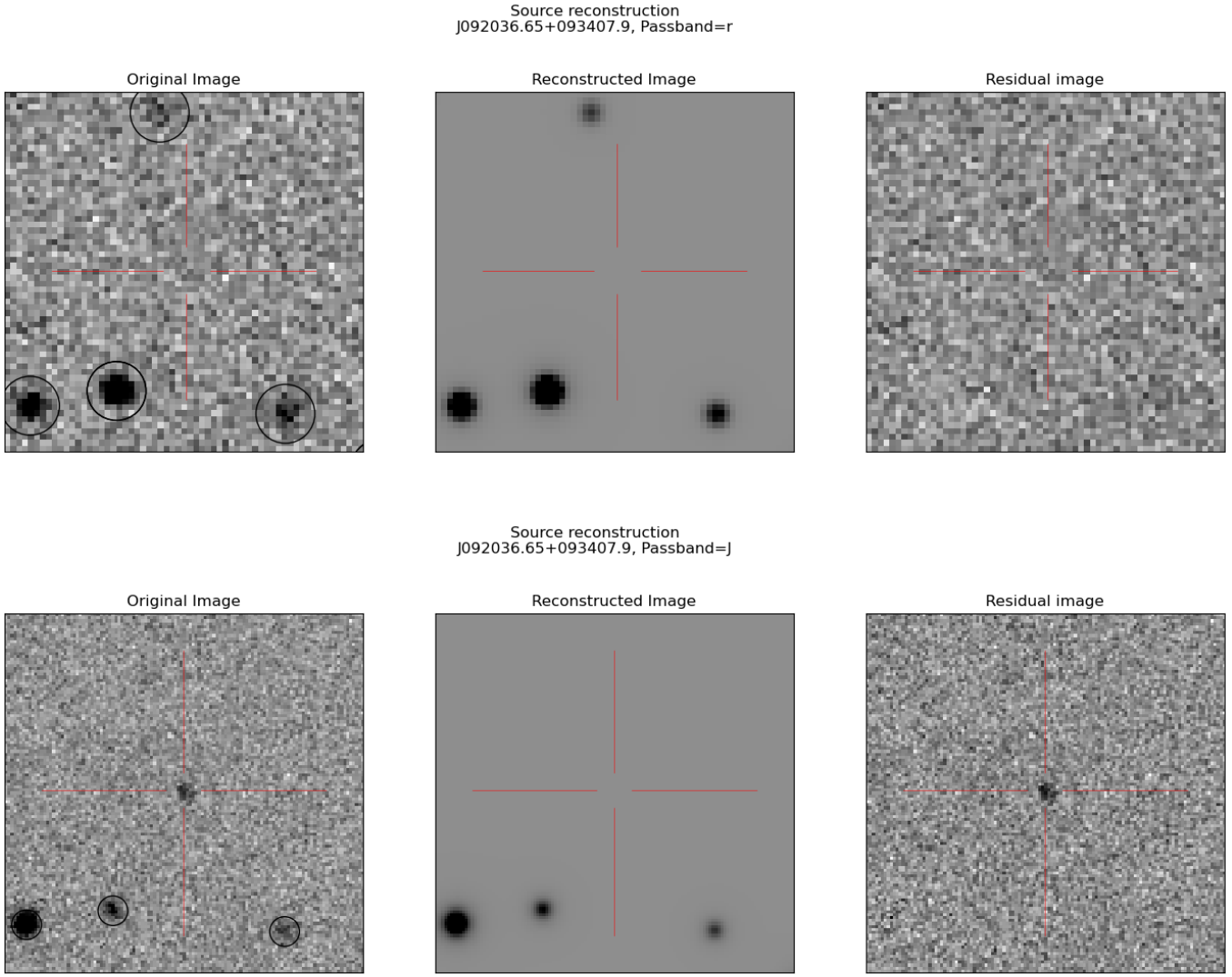


Figure 6.9: Reconstruction of nearby sources in the original postage stamp in r and J, before the central source gets reconstructed. The nearby sources are encircled in the plot on the left. The middle plot shows the reconstructed image before the central source flux is fitted, and the plot on the left shows the difference between the original image and the residual image.

number densities are derived from the luminosity function of Bochanski et al. (2010). For types $M7 - M9$, the number densities are extrapolated from Cruz et al. (2007). Cruz et al. (2007) arrive at a number density which combines $M7 - M9$. Euclid Collaboration (2019) then take this combined number density and approximate the individual number densities $M7 - M9$ by assuming a linear variation over the types $M7 - L0$. For types L and T ($L0 - T8$), the number densities are obtained from the UKIDSS LAS LT brown dwarf sample of Skrzypek et al. (2016).

For ETGs, Euclid Collaboration (2019) use the quiescent objects at redshifts $z = 1 - 2$ of the COSMOS dataset (Laigle et al., 2016), with $J > 22$ at an aperture of $1''$. From this data, the authors infer two populations of formation redshift in the ETG dataset, $z_1 = 3$ and $z_2 = 10$. Euclid Collaboration (2019) use this information together with the spectral evolution from Bruzual & Charlot (2003) to calculate their respective colours.
Their surface density as a function of J-band magnitude and redshift is obtained by fitting the maximum likelihood to the dataset.

6.1.7 Reconstruct postage stamps

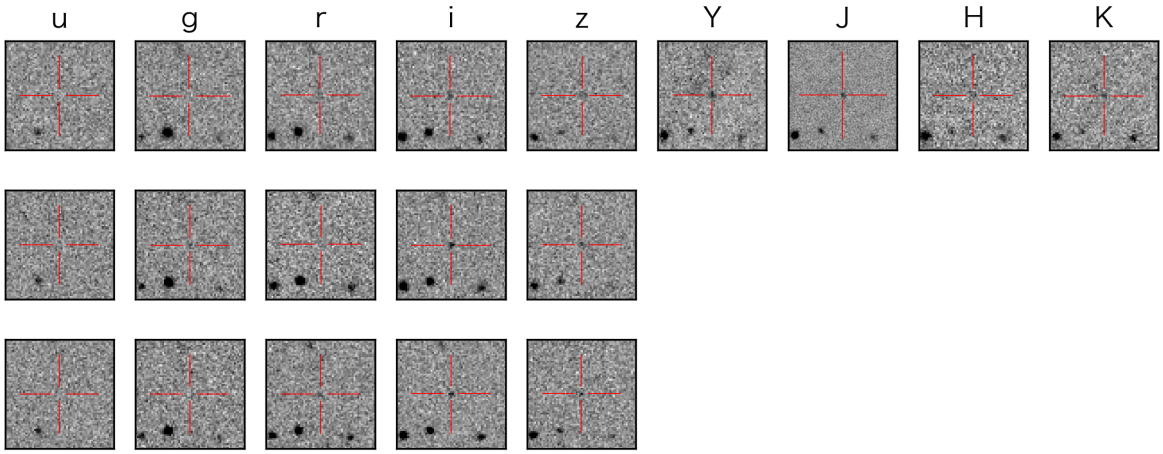
Once the best fit quasar SED has been obtained, we can completely reconstructed the postage stamps. We reconstruct each postage stamp as in equation (7.11) with K number of sources in the postage stamp. Note that for the central source of interest we are using the best fit quasar flux as obtained in the step above, whereas for the remaining sources in the postage stamp we are using the calculated PSF photometry. The reconstruction of nearby sources is shown in Figure 6.9.

Also note that we do not reconstruct sources that are within a radius of R_{keep} from the centre. The idea behind R_{keep} is that we do not want to fit sources that are too close to the central source. Rather, such candidates should be rejected as they make the photometric information of the central source unreliable.

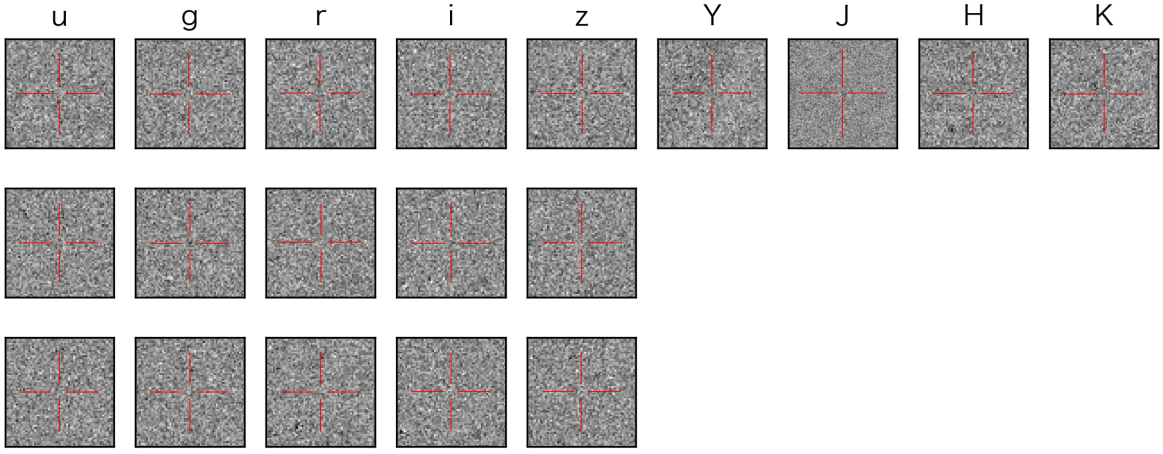
6.1.8 Calculate goodness of fit

The individual reduced $\tilde{\chi}^2_b$ and the overall reduced average $\tilde{\chi}^2$ values of the high-redshift quasar model are calculated as described in Section 7.2.1 in equations (7.8) and (7.9).

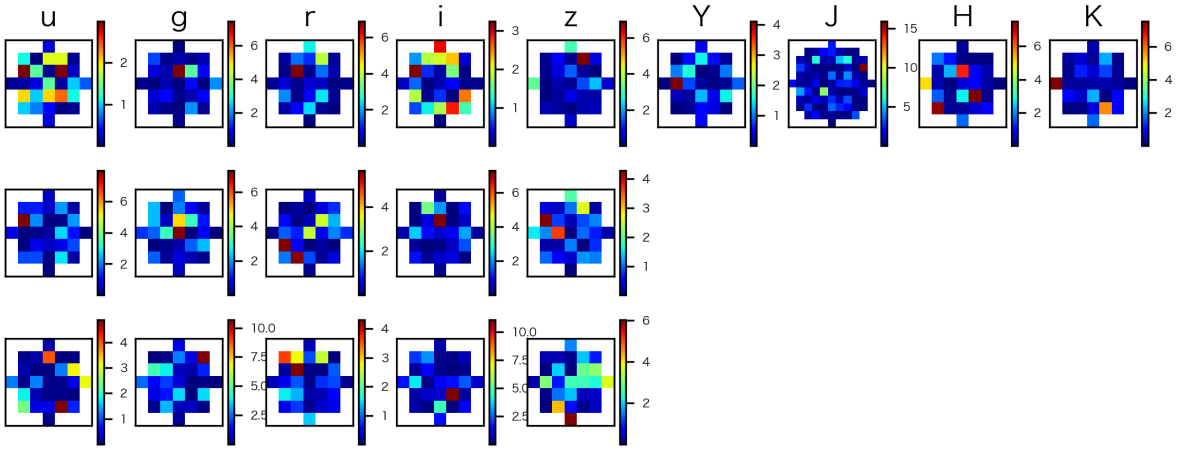
We calculate the goodness of fit of the quasar model starting from the centre of the postage stamp over a circular region of radius $R_{\tilde{\chi}^2} = 1''.2$. This radius gave us the best results in performance. If the radius is too large, the resulting chi-squared value gets blurred by the background pixels and hence is less informative. If the radius is too small, it does not encompass the central source and further is not able to detect any apparent motion of the source. This is shown in Figure 6.10, which shows the



(a) Original



(b) Residual



(c) Chi-squared pixel values

Figure 6.10: The original image (top), residual image (middle) and the chi-squared pixel values (bottom) of J092036.65+093407.9, using the best-fit quasar SED photometry. The chi-squared pixel values are shown over a circular region of radius $R_{\tilde{\chi}^2} = 1\farcs2$ from the centre.

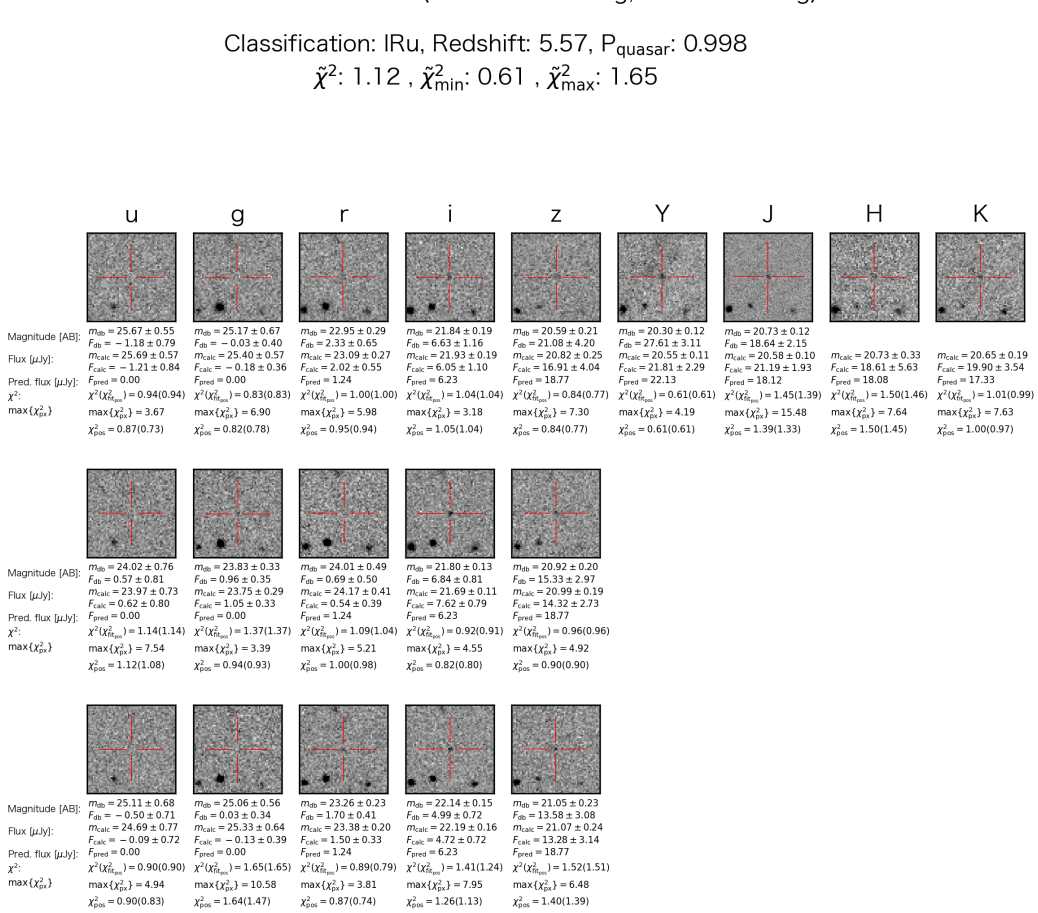


Figure 6.11: Summarised candidate J092036.65+093407.9. Photometry with the subscript 'db' is obtained from the database, and photometry with the subscript 'calc' is calculated with our data processing pipeline. F_{pred} is the best-fit quasar SED flux. χ^2 represents the chi-squared value (7.8) in each band, and $\chi^2_{\text{fit}_{\text{pos}}}$ represents the chi-squared value where the source centre is fitted along with the flux. (The chi-squared value χ^2_{pos} represents the chi-squared value where we use our calculated photometry instead of the best-fit quasar photometry, and hence this is a test against the morphology and stationarity of the source. We do not use this chi-squared value in any of our subsequent analysis.)

source J092036.65+093407.9 with its residual using the best-fit quasar SED photometry at the top, and the chi-squared pixel values at the bottom.

We repeat these steps with the difference that now the source centre of the central source is fitted as well. Comparing the two chi-squared values gives further insights into the stationarity of the candidate. The chi-squared value where the centre has been fitted will always be lower than the chi-squared value where it was not fitted, however a large discrepancy between the two strongly indicates that the source has moved.

6.1.9 Summarise candidate

Finally, we summarise each candidate. The information that is stored is listed in Table 6.2. Variables with the subscript `pos` include a fitted source centre, whereas variables without that subscript use the database source positions. In total, there are 70 summarised attributes per source.

We calculate $\tilde{\chi}^2_b$ as defined in (7.8). If there are multiple runs for a source then we average over them. For a source that has n_r number of single-epoch observations, we also obtain the minimum $\tilde{\chi}^2_{\text{min}} = \min\{\chi^2_{b,n}\}_{b\in B, n\leq n_r}$ and maximum $\tilde{\chi}^2_{\text{max}} = \max\{\chi^2_{b,n}\}_{b\in B, n\leq n_r}$ chi-squared values. We further calculate the maximum $\tilde{\chi}^2_{\text{pos,max}} = \max\{\tilde{\chi}^2_{\text{pos},b,n}\}_{b\in B, n\leq n_r}$ chi-squared value for the case when the source's position is fitted as well. We convert all fluxes into units of Janskys, and all magnitudes into AB magnitudes. These conversions are explained in Sections 6.4.4 and 6.5.2. Figure 6.11 shows the summarised photometry, chi-squared values and quasar probability for candidate J092036.65+093407.9.

6.2 Compatible surveys

We give the detailed pipeline implementation for SDSS DR12 and UKIDSS LAS DR10. Further, if one wants to add a new survey, one needs to adjust the database queries that are needed for the pipeline. These include retrieving PSF parameters (if given), photometry and image IDs for a given source location. On top of that, there will be different database flags for images that have data processing issues.

6.2.1 SDSS

SDSS uses the Sloan Foundation 2.5m telescope at the Apache Point Observatory in New Mexico. It has a wide-field CCD camera including five filters u, g, r, i, z. These range from the optical towards the near-infrared spectrum. SDSS has imaged 35% of the night sky. The broadband filters of SDSS together with their magnitude limits can be seen in Table 6.5. The data releases of SDSS with quasar discoveries can be seen in Table 6.6.

SDSS filters	*u*	*g*	*r*	*i*	*z*
Effective wavelengths (Å)	3511	4686	6165	7481	8931
Magnitude limits	22.0	22.2	22.2	21.3	20.5

Table 6.5: Effective wavelengths and magnitude limits of SDSS (Gunn et al., 1998)

Data Release	Discovered Quasars
DR7	105,783
DR12 (BOSS)	270,000
DR14 (eBOSS)	144,046

Table 6.6: SDSS Data Releases with quasar discoveries.

6.2.1.1 Image properties / source identifier

The camera consists of an array of 30 SITe/Tektronix CCDs each of 2048 by 2048 pixels (Gunn et al., 1998). The telescope moves along great circles with an exposure time in each filter of 54 seconds. Figure 6.12 shows the structure of the SDSS camera. The rows correspond to the five passbands *u*, *g*, *r*, *i* and *z* and the columns represent the six different scanlines. Each SDSS image can be uniquely

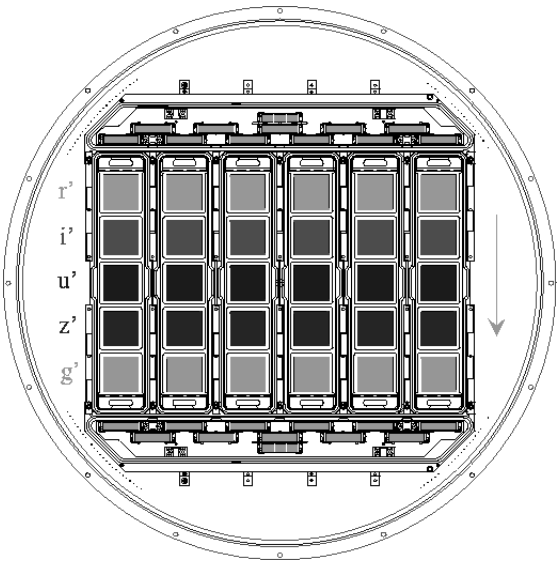


Figure 6.12: SDSS camera arrangement. Figure taken from Gunn et al. (1998)

identified by the following quantities

- run
- camera column (camcol)
- field
- passband

- rerun

The specific scanline in the camera is identified with the *camcol* number which ranges from 1 to 6. Within the scanline the *passband* defines a specific measurement image. The *rerun* number then defines how the original image was processed. Hence, for one original measurement there can be multiple images with different rerun numbers that have been processed in various ways. The several different scans can be identified with the *run* number. The *field* number corresponds to the index of the different measurements for each run. This is further illustrated in Figure 6.13.

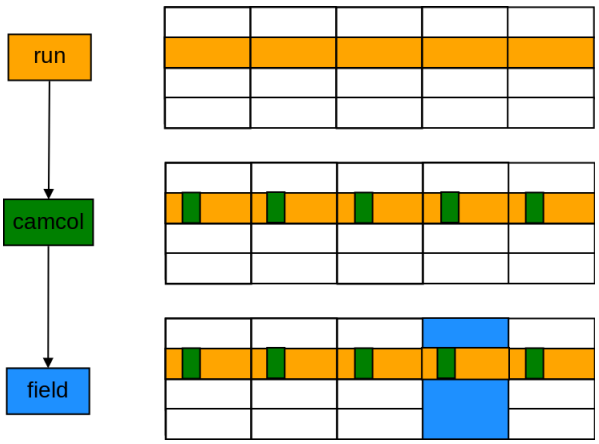


Figure 6.13: Illustration of SDSS' identifiers of run, camcol and field. They define a specific scanline within a measurement.

6.2.1.2 Querying the database

We use the SDSS Data Release 12 (DR12)[4]. Image identifiers and other catalog data can be retrieved via the Catalog Archive Server (CAS) through SkyServer[5]. The astronomical image for a given identifier can then be downloaded from the Science Archive Server (SAS)[6].

The following Python libraries are used to fetch the data from the servers:

- `urllib` [7]
- `wget` [8]

where `urllib` is a library that enables to parse and fetch URLs, while `wget` allows to download content from servers via different protocols.

[4] See `http://skyserver.sdss.org/dr12/en/help/browser/browser.aspx` for the schema browser
[5] `http://skyserver.sdss.org/dr12/en/home.aspx`
[6] `https://dr12.sdss.org/home`
[7] `https://docs.python.org/3/library/urllib.html`
[8] `https://pypi.org/project/wget/`

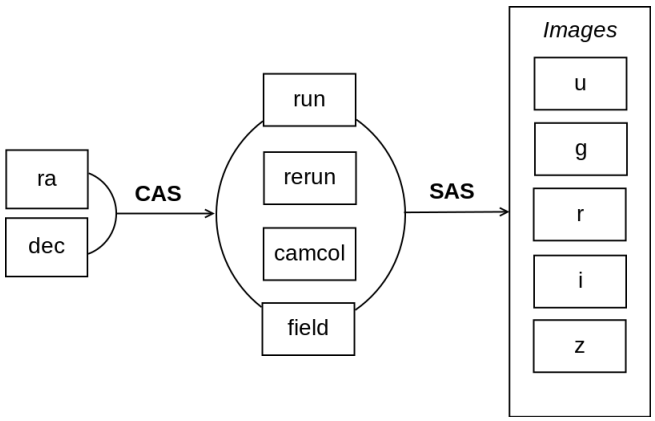


Figure 6.14: Communicating with SDSS' servers to retrieve images corresponding to a given location. First, the Catalog Archive Server (CAS) is queried in order to retrieve their identifiers. These quadruples then are used to create a URL for downloading the specific images from the Science Archive Server (SAS).

Retrieving image identifiers: The following query is used to retrieve the unique identifiers for a given right ascension (ra) and declination (dec) location,

```
SELECT DISTINCT TOP <max_entries> p.run, p.rerun, p.camcol,
 p.field, p.ra, p.dec
 FROM PhotoObjAll p
 WHERE (p.ra BETWEEN <ra> - <radius> AND <ra> + <radius>) AND
       (p.dec BETWEEN <dec> - <radius> AND <dec> + <radius>)
 GROUP BY p.run, p.rerun, p.camcol
```

where the radius is typically one arcsecond and the maximum amount of entries is 10. This gives the run, rerun, camcol and field number of unique images that contain the source at α, δ. In this case we identify a source image by their run, rerun and camcol number. Since each field overlaps by 128 pixels with adjacent fields, the same source can fall between two *field* numbers. As only one of these images is wanted, the *field* number is removed from the identifier for a given location.

In our database query we also test for database flags that would indicate data processing issues. For SDSS, images with the following flags are not downloaded: `PSF_FLUX_INTERP`, `BAD_COUNTS_ERROR`, `COSMIC_RAY`, `BADSKY`, `MAYBE_EGHOST`, `MAYBE_CR`, `EDGE`, `INTERP_CENTER` and `DEBLEND_NOPEAK`.

Downloading images: After a set of run, rerun, camcol and field numbers have been acquired from CAS, they are parsed to an URL that can subsequently be used to download the images in all broadband filters from the Science Archive Server (SAS). The URL has the following structure

```
https://dr12.sdss.org/sas/dr12/boss/photoObj/frames/<rerun>/
<run>/<camcol>/frame-<passband>-<run(6)>-<camcol>-<field(4).fits.bz2
```

where <> indicate the placeholders for the various variables, with (4) and (6) meaning that the number is filled up to 4/6 characters.

Photometry For all passbands, the magnitude and flux together with their errors for every source is retrieved via the following,

```
SELECT  p.u, p.g, p.r, p.i, p.z,
        p.err_u, p.err_g, p.err_r, p.err_i, p.err_z,
        modelFlux_u, modelFlux_g, modelFlux_r, modelFlux_i, modelFlux_z,
        modelFluxIvar_u, modelFluxIvar_g, modelFluxIvar_r,
        modelFluxIvar_i, modelFluxIvar_z
 FROM PhotoObj p
 WHERE  (p.ra BETWEEN <ra> - <radius> AND <ra> + <radius>) AND
        (p.dec BETWEEN <dec> - <radius> AND <ra> + <radius>)
        AND p.run=<run> AND p.rerun=<rerun> AND p.camcol=<camcol>
        AND p.field=<field>
```

SDSS stores the errors in the fluxes as inverse variances k, which have to be transformed to standard deviations σ via

$$k = \frac{1}{\sigma^2} \tag{6.11}$$

$$\Rightarrow \sigma = \sqrt{\frac{1}{k}} \tag{6.12}$$

The magnitude errors are already stored as standard deviations.

6.2.2 UKIDSS

UKIDSS is a near-infrared sky survey using the UKIRT telescope and WFCAM wide field camera (Lawrence et al., 2007). It is located in Hawaii where it has observed 7500 square degrees of the Northern sky. From that, the Large Area Survey (LAS) covers 4000 square degrees in the infrared wavelength bands Y, J, H, K. This area had previously been observed via SDSS in optical filters.

6.2.2.1 Image properties / Source identifier

The WFCAM consists of four Rockwell Hawaii-II 2048×2048 CCDs in the filter bands J, H, K and Y (Casali et al., 2001, 2007). The arrangement can be seen in Figure 6.15. The observations in the four

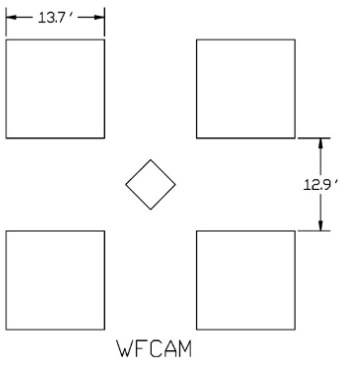


Figure 6.15: Alignment of the WFCAM. Figure taken from Casali et al. (2007). Four Rockwell cameras are mounted with an auto-guider in the centre.

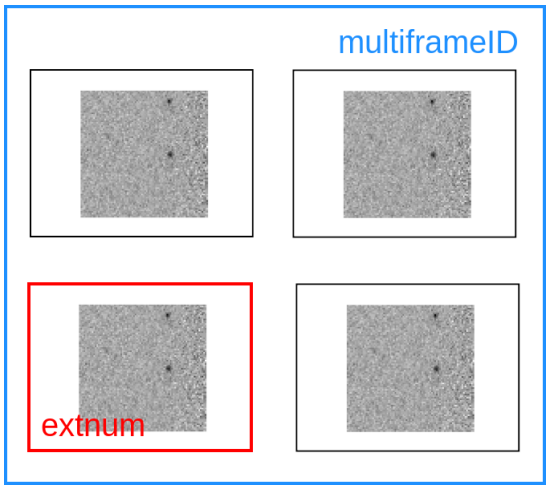


Figure 6.16: Image identifiers of UKIDSS

filter bands are carried out separately and are then tiled together to form a coverage of 0.75 square degrees. Each UKIDSS image can be uniquely identified via

- `multiframeID`
- `extnum`
- `filterID`

where `multiframeID` is the unique id of the relevant multiframe, `extnum` the extension number of the multiframe and `filterID` the id of the passband. Different extension numbers contain different metadata of an image. The first extension includes metadata that is true for all objects in the multiframe (e.g. the passband), whereas the subsequent extensions have metadata for individual detectors. Hence `extnum` from 2 to 5 give the image within a multiframe. This can be seen in Figure 6.16. In this example the extension number of the detector in the left bottom corner corresponds to number 4.

6.2.2.2 Photometry

UKIDSS includes a set of 13 circular-aperture fluxes and magnitudes of fixed aperture size. All of the fluxes in pixels have been scaled according to the fraction of the pixel enclosed. Typically the most accurate flux and magnitude are `aperFlux3` and `aperMag3`.

6.2.2.3 Querying the database

In order to retrieve information about the UKIDSS LAS survey, we query the WFCAM Science Archive (WSA) [9] (Hambly et al., 2008) with the specific database of UKIDSSDR10PLUS.

The following Python libraries are used to fetch the data:

- `urllib`
- `optparse`
- `wget`

The Python package `urllib` is used to fetch data via URL requests. `optparse` is used to parse the SQL queries into URL format. `wget` allows downloading content from the web.

Retrieving image identifiers: The following SQL query retrieves the UKIDSS LAS image identifiers for a given location at right ascension (ra) and declination (dec):

```
SELECT DISTINCT m.multiframeID, m.filterID, d.extnum
 FROM lasDetection s, Multiframe m, MultiframeDetector d
 WHERE (s.ra BETWEEN <ra> - <radius> AND <ra> + <radius>) AND
       (s.dec BETWEEN <dec> - <radius> AND <dec> + <radius>) AND
        s.multiframeID = m.multiframeID AND
        m.multiframeID = d.multiframeID  AND
        d.extnum=s.extnum
```

This gives us the identifiers of the multiframe id, filter id (passband) and the extension number. The filter id for each passband is encoded and listed in Table 6.7: For the LAS survey, we only make use of the numbers $2-5$.

In our query we also test for database flags that would indicate any data processing issues. For UKIDSS, we require that `ppErrBits < 256` and `errBits < 8`.

[9] `http://wsa.roe.ac.uk/`

Passband	Filter ID
Z	1
Y	2
J	3
H	4
K	5
H2	6
Br	7

Table 6.7: Filter ID for each UKIDSS passband

Downloading images:

1. First, a HTTP GET request is made in order to retrieve the web pages that contain information to download the images,

   `http://surveys.roe.ac.uk:8080/wsa/GetImage?database=UKIDSSDR10PLUS&programmeID=101&sys=J&ra=<ra>&dec=<dec>&filterID=all&obsType=object&frameType=stack&mfid=&xsize=<width(arcsec)>&ysize=<width(arcsec)>`

   where `sys=J` corresponds to the $J2000$ coordinate system and `filterID=all` allows to download all images in all passbands corresponding to that source. `<width>` is given in arcseconds, with a default value of width = 1. Note that the link only uses information about the location of the source in order to download their images. The following is an example of such an URL,

   `http://surveys.roe.ac.uk:8080/wsa/GetImage?database=UKIDSSDR10PLUS&programmeID=101&sys=J&ra=0.0165&dec=-1.1373799999999998&filterID=all&obsType=object&frameType=stack&mfid=&xsize=1.0&ysize=1.0`

2. The HTML response is then parsed to retrieve another web page for the image which contains information about the location and the final URL to download the FITS file.

3. The HTML result is then parsed to retrieve the URLs for downloading the images. The URLs have the following form:

   `http://wsa.roe.ac.uk/cgi-bin/getFImage.cgi?file=/disk<disk>/wsa/ingest/fits/<filename>.fit&mfid=<mfid>&extNo=<extno>&lx=<lx>&hx=<hx>&ly=<ly>&hy=<hy>&rf=<rf>&flip=1&uniq=<uniq>&xpos=<xpos>&ypos=<ypos>&band=Y&ra=<ra>&dec=<dec>`

   An example of such an URL is given below,

```
http://wsa.roe.ac.uk/cgi-bin/getFImage.cgi?file=/disk20/
wsa/ingest/fits/20050921_v1/w20050921_01495_st.fit&mfid=268655&extNo=1&
lx=946&hx=1096&ly=425&hy=575&rf=270&flip=1&uniq=1622_785_16_71151_1&
xpos=75.7&ypos=76.1&band=Y&ra=0.0165&dec=-1.13738
```

These can then be downloaded via `wget` or other libraries. Note that we cannot directly skip to step 3 since the last URL contains information that is not queriable from the database.

Photometry The following query is used to retrieve the fluxes and magnitudes of the UKIDSS sources,

```
SELECT s.ra, s.dec,
       s.aperMag3 as '<passband>', s.aperMag3err as 'err_'+'<passband>',
       s.aperFlux3 as 'modelFlux_'+<passband>,
       s.aperFlux3err as 'modelFluxVar_'+<passband>,
       d.aperCor3, d.skyCorrCat
 FROM lasDetection s, Multiframe m, MultiframeDetector d
 WHERE s.ra BETWEEN <ra> - <radius> AND <ra> + <radius> AND
       s.dec BETWEEN <dec> - <radius> AND <dec> + <radius> AND
       s.multiframeID = m.multiframeID AND m.multiframeID = d.multiframeID AND
       d.extnum = s.extnum AND
       m.multiframeID = <mfid> AND d.extnum = <ext_no> AND
       m.filterID = <filter_id>
```

The UKIDSS datbase stores both flux errors and magnitude errors as standard deviations. The photometry is calculated via successive sizes of circular apertures, where `aperMag3` and `aperFlux3` correspond to a diameter of 2″ which is the default aperture.

6.3 Implemented point spread functions

The Point spread function (PSF) is explained in detail in Section 6.1.5.1. The PSF models that are used here are illustrated together with their cumulative distribution in Figure 6.17 and are explained in the following.

For SDSS we use a Moffat-combined function for the PSF as stated below (Xin et al., 2018). The best-fit parameters are queried from the database for each source. Another PSF the SDSS database offers is a two-Gaussian fit, which we implement here as well but do not use. UKIDSS only stores the FWHM in its database, which we use to approximate the PSF with a single Moffat function.

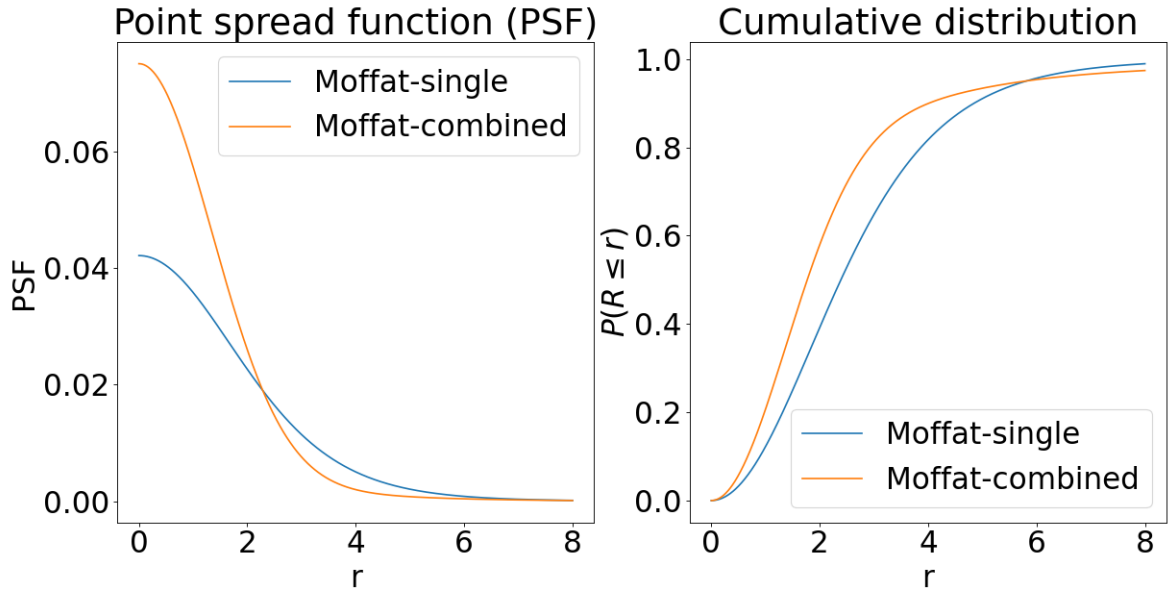


Figure 6.17: An example of the different point spread functions for SDSS (Moffat-combined) and UKIDSS (Moffat-single) that have been normalized. The following parameters were used for this plot: (1) Moffat-combined: $\sigma_1 = 1.3, \sigma_2 = 3.0, \sigma_P = 2.5, p_0 = 0.1, \beta = 3.6, b = 0.03$; (2) Moffat-single: $\alpha = 5.49, \beta = 5$. On the left the individual PSFs are plotted on top of each other. The right side shows the cumulative distributions of these PSFs.

We also show a Gaussian and Two-Gaussian PSF which we used initially as our PSFs before we had implemented the Moffat PSFs. A Moffat(-combined) PSF is a better representation of a point source than a Gaussian or Two-Gaussian and hence we use that. But our pipeline is able to model the other two PSFs as well.

If one wanted to use an empirical PSF instead, one could make use of `EPSFBuilder` from the `photutils` library (Anderson, 2016). Another library one could use is `PSFEx` (Bertin, 2011). In both cases, one could use a set of image cutouts of bright isolated stars to arrive at an empirical PSF for each band. Specifically, we could use a set of likely high-redshift quasar candidates to obtain an empirical PSF. This would make the flux calculation of high-redshift quasars more accurate. In return, this would make it easier to distinguish between quasar and contaminant models. However, the disadvantage is that one could not use that PSF for any other survey. The other disadvantage is that the PSF fluxes of the contaminant populations could be incorrect which could lead to false positives. Since we have already implemented the database accesses for both SDSS and UKIDSS, we decide to use the PSF information from their databases instead.

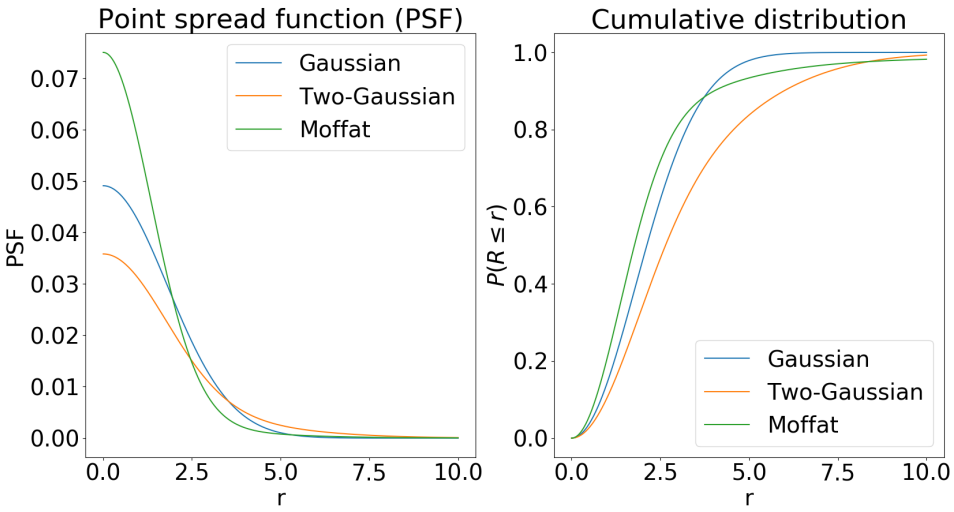


Figure 6.18: Examples of three different point spread functions that have been normalized. The following parameters were used for the PSFs: (1) Gaussian: $\sigma = 1.8$, (2) Two-Gaussian: $\sigma_1 = 1.7$, $\sigma_2 = 3.5$, $R = 0.2$, (3) Moffat: $\sigma_1 = 1.3, \sigma_2 = 3.0, \sigma_P = 2.5, p_0 = 0.01, \beta = 3.6, b = 0.03$. On the left the individual PSFs are plotted on top of each other. The right side shows the cumulative distributions of these PSFs.

6.3.1 Gaussian

The point spread function can be modelled via a Gaussian function as

$$S\,(r;\sigma) = \sqrt{\frac{\pi}{2}}\,\exp\left[-\frac{1}{2}\left(\frac{r}{\sigma}\right)^2\right]. \tag{6.13}$$

The standard deviation of the Gaussian can be obtained from the FWHM via

$$FWHM = 2\,\sqrt{2\ln 2}\sigma \tag{6.14}$$

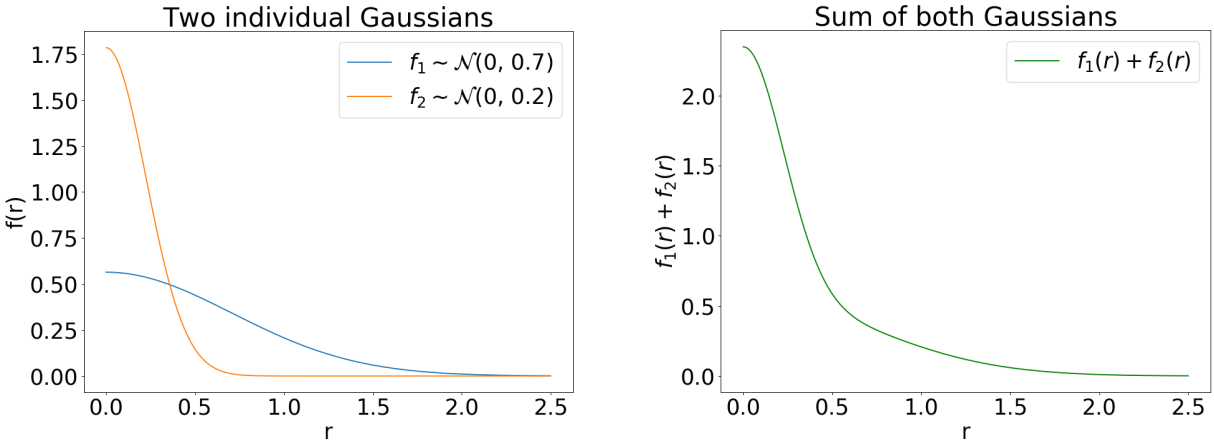
An example of a Gaussian PSF is shown in Figure 6.18.

6.3.2 Two-Gaussian

A representation with more heavy wings is given by a combination of two Gaussians as (Gunn et al., 1998)

$$S\,(r;K,\sigma_1,\sigma_2) = \frac{1}{2}\,(K+1)\left\{\frac{1}{\sqrt{2\pi}\sigma_2}K\,\exp\left[-\frac{1}{2}\left(\frac{r}{\sigma_2}\right)^2\right] + \frac{1}{\sqrt{2\pi}\sigma_1}\exp\left[-\frac{1}{2}\left(\frac{r}{\sigma_1}\right)^2\right]\right\}, \tag{6.15}$$

where $\{K,\sigma_1,\sigma_2\}$ the parameters of the Two-Gaussian, with K representing the ratio between the two overlapping Gaussian functions. This is shown in Figure 6.19*a*. The two Gaussian functions have the



(a) Two Gaussians with different standard deviation σ and same mean μ

(b) The sum of the two Gaussians gives rise to the unnormalized PSF

Figure 6.19: Two Gaussians with different σ form a point spread function.

same mean, but a different spread. Hence, together with the ratio K, the tails can be modelled to be more extended while the central peak can remain high. The PSF as the sum of the two Gaussians can be seen in 6.19*b* for the case that $K = 3$.

6.3.3 Moffat

We would like to make a more accurate fit via the Moffat distribution, which is given by

$$S\,(r;\alpha,\beta) = \frac{1}{\|S\|}\left[1 + \left(\frac{r}{\alpha}\right)^2\right]^{-\beta}, \tag{6.16}$$

where r the radius and $\{\alpha,\beta\}$ the parameters of the point spread function (Moffat, 1969). The normalization factor is defined as $\frac{1}{\|S\|} = \frac{\beta-1}{\alpha\pi^2}$. A comparison with a Gaussian PSF can be seen in Figure 6.20.

As S is a probability distribution, we have $S\,(r;\alpha,\beta) \geq 0$ and $2\pi\int_0^\infty r\,S\,(r;\alpha,\beta)\,dr = 1$. This gives

$$\frac{\beta-1}{\alpha\pi^2}\,2\pi\int_0^\infty r\left[1 + \left(\frac{r}{\alpha}\right)^2\right]^{-\beta} dr = 1 \tag{6.17}$$

In order for the left hand side to be finite, the integral must converge, and hence β cannot be negative or zero. Non-negativity of S implies that β has to be larger than 1. Further, α cannot be zero,

1. $\beta > 1$
2. $\alpha \neq 0$

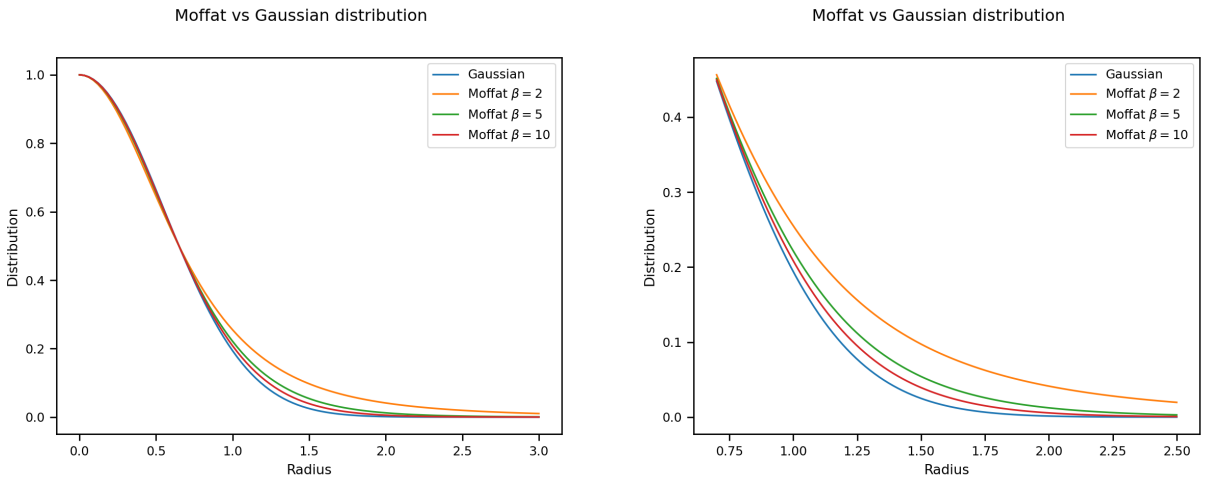


Figure 6.20: Moffat vs Gaussian PSF for the same FWHM. The Moffat distribution is shown for different β values. As β increases, the Moffat distributions gets closer to the Gaussian distribution.

The parameters α and β of the Moffat PSF can be obtained from the FWHM, which is retrieved from the FITS file. We select 2000 stars from UKIDSS LAS and for each we fit β in every passband. Then, we bin the FWHM values into 8 bins and create a map that gives the average β value for each of those bins. This map is then used to retrieve an interpolated β value for each FWHM. The parameter α can then be calculated via the FWHM,

$$\text{FWHM} = 2\alpha\sqrt{2^{1/\beta}-1} \tag{6.18}$$

$$\Longleftrightarrow \alpha = \frac{\text{FWHM}}{2\sqrt{2^{1/\beta}-1}} \tag{6.19}$$

Therefore, only β needs to be fitted for a fixed FWHM value.

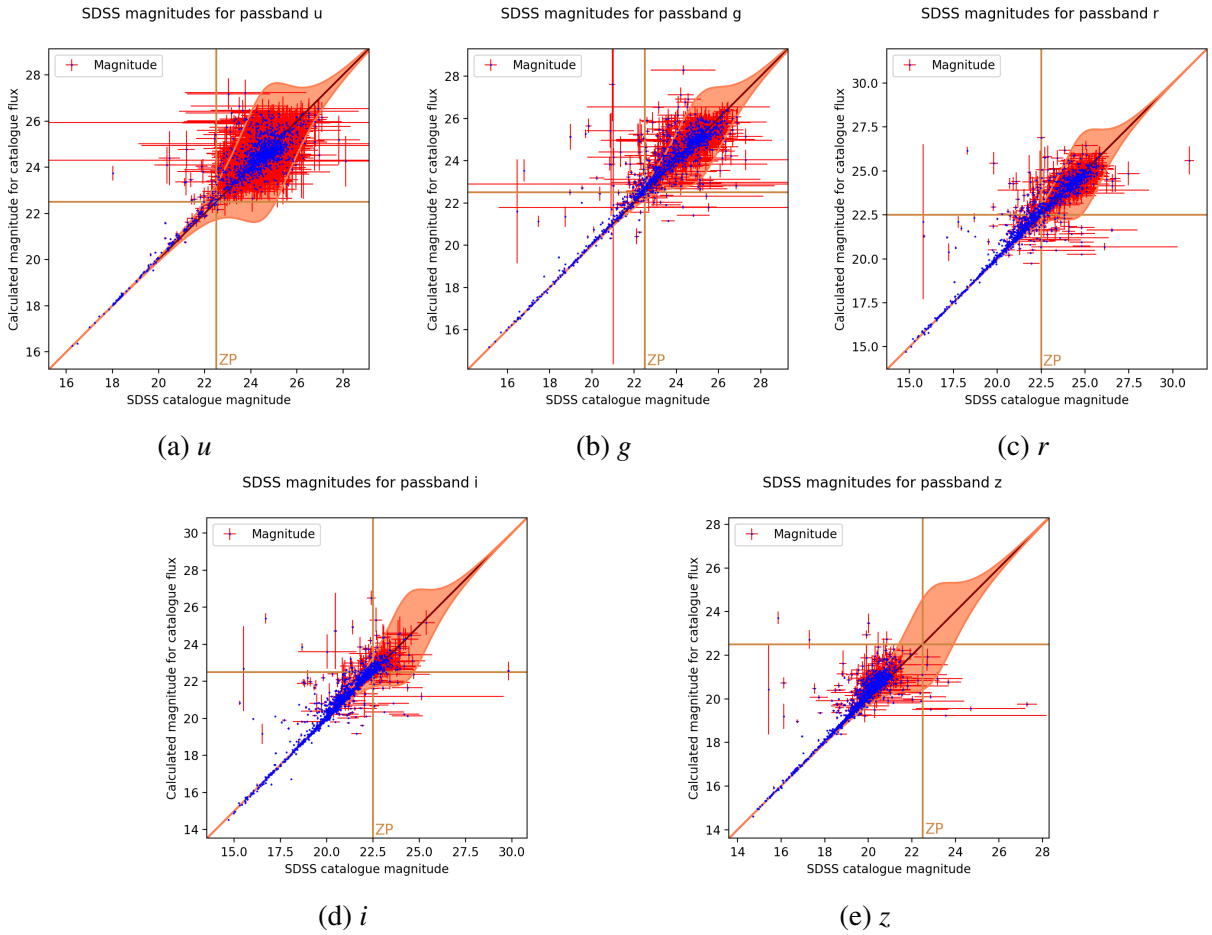
This can in theory be done for any survey that has seeing / FWHM information available. We do not use this for the SDSS dataset as information of a Moffat-combined PSF is already available in their database.

6.3.4 Moffat-combined

An even more sophisticated PSF is given by Xin et al. (2018),

$$S(r; b, p_0, \beta, \sigma_1, \sigma_2, \sigma_P) = \frac{1}{\|S\|}\frac{\exp\left[-\frac{1}{2}\left(\frac{r}{\sigma_1}\right)^2\right] + b\,\exp\left[-\frac{1}{2}\left(\frac{r}{\sigma_2}\right)^2\right] + p_0\left[1+\frac{1}{\beta}\left(\frac{r}{\sigma_P}\right)^2\right]^{-\beta/2}}{1+b+p_0}, \tag{6.20}$$

where again r is the radius and $\{b, p_0, \beta, \sigma_1, \sigma_2, \sigma_P\}$ the parameters of the point spread function. This function shows power-law wings similar to a Moffat function.



(a) *u* (b) *g* (c) *r*

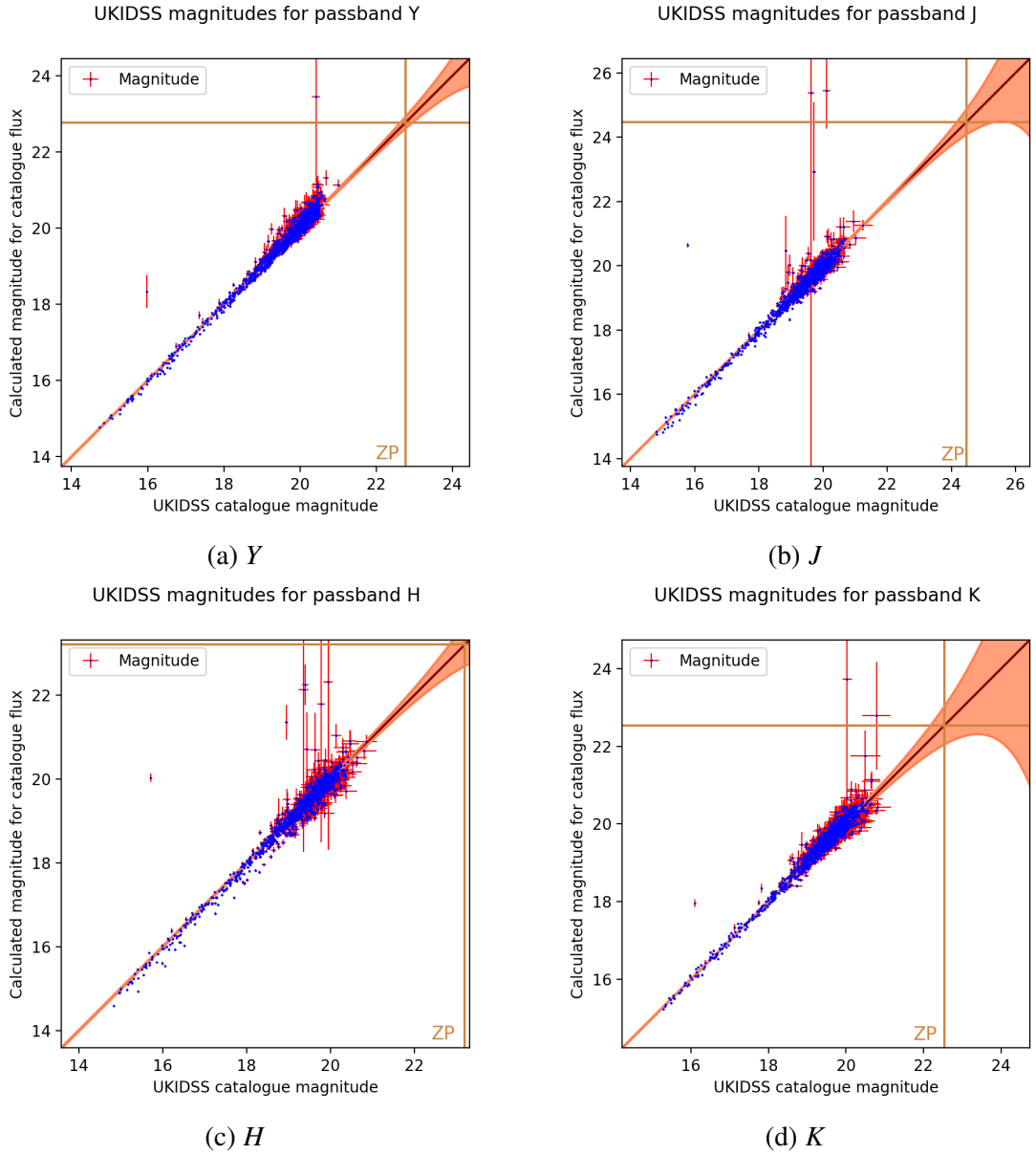
(d) *i* (e) *z*

Figure 6.21: Database magnitudes vs calculated magnitudes for SDSS. The set of ∼ 4300 source locations is obtained from Mortlock et al. (2012). The blue dots represent the magnitude and the red bars represent the error bars. ZP represents the zero-point magnitude. The shaded area represents the median standard deviation of the magnitudes of the central brown line.

6.4 Magnitude calculations

We explain how we obtain the magnitude from the calculated flux. The magnitudes are not used for the quasar probability calculation. Instead, linear fluxes are used for that. However, we add the magnitude calculations for completeness. They can also further be used to obtain spectral colours. In addition, asinh magnitudes allow us to represent negative fluxes.

Figures 6.21 and 6.22 show the database magnitudes plotted against our calculated magnitudes for each band in SDSS and UKIDSS respectively. These magnitudes have been plotted for a set of source locations that has been obtained from Mortlock et al. (2012). We can see that for bright sources $m < 21$ the calculated magnitudes are nearly the same to the database magnitudes, while the error increases for fainter magnitudes as expected. The majority of data points lie within the median standard deviation of the magnitudes of the central brown line.



(a) *Y* (b) *J*

(c) *H* (d) *K*

Figure 6.22: Database magnitudes vs calculated magnitudes for UKIDSS. The set of ~ 4300 source locations is obtained from Mortlock et al. (2012). The blue dots represent the magnitude and the red bars represent the error bars. ZP represents the zero-point magnitude. The shaded area represents the median standard deviation of the magnitudes of the central brown line.

6.4.1 Pogson vs asinh magnitude

We explain Pogson magnitudes and asinh magnitudes, which are used by UKIDSS and SDSS respectively.

6.4.1.1 Pogson magnitude

The Pogson magnitude (Pogson, 1856) is defined as

$$m = -2.5 \log_{10}\left(\frac{F}{F_0}\right), \tag{6.21}$$

where F the flux in some arbitrary units. F_0 is the reference flux (in the same flux units) that per convention corresponds to a magnitude of zero. We can see that if $F = F_0$, then $m = -2.5 \log_{10}(1) = 0$ as expected. The relation between the reference flux F_0 and the corresponding zero-point magnitude ZP is given by

$$ZP = -2.5 \log_{10}(F_0). \tag{6.22}$$

One can then rewrite equation (6.21) as

$$m = -2.5 \log_{10}(F) - ZP \tag{6.23}$$

Different magnitude systems use different calibrated zero-points.

Vega magnitude system The zero-point magnitudes in the Vega magnitude system are defined as the flux of Vega in a given band. That is, the zero-point magnitude is different in each band.

AB magnitude system The AB magnitude system is defined as the Pogson magnitude (6.23) with a zero-point magnitude of $ZP = -48.6$ in every band (Oke & Gunn, 1983),

$$m_{\mathrm{AB}} = -2.5 \log_{10}(F) + 48.6, \tag{6.24}$$

where the flux is in units of ergs cm^{-2} s^{-1} Hz^{-1}.

Rearranging equation (6.22), we can obtain F_0 to be

$$\begin{aligned} F_0 &= 10^{-\frac{48.6}{2.5}} \text{ ergs cm}^{-2} \text{ s}^{-1} \text{ Hz}^{-1} && (6.25)\\ &= 3.6307805\ldots \times 10^{-20} \text{ ergs cm}^{-2} \text{ s}^{-1} \text{ Hz}^{-1} && (6.26)\\ &\approx 3631 \times 10^{-23} \text{ ergs cm}^{-2} \text{ s}^{-1} \text{ Hz}^{-1} && (6.27)\\ &= 3631 \text{ Jy} && (6.28) \end{aligned}$$

A Jansky (Jy) is therefore defined as

$$1 \text{ Jy} = 10^{-23} \text{ ergs cm}^{-2} \text{ s}^{-1} \text{ Hz}^{-1} \tag{6.29}$$

Using this zero-point reference flux of $F_0 = 3631$ Jy in every filter, we can rewrite equation (6.24) to be

$$m_{\text{AB}} = -2.5 \log_{10}\left(\frac{F_{\text{Jy}}}{3631\ Jy}\right), \tag{6.30}$$

where F_{Jy} the flux in units of Jansky. Note that from the derivations above it follows that $F_0 = 3631$ Jy is an approximation to the true zero-point reference flux.

6.4.1.2 asinh magnitude

The asinh magnitude is defined as (Lupton et al., 1999)

$$m = -a\left[\sinh^{-1}\left(\frac{F}{2\,b}\right) + \ln(b)\right], \tag{6.31}$$

where a a constant, and b the softening parameter which corresponds to the flux above which linearity sets in.

6.4.1.3 A comparison between the two magnitudes

Since the Pogson magnitude takes the logarithm of the flux, it cannot handle negative fluxes. As $\lim_{x\to 0} \log_{10}(x) = -\infty$, the Pogson magnitude of faint objects varies a lot due to the steep gradient. In contrast to the Pogson magnitude, the asinh magnitudes can capture negative fluxes and does not vary too much at low S/N. This is illustrated in Figure 6.23. At high S/N, Pogson and asinh magnitudes are the same. While the Pogson magnitude tends to inifinity at low flux levels, the asinh magnitude rises steeply but stays finite.

6.4.2 SDSS asinh magnitude calculation

SDSS uses asinh magnitudes as in (6.31), using the constant $a = \frac{2.5}{\ln(10)}$ and a dimensionless softening parameter b that is measured relative to F_0 (Stoughton et al., 2002). The softening parameter b for each passband is given in Table 21 in Stoughton et al. (2002). This gives

$$m_{\text{asinh}} = -\frac{2.5}{\ln(10)}\left[\sinh^{-1}\left(\frac{F/F_0}{2\,b}\right) + \ln(b)\right]. \tag{6.32}$$

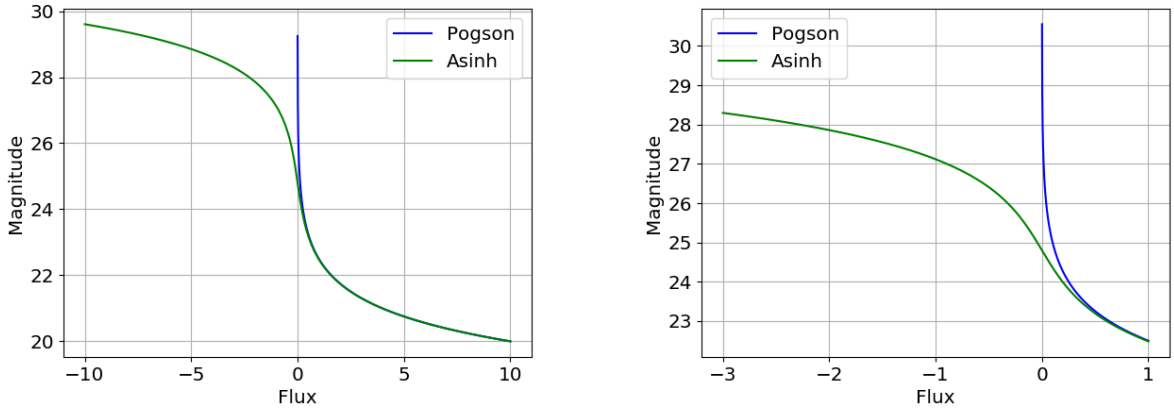


Figure 6.23: Comparison between Pogson (blue) and asinh (green) magnitudes. At higher flux levels, the two magnitudes coincide. At very low signal-to-noise ratios, the Pogson magnitude tends towards infinity, whereas the asinh magnitude keeps finite values. Moreover, the asinh magnitude is able to capture negative fluxes which is not possible with Pogson magnitudes.

6.4.2.1 Error in the asinh magnitude

Starting from equation (6.32) for the asinh magnitude, one can calculate the error in m_{asinh} via error propagation

$$\sigma^2_{m_{\text{asinh}}} = \left(\frac{dm_{\text{asinh}}}{dF}\right)^2 \sigma^2_F \tag{6.33}$$

where f, the flux, is the only variable parameter. The derivative of m with respect to f is given by

$$\frac{dm_{\text{asinh}}}{dF} = -\frac{2.5}{\ln(10)} \frac{1}{b\,F_0 \sqrt{\frac{F^2}{b^2 F_0^2} + 4}} \tag{6.34}$$

6.4.3 UKIDSS Vega magnitude calculation

UKIDSS uses the Vega magnitude system, where the Pogson magnitude is used as in equation (6.23) including various correction terms. The calibrated magnitude is then calculated via the following,

$$m_{\text{Vega}} = ZP - 2.5\,\log_{10}\left(\frac{F_{\text{ADU}}}{t}\right) - \texttt{aperCor3} - \texttt{skyCorrCat} - \texttt{extCorr} \tag{6.35}$$

where F_{ADU} the flux in Analog-to-digital unit (ADU) units, ZP the zero-point magnitude (flipped sign), t the exposure time, `aperCor3` the aperture correction, `skyCorrCat` the sky correction and `extCorr` the extinction correction.

The aperture correction term `aperCor3` corrects for using an aperture radius of 1″. Typically, we set `aperCor3` to zero, but we also have the option to use an aperture for our flux calculation. The

extinction correction is calculated as

$$\texttt{extCorr} = k\left(\frac{\text{amStart} + \text{amEnd}}{2} - 1\right) \tag{6.36}$$

The correction factors and information on how they are extracted are stated in Table 6.8.

Description	FITS Keyword (hdul index)	Database analogue (Multiframe - m, MultiframeDetector - d)
Exposure time	EXP_TIME (0)	expTime (m)
Airmass at start of observation	AMSTART (0)	amStart (m)
Airmass at end of observation	AMEND (0)	amEnd (m)
Zero point magnitude	MAGZPT (1)	photZPCat (d)
Zero point magnitude error	MAGZRR (1)	photZPErrCat (d)
Percentage sky correction	PERCORR (1)	skyCorrExt / skyCorrCat (d)
Extinction coefficient	EXTINCT (1)	extinctionExt / extinctionCat (d)
Stellar aperture correction (Aperture: r = 1 arcsec)	/	aperCor3 (d)

Table 6.8: Information on variables that are needed for correcting the measured flux. The second column represents the keyword in the image FITS file including the hdul index in which the keywords are stored. The third column shows the database analogues taken from the Multiframe and MultiframeDetector database table.

6.4.3.1 Error in the UKIDSS magnitude

The error in the UKIDSS magnitude can be obtained via

$$\sigma^2_{m_{\text{Vega}}} = \left(\frac{dm_{\text{Vega}}}{dF_{\text{ADU}}}\right)^2 \sigma^2_{F_{\text{ADU}}}. + \left(\frac{\partial m_{\text{Vega}}}{\partial \text{ZP}}\right)^2 \sigma^2_{\text{ZP}} \tag{6.37}$$

The first term is calculated as follows,

$$\begin{aligned}
\frac{\partial m_{\text{Vega}}}{\partial F_{\text{ADU}}} &= \frac{\partial}{\partial F_{\text{ADU}}}\Big[-2.5\log_{10}\Big(\frac{F_{\text{ADU}}}{t}\Big)\Big] && (6.38)\\
&= \frac{\partial}{\partial F_{\text{ADU}}}\Big[-2.5\log_{10}(e)\ \log_e\Big(\frac{F_{\text{ADU}}}{t}\Big)\Big] && (6.39)\\
&= -2.5\log_{10}(e)\,\frac{t}{F_{\text{ADU}}}\frac{1}{t} && (6.40)\\
&= \frac{-2.5\log_{10}(e)}{F_{\text{ADU}}} && (6.41)\\
&\approx -\frac{1.08574}{F_{\text{ADU}}}, && (6.42)
\end{aligned}$$

where we have used the mathematical identity

$$\log_{10}(x) = \log_{10}(e)\ \log_e(x). \tag{6.43}$$

The second term of equation (6.37) is obtained via

$$\frac{\partial m_{\text{Vega}}}{\partial \text{ZP}} = 1 \tag{6.44}$$

6.4.4 Conversion from survey magnitudes to AB magnitudes

For each source, we convert its SDSS and UKIDSS magnitudes to AB magnitudes.

The asinh magnitude of SDSS is intended to be on the AB magnitude system [10]. However, the system zero points could differ from the AB zero points by as much as 10% (Stoughton et al., 2002). One therefore needs to include small corrections in u and z (Doi et al., 2010).

We can convert the magnitude in a given survey to AB magnitudes by adding an offset,

$$m_{AB} = m_{survey} + AB_{\text{offset}} \tag{6.45}$$

For SDSS we have $m_{\text{survey}} = m_{\text{asinh}}$, and for UKIDSS we have $m_{\text{survey}} = m_{\text{Vega}}$. The AB offset AB_{offset} for the different survey filters can be seen in Table 6.9, where the offsets for the Vega magnitudes have been obtained from Table 7 in (Hewett et al., 2006).

Passband	AB offset
u	-0.04
g	0.0
r	0.0
i	0.0
z	0.02
Y	0.634
J	0.938
H	1.379
K	1.900

Table 6.9: AB offsets for SDSS (u, g, r, i, z) and UKIDSS (Y, J, H, K) (Doi et al., 2010; Hewett et al., 2006).

[10] `https://www.sdss.org/dr12/algorithms/fluxcal/#SDSStoAB`

6.4.4.1 Error in the AB magnitude

From equation (6.45) we obtain the error in the AB magnitude,

$$\sigma^2_{m_{AB}} = \left(\frac{\partial m_{AB}}{\partial m_{\rm survey}}\right)^2 \sigma^2_{m_{\rm survey}} + \left(\frac{\partial m_{AB}}{\partial AB_{\rm offset}}\right)^2 \sigma^2_{AB_{\rm offset}}. \tag{6.46}$$

We use $\sigma_{AB_{\rm offset}} = 0$ and hence, with $\left(\frac{\partial m_{AB}}{\partial m_{\rm survey}}\right) = 1$, we obtain the error in the AB magnitude

$$\sigma_{AB} = \sigma_{m_{\rm survey}} \tag{6.47}$$

6.4.4.2 SDSS: nanomaggy to AB

The SDSS unit for flux is given in nanomaggy (nmg). A nanomaggy is 10^{-9} of a maggy. A maggy is a linear measure of flux density. As mentioned above, the SDSS magnitudes are designed to follow the AB magnitude system. Looking at equation (6.30), this means that

$$1 \text{ mgy} = 10^9 \text{ nmgy} = 3631 \text{ Jy} \tag{6.48}$$

Thus, 10^9 nmgy correspond to an AB magnitude of about 0. This is the reference zero-point flux $F_0 = 10^9$ nmgy, which gives

$$m_{\rm AB} = -2.5 \log_{10}\left(\frac{f}{10^9 \text{ nmgy}}\right). \tag{6.49}$$

Via equation (6.22) we calculate the zero-point magnitude to be $ZP = -22.5$. We then arrive with equation (6.23) at

$$m_{\rm AB} = -2.5 \log_{10}\left(\frac{f}{1 \text{ nmgy}}\right) + 22.5, \tag{6.50}$$

where f the flux in units of nanomaggies.

6.5 Flux calculations

We explain how we calculate the flux for a source in a postage stamp in SDSS and UKIDSS. We start from the postage stamp, calculate the raw ADU flux, and convert this flux to standard flux units. The standard flux units we use throughout this project are Janskies (Jy).

6.5.1 The raw ADU flux

The flux contained within a radius r is given by

$$F_r = F \int_0^r 2\pi r' \mathcal{S}(r')\, dr', \tag{6.51}$$

where F the total flux of a source.

The flux error can be obtained via

$$\sigma^2_{F_r} = \left(\frac{\partial F_r}{\partial F}\right)^2 \sigma^2_F \tag{6.52}$$

$$= \left[\int_0^r 2\pi r' \mathcal{S}(r')\, dr'\right]^2 \sigma^2_F \tag{6.53}$$

We use a radius of infinity, $r = \infty$, thus giving us $F_\infty = F$. Therefore, $\sigma_{F_\infty} = \sigma_F$.

Optionally, for UKIDSS we can also use a radius of $r = 1''$ which gives us the flux F_1. This mimics the aperture flux calculation of the same radius used in UKIDSS (`aperFlux3`). The error is obtained from (6.52) to be

$$\sigma_{F_1} = \left[\int_0^{1[\text{arcsec}]} 2\pi r' \mathcal{S}(r')\, dr'\right] \sigma_F \tag{6.54}$$

In this case, we need to include an aperture correction term `aperCor3` to correct for the missing flux.

6.5.1.1 Analytical calculation of the flux

We can obtain the flux F by calculating the log-maximum likelihood value for F. We therefore want to solve

$$\frac{\partial \log P(\mathbf{d}|M_i, \boldsymbol{\theta}_i)}{\partial F} \stackrel{!}{=} 0, \tag{6.55}$$

where F contained in the parameters $\boldsymbol{\theta}_i$ of model M_i. From equation (7.6) it follows that

$$\log P(\mathbf{d}|M_i, \boldsymbol{\theta}_i) \propto -\frac{1}{2} \sum_{i^2+j^2 \le r^2} \left(\frac{d_{ij} - F_{ij}(\boldsymbol{\theta}_i)}{\sigma}\right)^2, \tag{6.56}$$

where the product of the Gaussians has been converted to a sum in log-space.

Next, we will try to bring the log-likelihood to a polynomial form of

$$\log P(\mathbf{d}|M_i, \boldsymbol{\theta}_i) = a\, F^2 + b\, F + c \tag{6.57}$$

where a, b, c are called *sufficient statistic*. These sufficient statistics can be considered as weighted

sums of the data, with increasing weight towards the source centre. Differentiating (6.57) with respect to F and setting the result to zero to find the maximum likelihood leads to

$$\frac{\partial \log P(\mathbf{d}|M_i, \boldsymbol{\theta}_i)}{\partial F} = 2\,a\,F + b \stackrel{!}{=} 0 \tag{6.58}$$

We obtain the maximum likelihood of flux F to be

$$F_{ML} = -\frac{b}{2\,a}. \tag{6.59}$$

Re-arranging the log-likelihood to the form in (6.57) gives

$$\log P(\mathbf{d}|M_i, \boldsymbol{\theta}_i) = c_1 - \frac{1}{2}\sum_{ij}\frac{\left(F_{ij} - d_{ij}\right)^2}{\sigma^2} \tag{6.60}$$

$$= c_1 - \frac{1}{2}\sum_{ij}\frac{\left[\left(\mu_0\theta_{\rm px}^2 + F\,S(i,j)\,\theta_{\rm px}^2\right) - d_{ij}\right]^2}{\sigma^2} \tag{6.61}$$

$$\begin{aligned} &= c_1 - \frac{1}{2\sigma^2}\sum_{ij}\Big[\mu_0^2\,\theta_{\rm px}^4 + F^2\,S(i,j)^2\,\theta_{\rm px}^4 + 2\mu_0 F\,S(i,j)\,\theta_{\rm px}^4 + \\ &\quad d_{ij}^2 - 2\mu_0\,\theta_{\rm px}^2\,d_{ij} - 2F\,S(i,j)\,\theta_{\rm px}^2\,d_{ij}\Big] \\ &= c_1 - \frac{1}{2\sigma^2}\sum_{ij}\Big\{F^2\left[S(i,j)^2\,\theta_{\rm px}^4\right] - 2F\left[S(i,j)\,\theta_{\rm px}^2\,d_{ij} - \mu_0\,S(i,j)\,\theta_{\rm px}^4\right] + \\ &\quad \left(\mu_0^2\theta_{\rm px}^4 + d_{ij}^2 - 2\mu_0\,\theta_{\rm px}^2\,d_{ij}\right)\Big\} \\ &= c_1 - \frac{1}{2\sigma^2}\left\{F^2\left[\sum_{ij}S(i,j)^2\,\theta_{\rm px}^4\right] - 2F\left[\sum_{ij}S(i,j)\,\theta_{\rm px}^2\,(d_{ij} - \mu_0\,\theta_{\rm px}^2)\right] + \right. \\ &\quad \left.\left[\sum_{ij}\left(d_{ij} - \mu_0\,\theta_{\rm px}^2\right)^2\right]\right\} \\ &= F^2\left[-\frac{1}{2\sigma^2}\sum_{ij}S(i,j)^2\,\theta_{\rm px}^4\right] + F\left[\frac{1}{\sigma^2}\sum_{ij}S(i,j)\,\theta_{\rm px}^2\,(d_{ij} - \mu_0\,\theta_{\rm px}^2)\right] + \\ &\quad \left[c_1 - \frac{1}{2\sigma^2}\sum_{ij}\left(d_{ij} - \mu_0\,\theta_{\rm px}^2\right)^2\right]. \end{aligned} \tag{6.62}$$

Thus, a, b and c are retrieved to be

$$a = -\frac{1}{2\sigma^2}\sum_{ij}S(i,j)^2\,\theta_{\rm px}^4 \tag{6.63}$$

$$b = \frac{1}{\sigma^2}\sum_{ij}S(i,j)\,\theta_{\rm px}^2\,(d_{ij} - \mu_0\,\theta_{\rm px}^2) \tag{6.64}$$

$$c = c_1 - \frac{1}{2\sigma^2}\sum_{ij}\left(d_{ij} - \mu_0\,\theta_{\rm px}^2\right)^2 \tag{6.65}$$

which can be plugged into equation (6.59) to obtain the maximum likelihood value of F,

$$F_{ML} = -\frac{\frac{1}{\sigma^2}\sum_{ij} S(i,j)\,\theta_{\rm px}^2\,(d_{ij}-\mu_0\,\theta_{\rm px}^2)}{-\frac{1}{\sigma^2}\sum_{ij} S(i,j)^2\,\theta_{\rm px}^4} \tag{6.66}$$

$$= \frac{\sum_{ij} S(i,j)\,\theta_{\rm px}^2\,(d_{ij}-\mu_0\,\theta_{\rm px}^2)}{\sum_{ij} S(i,j)^2\,\theta_{\rm px}^4} \tag{6.67}$$

6.5.1.2 Analytical calculation of the flux error

Starting from equation (6.62), we can rewrite the product of univariate Gaussians as a single Gaussian to obtain the error in the flux,

$$\log P(\mathbf{d}|M_i,\boldsymbol{\theta}_i) \overset{\left[\frac{\frac{1}{\sum_{ij} S(i,j)^2\,\theta_{\rm px}^4}}{\frac{1}{\sum_{ij} S(i,j)^2\,\theta_{\rm px}^4}}\right]}{=} c_2 - \frac{1}{2}\frac{\left\{F^2 - 2F\left[\frac{\sum_{ij} S(i,j)\,\theta_{\rm px}^2\,(d_{ij}-\mu_0\,\theta_{\rm px}^2)}{\sum_{ij} S(i,j)^2\,\theta_{\rm px}^4}\right] + \left[\frac{\sum_{ij}(d_{ij}-\mu_0\,\theta_{\rm px}^2)^2}{\sum_{ij} S(i,j)^2\,\theta_{\rm px}^4}\right]\right\}}{\left(\frac{\sigma^2}{\sum_{ij} S(i,j)^2\,\theta_{\rm px}^4}\right)} \tag{6.68}$$

$$= c_3 - \frac{1}{2}\frac{(F-F_{\rm ML})^2}{\sigma_F^2} \tag{6.69}$$

where we have used $S(i,j) := S\left[\sqrt{(x_i-x_s)^2+(y_j-y_s)^2}\right]$ for readability, and c_1, c_2, c_3 are constants. The variance in F is given by

$$\sigma_F^2 = \frac{1}{\sum_{ij} 1/(\frac{\sigma}{S(i,j)})^2\,\theta_{\rm px}^4} = \frac{\sigma^2}{\sum_{ij} S(i,j)^2\,\theta_{\rm px}^4}, \tag{6.70}$$

with σ being the variance of the background flux.

6.5.2 Conversion from survey flux units to Jansky via AB magnitudes

We convert the flux of a given source from the units of the corresponding survey to Jansky. We make use of the converted AB magnitudes within that process. The database flux units in SDSS are given by nanomaggies (nmgy), and in UKIDSS by Analog-to-digital unit (ADU) values.

6.5.2.1 SDSS: nanomaggy to Jansky

Since 1 nmgy = 3631×10^{-9} Jy, we get the conversions

$$F_{\rm Jy} = \frac{3631\times10^{-9}\ {\rm Jy}}{1\ {\rm nmgy}}\,F_{\rm nmgy} \tag{6.71}$$

$$\sigma_{F_{\rm Jy}} = \frac{3631\times10^{-9}\ {\rm Jy}}{1\ {\rm nmgy}}\,\sigma_{F_{\rm nmgy}} \tag{6.72}$$

6.5.2.2 UKIDSS: ADU to Jansky

We find a conversion that goes directly from ADU flux units to flux in Janskies.

In order to arrive at the converted flux in Jansky, we use equation (6.45). Plugging in equations (6.35) and (6.30) then gives

$$-2.5\ \log_{10}\left(\frac{F_{\mathrm{Jy}}}{3631\mathrm{Jy}}\right) = \left(ZP - 2.5\ \log_{10}\frac{F_{\mathrm{ADU}}}{t} - \mathrm{aperCor} - \mathrm{skyCorr} - \mathrm{extCorr}\right) + \mathrm{AB}_{\mathrm{offset}} \tag{6.73}$$

$$\Rightarrow \left(\frac{F_{\mathrm{Jy}}}{3631\mathrm{Jy}}\right)^{-2.5} = 10^{ZP-\mathrm{aperCor}-\mathrm{skyCorr}-\mathrm{extCorr}+\mathrm{AB}_{\mathrm{offset}}} \left(\frac{F_{\mathrm{ADU}}}{t}\right)^{-2.5} \tag{6.74}$$

$$\Rightarrow F_{\mathrm{Jy}} = \frac{F_{\mathrm{ADU}}}{t}\, 3631\mathrm{Jy} \times 10^{-\frac{2}{5}\left(ZP-\mathrm{aperCor}-\mathrm{skyCorr}-\mathrm{extCorr}+\mathrm{AB}_{\mathrm{offset}}\right)} \tag{6.75}$$

The last equation gives us the conversion between flux in ADU units to flux in units of Janskies.

Error in flux We take our flux conversion from equation (6.75) and calculate the error in flux via the following

$$\sigma^2_{F_{\mathrm{Jy}}} = \left(\frac{\partial F_{\mathrm{Jy}}}{\partial F_{\mathrm{ADU}}}\right)^2 \sigma^2_{F_{\mathrm{ADU}}} + \left(\frac{\partial F_{\mathrm{Jy}}}{\partial \mathrm{ZP}}\right)^2 \sigma^2_{\mathrm{ZP}}, \tag{6.76}$$

where $\sigma_{F_{\mathrm{ADU}}}$ the error in ADU flux and σ_{ZP} the error in the zero point magnitude. σ_{ZP} is obtained from the FITS header with the keyword `MAGZRR`, which is equivalent to `photZPErrCat` of the database table `MultiframeDetector`. We take the error in the other parameters of equation (6.75) to be zero. We obtain the conversion from flux error in ADU units to flux error in Janskies,

$$\begin{aligned}\sigma_{F_{\mathrm{Jy}}} = \Bigg\{\left[\frac{1}{t}\, 10^{-\frac{2}{5}\left(ZP-\mathrm{aperCor}-\mathrm{skyCorr}-\mathrm{extCorr}+\mathrm{AB}_{\mathrm{offset}}\right)} 3631\mathrm{Jy}\right]^2 \sigma^2_{F_{\mathrm{ADU}}} + \\ \left[-\ln(10)\,\frac{2}{5}\,\frac{F_{\mathrm{ADU}}}{t}\, 10^{-\frac{2}{5}\left(ZP-\mathrm{aperCor}-\mathrm{skyCorr}-\mathrm{extCorr}+\mathrm{AB}_{\mathrm{offset}}\right)} 3631\mathrm{Jy}\right]^2 \sigma^2_{\mathrm{ZP}}\Bigg\}^{\frac{1}{2}},\end{aligned} \tag{6.77}$$

where we have used the following mathematical identity,

$$\frac{\partial a^u}{\partial x} = \ln(a)\, a^u\, \frac{\partial u}{\partial x}. \tag{6.78}$$

Analogously we can obtain the error in ADU flux when converting back from Janskies. Following from equation (6.76), we have for the error in ADU flux,

$$\sigma^2_{F_{\mathrm{ADU}}} = \frac{\sigma^2_{F_{\mathrm{Jy}}} - \left(\frac{\partial F_{\mathrm{Jy}}}{\partial \mathrm{ZP}}\right)^2 \sigma^2_{\mathrm{ZP}}}{\left(\frac{\partial F_{\mathrm{Jy}}}{\partial F_{\mathrm{ADU}}}\right)^2} \tag{6.79}$$

$$= \left(\frac{\partial F_{\mathrm{ADU}}}{\partial F_{\mathrm{Jy}}}\right)^2 \sigma^2_{F_{\mathrm{Jy}}} - \left(\frac{\partial F_{\mathrm{ADU}}}{\partial \mathrm{ZP}}\right)^2 \sigma^2_{\mathrm{ZP}}, \tag{6.80}$$

which gives us

$$\sigma_{F_{\mathrm{ADU}}} = \left\{ \left[\frac{1}{3631\mathrm{Jy}}\, 10^{\frac{2}{5}\left(ZP-\mathrm{aperCor}-\mathrm{skyCorr}-\mathrm{extCorr}+\mathrm{AB}_{\mathrm{offset}}\right)} t \right]^2 \sigma^2_{F_{\mathrm{Jy}}} - \left[\ln(10)\, \frac{2}{5}\, \frac{F_{\mathrm{Jy}}}{3631\mathrm{Jy}}\, 10^{\frac{2}{5}\left(ZP-\mathrm{aperCor}-\mathrm{skyCorr}-\mathrm{extCorr}+\mathrm{AB}_{\mathrm{offset}}\right)} t \right]^2 \sigma^2_{\mathrm{ZP}} \right\}^{\frac{1}{2}}. \tag{6.81}$$

Conversion from Vega magnitude to flux in Janskies We sometimes want to go from Vega magnitudes directly to flux in Janskies. For this we need equation (6.30) and re-arrange it for the flux. We then plug in equation (6.45) for the AB magnitude. This gives us the following conversion from Vega magnitude to flux in Janskies,

$$F_{\mathrm{Jy}} = 3631\mathrm{Jy} \times 10^{-\frac{2}{5}\left(m_{\mathrm{Vega}}+\mathrm{AB}_{\mathrm{offset}}\right)} \tag{6.82}$$

We can calculate the flux error directly from the equation above to obtain

$$\sigma^2_{F_{\mathrm{Jy}}} = \left(\frac{\partial F_{\mathrm{Jy}}}{\partial m_{\mathrm{Vega}}}\right)^2 \sigma^2_{m_{\mathrm{Vega}}} + \left(\frac{\partial F_{\mathrm{Jy}}}{\partial AB_{\mathrm{offset}}}\right)^2 \sigma^2_{AB_{\mathrm{offset}}}. \tag{6.83}$$

If we set $\sigma_{AB_{\mathrm{offset}}}$ equal to zero, we receive the flux error to be

$$\sigma_{F_{\mathrm{Jy}}} = \left(-\frac{5}{2}\, \ln(10)\, F_{\mathrm{Jy}}\right) \sigma_{m_{\mathrm{Vega}}} \tag{6.84}$$

6.6 Summary

We have shown how we process each candidate by starting from its location in the sky and ending at a set of summarising attributes (see Table 6.2) including the flux and magnitude of the source. We have explained how we communicate with the database of each survey to obtain images and database photometry in every band, and how we discard images with data processing issues. We have further shown how we calculate the flux and magnitude of each source, and how we convert them to standardized units. This pipeline can be used for visually inspecting candidates through plotting their postage stamps together with their summary information. Its summarising attributes can further be used for automated high-redshift quasar classification methods such as heuristic cuts or as input features in machine learning.

Chapter 7

Two-metric threshold classification

This chapter builds up on the work of Mortlock et al. (2012) and Euclid Collaboration (2019), where we use Bayesian model comparison to compare models of high-redshift quasars and contaminants based on colour information in the optical and NIR. This gives us a probability that a source is a high-redshift quasar based on photometry alone. In addition to that, we want to test for the morphology and the stationarity of a source, which we cannot do with photometry information alone. This is why we apply a chi-squared test on the source images themselves, where we test if a source has moved and if it looks like a point source. A brief summary is also given in 3.2.2.

Table 7.1: Two metrics that test different quasar properties. Bayesian model comparison as an augmentation of the work in Mortlock et al. (2012) and Euclid Collaboration (2019) compares the input colour information to the colour templates of the different populations and returns a quasar probability $P_{\rm q}$. The residual chi-squared $\tilde{\chi}^2$ method tests for the stationarity and morphology, and that the obtained photometry follows the colours of one of the quasar templates.

Input	Tests quasar properties	**Output**
Photometry	SED colour templates compared to that of contaminants	$P_{\rm q}$ (7.1)
Pixel/Image	Stationarity, point source, SED colour templates	$\tilde{\chi}^2$ (7.9)

We make use of the available raw optical and NIR images in SDSS and UKIDSS in order to receive a better quasar selection. From these images we extract information on the colour, motion and morphology of possible high-redshift quasars. We do not take into account variability or radio-loudness.

In order to do this, we define a model on the postage stamp. Specifically, we model the pixel value of each pixel in the postage stamp. We will refer to this as the 'pixel-level' model. We leverage the following properties in our pixel-level model:

- Objects that are far away appear to be stationary
- Quasars are point-like

- Quasars have a distinct SED

Ideally, we want to develop a Bayesian model comparison approach where we include pixel-level models of quasars and contaminant populations. This would then give us a quasar probability based on the properties of quasars and that of contaminants. However, this is not trivial since it would be difficult to model every population down to the pixels. For example, the light profile and morphology of extended sources are complicated and not always known well (e.g., Anders et al. (2006)). On top of that, calculating the likelihood over each set of model parameters would be computationally expensive.

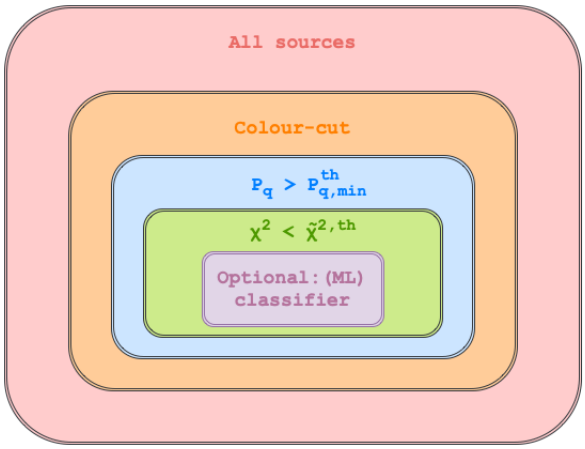


Figure 7.1: An illustration of a sequential process that sifts out contaminants from of a pool of candidates before having to manually inspect the remaining sources. This includes colour cuts, cuts in quasar probability based on e.g. Bayesian model comparison and cuts in residual chi-squared based on a mathematical model of a quasar. Optionally, one can optimise the classification via machine learning techniques.

Instead, we develop two metrics that extract complementary information and approximate the ideal solution above. The two metrics are listed in Table 7.1. By applying cuts on these two metrics, we can classify whether a source is expected to be a high-redshift quasar or not. These cuts are subsequently tuned to give the best possible classification performance.

On the flux level, we make use of methods from (Mortlock et al., 2012) and (Euclid Collaboration, 2019): We include flux models of contaminants and use Bayesian model comparison to find the model that fits the data best. This gives us a probability (P_q) that a source is a quasar. The approach tests against the SED of a source but does not take into account the stationarity or a point-source like appearance.

On the pixel level, we model a stationary point source whose SED follows that of a high-redshift quasar and is tested by its chi-squared ($\tilde{\chi}^2$) goodness of fit on image cutouts. This tests against sources that are extended or move between filters, and sources that have an SED different to that of our quasar templates. If we had modelled contaminants on the pixel level as well, we would additionally have to

model extended sources and moving point sources. Due to this added complexity, we have decided to only model the quasar population at this time.

We can apply thresholds to both of these techniques in order to get rid of contaminants. This is illustrated in Fig. 7.1, which shows a selection process that only considers candidates which have colours similar to quasars, are above a certain quasar probability threshold $P^{\rm th}_{\rm q,min}$ and below a specific reduced average chi-squared value $\tilde{\chi}^{2,\rm th}$ or reduced maximum chi-squared value $\tilde{\chi}^{2,\rm th}_{\rm max}$. Applied to a new dataset, this allows to sift out more contaminants than by colour cuts alone, before having to do any visual inspection. Our method is designed to be redshift and survey independent and thus can be applied to future surveys such as Euclid and LSST.

In Sections 7.1 and 7.2, we describe our metrics to classify high-redshift quasars. This includes our Bayesian model comparison approach from which we obtain a quasar probability, and our quasar model that is tested on image cutouts. Section 6 outlines our data processing pipeline in detail. In Section 7.3, we analyse how well our method works on a set of simulated quasars and contaminants. We then test on real data in Section 7.4. Our classification results and tuned thresholds of our metrics are presented in Section 7.5. The discussion and conclusion can be found in Section 7.6.
Throughout this work we use AB magnitudes (Oke & Gunn, 1983).

7.1 Metric 1: flux-level population modelling

Bayesian model comparison enables the testing of multiple hypotheses by comparing a set of statistical models to observed data (Burnham, 2002; Sivia & Skilling, 2006; Neath et al., 2017). Given a set of population models $\{M_j\}$, data $\mathbf{d}$ and additional background prior information I, the probability that a source is of type M_i given input data $\mathbf{d}$ is

$$P\left(M_i|\mathbf{d}, I\right) = \frac{P\left(\mathbf{d}|M_i, I\right)\ P\left(M_i|I\right)}{\sum_j P\left(\mathbf{d}|M_j, I\right)\ P\left(M_j|I\right)}, \tag{7.1}$$

where $P\left(M_i|I\right)$ is the prior probability that a source belongs to model M_i. Through marginalization we can express $P\left(\mathbf{d}|M_i, I\right)$ as the product of likelihood and prior,

$$P\left(\mathbf{d}|M_i, I\right) = \int P\left(\mathbf{d}|M_i, \boldsymbol{\theta}_i, I\right)\ P\left(\boldsymbol{\theta}_i|M_i, I\right) d\boldsymbol{\theta}_i, \tag{7.2}$$

where $\boldsymbol{\theta}_i$ are the model parameters of model i. In the simplest case we only have one model M_1, for which the posterior then will be of probability 1. Here, we are interested in the models of quasars and contaminants.

Mortlock et al. (2012) implement a Bayesian model comparison approach to rank colour selected candidates by their probability of being a high-redshift quasar. They compare the observed photometry

against SED templates. This includes not just SED models of quasars, but also that of contaminants. As these contaminants are far more numerous than quasars, any source that is not a clear quasar will more likely be a contaminant. They therefore use the luminosity function as a prior.

In Mortlock et al. (2012), the likelihood $P(\mathbf{d}|M_i, \boldsymbol{\theta}_i, I)$ is a product of $|B|$ Gaussians, where $|B|$ the number of passbands. The individual Gaussians are given as $\mathcal{N}(F_b, \sigma^2_{F_b})$ for each passband $b \in B$, with mean being the model flux F_b as a function of parameters $\boldsymbol{\theta}_i$, and standard deviation being the uncertainty in the predicted flux,

$$P(\hat{F}_b|M_i, \boldsymbol{\theta}_i) \propto \exp\left[-\frac{1}{2}\left(\frac{\hat{F}_b - F_b(\boldsymbol{\theta}_i)}{\sigma_{\hat{F}_b}}\right)^2\right], \tag{7.3}$$

where $\hat{F}_b \pm \sigma_{\hat{F}_b}$ the observed flux in band b, M_i the population model (e.g. quasar or contaminant) and $\boldsymbol{\theta}_i$ the parameters of the population model. The likelihood is then given by the product of individual Gaussians over the bands B,

$$P(\mathbf{d} = \hat{\mathbf{F}}|M_i, \boldsymbol{\theta}_i) = \prod_{b \in B} P(\hat{F}_b|M_i, \boldsymbol{\theta}_i). \tag{7.4}$$

The priors $P(M_i|I)$ and $P(\boldsymbol{\theta}_i|M_i, I)$ are combined into a surface density of model population i. The models $\{M_j\}$ included in this are that of quasars (M_1 = quasar) and that of MLT dwarfs (M_2 = MLT). A threshold is then applied to only visually inspect the most probable quasar candidates. By including visual inspection, many candidates whose data does not follow the model assumptions are rejected.

Euclid Collaboration (2019) refine the Bayesian model comparison explained above by adapting the contaminant populations to include both MLT dwarfs and early type galaxies (M_3 = ETG). Previously, zero flux was assumed in the optical u, g, r bands.

In this thesis, we develop this technique further by including optical SDSS u, g, r bands in the 9 quasar SED templates. This helps to additionally identify quasars at $5.2 \lesssim z \lesssim 6.5$. This is implemented as the majority of visually accepted candidates in our dataset lie within this range, and would otherwise lead to large residual chi-squared values and low probabilities. In total, this gives quasar templates in the redshift range $5.2 \leq z \leq 12$. However, it should be noted that in the quasar models the flux blueward of Lyα is set to zero due to the Gunn-Peterson trough. However, at $5.2 \lesssim z \lesssim 6.5$ one will see a Lyman-alpha forest in the observed spectrum instead. Therefore, the SED models at these lower redshifts are not accurate enough. In future, one could attempt to incorporate the IGM transmission for a given population of absorbers as a function of redshift for the quasar SED templates.

The quasar templates are taken from Hewett et al. (2006) and encompass SEDs with a standard equivalent line width of $EW(C_{IV}) = 39.1$ Å, half line widths and double line widths. Furthermore, they include a standard UV continuum slope ($f_{1315}/f_{2225} = 1.0$), a blue ($f_{1315}/f_{2225} = 1.16$) and a red continuum ($f_{1315}/f_{2225} = 0.84$), where f_{1315}/f_{2225} the flux ratio between the rest frame wavelength at 1315

Å and at 2225 Å. Combining these line width and continuum options with each other gives 9 distinct quasar SED templates.

This is implemented in the Python package `Pq_server` [1] (Euclid Collaboration, 2019). Using `Pq_server`, we extract the probability that the source is a quasar $P_{\rm q}$. Moreover, we obtain the best fit quasar SED template with the fluxes adjusted to the best fit redshift of the candidate. Note that `Pq_server` works best for $z \gtrsim 6.5$ as that is the redshift where the contaminant populations (dwarf stars M0 – T8 and early type galaxies) become relevant. Thus, the probabilities below that redshift are less reliable as they might be higher than they should be.

Note that this approach only tests the SED of a source, but not any point source-like appearance nor any stationarity. Rather, it makes the assumption that each source is a point source and does not move between bands. The flux $\hat{\mathbf{F}}$ is obtained from the database PSF photometry of each survey. Since the database flux calculation adjusts for any movement between bands, we cannot spot any moving sources with the flux information alone. Further, it is possible that the calculated PSF flux of a non point source-like object follows the SED of a high-redshift quasar, since the true flux is different to the PSF flux in that case. In order to test these assumptions, we have to go to the pixel level.

7.2 Metric 2: pixel-level goodness of fit

We model postage stamp images of size $l \times l$. The flux in the pixels of the background are assumed to be Gaussian distributed with mean μ_0 and variance σ^2. The noise model of the ADU value in pixel (i, j) is defined as

$$d_{ij} = F_{ij} + n_{ij}, \tag{7.5}$$

where $\mathbf{d}$ the i.i.d. ADU values of the pixels of the postage stamp of size $N \times N$, F_{ij} the flux in pixel (i, j) and $n_{ij} \sim \mathcal{N}(0, \sigma^2)$ the additive Gaussian noise.

We assume the flux uncertainty in the pixel elements $\{d_{ij,b}\}$ to be Gaussian distributed $\mathcal{N}(F_{ij,b}, \sigma_b^2)$. We neglect Poisson effects as we assume the uncertainty to be background dominated. The individual likelihood for band $b \in B$ for a model M_i with model parameters $\boldsymbol{\theta}_{i,b}$ is then given by

$$P(\mathbf{d}_b | M_i, \boldsymbol{\theta}_{i,b}) \propto \exp\left[-\frac{1}{2} \sum_{i^2+j^2 \leq r^2} \left(\frac{d_{ij,b} - F_{ij,b}(\boldsymbol{\theta}_{i,b})}{\sigma_b}\right)^2\right]. \tag{7.6}$$

The full likelihood is then again given by the product of $P(\mathbf{d}_b | M_i, \boldsymbol{\theta}_{i,b})$ over bands $b \in B$.

[1] `https://github.com/rhysrb/Pq_server`

7.2.1 Chi-squared

Our second metric is a test of goodness of fit. Since the likelihood is Gaussian, we can use the method of chi-squared, where equation (7.6) can be rewritten as

$$P(\mathbf{d}_b|M_i, \boldsymbol{\theta}_{i,b}) \propto \exp\left(-\frac{\chi_b^2}{2}\right). \tag{7.7}$$

Maximising the likelihood is equivalent to minimising the chi-squared value χ_b^2. Using equation (7.7), the goodness of fit in passband $b \in B$ is tested via reduced chi-squared fitting where we divide χ_b^2 by the number of degrees of freedom,

$$\tilde{\chi}_b^2 = \frac{1}{N_{\text{dof}}} \sum_{i^2+j^2 \leq R_{\tilde{\chi}^2}^2} \left(\frac{d_{ij,b} - F_{ij,b}}{\sigma_b}\right)^2, \tag{7.8}$$

where $R_{\tilde{\chi}^2}$ the radius and $N_{\text{dof}} \sim \pi R_{\tilde{\chi}^2}^2$ the number of degrees of freedom.

Additionally, we want to create a more redshift and survey independent feature, and thus we take the average over all passbands,

$$\tilde{\chi}^2 = \frac{1}{|B|} \sum_{b \in B} \tilde{\chi}_b^2. \tag{7.9}$$

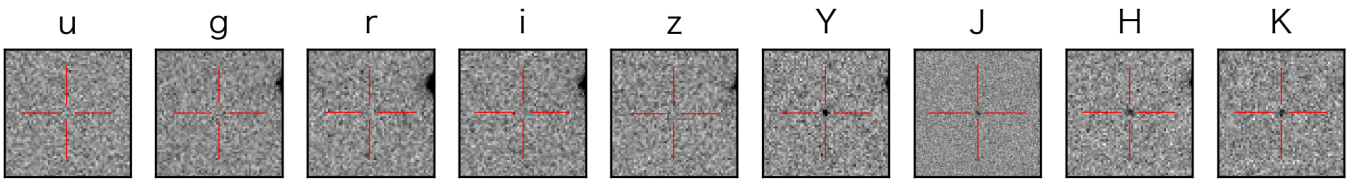
This gives the reduced average chi-squared value for a given source. Since our data $\mathbf{d}$ are the pixel values, $\tilde{\chi}^2$ is tested on the pixel level of the image cutouts.

As mentioned earlier, doing a Bayesian model comparison on the pixel level would be far too complex. We can see that the integral in (7.2) over model parameters $\boldsymbol{\theta}_{i,b}$ using our pixel likelihood $P(\mathbf{d}_b|M_i, \boldsymbol{\theta}_{i,b})$ would be very computationally expensive. Instead, we use chi-squared statistics from above to test how well our pixel-level model of a high-redshift quasar fits the observed pixel data.

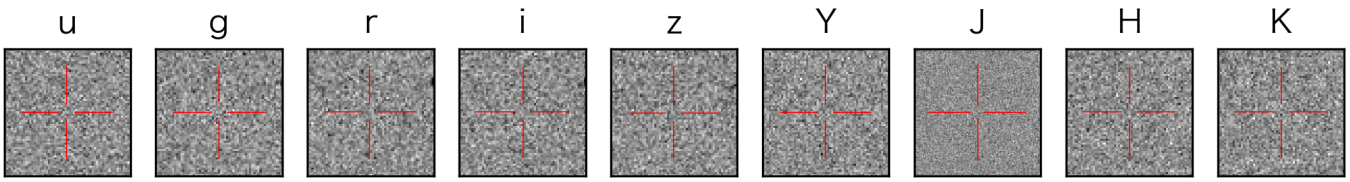
7.2.2 Quasar model

We compare the pixel data $\{d_{ij}\}$ of an image to the predicted image data $\{F_{ij}\}$ of a quasar at redshift $\hat{z}$. The residual chi-squared $\tilde{\chi}^2$ (7.9) then gives us an estimate of how well our predictions fit the data.

The pixel flux F_{ij} in a quasar postage stamp comes from a point source that is located in the centre of the image and is stationary between passbands. The point source is modelled via a point spread function that is specific to each survey. The PSFs that we use for each survey to calculate the fluxes are explained in Section 6.1.5.1.



(a) Original



(b) Residual using our calculated PSF flux

Figure 7.2: Quasar ULAS J1120+0641 (Mortlock et al., 2011) at redshift $z = 7.09$. a) Original image data in SDSS and UKIDSS LAS, b) Residual images using our calculated PSF fluxes. The residual images consist of noise only, which means that our model is a great fit to this quasar. Indeed, the chi-squared values of $\tilde{\chi}^2 = 0.94$ and $\tilde{\chi}^2_{\rm max} = 1.28$ confirm this. The size of each plotted postage stamp is $24\farcs4 \times 24\farcs4$.

The pixel flux F_{ij} is given by

$$F_{ij} = \mu_0\theta^2_{\rm px} + F \int_{x_i-\frac{\theta_{\rm px}}{2}}^{x_i+\frac{\theta_{\rm px}}{2}} {\rm d}x' \int_{y_j-\frac{\theta_{\rm px}}{2}}^{y_j+\frac{\theta_{\rm px}}{2}} {\rm d}y'\, S\left[\sqrt{(x'-x_s)^2+(y'-y_s)^2}\right] \tag{7.10}$$

$$\simeq \mu_0\theta^2_{\rm px} + F\, S\left[\sqrt{(x_i-x_s)^2+(y_j-y_s)^2}\right]\theta^2_{\rm px}, \tag{7.11}$$

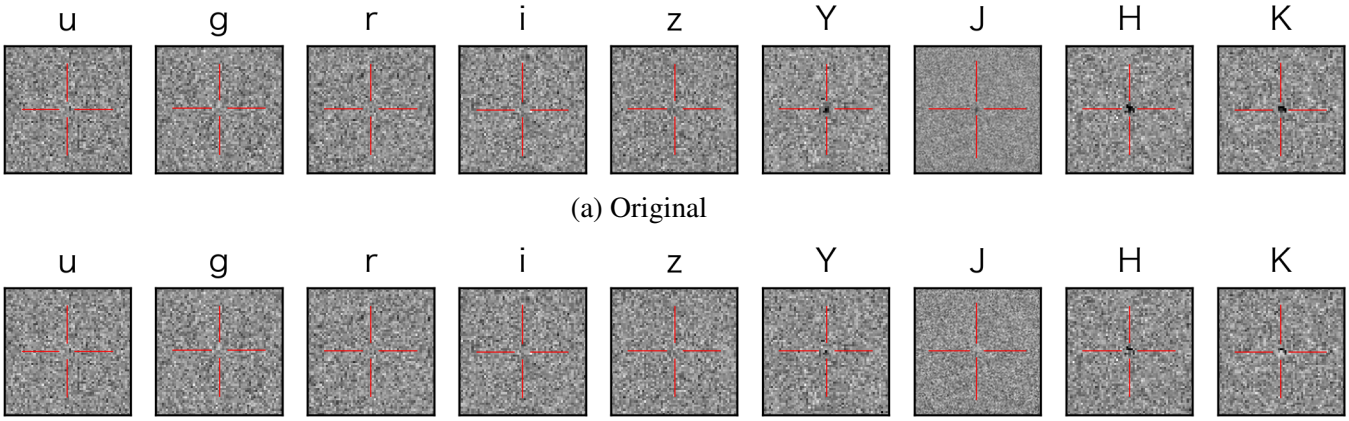
where F the flux from the quasar SED template that best fits the observed flux, $\theta^2_{\rm px}$ the area of a pixel which is specific to each survey, S the point spread function and $(x_s, y_s) \in \Re^2$ the centre of the source within the image. Here, we have approximated the result by assuming that the surface brightness is constant over the pixel area.

The likelihood is given by (7.6), using our quasar model M_1 and quasar model parameters $\boldsymbol{\theta}_{1,b} = \{\mathbf{F}_b, (x_s, y_s), \theta_{px,b}, S_b, \mu_{0,b}, \sigma_b\}$.

7.3 Simulations

We test our pipeline by simulating a diverse set of sources: quasars; MLT dwarfs; and ETGs. The noise level per pixel in each passband approximately corresponds to the average noise level over the sky region of our dataset. The same applies to the parameters of the point spread functions. Specifically, we simulate the following:

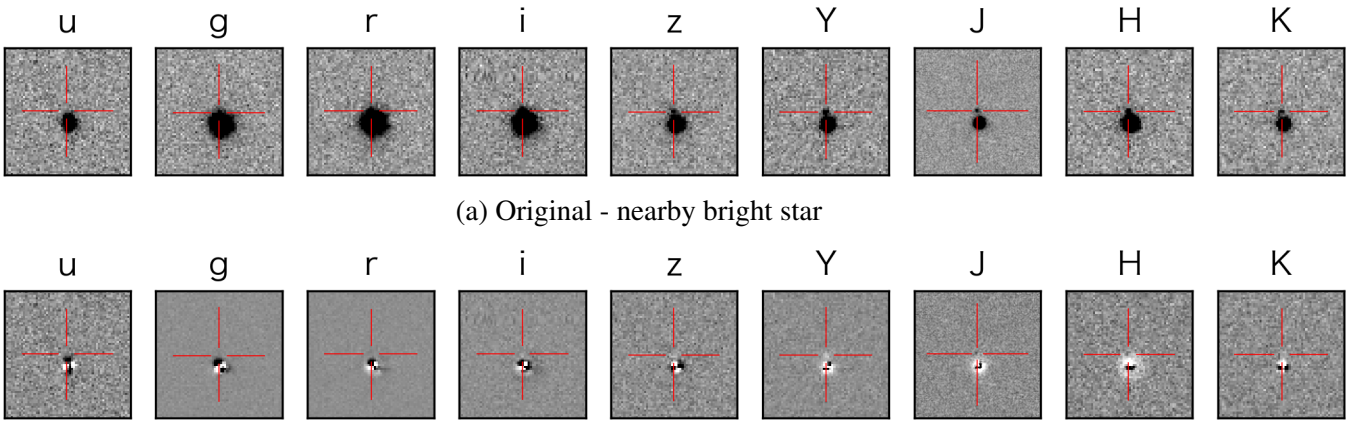
1. Quasars:
    - Quasar



(a) Original

(b) Residual using the best fit quasar SED

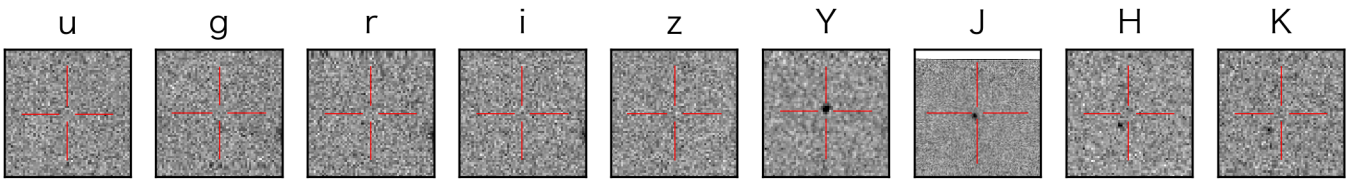
Figure 7.3: Candidate J000042.15+072229.9 has a probability of being an ETG of $P_{\mathrm{ETG}} = 1.00$. We therefore expect the residual to have remaining positive/negative flux and thus the source to have a large average chi-squared value. This is confirmed in the residual in b). The largest chi-squared value of $\tilde{\chi}^2_{\mathrm{max}} = 6.54$ can be found in H, due to the observed flux being incompatible with the best fit quasar template, and due to additional cross-talk issues.



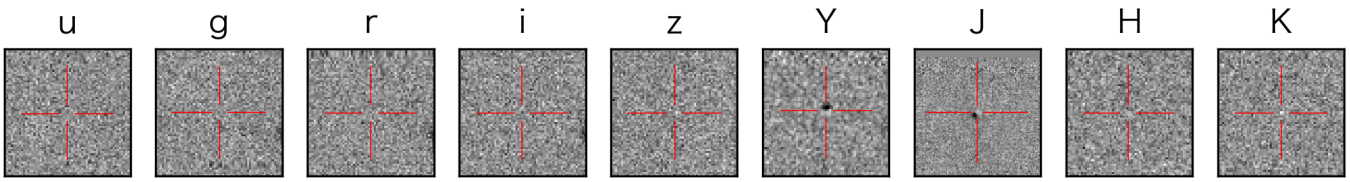
(a) Original - nearby bright star

(b) Residual using the best-fit quasar SED.

Figure 7.4: A bright star in the vicinity of the source J000226.75−005604.5. a) Original image data in SDSS and UKIDSS LAS, b) Residual using the best-fit quasar SED. As the residual does not consist of noisy images only, this gives a low quasar probability $P_{\mathrm{q}} = 0.00$ and a high residual chi-squared value $\tilde{\chi}^2 = 23.86$ as expected.



(a) Original - a moving source



(b) Residual using the best-fit quasar SED.

Figure 7.5: A moving source J002318.32+000435.8. a) Original image data in SDSS and UKIDSS LAS, b) Residual using the best-fit quasar SED. The model picks up that the source is moving and hence the pipeline obtains values of $\tilde{\chi}^2 = 2.72$ and $P_{\rm q} = 0.00$.

- Quasar with a bright nearby source

2. MLT dwarfs:
   - Stationary dwarf
   - Non-stationary dwarf
3. Early-type galaxies (ETGs):
   - Stationary ETG

The first set of simulations tests if our method correctly gives high probabilities and low $\tilde{\chi}^2$ values to simulated quasars. By varying the flux levels we can detect at what point our method breaks down. The second set of simulation tests if our method can detect a moving source, e.g. a local star, by obtaining large values in $\tilde{\chi}^2_b$ in at least one passband b. Lastly, simulating ETGs as extended sources should again give low quasar probabilities and large $\tilde{\chi}^2$ values.

We perform all tests at J-band fluxes between 10 μJy and 100 μJy. The flux in the other bands is then calculated from the SED templates with J as the reference band. We use these flux levels since the majority of both visually accepted and rejected candidates lie within that range, as shown in Fig. 7.6. The maximum flux in J for the accepted candidates is $F_J = 128$ μJy. The median flux for the visually accepted candidates lies at $F^{\rm accepted}_{J,\rm median} = 36.4$ μJy, while the median flux of the visually rejected candidates is $F^{\rm rejected}_{J,\rm median} = 50.2$ μJy. This makes sense as most of the J-band brightest sources are visually rejected.

The minimum flux in J of the visually accepted candidates lies at $F^{\rm accepted}_{J,\rm min} = 12$ μJy. Using a minimum of 10 μJy in our simulations therefore makes sense. Generally, having a set of simulated faint sources helps us to analyse how the performance decreases when going to lower and lower SNRs.

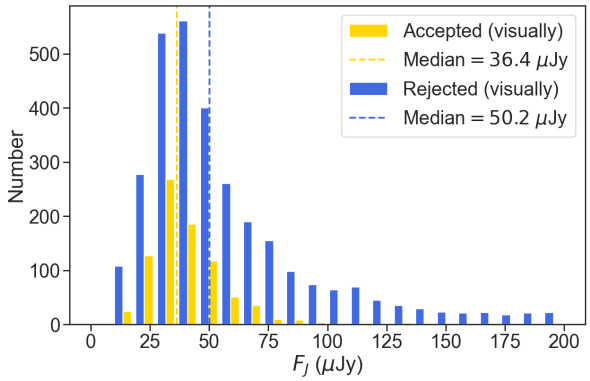


Figure 7.6: Distribution of flux in the J-band of our validation set (see Sec. 7.4.2). The orange histogram corresponds to the visually accepted candidates, and the blue histogram to the visually rejected ones.

We expect the difficulty to distinguish quasars from contaminants to lie in low SNRs for the residual chi-squared calculation, whereas the quasar probability will depend more on how realistic a certain flux is at a given redshift.

Table 7.2: Simulation results of simulated quasars, MLT dwarfs and ETGs. We show the $(\tilde{\chi}^2, P_\mathrm{q})$ pair for both the minimum $\tilde{\chi}^2$ and the maximum P_q. The minimum $\tilde{\chi}^2$ pair represents the source out of each model with the smallest $\tilde{\chi}^2$ value, together with its corresponding P_q value. Similarly, the minimum P_q pair represents the source out of each model with the smallest P_q value, together with its corresponding $\tilde{\chi}^2$ value.

Simulation		$\min\{\tilde{\chi}^2\}$ pair		$\max\{P_\mathrm{q}\}$ pair	
		$\tilde{\chi}^2$	P_q	$\tilde{\chi}^2$	P_q
Quasars	Model quasar	0.7	1.0	0.9	1.0
	Nearby source	0.8	1.5×10^{-3}	1.0	1.0
MLT dwarfs	Stationary	0.8	1.3×10^{-14}	1.1	2.9×10^{-3}
	Non-stationary	0.8	8.8×10^{-4}	1.0	1.0
ETGs	Model ETG	0.8	4.3×10^{-5}	1.0	6.2×10^{-5}

The results are summarised in Table 7.2. This shows the best $\{\tilde{\chi}^2, P_\mathrm{q}\}$ pairs for each model population. Namely, it shows the smallest $\tilde{\chi}^2$ value for each population together with their respective quasar probability, plus it shows the maximum quasar probability P_q together with its corresponding chi-squared value. That means that all other simulated model sources are worse or equal to these two values, where worse means a smaller quasar probability compared to $\max\{P_\mathrm{q}\}$, and a larger chi-squared value compared to $\min\{\tilde{\chi}^2\}$. We can see that whenever $\tilde{\chi}^2$ is small for the simulated contaminants, the quasar probability P_q is close to zero. Vice versa, when P_q is large then $\tilde{\chi}^2$ is large too. Thus, even the most quasar-like points of our contaminant simulations are a very bad fit to our quasar model. The exception are simulated non-stationary MLT dwarfs, where both P_q and $\tilde{\chi}^2$ can have perfectly acceptable

quasar values when they have moved too far away from the centre. In that case, it is as if there was zero flux in the SDSS bands, which then looks similar to a Gunn-Peterson trough and hence gives a high quasar probability. Realistically, we expect surveys to be close enough in time that the source would not move outside $R_{\tilde{\chi}^2}$ and hence would still be detected by our mechanism.

7.3.1 Quasars

Simulating quasars whose SED follows that of our 9 templates at different redshifts and with varying flux gives us a clue to how well our method solves our most important task - to detect quasars. Including bright nearby sources will make this more difficult, and we expect some sources with low probabilities and high chi-squared values in that case. Simulating bright nearby sources also allows us to check how it affects the background pixel flux estimate and the overall source flux estimate.

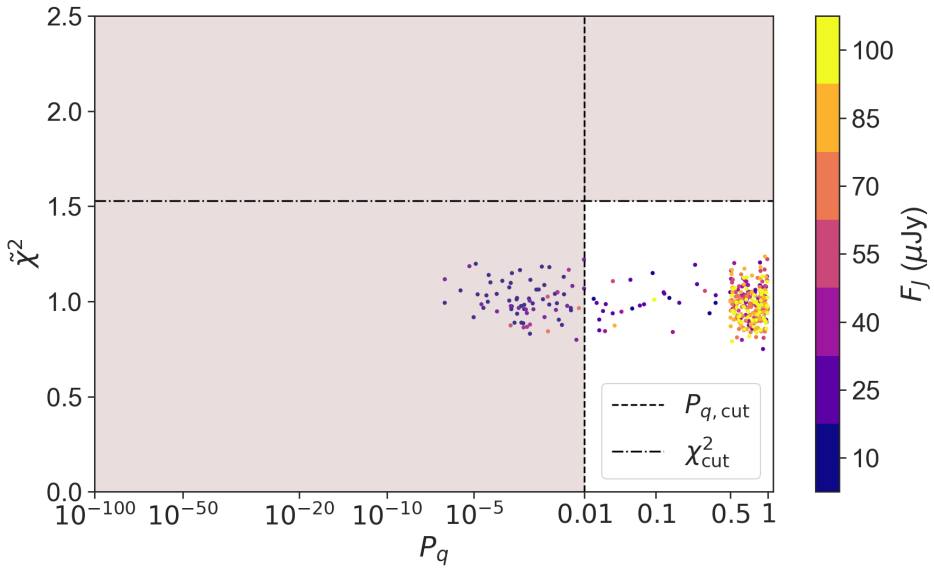


Figure 7.7: A set of 378 simulated quasars with varying J-band flux between 10 μJy and 100 μJy. These are simulated for all 9 templates and for redshifts between 5.2 and 8.5. For plotting purposes, we take a uniform distribution in P_{q} between 0.5 and 1.0 for sources with $P_{\mathrm{q}} \geq 0.5$.

The redshift range we use is between $z = 5.2$ and $z = 8.0$. Further, we make use of all 9 SED templates, representing different continua and line emissions.

The results for simulated quasars are shown in Figure 7.7. As expected, the residual chi-squared values are close to 1 or below. If $\tilde{\chi}^2 < 1$, then the uncertainties are smaller than expected by our model. While $\tilde{\chi}^2$ stays around 1 for most J-band flux levels, the probability starts to decrease for faint sources with $F_J \lesssim 25\,\mu$Jy. This makes sense, as the smaller the signal-to-noise ratio, the more difficult it gets to extract the source signal. Therefore, we might miss a small portion of faint quasars. Keep in mind that we expect strong emission in J for quasars at redshifts $z \gtrsim 6.5$. E.g. the majority of sources that have a quasar probability below $P_{\mathrm{q}} \sim 10^{-3}$ are at simulated redshifts $z \geq 7.5$, while having a faint

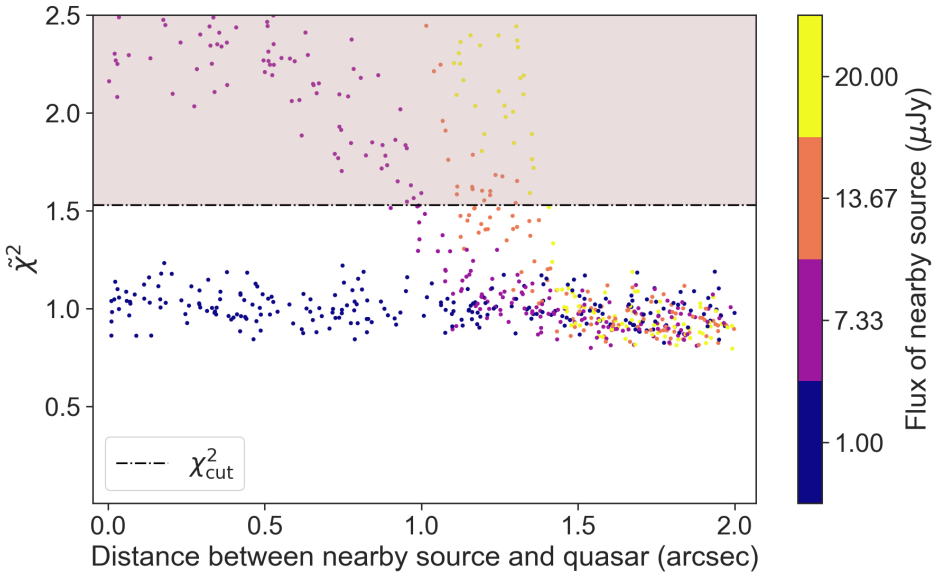


Figure 7.8: A set of 950 simulated quasars with a nearby source of varying brightness and distance to the central quasar. The nearby source is modelled to have constant flux across all bands, which is simulated for fluxes $F = 1, 7.3, 13.7, 20\,\mu$Jy.

J-band flux. In this redshift range, a low flux in J can result in negative fluxes in the optical bands. Hence, a low probability is preferred in those cases.

The results for simulated quasars with a nearby source can be seen in Fig. 7.8. Again, the quasar's flux is varied, using the J band flux to calculate the flux in the other bands from our templates. The nearby source is of constant flux over all passbands, in the range $1\,\mu\text{Jy} < F_J < 20\,\mu\text{Jy}$. We do not include sources brighter than $20\,\mu$Jy as these have even larger reduced chi-squared values and therefore are a clear rejection. The distance between the source and the central quasar is up to 2 arcseconds. As expected, a constant flux of only $1\,\mu$Jy gives a low $\tilde{\chi}^2$ independent of where the source is located. This is because the flux is within the uncertainty of the background pixel flux. At $F = 7.3\,\mu$Jy, we already see $\tilde{\chi}^2 > 2$, which starts to go down at around a distance of 0.75 arcseconds. Generally, the reduced chi-squared values decrease with increasing distance of the source from the central quasar. This makes sense, as the further the nearby source moves away from the central source, the less impact it has on the central flux and chi-squared calculation, and thus the smaller $\tilde{\chi}^2$ gets. As the sources moves outside $R_{\tilde{\chi}^2} = 1\farcs2$, the radius for the $\tilde{\chi}^2$ and flux calculation, it has less of an impact on the central source which means that the central flux can be recovered. It gets to values of around $\tilde{\chi}^2 \sim 1$ at a distance of about 1.5 arcseconds for the source fluxes we considered.

Note that in our pipeline we only reconstruct sources that are at least R_{keep} away from the central source. This is exactly such that we can capture moving sources, while at the same time not letting actual nearby objects get in the way of the central source calculation.

7.3.2 MLT dwarfs

Next, we simulate MLT dwarfs. We do this by generating point sources with colours of that of MLT dwarfs. Since the images of SDSS and UKIDSS have been released several years apart, a local star is located at a different location in each survey. We therefore need to simulate motion of MLT dwarfs. In our case, we use the overall source locations as reported in the database of UKIDSS LAS DR10PLUS. For this reason we only simulate offset locations in the SDSS bands. Had we taken the source locations from SDSS, we would have simulated the offsets in UKIDSS instead.

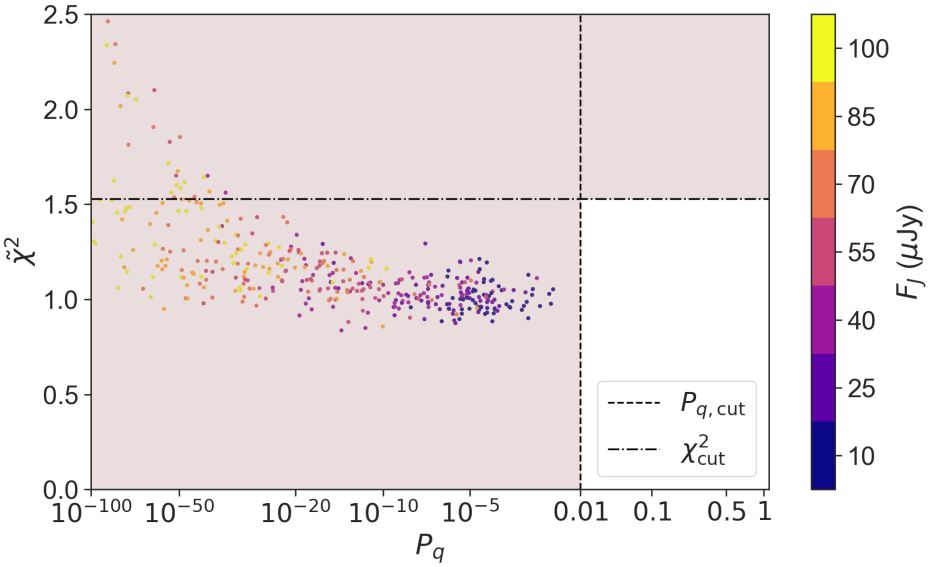
We test if our $\tilde{\chi}^2$ method can reveal sources with apparent motion and flag up such candidates. We use all available MLT dwarf types in `Pq_server` (Euclid Collaboration, 2019), from M star M0 up to T dwarf T8.

Fig. 7.9 shows the simulated stationary MLT dwarfs as a function of flux in J and spectral type respectively. All simulated sources have a quasar probability below 10^{-3}, with the majority having a quasar probability of zero. Some sources have low $\tilde{\chi}^2$ values, particularly the ones with faint J-band photometry of $F_J = 10\,\mu$Jy. Again, this is expected as at such low flux levels a tiny difference in the predicted quasar SED will not account for a large difference in $\tilde{\chi}^2$, especially if the source flux is within the uncertainty of the background pixel flux. In terms of spectral types, T and L type dwarfs have lower residual chi-squared values, while M dwarfs give the worst fit. The plots have been cropped to be standardised, but the residual chi-squared values for MLT dwarfs continue to grow and go up to $\tilde{\chi}^2 \approx 20$. However, no sources simultaneously have $\tilde{\chi}^2 \sim 1$ and $P_\mathrm{q} > 10^{-3}$. Hence, the combination of P_q and $\tilde{\chi}^2$ can easily distinguish between quasars and MLT dwarfs when the sources follow our MLT SED templates.

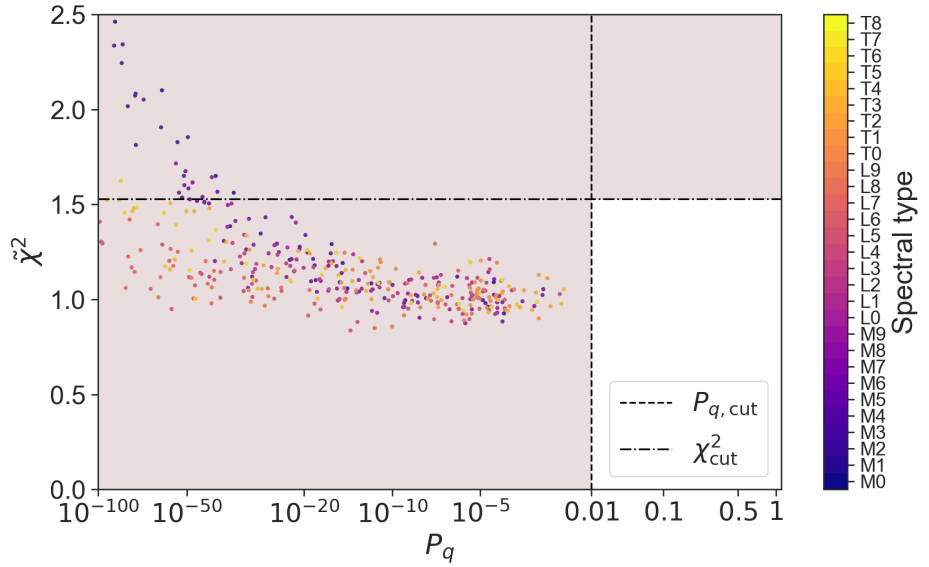
The results for the non-stationary MLT dwarfs can be seen in Figure 7.10. As expected, the brighter the source, the higher the chi-squared value. For dwarfs located in the centre, the maximum quasar probability is around 10^{-3}. This holds for the faintest dwarfs of $F = 10\,\mu$Jy only. For brighter MLT dwarfs, the probability goes down towards $P_\mathrm{q} \sim 10^{-100}$.

For a subset of dwarfs, the quasar probability starts to increase once the MLT dwarf moves away from the centre. It gets close to a probability of 1 at around a distance of $0\farcs8$ from the central quasar, which corresponds to 2 pixels on the postage stamp.

Once the source has moved outside the radius $R_{\tilde{\chi}^2} = 1\farcs2$ within which the chi-squared calculation takes place, $\tilde{\chi}^2$ returns to normal values. This is because our pipeline would essentially calculate zero flux for such sources, as the central source has moved too far outwards. In this low flux regime it is difficult to make accurate predictions about what source population the object belongs to.



(a) Simulated stationary MLT dwarfs as a function of flux in J.



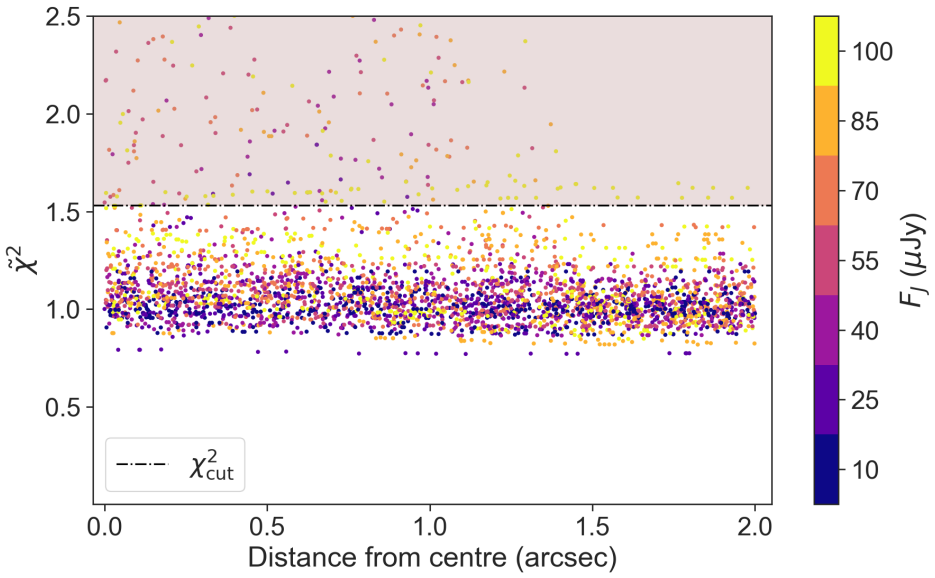
(b) Simulated stationary MLT dwarfs as a function of spectral type

Figure 7.9: A set of 406 simulated stationary MLT dwarfs. The source location is fixed in the centre across all bands. The MLT SED models range from $M0$ to $T8$, and have a J-band flux between 10 μJy and 100 μJy.

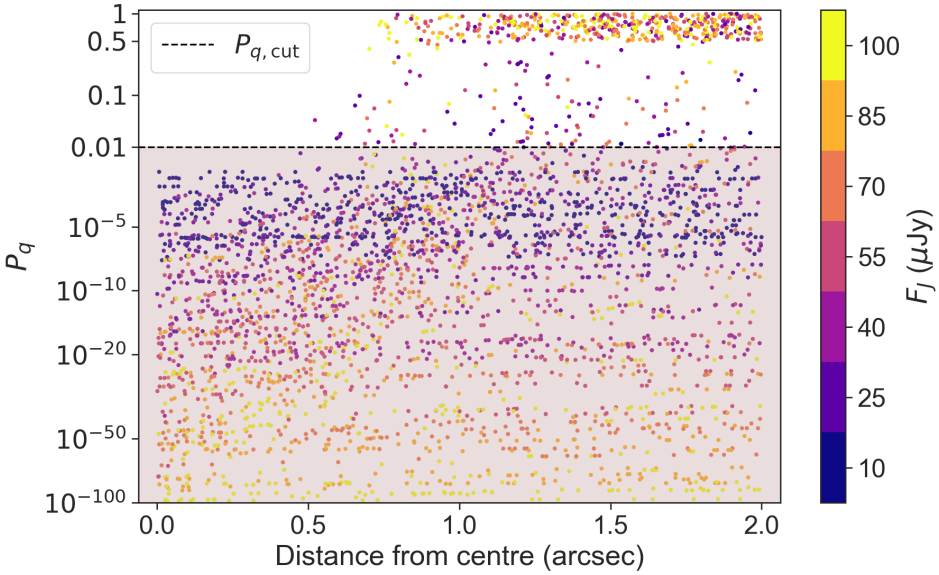
7.3.3 Early type galaxies

We simulate early type galaxies as extended sources with colours of that of ETGs. However, we don't want the simulated ETGs to be too extended as this would make the differentiation from quasars too easy. We do this by using a PSF that is a mixture of 2 Gaussians,

$$S(r;R,\sigma_1,\sigma_2) = \frac{1}{2}(R+1)\cdot\left\{\frac{1}{\sqrt{2\pi}\sigma_2}R\exp\left[-\frac{1}{2}\left(\frac{r}{\sigma_2}\right)^2\right]+\frac{1}{\sqrt{2\pi}\sigma_1}\exp\left[-\frac{1}{2}\left(\frac{r}{\sigma_1}\right)^2\right]\right\}, \tag{7.12}$$



(a) Moving MLT dwarfs with evolving $\tilde{\chi}^2$.



(b) Moving MLT dwarfs with evolving P_{q}.

Figure 7.10: A set of 3587 simulated non-stationary MLT dwarfs, where the source location is offset from the centre in SDSS bands only. The source location in the UKIDSS bands remains in the centre.

with parameters $\sigma_1 = 0.4''$, $\sigma_2 = 0.6''$ and $R = 0.5$. Neither σ_1 nor σ_2 can be too large, otherwise the flux contribution only comes from one Gaussian which makes it a point source again. For the point sources in our dataset we typically have $\sigma \sim 0.3''$ if we use a 1-Gaussian PSF. For the 2-Gaussian PSF simulating the extended galaxies, we chose one of the Gaussian uncertainties to be two times that, hence $\sigma_2 = 0.6''$. We have found that this choice gives good classification results, though the PSF model could be improved in future.

This violates our model in both PSF (extended, not point source) and colours (ETG, not quasar) and thus we expect a large $\tilde{\chi}^2$ and a low P_{q}.

We vary the parameters available in `Pq_server` (Euclid Collaboration, 2019), using two different formation redshifts at $z = 3$ and $z = 10$, and a redshift range $1 \leq z \leq 2$.

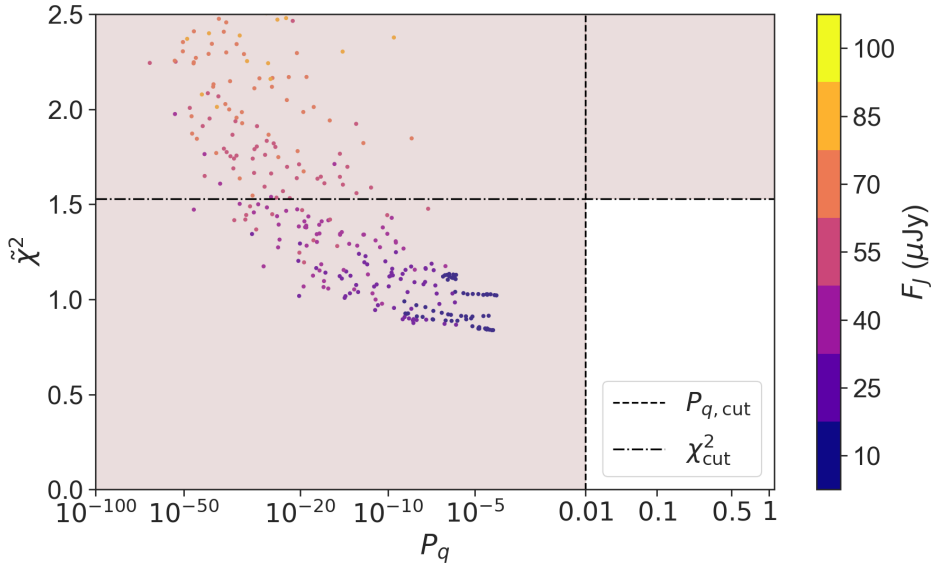


Figure 7.11: A set of 420 simulated ETGs as a function of quasar probability, residual chi-squared and flux in J. These lie in the redshift range between $1 \leq z \leq 2$ and at formation redshifts of $z = 3$ and $z = 10$.

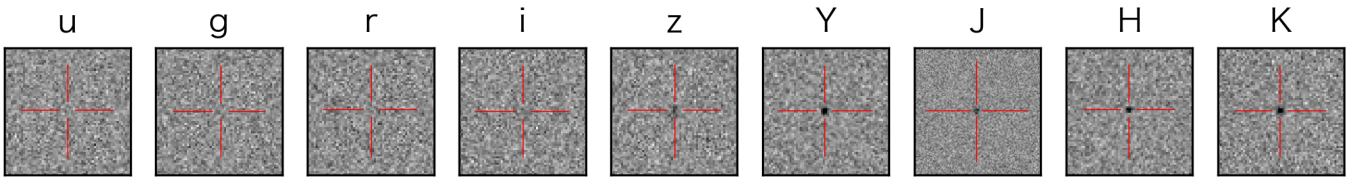
As we can see in Fig. 7.11, the quasar probabilities are close to zero as expected, with a maximum probability of $P_{\mathrm{q}} = 6.2 \times 10^{-5}$. Our residual chi-squared $\tilde{\chi}^2$ increases with increasing flux levels as expected. The $\tilde{\chi}^2$ are large except in the case for simulations of faint ETGs. This occurs below a threshold of ≈ 25 μJy, where it becomes more difficult to distinguish between quasars and contaminants. Overall, this shows that our method can distinguish between ETGs and quasars whenever the source follows the colours of either of our templates and is sufficiently bright.

7.4 Application to real quasar candidates

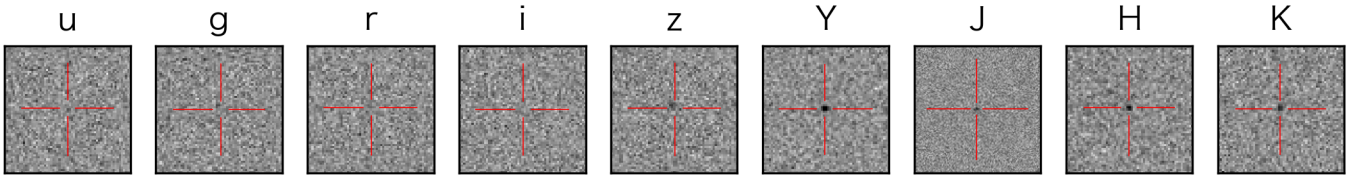
For our target of $z \gtrsim 6.5$ quasars, we require information in the optical and NIR. Optical data is needed to 'see' the absorption bluewards of Lyα, and NIR data is needed to observe the strong emission at Lyα.

The set of sources is taken from Mortlock et al. (2012). Mortlock et al. (2012) obtain this set from DR8 of UKIDSS LAS (Lawrence et al., 2007), cross-matched with SDSS DR7 (York et al., 2000), out of which they expect only ~ 15 to be high-redshift quasars with $Y \leq 19.8$. The set of sources have been selected via cuts in colour and have been examined for flags that would indicate that the source is extended, has proper motion or unreliable photometry. Specifically, the following cuts were made

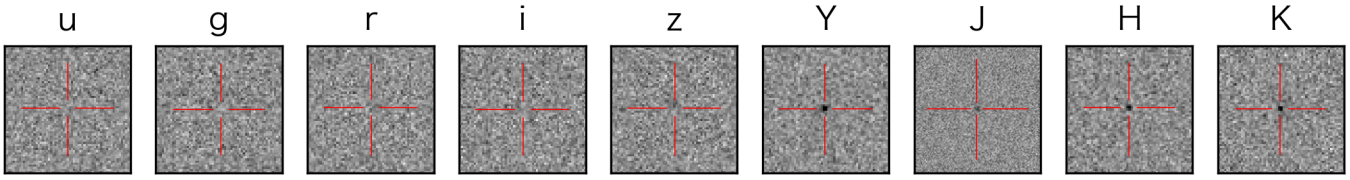
- $\hat{i} - \hat{Y} \geq 2.8$



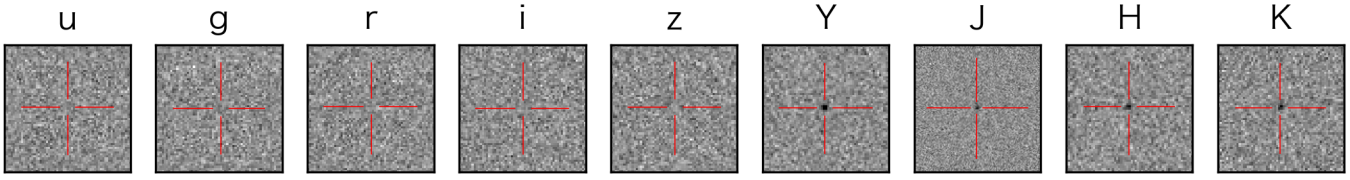
(a) Offset: 0 arcsec



(b) Offset: 0.6 arcsec

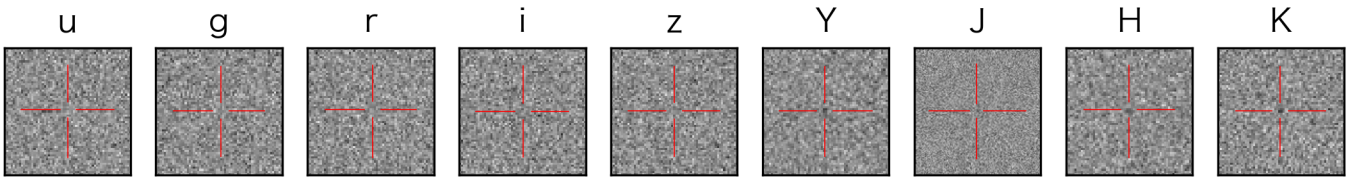


(c) Offset: 1.1 arcsec

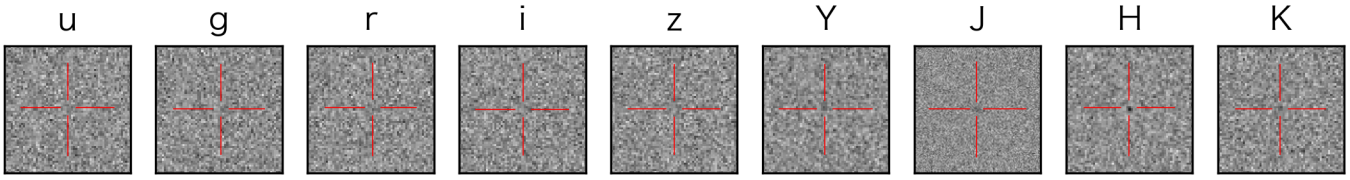


(d) Offset: 1.7 arcsec

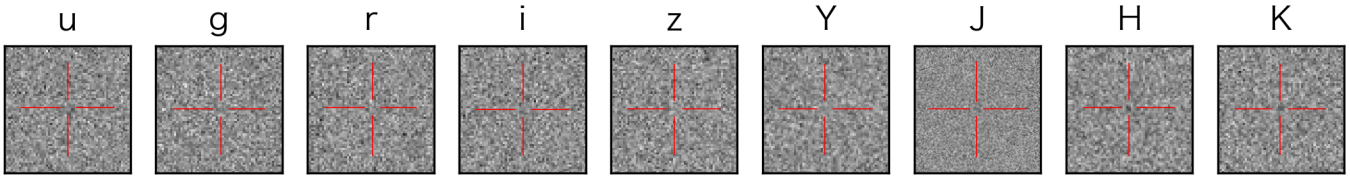
Figure 7.12: A simulated source at redshift $z = 6.0$ with correct quasar template colours but with various location offsets in SDSS.



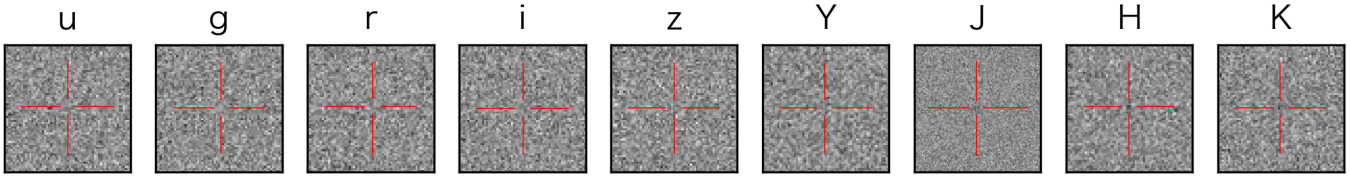
(a) $z = 6.5$



(b) $z = 7.0$



(c) $z = 7.5$



(d) $z = 8.0$

Figure 7.13: A simulated faint quasar of magnitude $m_J = 28.90$ mag (or flux $F_J = 0.01\mu$ Jy), standard line equivalent width and a red continuum at different redshifts

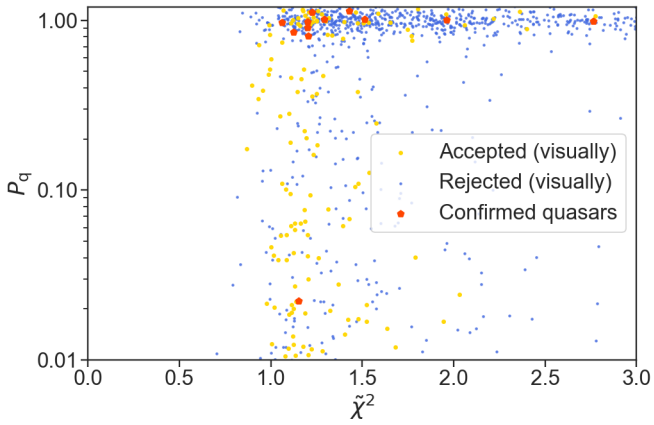


Figure 7.14: Scatter plot of $\tilde{\chi}^2$ vs P_q based on the classification labels of the visual inspection. Gaussian noise is added for candidates with $P_q > 0.9$ to increase readability ($\mathcal{N}(P_q, 0.01)$ and $\mathcal{N}(\tilde{\chi}^2, 0.01)$). The orange crosses correspond to visually accepted candidates. The blue dots correspond to visually rejected candidates. The majority of rejected candidates are not visible on the plot due to range of axes chosen. The red dots show spectroscopically confirmed quasars (Fan et al., 2000, 2001, 2004; Goto, 2006; Venemans et al., 2007; Mortlock et al., 2009, 2011; Jiang et al., 2016). The low probability confirmed quasar is assigned a probability of $P_q = 1.00$ when including the SDSS photometry (see Section 7.5.3.2). We test the manual threshold classifier on a set of different quasar probability thresholds: $P^{th}_{q,min} = 0$, $P^{th}_{q,min} = 0.01$ and $P^{th}_{q,min} = 0.1$.

- $\hat{Y} - \hat{J} \leq 0.88$
- $\hat{i} - \hat{Y} \geq 1.20 - 2.40\left(\hat{Y} - \hat{J}\right)$
- non-extended objects (in UKIDSS): `MergedClassStat` ≤ 4.0
- no unreliable photometry
- close to zero proper motion

This leaves a set of $\sim 9 \times 10^3$ high-redshift quasar candidates.

7.4.1 Survey data

Since the set of quasar candidates in Mortlock et al. (2012) had been visually inspected in SDSS and UKIDSS LAS data, we use the same two surveys but use updated data releases. We receive optical information from the SDSS Data Realease 12 (DR12) and obtain NIR data from the UKIDSS LAS Data Release 10 PLUS (DR10PLUS). SDSS uses the Sloan Foundation 2.5m telescope at the Apache Point Observatory in New Mexico (Gunn et al., 1998). It has a wide-field CCD camera including five optical to NIR filters u, g, r, i, z. UKIDSS LAS makes use of the WFCAM wide field camera and has four NIR wavelength bands Y, J, H, K (Lawrence et al., 2007). Its observing area is matched to the SDSS area.

7.4.2 Validation set

Out of the $\sim 9 \times 10^3$ high-redshift quasar candidates from Mortlock et al. (2012), ≈ 4300 have been visually labelled. The labels are binary, where a positive (accepted) label corresponds to the source resembling a quasar in all of the bands, and a negative (rejected) label corresponds to a source that either looks like a contaminant or to an image with data processing issues around the nominal source position. Details on the labels can be found in Appendix A.

From the pool of candidates, only ~ 800 have a positive label, out of which 14 are spectroscopically confirmed quasars at $z > 5.7$ ((Fan et al., 2000, 2001, 2004; Goto, 2006; Venemans et al., 2007; Mortlock et al., 2009, 2011; Jiang et al., 2016). This gives a ratio of approximately $1 : 6$ between probable quasars and contaminants.

Further, out of the complete sample of ≈ 4300 candidates, only ≈ 450 have $P_\mathrm{q} > 0.01$, and ≈ 330 have $P_\mathrm{q} > 0.1$. Thus, requiring $P^\mathrm{th}_\mathrm{q,min} = 0.01$ already reduces the pre-selected candidate list n order of ~ 10. Requiring $\tilde{\chi}^{2,\mathrm{th}} = 1.5$ reduces the candidate list to ≈ 2300. Requiring $\tilde{\chi}^{2,\mathrm{th}} = 1.5$ and $P^\mathrm{th}_\mathrm{q,min} = 0.1$ gives ≈ 160 sources, thus less than half than by just requiring $P^\mathrm{th}_\mathrm{q,min} = 0.1$.

7.4.3 Manual threshold selection

We manually select thresholds for the residual average chi-squared value, the residual maximum chi-squared value and the quasar probability. In particular, we manually accept a candidate if all of the following holds,

- $P_\mathrm{q} > P^\mathrm{th}_\mathrm{q,min}$
- $\tilde{\chi}^2 < \tilde{\chi}^{2,\mathrm{th}}$
- $\tilde{\chi}^2_\mathrm{max} < \tilde{\chi}^{2,\mathrm{th}}_\mathrm{max}$

We compare our predicted results to the visually labelled validation set. These thresholds are optimised in Section 7.5.

We do expect that these thresholds need to be adapted for future surveys. In addition, one will need to add more quasar SED models that cover a larger range in order to acquire a similar $\tilde{\chi}^2$ value. This is because future surveys will have more precise photometry and hence smaller uncertainties. Due to these smaller uncertainties, the $\tilde{\chi}^2$ value would increase if we did not add more quasar template SEDs.

That means, for application on future surveys we would not require a pre-labelled data set. Visual inspection would only be needed for testing the algorithm on a new data set. Thresholds can be set

more conservatively to find the most likely quasar candidates even if this means that other quasars will be missed.

One could also select the thresholds manually based on the summary values we get for our candidates such as in Fig. 7.14. The figure shows a scatter plot of $\tilde{\chi}^2$ vs $P_{\rm q}$, where the values of spectroscopically confirmed quasars can be seen in red. All of the quasars have $P_{\rm q} > 0.1$, where the quasars with the lowest probability happen to be the ones at the lowest redshift. If $P_{\rm q}$ is close to 1 but $\tilde{\chi}^2$ is large then this indicates that even though the database/calculated photometry agree very well with a quasar SED, the source either does not resemble a point source or it has moved. On the other hand, if $P_{\rm q}$ is close to 0 and $\tilde{\chi}^2$ is low, then even though the source is a stationary point source and follows quasar colours, it is much more likely that the source is a contaminant.

As the classification labels are based on visual inspection, some of the accepted sources will have a quasar probability close to zero. These are candidates that look like quasars but are actually much more likely to be contaminants. Thus, it could be argued that these very low probability candidates should have been rejected instead. This can make it slightly difficult to compare our results to the visual classification labels. For this reason the cut in the quasar probability $P^{\rm th}_{\rm q,min}$ actually completely removes candidates below that threshold, such that the classification is based solely on $\tilde{\chi}^{2,\rm th}$ and $\tilde{\chi}^{2,\rm th}_{\rm max}$ applied on sources with $P_{\rm q} > P^{\rm th}_{\rm q,min}$. Generally, we do not expect this manual threshold classifier to perfectly re-create the visually labelled data set. However, it will be useful for pinning down the most likely candidates on which follow-up observations can be done.

Better results will be achieved if the thresholds are optimized for each new survey. For instance, the better the survey PSF can fit a point source, the lower/larger the chi-squared values will be for positive/negative candidates. In addition, a more sophisticated PSF will be able to measure the photometry more accurately, resulting in a better choice of SED template. Therefore, if other surveys use different PSFs to the ones described in Sec. 6.3, it is advised to adjust the parameter cuts accordingly. This means a new training set would be needed. Furthermore, for sources with zero flux in u, g, r, i, the chi-squared value in those bands will be close to 1. This will bias the average chi-squared value towards lower values. Therefore, future surveys that have fewer/more bands to cover the Gunn-Peterson trough will have candidates with larger/lower average chi-squared values. Similarly, surveys in which candidates have non-zero flux in fewer/more bands compared to SDSS and UKIDSS LAS will have a bias towards lower/larger chi-squared values.

7.5 Results

We discuss our classification results, possible flaws with our method and how it can act as a cleaning tool.

7.5.1 Performance measure

As the number of positive samples in our dataset is very small, classifying nearly every candidate as a contaminant (negative label) would already give a high accuracy, giving us a false sense of the effectiveness of our classifier. We therefore need a different performance measure, one that shows how many of our quasar candidates have correctly been accepted and how many have been incorrectly rejected.

For this we need to define precision and recall,

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{7.13}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{7.14}$$

where TP the number of positive samples that have been correctly classified as positive, and FN (FP) the number of positive (negative) samples that have been falsely classified as negative (positive) by our model.

Precision therefore describes the fraction of correctly accepted candidates versus the total number of accepted candidates. The latter includes incorrectly accepted sources as well. Hence, precision does not take into account how many actual positive samples have been missed by the classifier, it just tells us that if the classifier predicts a positive sample then how accurate is that it is actually a positive sample. The missed positives are taken into account by the recall, which is defined as the fraction of correctly accepted candidates versus the total number of actual positive sources.

In our case we want to detect as many quasars as possible. We would rather include a contaminant in our result than to miss out on a quasar, and are therefore more interested in the recall of our classifier.

We can skew the weight towards the recall by using the Fβ performance measure (Baeza-Yates, 2011),

$$F_\beta = \left(1 + \beta^2\right) \frac{\text{precision} \times \text{recall}}{[(1 + \beta^2) \times \text{precision}] + \text{recall}}. \tag{7.15}$$

A perfect classifier has an F_β score of 1. The more F_β approaches 0, the worse the classifier becomes. For $\beta = 1$, the equation above becomes the harmonic mean between precision and recall,

$$F_1 = \frac{2}{(1/\text{precision}) + (1/\text{recall})}. \tag{7.16}$$

If we set $\beta > 1$, then more weight is given to the recall compared to the precision. Similarly, a value of $0 \leq \beta < 1$ gives more weight to the precision.

Thus, we will focus on $\beta > 1$, but will still include $\beta = 1$ for reference. In total, we test for values of $\beta = 1, 2, 3$. The larger β, the more fractional false positives and thus the more candidates one

would have to visually inspect. How much contaminants one is willing to include thus depends on the sample size. For a dataset that contains hundreds of thousand sources, a value of $\beta = 1 - 2$ might be better than $\beta = 3$. Similarly, for a small dataset of a few hundred sources one might be more willing to allow for $\beta = 3$.

7.5.2 Classification results

We present the classification results of our validation set. Figures 7.15,7.16 and 7.17 show contour plots of the F_β score for $\beta = 1, 2, 3$ as a function of threshold in the average and maximum residual chi-squared values $\tilde{\chi}^{2,\mathrm{th}}, \tilde{\chi}^{2,\mathrm{th}}_{\mathrm{max}}$. The manual threshold classifier is applied on subsets of the validation set for quasar probability thresholds of $P^{\mathrm{th}}_{\mathrm{q,min}} = 0$, $P^{\mathrm{th}}_{\mathrm{q,min}} = 0.01$ and $P^{\mathrm{th}}_{\mathrm{q,min}} = 0.1$ respectively. Notice that $P^{\mathrm{th}}_{\mathrm{q,min}} = 0$ includes the full validation set, whereas $P^{\mathrm{th}}_{\mathrm{q,min}} = 0.01$ and $P^{\mathrm{th}}_{\mathrm{q,min}} = 0.1$ lead to subsets of the validation set. The results are summarised in Table 7.3. The overall largest F_β score is achieved by $\beta = 3$ and $P^{\mathrm{th}}_{\mathrm{q,min}} = 0$. It is given by $F_3 = 0.79$ and uses the thresholds $\tilde{\chi}^{2,\mathrm{th}} = 2.39$ and $\tilde{\chi}^{2,\mathrm{th}}_{\mathrm{max}} = 6.83$.

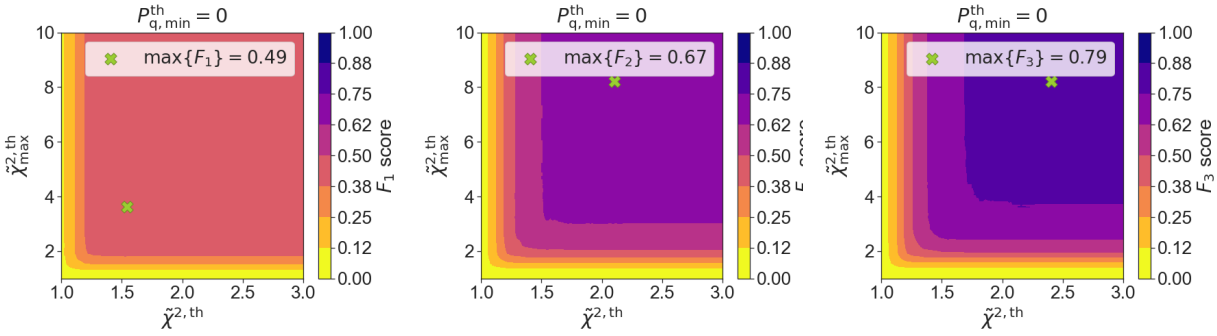
For all three probability cuts the maximum F_β score increases with increasing β, for $\beta = 1, 2, 3$, such that it is at its highest value at $\beta = 3$. This means that the recall has a higher value than the precision, and that we therefore miss less quasars with the downside of including some contaminants.

Further, the maximum scores of F_1, F_2 and F_3 decrease with increasing probability cut $P^{\mathrm{th}}_{\mathrm{q,min}}$. F_1 has its highest value of $F_1 = 0.48$ at $P^{\mathrm{th}}_{\mathrm{q,min}} = 0$. This is low compared to the maximum F_3 score of $F_3 = 0.79$, also at $P^{\mathrm{th}}_{\mathrm{q,min}} = 0$. This suggests that we have quite a lot of false positives in our sample set. This is confirmed by very low precision values for all F_β values. The precision decreases with increasing $P^{\mathrm{th}}_{\mathrm{q,min}}$ for F_2 and F_3. For F_1. there is a small increase in precision of 0.01 between $P^{\mathrm{th}}_{\mathrm{q,min}} = 0.01$ and $P^{\mathrm{th}}_{\mathrm{q,min}} = 0.1$.

In addition to the precision, the recall also decreases with increasing $P^{\mathrm{th}}_{\mathrm{q,min}}$. A drop in recall with increasing $P^{\mathrm{th}}_{\mathrm{q,min}}$ means that the recall is larger for sources with $P_{\mathrm{q}} < P^{\mathrm{th}}_{\mathrm{q,min}}$ than it is for sources with $P_{\mathrm{q}} \geq P^{\mathrm{th}}_{\mathrm{q,min}}$. Even though the F_β scores are smaller for non-zero thresholds in the probability than they are for $P^{\mathrm{th}}_{\mathrm{q,min}} = 0$, the visual classification labels will be more accurate for the set of $P^{\mathrm{th}}_{\mathrm{q,min}} > 0$. This is because the majority of candidates with $P_{\mathrm{q}} < 0.01$ are contaminants, and therefore any source with a positive label as per visual classification is actually much more likely to be a contaminant.

As the validation set gets smaller with increasing quasar probability threshold $P^{\mathrm{th}}_{\mathrm{q,min}}$, the regions of maximum F_β score narrow down to smaller areas, and hence a smaller subset of possible thresholds in the residual average and maximum chi-squared $\tilde{\chi}^{2,\mathrm{th}}, \tilde{\chi}^{2,\mathrm{th}}_{\mathrm{max}}$ remains. Generally, as $P^{\mathrm{th}}_{\mathrm{q,min}}$ increases, the cuts in the average and maximum residual chi-squared are decreasing overall for each β individually.

At $\tilde{\chi}^{2,\mathrm{th}} = 0.1$, only ≈ 160 sources are left in the validation set (see 7.4.2). At such a low source count we can use $\beta = 3$ which gives us a score of $F_3 = 0.72$ using the thresholds of $\tilde{\chi}^{2,\mathrm{th}} = 1.70$ and



(a) $\beta = 1$ (b) $\beta = 2$ (c) $\beta = 3$

Figure 7.15: The F_β score of the manual threshold classifier as a function of the thresholds in the average and maximum residual chi-squared for (a) $\beta = 1$, (b) $\beta = 2$, (c) $\beta = 3$. It is applied to the complete validation set ($P_{\rm q} \geq 0$).

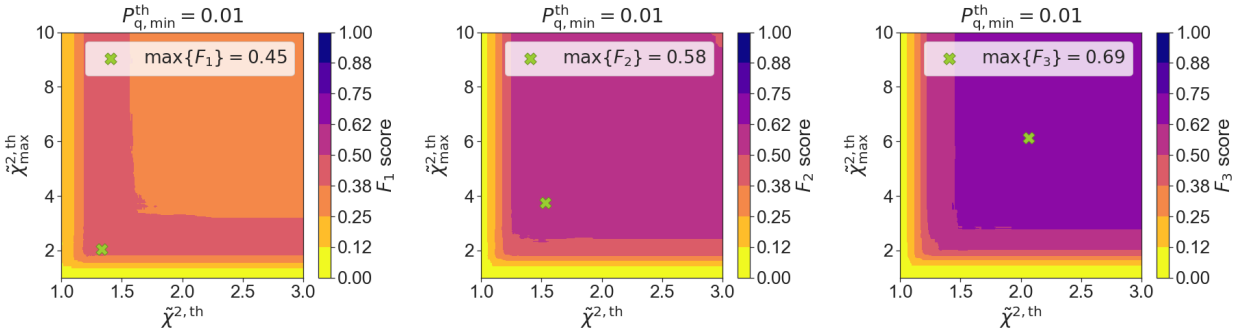
$\tilde{\chi}^{2,\rm th}_{\rm max} = 5.96$. Even though we get a higher value of $F_3 = 0.79$ at $\tilde{\chi}^{2,\rm th} = 0$, the number of accepted candidates would include much more contaminants.

Table 7.3: The thresholds in residual chi-squared $\tilde{\chi}^{2,\rm th}$,$\tilde{\chi}^{2,\rm th}_{\rm max}$ that give the largest F_β score for $\beta = 1, 2, 3$. The thresholds are applied to a subset of the validation set with quasar probability cuts of $P^{\rm th}_{\rm q,min} = 0, 0.01, 0.1$.

	$P^{\rm th}_{\rm q,min}$	$\tilde{\chi}^{2,\rm th}$	$\tilde{\chi}^{2,\rm th}_{\rm max}$	Precision	Recall	F_β
$\beta = 1$	0.00	1.52	3.96	0.34	0.83	0.48
	0.01	1.35	2.39	0.36	0.66	0.47
	0.10	1.43	2.04	0.37	0.55	0.44
$\beta = 2$	0.00	2.04	6.22	0.31	0.94	0.67
	0.01	1.43	5.96	0.29	0.86	0.62
	0.10	1.43	5.96	0.26	0.84	0.58
$\beta = 3$	0.00	2.39	6.83	0.30	0.96	0.79
	0.01	1.70	5.96	0.25	0.92	0.72
	0.10	1.70	5.96	0.22	0.92	0.69

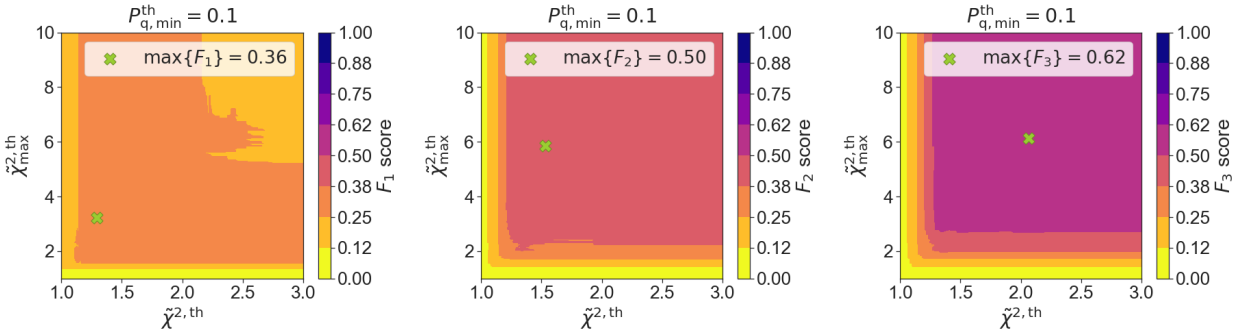
7.5.3 Discussion

We show sources where the combination of $\tilde{\chi}^2$ and $P_{\rm q}$ correctly accept high-redshift quasar candidates, and reject likely contaminants. We also show a subset of sources that had been accepted via the original visual inspection of Mortlock et al. (2012) but rejected by our classifier and vice versa. These exemplify the limits of our method but also its strengths when it correctly flags up candidates that originally had been assigned incorrect labels.



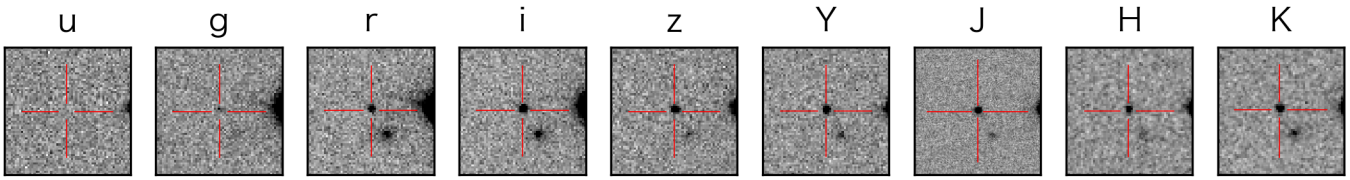
(a) $\beta = 1$ (b) $\beta = 2$ (c) $\beta = 3$

Figure 7.16: The F_β score of the manual threshold classifier as a function of the thresholds in the average and maximum residual chi-squared for (a) $\beta = 1$, (b) $\beta = 2$, (c) $\beta = 3$. It is applied to the subset of the validation set where $P_\text{q} \geq 0.01$.

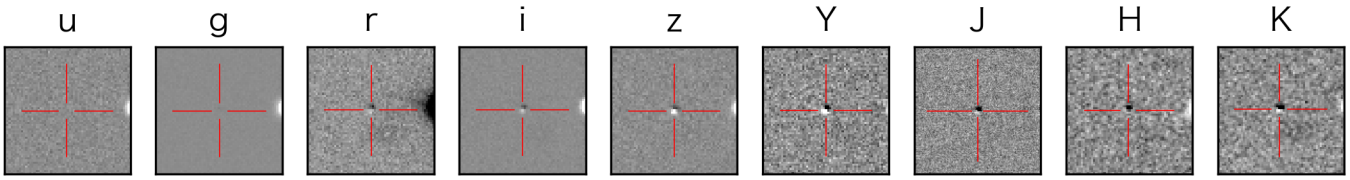


(a) $\beta = 1$ (b) $\beta = 2$ (c) $\beta = 3$

Figure 7.17: The F_β score of the manual threshold classifier as a function of the thresholds in the average and maximum residual chi-squared for (a) $\beta = 1$, (b) $\beta = 2$, (c) $\beta = 3$. It is applied to the subset of the validation set where $P_\text{q} \geq 0.1$.



(a) Original - pair



(b) Residual using the best-fit quasar SED.

Figure 7.18: This source J002101.25+004352.4 is one of a close pair. In addition, it has a bright source nearby. a) Original image data in SDSS and UKIDSS LAS, b) Residual using the best-fit quasar SED. The residual indicates that either the source's position is not completely in the centre or the source is not point-like and spherical symmetric.

7.5.3.1 Correct classification

Fig. 7.2 shows quasar ULAS J1120+0641 (Mortlock et al., 2011) at redshift $z = 7.09$. The residual image is the difference between the original data and our reconstructed data using our calculated PSF fluxes and background estimates. It can be seen that the residuals in each band are consistent with being noise. This means firstly that our PSF is a good model of a point source. Secondly, our model of a stationary point source fits well to this quasar, which is exactly what we intended to achieve. Note that in this plot we do not include the best fit SED yet, but rather use our calculated PSF photometry.

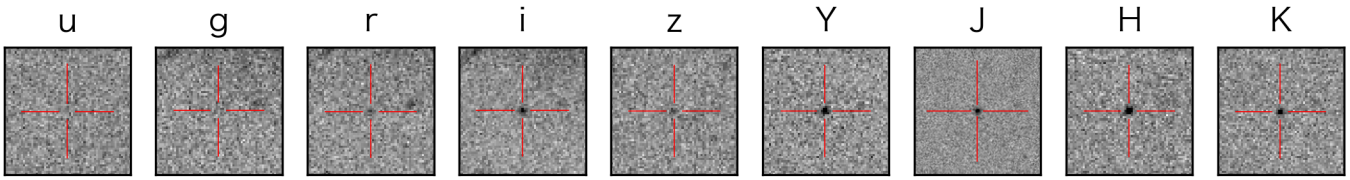
Fig. 7.3 shows candidate J000042.15+072229.9 that has a probability of being an early type galaxy of $P_{\mathrm{ETG}} = 1.00$. The fact that this source is very unlikely to be a quasar is confirmed by a large maximum chi-squared value of $\tilde{\chi}^2_{\mathrm{max}} = 6.54$. The residuals show the remaining flux, which is indicative of that the best fit quasar SED template does not fit very well to the observed flux.

Other examples such as nearby bright sources, diffraction spikes and persistence issues can be seen in Figures 7.4, 7.18, 7.19 and 7.5.

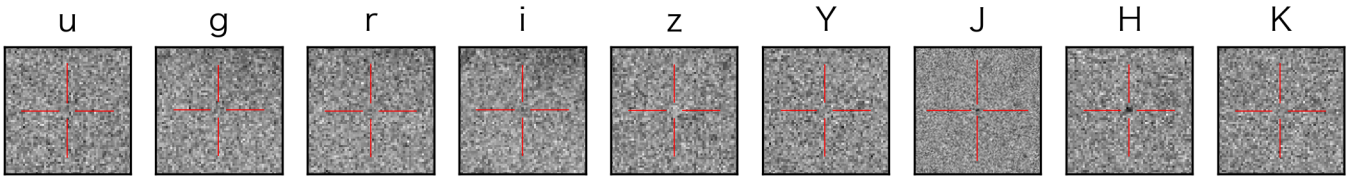
7.5.3.2 Limitations of this method

If we use $P^{\mathrm{th}}_{\mathrm{q,min}} = 0.01$, two quasars (J1148+0702 and J1044−0125) get misclassified.

Quasar J1148+0702 (Jiang et al., 2016) is with $P_{\mathrm{q}} = 1.22 \times 10^{-6}$ the quasar with the lowest probability due to missing SDSS data. This is because this source had an unwanted flag (`COSMIC_RAY`) in the SDSS database. Due to this missing data, the likelihood will be similar for the three different populations, and hence the probability falls back to the prior. Since the prior is the lowest for quasars,



(a) Original - close to diffraction spike



(b) Residual using the best-fit quasar SED.

Figure 7.19: This source J091108.35+092732.0 is close to a diffraction spike. a) Original image data in SDSS and UKIDSS LAS, b) Residual using the best-fit quasar SED. In this case the model is a good representation again since the diffraction spikes do not occur within the cross-hair (where the calculation is made). The largest chi-squared value occurs in the H-band with $\chi^2_H = 2.86$. However, the overall reduced chi-squared is low at $\tilde{\chi}^2 = 1.18$. Nevertheless, this source has a zero quasar probability $P_\mathrm{q} = 0.00$ and hence would be rejected with a cut in the quasar probability.

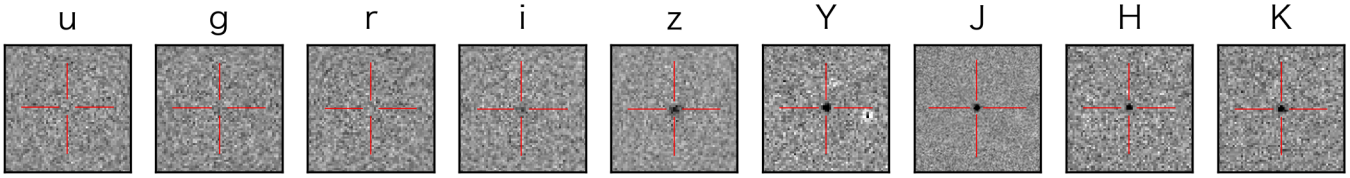


Figure 7.20: Quasar J1044−0125 (Fan et al., 2000) at redshift $z = 5.74$ that should have been accepted but was rejected by the manual threshold classifier. This is due to a large $\tilde{\chi}^2$ value in the Y-band, $\chi^2_Y = 4.56$, due to incompatible observed flux with the quasar SED template.

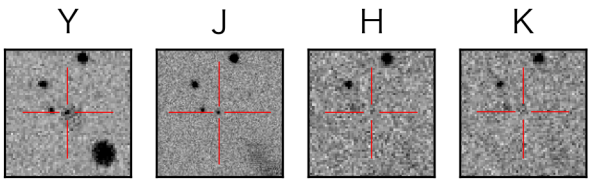
we received a very small quasar probability. Upon including the SDSS images as well, the probability drastically increases towards $P_\mathrm{q} = 1.00$.

The other quasar, J1044−0125 (Fan et al., 2000), is shown in Fig. 7.20. This quasar is at redshift $z = 5.74$ (Goodrich et al., 2001) but has a predicted redshift of $\hat{z} = 6.00$. We therefore expect some mismatch in the template flux, which is agreed with a large Y-band chi-squared value of $\chi^2_Y = \tilde{\chi}^2_\mathrm{max} = 4.56$.

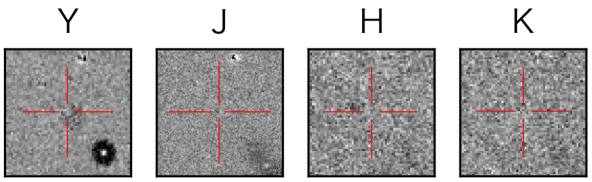
This indicates that none of our 9 templates are a good match for this quasar SED. Therefore, a possible extension to this project would be to add more quasar SED templates.

It should be noted that `Pq_server` works best for redshifts $z \gtrsim 6.5$, and as such we expect our pipeline to work especially well for redshifts beyond 6.5, and to miss out on some quasars below that limit. The one confirmed quasar at $z \sim 7$ that we have in our sample has a promising maximum chi-squared value of $\tilde{\chi}^2_\mathrm{max} = 1.28$.

Some data processing issues can also be difficult to detect. This is exemplified in Figure 7.21, where



(a) Original - persistence image



(b) Residual using the best-fit quasar SED.

Figure 7.21: Candidate J073749.93+261844.9 is the persistence image. SDSS images have been neglected as they have data processing issues flagged in the database. a) Original image data in SDSS and UKIDSS LAS, b) Residual using the best-fit quasar SED. The pipeline fails to flag this up as it does conform our model of a stationary point-source that follows the SED of one of our quasar templates. Thus, it gives a residual chi-squared of $\tilde{\chi}^2 = 1.00$ and a quasar probability of $P_{\rm q} = 0.06$.

a persistence image gives a high quasar probability.

7.5.3.3 Cleaning tool

Our classifier further acts as a cleaning tool by identifying candidates that have been labelled incorrectly in the original visual inspection. E.g. a moving source or extended object that should have been rejected, or a quasar-like source that should have been accepted. This makes our method is more reliable than (the initial) visual inspection. In future surveys, our method will pre-select probable quasar candidates before any visual inspection has to be done, and hence it won't have labels to compare its results with. The comparison only occurs whenever we want to test our method on a new survey against a (visually) labelled validation set.

An example of a candidate that had been accepted based on visual inspection but should have been rejected is shown in Fig. 7.22. This candidate has moved and thus should be rejected. The candidate has a maximum chi-squared value of $\tilde{\chi}^2_{\rm max} = 10.09$ in Y because the predicted flux is half the observed value. The next largest chi-squared value is in J since the source has moved. The radius in H and K for the chi-squared calculation is too small to encompass the moving source in those bands, however the chi-squared value is still large ($\chi^2_{H,K} \sim 4$) due to the observed flux being inconsistent with the template flux. Thus, our classifiers correctly reject this candidate.

Candidate J115132.21+07518.8, shown in Fig. 7.23, is a candidate that has been identified as a good candidate via manual threshold selection. It had originally been rejected based on visual inspection due to missing H and K data back in 2011 when the visual inspection had been made. Our pipeline gave this candidate a quasar probability of $P_{\rm q} = 0.56$ and a residual chi-squared value of $\tilde{\chi}^2 = 1.24$.

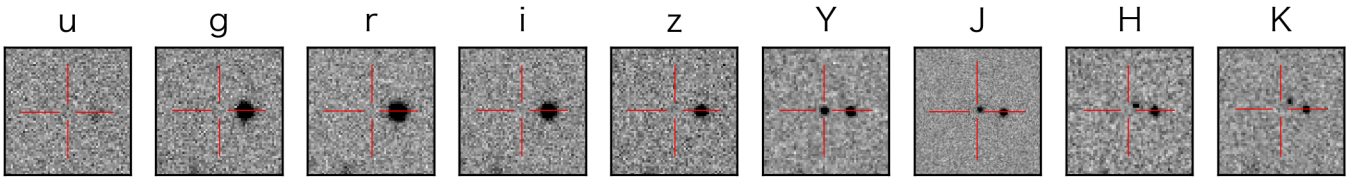


Figure 7.22: Candidate J002000.71−003412.8 had originally been accepted via visual inspection, however it is clear that the source is moving. Consequently, large chi-squared values in the NIR bands led to the correct rejection of this candidate by our classifiers.

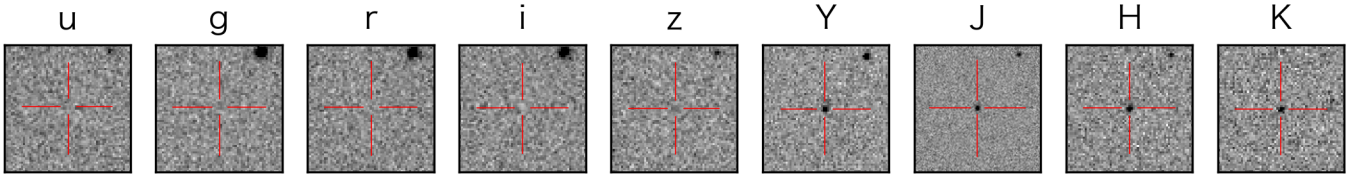
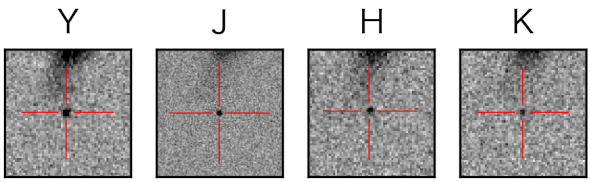


Figure 7.23: Candidate J115132.21+07518.8 had originally been rejected based on visual inspection due to missing data in bands H and K back in 2011 when the visual inspection had been made. A high quasar probability and low $\tilde{\chi}^2$ indicates that this candidate should have been accepted. After including Pan-STARRS photometry of $i = 23.00 \pm 0.19$ and $z = 21.33 \pm 0.08$, the quasar probability shrank from $P_{\rm q} = 0.56$ to $P_{\rm q} = 0.01$. Nevertheless, the classifier only has information in SDSS and UKIDSS and thus correctly accepts this candidate.

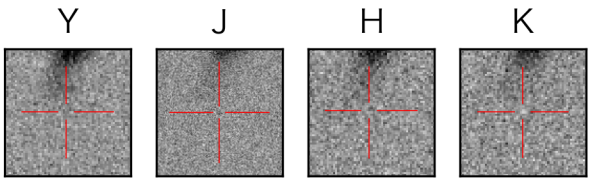
Indeed this candidate does look like a good quasar candidate. Subsequent analysis of Pan-STARRS photometry in i and z ($i = 23.00 \pm 0.19$ and $z = 21.33 \pm 0.08$) decreased the probability to $P_{\rm q} = 0.01$. However, based on data of SDSS and UKIDSS alone this candidate should have been accepted. Thus, this candidate is correctly accepted by our classifier.

Another such example is shown in Figure 6.10, which shows the original and residual image together with the chi-squared pixel values of the visually rejected candidate J092036.65+093407.9. The overall reduced chi-squared of this source is $\tilde{\chi}^2 = 1.12$, with a quasar probability of $P_{\rm q} = 1.00$ and a predicted quasar redshift of $\hat{z} = 5.57$. This source was likely visually rejected due to the non-zero flux in r which indicates a redshift below 6. As the searched redshift range in Mortlock et al. (2012) was $\hat{z} \gtrsim 6.5$, this candidate was subsequently visually rejected. In this project, however, we are able to model quasars at $z \geq 5.2$, and hence our methods do not discard this candidate. If we are only interested at possible high-redshift quasars at $z > 7$, we can simply remove candidates with a predicted redshift lower than that.

Another instance is candidate J143135.16+102210.9, which is shown in Figure 7.24. This only shows postage stamps in UKIDSS since error flags present in SDSS meant that those images were not downloaded. The complete postage stamps are shown in Figure A.17. This candidate resembles a high-redshift quasar with $\hat{z} > 7$, and was originally visually accepted by (Mortlock et al., 2012). After additional photometric data was collected, the candidate was labelled as variable, being a supernova hosted by the nearby galaxy. It is great that our methods were able to pick up this source, which, based on the available information alone, could have been a possible high-redshift candidate.



(a) Original



(b) Residual using the best fit quasar SED

Figure 7.24: Candidate J143135.16+102210.9 was originally visually accepted, but upon closer inspection and after further photometric data was collected, the source was visually rejected.

7.6 Conclusions

We have developed an automated tool to find high-redshift quasars at $z \gtrsim 6.5$ from images and catalog data of broad-band surveys. The results can then be used for visual inspection and spectroscopic follow-up. We validated our pipeline via $\sim$ 4300 candidates taken from Mortlock et al. (2012), which have been visually inspected and labelled in SDSS and UKIDSS.

Mortlock et al. (2012), followed by Euclid Collaboration (2019), developed a Bayesian model comparison based on colour information that includes contaminant populations of MLT dwarfs and early type galaxies. We have updated this approach to include colour templates in the u, g, r bands which previously had been set to zero flux. The quasar SED models are not designed for quasars at $5.2 \lesssim z \lesssim 6.5$ since the flux blueward of Lyman-alpha is set to zero. Hence, this leads to some limitations in modelling the template in u, g, r at these lower redshifts.

Photometry cannot test the morphology and stationarity of high-redshift quasars, and therefore we analyse image cutouts on the pixel level. We do not use a full Bayesian model comparison on the pixel level as that would be too computationally expensive. Rather, we use chi-squared statistics to test against our model of a quasar. The model is that of a stationary point source that follows the SED of a quasar.

Our automated pipeline extracts these and other features for a given source. Starting from sky locations of a list of $\sim$ 4300 cross-matched SDSS and UKIDSS LAS sources, selected and labelled by Mortlock et al. (2012), our pipeline first processes the candidates and extracts information such as database and PSF photometry, residual chi-squared goodness of fit $\tilde{\chi}^2$ and quasar probability P_q. Next, we discard candidates with $P_\mathrm{q} < P^\mathrm{th}_\mathrm{q,min}$ in order to get rid of the majority of contaminants. After that, setting cuts in our reduced chi-squared values further remove contaminants from our list

of candidates. Our manual cuts in chi-squared and the quasar probability work well in reproducing the previous classification results of Mortlock et al. (2012). Moreover, our method acts as a cleaning tool and flags up candidates that originally were assigned the wrong visual inspection classification label. The remaining candidates can either be visually inspected and classified or they can be fed to a pre-trained machine learning classifier. If the former is applied and they get visually classified, the sample can be used as a training set for another machine learning classifier. In the latter case, the ML classifier should be able to sift out much more contaminants than the manual threshold selection achieves on its own.

This method can be adapted for use for upcoming surveys such as LSST and Euclid. This does not even require a labelled training set and can be applied on-the-go, with summary features that should not vary too much over different surveys if the survey PSFs are similar to the ones described here. Otherwise, re-optimizing the parameter cuts on a small sample set is advised.

Thus, our method allows not only to get rid of more contaminants than previously, but also to scale up the number of candidates that can be processed in a given time frame. All of this is done automatically. The pipeline so far works for SDSS and UKIDSS LAS and can be easily adapted to future surveys.

Chapter 8

Machine learning classification pipeline

In the classification pipeline, we use machine learning techniques to increase both completeness and efficiency to optimise our quasar classification. The classifiers are trained on the summarised features from our data processing pipeline as described in Section 6. Running the trained classifiers on a new set of sources gives back a set of probable quasar candidates. The classification result is more accurate than that of our two-metric threshold classifier (see Section 7), though far less easy to interpret.

The mechanism is illustrated in Figure 8.1. The classification pipeline starts from the output of the data processing pipeline, and feeds the features of candidates with $P_{\mathrm{q}} > 0.01$ to the classifier. The classifier then decides which sources are likely to be high-redshift quasars and which are not.

The classifiers we use are a decision tree for interpretability (see Section 5.4) and a random forest for increased performance (see Section 5.5). We test our classifiers on ~ 4300 visually inspected and labelled candidates from Mortlock et al. (2012). They have been classified in the optical and NIR using data from SDSS and UKIDSS.

All our classifiers work well in reproducing the previous classification results of Mortlock et al. (2012). In addition, our classifiers are able to flag mislabeled candidates that were assigned the wrong visual inspection classification label.

Compared to the previous visual classification in Mortlock et al. (2012), this allows to sift out most contaminants before having to do any visual inspection. We separately test our classifiers with features independent of redshift and of survey specifics, and with features that are dependent on redshift and on survey specifics, and compare the results. The advantage of using redshift- and survey- independent features is that they can be applied to future surveys such as Euclid and LSST. Though, for the same reasons as in the previous chapter, some re-training will still have to be applied. In short, our method drastically reduces the amount of visual classification that is needed. Its results can then be used for visual inspection and spectroscopic follow-up.

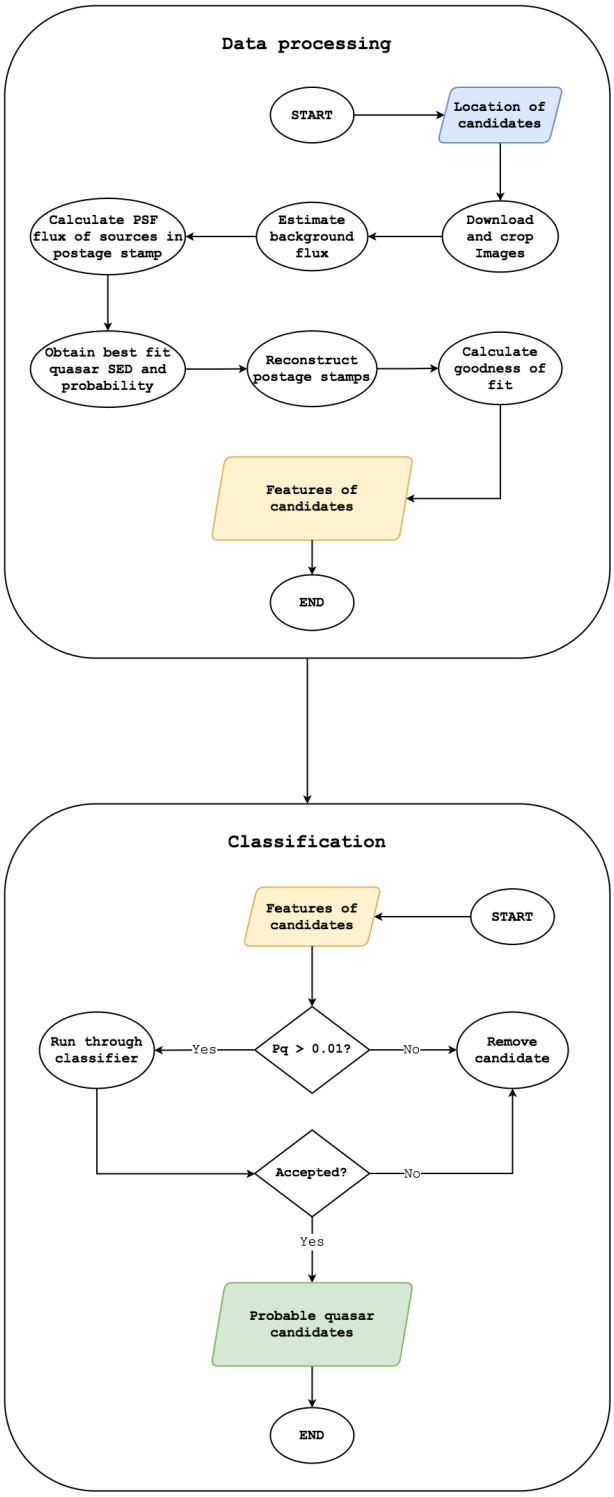
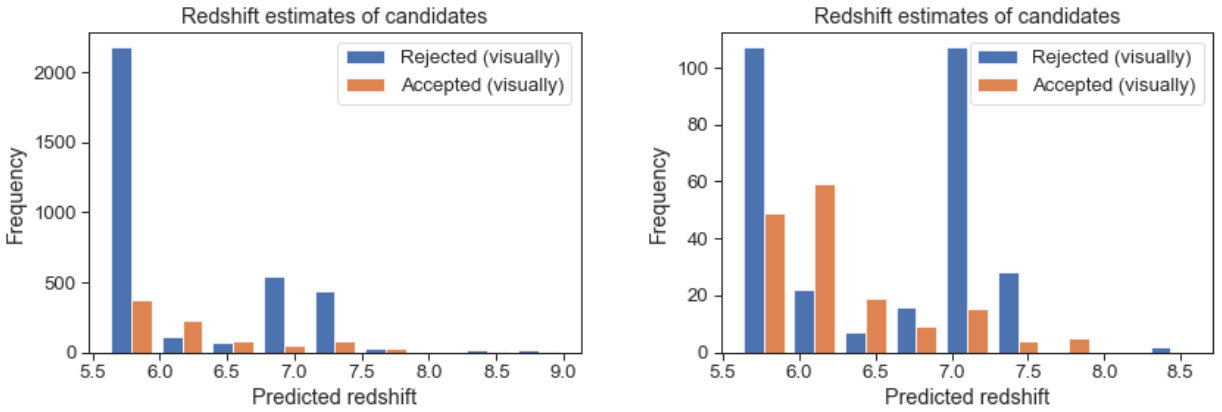


Figure 8.1: Flowchart of the data processing and the classification pipeline.
1) Data processing (Chapter 6): First, for a given source location, all available image data is downloaded and processed. Next, the background pixel flux is estimated via sigma clipping. After that, the flux of each source is calculated via the fitting of point spread functions. The quasar probability and best fit quasar SED are obtained from the observed photometry. Finally, each individual postage stamp is reconstructed and the goodness of fit of our model is obtained.
2) Classification: only candidates with $P_{\mathrm{q}} > 0.01$ are fed to the classifier. Candidates that are labeled as quasars by the classifier can then be further analysed via visual inspection.



(a) Redshift estimates of training set 1.

(b) Redshift estimates of training set 2 ($P_{\mathrm{q}} > 0.01$).

Figure 8.2: Histogram of redshift estimates for a) training set 1, b) training set 2 ($P_{\mathrm{q}} > 0.01$).
a) The candidates have estimated redshifts between $\hat{z} = 5.60$ and $\hat{z} = 9.00$. Most sources (quasars and contaminants) have a predicted quasar redshift of $z \sim 5.6$.
b) The range of estimated redshifts is between $\hat{z} = 5.60$ and $\hat{z} = 8.60$. The requirement in P_{q} mostly gets rid of quasars around $z \lesssim 5.6$. This is because the template flux is zero or close to zero in u, g, r and hence candidates with non-zero flux in those bands will have a low quasar probability. Another peak of the visually rejected candidates is at $z \sim 7$. This suggests that many contaminants look similar to quasars at $z \sim 7$.

8.1 Training data

We start with the same set of $\sim$ 4300 labelled sources from Mortlock et al. (2012) as described in Section 7.4.2. The survey data is explained in Section 7.4.1.

The difference here is that we use the set from 7.4.2 as the training set instead of the validation set. This is because for the two-metric threshold classifier we only needed a validation set to validate our method, but no training was required. In machine learning classifiers, however, training data is needed for the classifier to learn the classification function.

8.1.1 Training set 1

Training set 1 consists of the complete candidate set. The distribution of derived redshift estimates can be seen in Fig. 8.2a. The reason for the large number of candidates in the first redshift bin is because `Pq_server` only works for redshifts above 5.6. Hence, any quasar that has a redshift below that limit will be thresholded into a predicted redshift of $\hat{z} \geq 5.6$. Similarly, any contaminant gets a predicted redshift that fits best to the quasar SED templates. In this case, most contaminants look similar to quasars at redshift $z \sim 5.6$.

8.1.2 Training set 2

Training set 2 consists of candidates with a quasar probability of $P_q > 0.01$. This is because even though some candidates were visually classified as accepted, they are much more likely to be contaminants. Thus we would not consider these candidates for follow-up photometry. This constraint reduces the training set to 449 candidates, out of which 159 are visually accepted candidates. The smaller sample set will make training more difficult, however, overall the visual classification labels will be more accurate than before. We also could have relabelled the candidates with $P_q \leq 0.01$ as rejected, however then we could not have used P_q as a feature in the classifier since it would have quickly figured out the user-defined threshold. Indeed, using 10-fold cross validation, this gives AUC $= 0.99 \pm 0.01$ as ~ 4000 candidates can correctly be rejected by re-discovering the P_q threshold. The cut in P_q removes a large quantity of estimated low-redshift candidates, as can be seen in Fig. 8.2b, on which an SED template with zero or close to zero flux in u, g, r will not be a good fit. After the cut in P_q has been applied, we can see that another peak appears at $z \sim 7$ for the visually rejected candidates. This suggests that a significant proportion of contaminants look similar to quasars at $z \sim 7$ compared to redshifts $z \sim 6.5$.

8.1.3 Purity of the training set

The visual classification by Mortlock et al. (2012) contained information on why candidates had been rejected, including issues such as cross-talk, detector flaws, diffraction spikes, satellite trails and persistence. As discussed above, one of the reasons a source is visually rejected is if it has at least one of those problems. This implies that if survey images of an actual quasar have data processing issues, we will miss out on that quasar. The data processing issues of UKIDSS are explained in detail in Dye et al. (2006). Figure 8.3 shows their analysis of some of the data processing issues that can arise in the WFCAM of UKIDSS.

For training set 2, around 100 candidates have some form of processing issues in at least one passband. This is nearly a quarter of our sample set! This gives a purity of about 78% for training set 2. For training set 1, we obtain a higher purity of 86%. The difference between training set 1 and 2 is likely because the probability code itself does not take into account any morphological information but rather only flux values. As long as the input fluxes are close to one of the 9 SED templates, the probability P_q will be high. Hence, the requirement that $P_q > 0.01$ does not get rid of candidates with data processing issues.

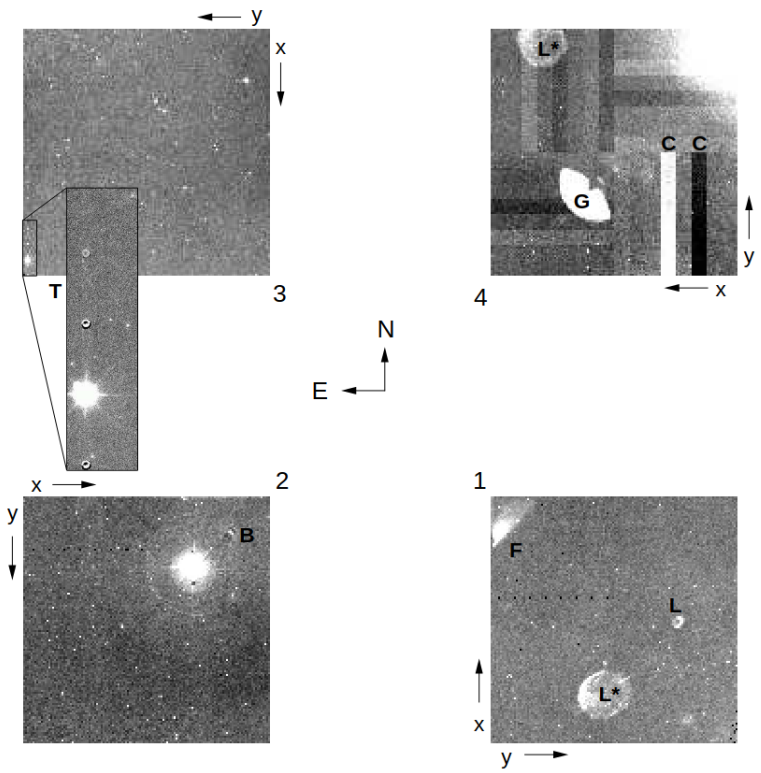


Figure 8.3: Data processing issues of UKIDSS' WFCAM. B: 'bow-tie' feature in the halos of bright stars; C: channel bias offsets; F: flare persistence; T: cross-talk image due to a bright star, G: bright ghost due to the moon shining directly onto the field lens, L: due to marks/dust on the field lens, L*: marks due to the lens handling jig. Figure taken from Dye et al. (2006).

8.2 Classifiers

We test our candidates with two different classifiers, which we train on our pre-labelled set as described in Sec. 8.1. The features extracted and summarised in the data processing pipeline (see Table 6.2) serve as the input for the classifiers. First, we implement a simple decision tree of depth 4 in order to interpret the results and identify important features. It further helps us to apply user-defined thresholds for our two-metric threshold classifier. To optimise the classification we make use of a random forest with balanced weights for each class. This gives us a much better classification result, with the caveat of being less interpretable.

All of these methods can be used for upcoming surveys such as Euclid, which will enable discoveries of fainter quasars and at even higher redshifts (Laureijs et al., 2011; Euclid Collaboration, 2019).

8.2.1 Feature sets

We test two different sets of features for the classifiers. In feature set A we only include the summary statistics (goodness of fit and the quasar probability) to test how well these can interpret the results on their own. In feature set B we include most of the available information from pipeline 1, thus including the PSF photometry. The two feature sets that we use for the classification are summarised in Table 8.1.

8.2.1.1 Feature set A

The goal for feature set A is to construct a set of features that are independent of redshift and passband. Thus, the set is supposed to be survey-independent. In reality, optimizing the features for each survey individually and for a given redshift range will always give better results. For example, as we discussed in the previous chapter, the chi-squared values will likely increase for upcoming surveys since the photometric errors will decrease. Therefore, using the trained classifier on those new dataset might not give as good results. However, one could offset all the chi-squared values by a certain amount to try to circumvent having to re-train the model. Re-training itself is very fast, it is more that it requires time to visually classify a set of sources, where the sources have to be carefully selected as well.

The only features we are using are the reduced average chi-squared value $\tilde{\chi}^2$, the overall maximum chi-squared value of all runs over all passbands $\tilde{\chi}^2_{\mathrm{max}}$, the reduced average chi-squared value where the centre has been fitted plus its maximum, and the quasar probability P_{q}. That is, $\tilde{\chi}^2$,$\tilde{\chi}^2_{\mathrm{max}}$,$\tilde{\chi}^2_{\mathrm{pos}}$,$\tilde{\chi}^2_{\mathrm{pos,max}}$ and P_{q}. This is our feature set A.

Set	Features
A	$\{\tilde{\chi}^2, \tilde{\chi}^2_{\rm max}, \tilde{\chi}^2_{\rm pos}, \tilde{\chi}^2_{\rm pos,max}, P_{\rm q}\}$
B	$\{\tilde{\chi}^2, \tilde{\chi}^2_{\rm max}, \tilde{\chi}^2_{\rm pos}, \tilde{\chi}^2_{\rm pos,max}, P_{\mathbf{q}}, \{\chi^2_b, \mu_{0,b}, \sigma_b, F_b, \sigma_{F_b}, {\rm SNR}_b\}_{b\in B}, \hat{z}\}$

Table 8.1: Feature sets A and B that are used for classification.

We first used the individual chi-squared values for each band $\{\chi^2_b\}$ as well. However, this made the results more redshift-dependent. For instance, in our sample set there are many candidates at redshift $z \simeq 6$, which will drop out in the r-band. The templates used in `Pq_server` always give zero flux in u, g, r. Hence, any flux detected in those three passbands will result in a large reduced average chi-squared value. Therefore, χ^2_r is one of the most important features. In reality, our goal is to find quasars at $z > 6.5$. Future surveys such as Euclid will have the capacity to detect ~ 10 quasars at $z \gtrsim 8$ (Euclid Collaboration, 2019). At such large redshifts the Lyman-break shifts further into the NIR band. Therefore the band that encompasses the most important features will change.

Instead, we want to find any quasar above redshift 6.5, and so it should be possible to detect a $z \sim 6.5$ quasar and a $z \sim 8$ quasar in the same instance of a classifier. In addition, future surveys will have different filter sets of different wavelengths and widths and therefore will be different from the SDSS and UKIDSS bands. We therefore opt for a more redshift and passband independent feature set. However, as mentioned above, for surveys with very different magnitude limits and photometric properties compared to SDSS/UKIDSS, re-training is still advised.

8.2.1.2 Feature set B

In comparison to feature set A, feature set B contains redshift and passband dependent data. That is, we include flux information F_b, χ^2_b values and background pixel flux $\mu_{0,b}, \sigma_b$ for all passbands b, in addition to the predicted redshift $\hat{z}$. We expect this feature set to give better classification results than feature set A as it includes more useful information than set A.

8.2.2 Decision tree

With the aim of being able to interpret our results, we first test our candidate list on a DT of depth $d = 4$. The mechanisms of a DT are explained in detail in Section 5.4. The difference in classification performance between $d = 3$ and $d = 4$ was significant, whereas the difference between $d = 4$ and $d = 5$ was not, leading us to use a depth of 4. Further, a larger depth would have drastically decreased interpretability.

The results of the DT are then used to find the most important features and to possibly find a good threshold for the two-metric threshold classifier.

8.2.3 Random forest

Next, we choose a random forest classifier, which is more complex than a DT. The RF is explained in detail in Section 5.5. The RF increases the accuracy of our classification results due to the added complexity. However, the RF will also make the results less interpretable. The RF will be survey-dependent and thus will need to be trained on a different training set for future surveys. However, it will increase the true positive rate and decrease the false positive rate compared to the DT.

8.2.3.1 Reducing the complexity of the random forest

We optimize the parameters of the random forest (maximum number of trees, minimum number of samples required to split a node etc) via analysing the stratified cross-validated classification results for a range of values that cover the parameter space. We circumvent the problem of our unbalanced dataset by giving a higher penalty for misclassifying quasars and by undersampling the majority class during training. In order to find a minimal feature set out of a total of 70 features, we try out different dimensionality reduction mechanism such as PCA with different kernels and use diverse feature importance measures (see Section C.3).

8.2.3.2 Tuning the parameters

In order to avoid overfitting, we tune the parameters of each classifier for each feature set. This is done by observing how the AUC score evolves for the training and validation set over a range of parameter sets. By iteratively increasing the complexity of the classifier, there will be a point where the AUC score increases for the training set but decreases for the validation set. Let us assume this is at iteration $n + 1$. The parameters of the previous iteration n then give the optimal parameter set for the classifier. Further increasing the complexity would overfit the classifier, whereas decreasing the complexity would underfit it. The parameter set corresponding to iteration n gives the sweet spot in between.

8.2.3.3 Minimal feature set for a maximum classification score

In addition, we would like to find the minimal feature set such that the performance of our classifier does not change. This further diminishes overfitting and removes any unnecessary complexity. We try different techniques in order to find the most important features of our classifiers.

First, we use PCA (see Section C.3.2) that projects the features on the eigenvectors with the largest eigenvalues. In order to model non-linearities, we try out different kernel projections such as the radial basis function (RBF) kernel, a polynomial and a cosine kernel.

Next, we use the drop-column feature importance measure (see Section 5.5.3.3). However, since a subset of our feature set is correlated we do not expect this to work well. We could manually remove any redundant features but that would defeat the purpose of using an algorithm that is supposed to give the most important features.

We then try further feature importance measures including permutation importance (see Section 5.5.3.2) and mean decrease impurity (MDI) (see Section 5.5.3.1).

Finally, we use a conditional random forest with the conditional permutation importance measure as described in Strobl et al. (2008) and Section 5.5.3.4. The RF favours features that have more possible cuts like (countably) inifinite sets do. Continuous variables are inifinite sets and therefore are more likely to get selected as an important feature than variables with finite values. Our feature sets consists mostly of continuous variables, however one could argue that the redshift estimate $\hat{z}$ is not continuous as it has been calculated at at finite step size of 0.01. We therefore expect the previous importance measures to disfavour the potentially important $\hat{z}$ feature, whereas the conditional random forest should remove that bias.

We make use of the `party` package in R which includes `cforest`, an implementation of the conditional random forest, and a conditional permutation importance measure `varimp` (Hothorn et al., 2006; Strobl et al., 2009). We further use the `varImp` package to use `varImpAUC`, a conditional permutation importance measure based on the AUC score for unbalanced data sets (Janitza et al., 2013).

8.2.3.4 Balancing the dataset

In order to tackle an imbalanced dataset, one can either undersample the majority class or upsample the minority class. The former is done by the `BalancedRandomForestClassifier` of the Python package `imblearn` (Lemaitre et al., 2016). The latter can be achieved by creating new samples of the minority class. We use the Python package `SMOTE` (Chawla et al., 2011) that uses k-nearest neighbours of the minority class to create artificial data. We use it to upsample the quasar (minority) class for feature set A in the training set. This is done separately for each iteration in the stratified 10-fold cross validation. It is not used for the whole dataset in order to not create any overoptimism in the classification results (Blagus & Lusa, 2015; Santos et al., 2018). It is further not used for feature set B as k-nearest neighbours does not work well in high dimensions (Weber et al., 1998).

In addition to using the `BalancedRandomForestClassifier` to undersample the majority class, we also use it to balance the weights in the loss function of the classifier such that the classifier is given a larger penalty for misclassifying any of the candidates in the quasar sample set. This is because we

do not want to miss out on any possible quasar candidate, even if that means that some non-quasars will contaminate the results.

One should note that these are artificially created positive candidates and are not a function of the quasar's SED templates or of the astronomical images. Hence, using k-nearest neighbour could create samples that do not actually represent quasar candidates but rather contaminants. In our case the performance was improved by these steps, but it might be worth considering to change this approach in future.

8.2.3.5 Testing against artificial positive candidates

Finally, we generate artificial postage stamps that follow the SED of a quasar, as is described in Section 7.3.1. These artificial candidates are then run through the pipelines and form a separate testing set. We expect the majority of these generated candidates to be accepted as possible quasar candidates. Since the templates are scaled on the $J-$band flux, an unrealistic large/small flux in J can result in a low quasar probability. However, the chi-squared values should be small. These artificial candidates help us to find weaknesses in our classifiers.

8.3 Results

We describe how well our classifier can reproduce and enhance the results by Mortlock et al. (2012). It is important to point out that the labels are not ground truths except for the 14 spectroscopically confirmed quasars in the set. The human classification error in the ImageNet Large Scale Visual Recognition Challenge was estimated to be 5.1% (Russakovsky et al., 2014). Due to the complexity of quasar classification we expect an even larger error rate in this dataset. Thus, for our training set 1, it is not desirable that the classifier reaches an accuracy of 100% since this would indicate overfitting. For our training set 2, however, candidates with a low quasar probability $P_{\mathrm{q}} \leq 0.01$ have been removed and as such we expect the error rate to decrease. This is because we assume that most errors will have been made for candidates that look like quasars but are actually much more likely to be contaminants.

8.3.1 Performance measure

As the number of positive samples in our dataset is very small, just classifying nearly every candidate as a contaminant (non-quasar) would already give a high accuracy, giving us a false sense of the effectiveness of our classifier. We therefore need a different performance measure, one that shows

how many of our quasar candidates have correctly been accepted and how many have been incorrectly rejected.

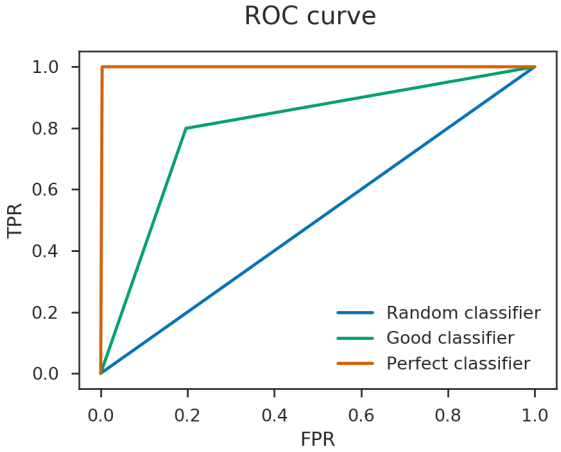


Figure 8.4: A Receiver Operating Characteristic (ROC) curve. The x-axis shows the False Positive Rate (FPR), and the y-axis the True Positive Rate (TPR). In blue a model that assigns random labels, giving an AUC score of 0.5. Orange shows a good classifier with AUC > 0.5 and green a perfect classifier with AUC = 1.0.

Hence, we use ROC AUC score as our performance measure. AUC stands for 'area under the curve', and ROC stands for 'receiver operating characteristic curve'. Therefore, ROC AUC is the area under the receiver operating characteristic curve. The ROC AUC is the integral over the ROC curve that plots the true positive rate (TPR) vs. the false positive rate (FPR) for a range of classification thresholds. The TPR and FPR rates are defined as

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{P}} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{8.1}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{N}} = \frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}}, \tag{8.2}$$

where N (P) the actual number of negative (positive) samples and FN (FP) the number of positive (negative) samples that have been falsely classified as negative (positive) by our model. A classifier that assigns random labels has an AUC score of 0.5. A perfect classifier has AUC = 1. This is illustrated in Figure 8.4. A classifier can have a high accuracy but a low AUC score if e.g. the fractional number of quasars that have been accepted is low.

8.3.2 Classification results

The results for the RF, DT and two-metric threshold threshold classifier (from Chapter 7) on training set 1 are shown in Table 8.2, where the results for the RF and the DT were obtained using 10-fold stratified cross validation. The results are shown for feature set A and feature set B. The cross-validation results for training set 2 are shown in Table 8.3. The tuned parameters of the RF are shown in Table 8.4.

	AUC	
Classifier	Feature set A	Feature set B
RF	0.82 ± 0.02	0.93 ± 0.01
DT	0.79 ± 0.03	0.87 ± 0.02
Two-metric	0.57	/

Table 8.2: The classification results (AUC score) for the random forest, the decision tree and the two-metric threshold classifier on training set 1. For the RF and the DT 10-fold stratified cross validation has been applied. The second column shows the AUC scores using the survey-independent feature set (set A). The third column shows the AUC scores for the complete feature set (set B).

	AUC	
Classifier	Feature set A	Feature set B
RF	0.83 ± 0.06	0.94 ± 0.04
DT	0.80 ± 0.05	0.87 ± 0.06
Two-metric	0.73	/

Table 8.3: The classification results (AUC score) for the random forest, the decision tree and the two-metric threshold classifier on training set 2 ($P_q > 0.01$). For the RF and the DT 10-fold stratified cross validation has been applied. The second column shows the AUC scores using the survey-independent feature set (set A). The third column shows the AUC scores for the complete feature set (set B).

The best results are achieved with the RF using training set 2 and feature set B, giving an AUC score of 0.94 ± 0.05. Intuitively it makes sense that using training set 2 gives better results, since a large fraction of the visually accepted candidates had a low quasar probability. The ROC curve for different instances of the classifiers for training set 2 and both feature sets are shown in Fig. 8.5, together with the classification threshold that is furthest away from the diagonal. The confusion matrix of the DT and RF for the corresponding thresholds can be seen in Figures 8.6 and 8.7. A low number of FN is desired as we do not want to miss out on any quasar candidate. The first row defines the pool of negative candidates (N), and the second row the good candidates (P). The diagonal elements correspond to the correctly classified candidates (TN and TP). The off-diagonal elements correspond to the incorrectly classified samples (FN and FP). The thresholds can be chosen along the ROC curve depending on the level of completeness and consistency one would like to achieve. We are willing to pay the price for a slightly larger amount of FP if we can keep the amount of FN low. Compared to training set B, the number of false positives and false negatives would approximately double for the RF applied on training set A.

The redshift histogram of the false positives and false negatives is shown in Figure 9.6. The FP and FN have been obtained by running 20 random forest classifiers and collecting the FP and FN of their respective test sets. In a), we do not remove any duplicates. In b), we do remove the duplicates. In both figures, we can see a significant peak at redshift bin $z \sim 5.5$ for the false negatives. This is likely due to the face that a significant proportion of our input candidate list is predicted to be at around that redshift. It is also better to miss out on quasars that are at the lower redshift end of our classification compared to higher redshift ones. There is another strong peak at $7 \leq z < 7.5$ when we

Parameter	Value
max_depth	10
class_weight	'balanced'
criterion	'gini'
bootstrap	True
max_features	'sqrt'
min_samples_leaf	4
min_samples_split	2
warm_start	False
n_estimators	500

Table 8.4: Tuned parameters of the RF as described in Sec. 8.2.3.2, implemented via the `BalancedRandomForestClassifier` of the Python package `imblearn` (Lemaitre et al., 2016).

do not remove the duplicates, which means that there are a few high-redshift quasar candidates that are often misclassified. This peak is reduced when the duplicates are removed. For the false positive candidates, the predicted quasar redshift is meaningless since these sources are no quasars. But it is still interesting in which predicted quasar redshift bins our sources of contamination lie. Here, there are peaks at the around the same redshifts as for the false negatives, but they are greatly reduced and are not significant.

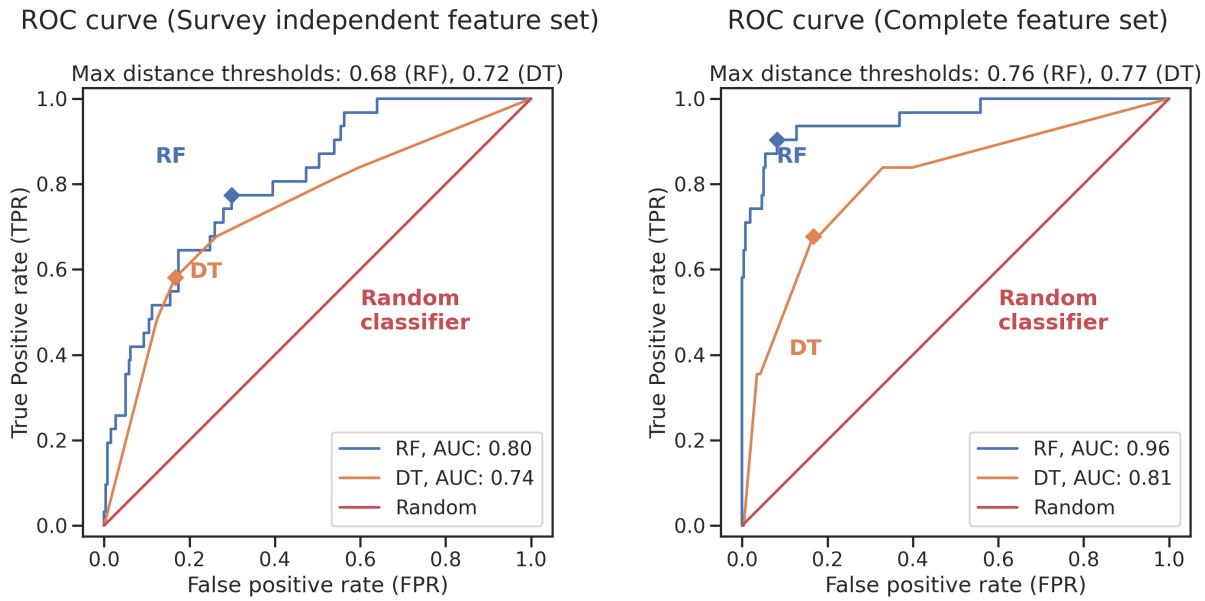
The RF outperforms the other classifiers on both feature sets and on both training sets. The DT consistently has a lower AUC score than the RF, although the performance of the DT on feature set B is better than the performance of the RF on feature set A. The two-metric threshold classifier has the lowest classification performance in both training sets but in return has reproducible and interpretable results.

Looking at the summary results in Tables 8.2 and 8.3, there is about a 10% difference in the AUC score between using feature set A and feature set B. This indicates that feature set A already provides the most important information, but that having more redshift-dependent information in feature set B will always improve the classification results.

Further, there is about a 1% increase in performance for the machine learning classifiers in training set 2 compared to training set 1. The two-metric threshold classifier, however, increases substantially by 16% from AUC = 0.57 to AUC = 0.73. This indicates that

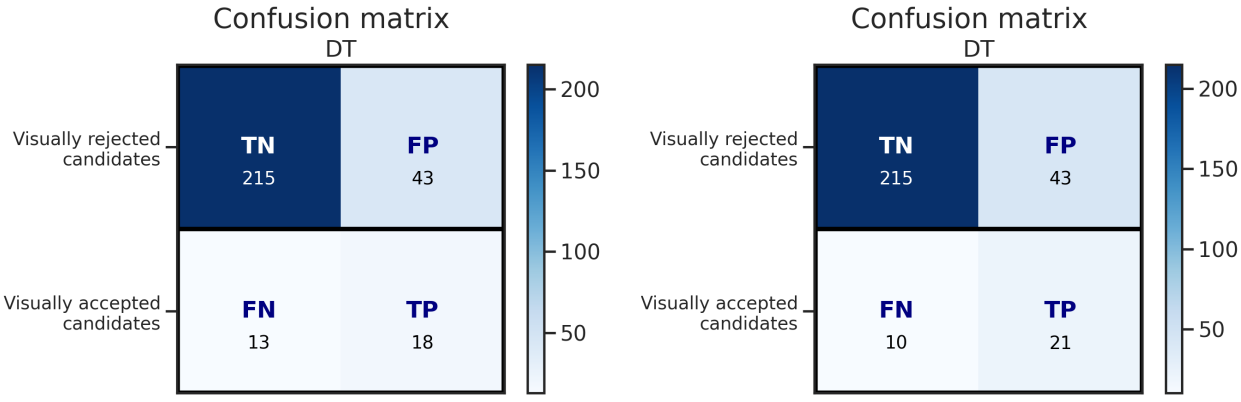
- Most of the visual misclassifications occur at low quasar probability. Thus, the accuracy of the visual labels in training set 2 is higher than for the labels in training set 1.
- The substantial decrease in training candidates from training set 1 to training set 2 makes training the machine learning classifiers (DT, RF) more difficult. Thus, for training set 2, the effect of increased visual labelling accuracy is outweighed by the decrease in candidates.

We pre-train the RF on a set that contains sources from training set 1 but not any sources from training



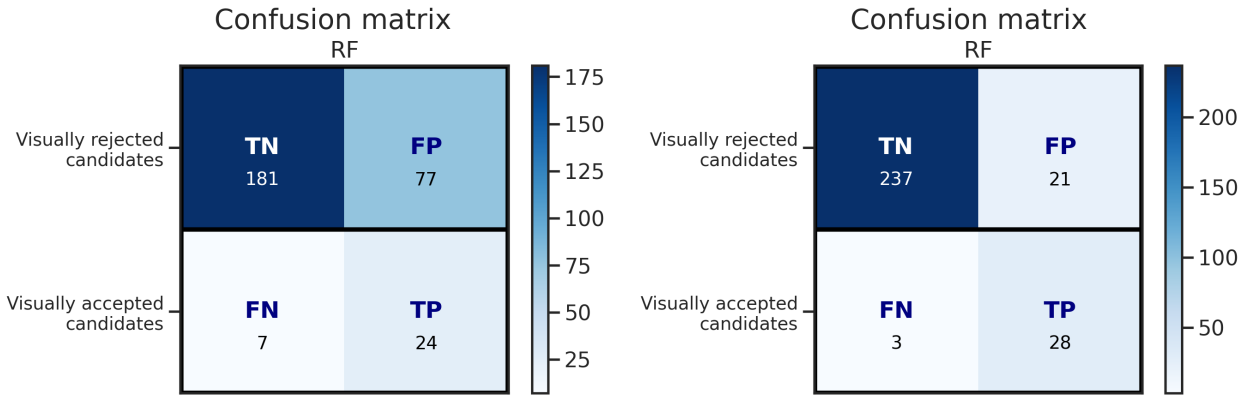
(a) Feature set A (b) Feature set B

Figure 8.5: ROC curve for an instance of the random forest in blue and the decision tree in orange using training set 2 (consisting of candidates with $P_q > 0.01$) with a) Feature set A b) Feature set B. In red a classifier that assigns random labels (AUC = 0.5). The RF gives the best classification performance with feature set B with an AUC score of AUC = 0.96. The DT gives AUC = 0.88. The classification thresholds that achieve the greatest distance from the diagonal are marked on the curves.



(a) Feature set A (b) Feature set B

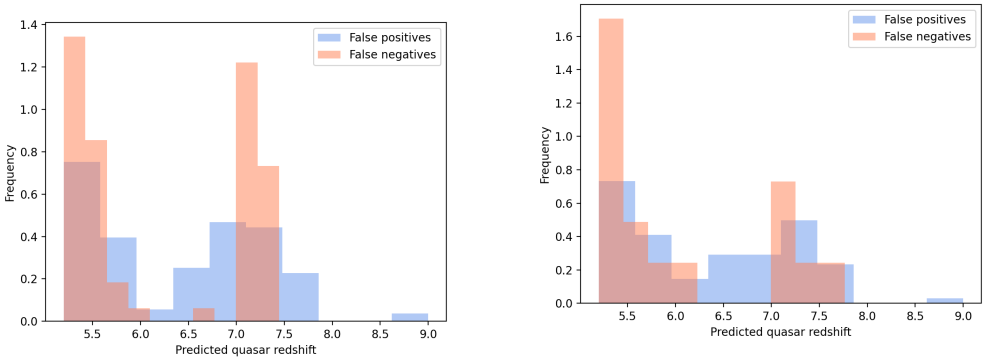
Figure 8.6: Confusion matrix for the DT threshold as in Fig. 8.5 for a) Feature set A and b) Feature set B. While the diagonal represents the correctly classified candidates, the off-diagonal elements are incorrectly classified candidates. The top row represent the overall visually rejected candidates and the bottom row represent the visually accepted candidates.



(a) Feature set A

(b) Feature set B

Figure 8.7: Confusion matrix for the RF thresholds as in Fig. 8.5 for a) Feature set A and b) Feature set B. While the diagonal represents the correctly classified candidates, the off-diagonal elements are incorrectly classified candidates. The top row represent the overall visually rejected candidates and the bottom row represent the visually accepted candidates.



(a) Predicted quasar redshifts for a set of ~ 600 false positives and ~ 80 false negatives. Here, we do not remove any duplicates.

(b) Predicted quasar redshifts for a set of 90 false positives and 16 false negatives. These are the same set as in a), but with duplicates removed.

Figure 8.8: Predicted quasar redshifts for a set of false positives and false negatives. These have been obtained by accumulating the false positives and false negatives of each test set of 20 random forest classifiers which each have been trained on a random subset of the candidate list. In a), the list of FP and FN contain duplicates since some of the test sets of the individual classifiers will overlap. In b), duplicates are removed. Note that the predicted quasar redshift is not meaningful for the false positive sources.

set 2 (the algebraic difference between the two) to see if the performance increases by including more candidates. We set the classification labels of the pretrained set to '0', although we leave a handful of positive candidates such that the classifier can actually train on the data. However, we do not see an enhanced performance from pretraining. This is likely because the pretrained data is not representative of training set 2 and thus does not help in the classification performance.

Independent of the feature and training set, the RF and DT correctly classify all of the 14 confirmed quasars in the set. This is discussed in Sec. 8.3.4.

In addition, 100% of our generated mock quasars are accepted by the classifiers. The reduced chi-squared value of the generated quasars is with $\tilde{\chi}^2 = 0.997$ close to 1 as expected. This means that the quasar candidates in training set 2 that get rejected by the classifier either do not follow any of the 9 SED templates or cannot be reconstructed with the PSF of the specific survey.

8.3.3 Dimensionality reduction

As many of our features are correlated, there is no unique minimum set that achieves the same result as our baseline performance (see Table 8.2 and 8.3). Thus, it is not possible to rank all the features by their importance. However, some features will be present in all minimum feature sets, which can be used as a measure of their significance.

We run the permutation feature importance measure on the DT classifier and one of the feature sets we obtain is $\{\tilde{\chi}^2_{\text{max}}, \mu_r, \text{SNR}_z, F_J, F_K\}$, with the first feature in the set being the most important feature and decreasing in importance after that. Running the DT classifier with just these 5 features gives us our baseline DT performance of AUC = 0.87. We then plot several DT to make sure that the cuts on these features are sensible. We note that at least one of the chi-squared summary values is needed in all of the feature sets to give AUC = 0.87, indicating that these are part of the most important features.

For the RF with training set 2 we tested several different methods in order to reduce the number of features used by the classifier. Firstly, using linear PCA, we need the 12 eigenvectors that correspond to the largest eigenvalues to explain 95% of the total variance in the data. This gives us an AUC score of AUC = 0.83 ± 0.12, about 10% less than our baseline performance. Using 20 features to explain 99% of the variance gives AUC = 0.84 ± 0.07, only slightly better than before. This indicates that either only a subset of our candidates are linearly separable or that some of the important features have very low variance and thus are not included.

Using again 20 eigenvectors but with a cosine kernel projection, we retrieve AUC = 0.88 ± 0.09. Using a polynomial kernel we obtain AUC = 0.89 ± 0.11. This is an improvement from linear PCA, suggesting that we do need non-linear components in our feature set. However, it is still below our baseline performance of around AUC = 0.94. Using more components does not significantly improve the performance of the RF. This means that some information is lost by using eigenvectors as our

feature set. Consequently, none of the transformations as described in 8.2.3.3 can linearly separate our two classes.

With the built-in MDI `scikit-learn` random forest importance measure we achieve AUC = 0.90 ± 0.08 with the 20 most important features. The top 5 ranked features are $F_z, \mathrm{SNR}_z, \tilde{\chi}^2, \tilde{\chi}^2_{\mathrm{max}}$ and χ^2_Y in decreasing order. This seems plausible as the Lyman-alpha emission line will occur in Y for $z \sim 6.5$, and the Gunn-Peterson trough should be 'visible' in z and the bands below. We obtain similar results when applying the permutation importance measure, where the redshift estimate has been selected as the most important feature.

We have seen that by either compressing or down-scaling the data in training set 2 to 20 features, we decrease the AUC score by up to 10%. This is about the same decrease as when we use feature set A, which consists of only 5 features. In those cases it therefore makes sense to use feature set A instead.

The best results are achieved by using the conditional random forest combined with the conditional feature importance measure, where the 10 most important features can be seen in Fig. 8.9. The fact that 5 of our chi-squared measures are selected in the top 6 features gives us confidence that our model is a good way to test if a source is a quasar. $\tilde{\chi}^2_{\mathrm{pos,max}}$ is listed as the most important feature. The idea behind $\tilde{\chi}^2_{\mathrm{pos,max}}$ is that if the maximum chi-squared value is large even when the position is fitted, then that source must be a bad candidate. In return, if $\tilde{\chi}^2_{\mathrm{pos}}$ is much smaller than $\tilde{\chi}^2$ then this would imply that the source has moved. $\tilde{\chi}^2_{\mathrm{max}}$ can detect outliers in the sense that an SED might be a great fit in all bands except for one, where the fit is really bad. The reduced chi-squared value might still be relatively small, whereas $\tilde{\chi}^2_{\mathrm{max}}$ will be large and thus this source would be rightly rejected.

Using these 10 features (Fig. 8.9) we already achieve a classification score of AUC = 0.92 ± 0.05. Using the top 5 features $\{\tilde{\chi}^2, \tilde{\chi}^2_{\mathrm{pos}}, \tilde{\chi}^2_{\mathrm{max}}, \tilde{\chi}^2_{\mathrm{pos,max}}, F_z\}$ gives us AUC = 0.89 ± 0.06. Hence, using flux in the z band, F_z, instead of the quasar probability P_{q} in feature set A gives a 7% increase in the classification score (see Table 8.2).

We do not see any significant difference between the conditional permutation importance measure based on the mean decrease in accuracy or the AUC score. We assume that this is due to the fact that training set 2 is balanced enough for the positive samples to have an impact on the accuracy. We still recommend to use `varImpAUC` (with `conditional=TRUE`) in future settings.

As a result, we cannot find a small subset of features that gives us back the same classification score as when we use all features. We therefore conclude that all of our features contribute to some extent to the overall result and that none of our features is redundant. We see that only 10 features are required to get an AUC score of ~ 0.92, but that for the last 2 percent we need the remaining features. Some of the sources that the remaining features can capture are outliers in the sense that they have a low (large) reduced chi-squared value but actually are visually rejected (accepted). And essentially if the flux and flux errors are known in all passbands, the random forest is able to approximately reconstruct

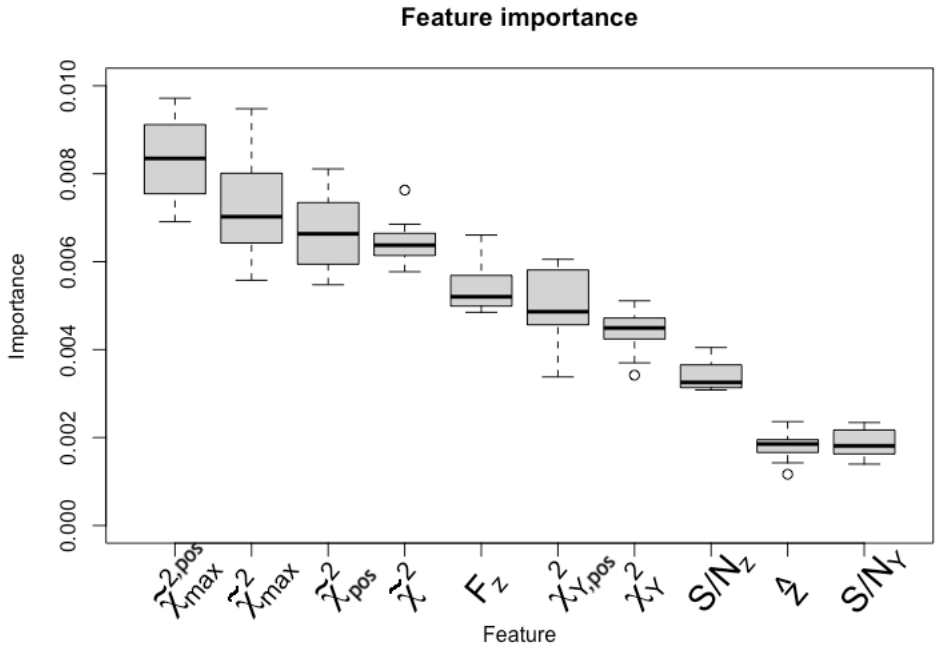


Figure 8.9: The 10 most important features for feature set B and training set 2, as obtained by the conditional permutation importance measure of 10 instances of a conditional random forest. The lines in the grey boxes represent the median values. The grey boxes themselves represent values between the 25th percentile and the 75th percentile. This box contains 50% of the data, and its length along the y-axis is called the interquartile range (IQR). The lines emerging from the box are the so-called 'whiskers'. The end of the lines represent the minimum and maximum points within 1.5× IQR from the grey box. Points exceeding that are marked as dots.
The most important feature is given by $\tilde{\chi}^2_{\text{pos,max}}$, followed by $\tilde{\chi}^2_{\text{max}}$. The top 4 features are all redshift-independent which is a sign that our method can distinguish well between quasars and non-quasars. The features of passbands z and Y are important as they indicate the boundary between the Lyman-alpha emission line and the Gunn-Peterson trough. E.g., you would expect a large flux in z for redshifts $z \lesssim 6$, whereas you would expect a low flux in z and a large flux in Y for redshifts $z \gtrsim 7.0$. The overall small importance values indicate that the features are correlated as expected.

the SED templates. In addition, any feature that helps to detect the Lyman-alpha emission line and the Gunn-Peterson trough will help in the identification of quasars.

8.3.4 Discussion

We show a subset of sources that had been accepted via the original visual inspection of (Mortlock et al., 2012) but rejected by our classifier and vice versa. These exemplify the limits of our method but also its strengths when it correctly flags candidates that originally had been assigned incorrect labels.

8.3.4.1 Limitations of this method

The RF and DT correctly classify all confirmed quasars in the set, independent of which feature set is chosen. However, there is a subset of candidates that get misclassified. We only show misclassifications from the RF since that classifier is more accurate than the DT.

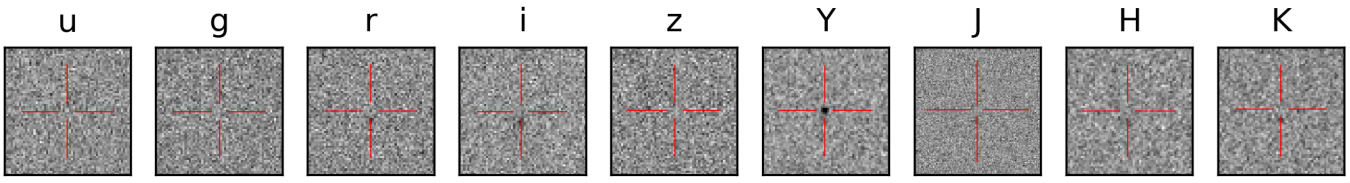
Examples of false negatives We show some representative examples of candidates that were accepted by visual inspection but were rejected by our ML classifier.

An example of a false negative can be seen in Figure 8.10. This source neither looks like a typical quasar nor a contaminant, and hence gives a large maximum residual chi-squared of $\tilde{\chi}^2_{\rm max} = 4.36$. However, the model of a high-redshift quasar fits better than that of the contaminants, which is confirmed by a high quasar probability of $P_{\rm q} = 0.98$. It is possible that this is a moving source that has moved outside the postage stamp, in which case the rejection by the RF would be correct.

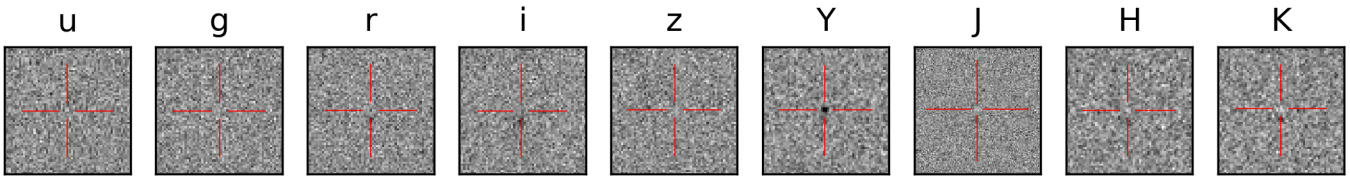
Another example can be seen in Figure 8.11. This source was rejected by the RF as the central source is so bright that it cannot be modelled by our point spread functions anymore. This and the difference between predicted and observed fluxes lead to a high residual chi-squared value of $\tilde{\chi}^2 = 217.37$.

Some candidates will be rejected based on one 'bad' run of a source, e.g. where the photometry is unusable. We mostly mitigate this effect by discarding images with any data processing issues, but images with processing issues that have not been flagged by either the database or our pipeline will make the residual chi-squared values larger. One could also remove runs that significantly deviate from the other runs, e.g. large deviations in magnitude, as this would be an indication that the photometry of that run is wrong rather than the whole object being discarded. However, it could also indicate a moving source in which case we do want to keep the images.

Examples of false positives Using our RF, the number of FP always outnumbers the number of FN since we penalise the classifier more for missing a quasar compared to classifying a contaminant as a quasar. We show several examples of likely false positives.

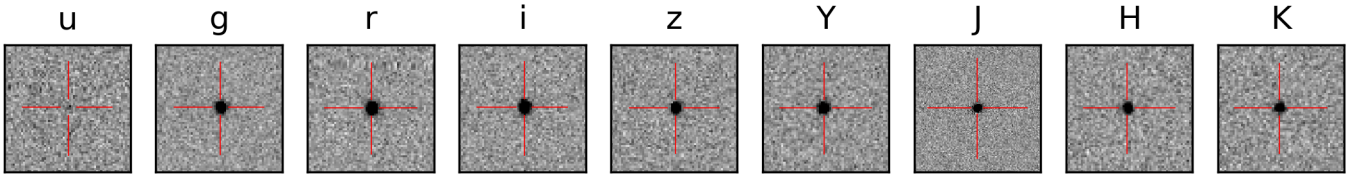


(a) Original

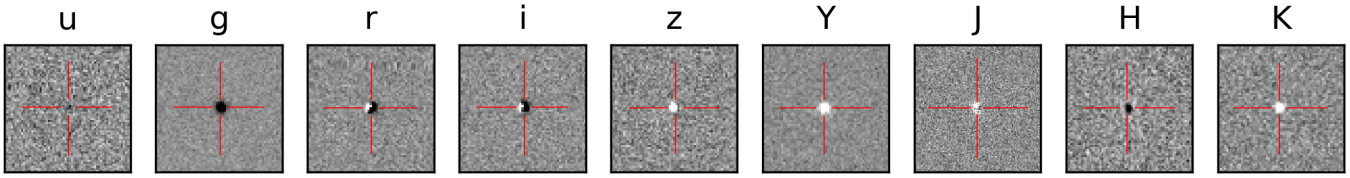


(b) Residual using the best-fit quasar SED

Figure 8.10: Candidate J000405.89−005342.6 has a positive visual classification label but often gets rejected by the RF. a) Original image data in SDSS and UKIDSS LAS, b) Residual using the best-fit quasar SED. It has a high quasar probability of $P_{\mathrm{q}} = 0.98$, but also a large maximum chi-squared value of $\tilde{\chi}^2_{\mathrm{max}} = 4.36$.



(a) Original



(b) Residual using the best-fit quasar SED

Figure 8.11: Candidate J232822.85+151813.3 has a positive visual classification label but often gets rejected by the RF. a) Original image data in SDSS and UKIDSS LAS, b) Residual using the best-fit quasar SED. This source has a an average and maximum residual chi-squared value of $\tilde{\chi}^2 = 217.37$ and $\tilde{\chi}^2_{\mathrm{max}} = 1070.73$, and hence is rejected by the RF, DT and two-metric threshold classifier. This is due to a difference in observed and predicted flux, and due to a source shape that we cannot properly capture with our point spread function in neither SDSS nor UKIDSS.

For instance, candidates with cross-talk artifacts that are not necessarily visible to the naked eye sometimes do get accepted by the classifier when they should not. Cross-talk processing issues have an effect on the measured photometry of a source and hence paints the observed fluxes unreliable. Examples of candidates with cross-talk issues that are accepted by our machine learning classifiers are shown in Figures 8.12 - 8.13. J033404.44+001003.4 in Figure 8.12 has a very low quasar probability due to the dropout in the optical. However, our ML classifiers sometimes accept this candidate. In comparison, the candidate in Figure 8.13 has a high quasar probability of $P_{\mathrm{q}} = 1.00$ at predicted redshift $\hat{z} = 5.50$, and low residual chi-squared values of $\tilde{\chi}^2 = 1.16$ and $\tilde{\chi}^2_{\mathrm{max}} = 1.86$. However,

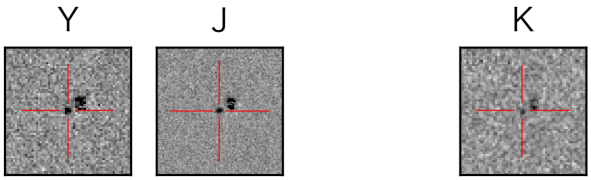
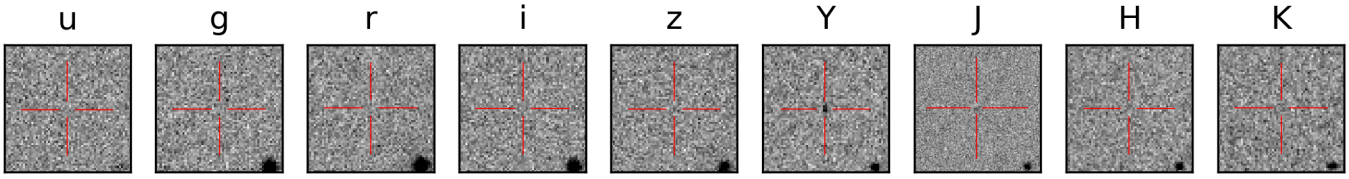
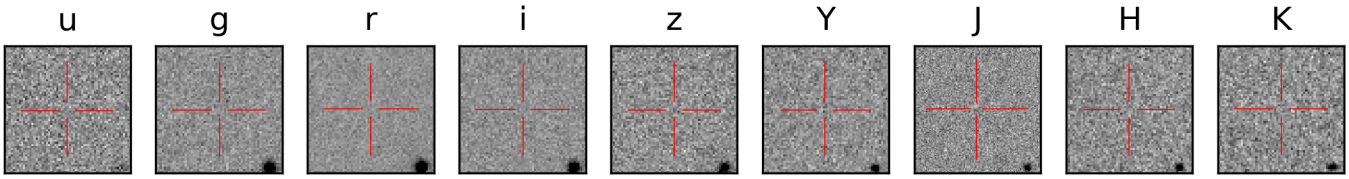


Figure 8.12: Candidate J033404.44+001003.4 had originally been rejected via visual inspection due to cross-talk. This candidate often gets accepted by the machine learning classifiers.

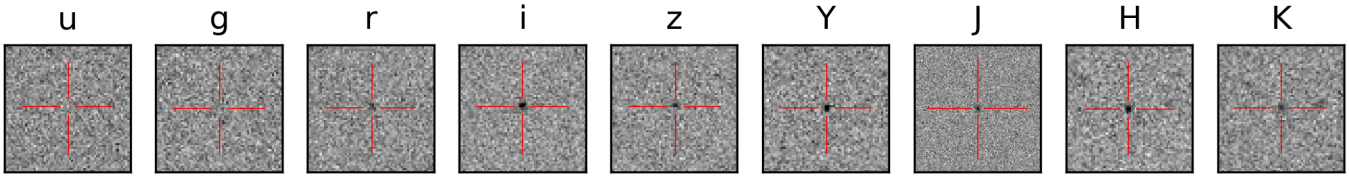


(a) Original

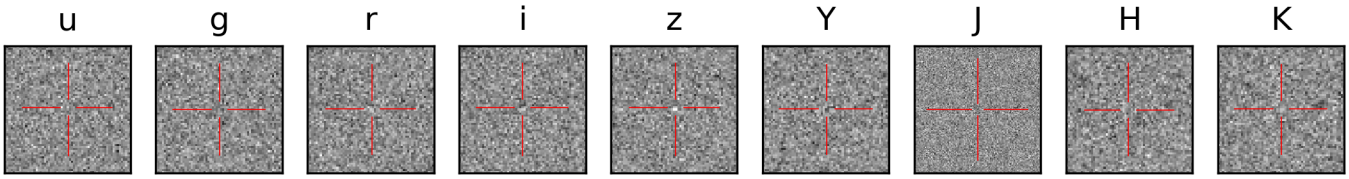


(b) Residual using the best-fit quasar SED

Figure 8.13: Candidate J085003.71−010507.8 has been rejected by visual inspection due to cross-talk issues. However, the RF often accepts this candidate. a) Original image data in SDSS and UKIDSS LAS, b) Residual using the best-fit quasar SED. It has a high quasar probability $P_{\rm q} = 1.00$ and a low residual chi-squared value $\tilde{\chi}^2 = 1.16$.



(a) Original

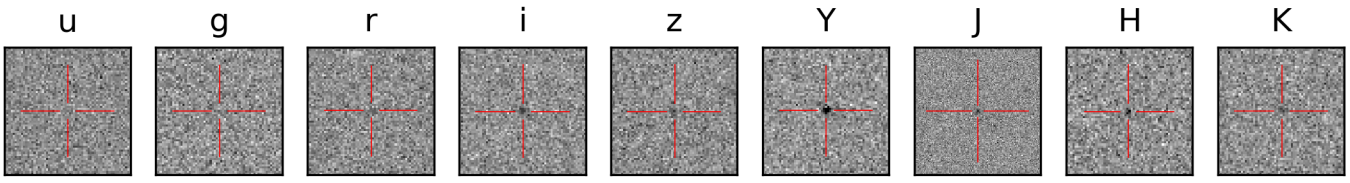


(b) Residual using the best-fit quasar SED

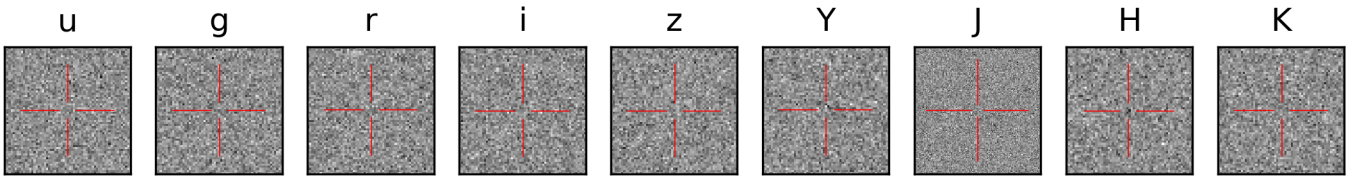
Figure 8.14: Candidate J095350.69+112449.8 has been rejected by visual inspection but often gets accepted by the RF. a) Original image data in SDSS and UKIDSS LAS, b) Residual using the best-fit quasar SED. The quasar probability is high at $P_{\rm q} = 1.00$ with $\hat{z} = 5.32$, and low chi-squared values of $\tilde{\chi}^2 = 1.28$ and $\tilde{\chi}^2_{\rm max} = 2.18$.

cross-talk paints the images as unreliable.

Candidate J095350.69+112449.8 in Figure 8.14 has a quasar probability of 1 and low chi-squared

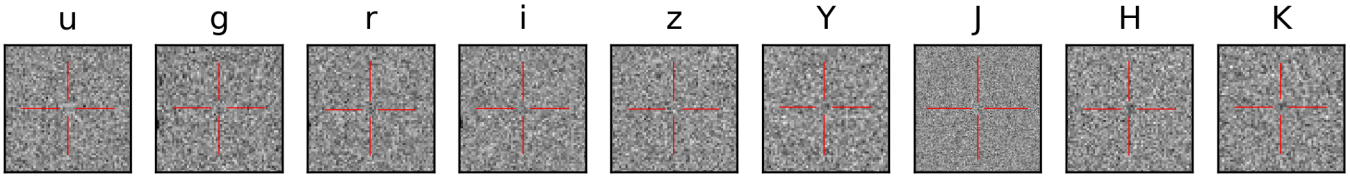


(a) Original

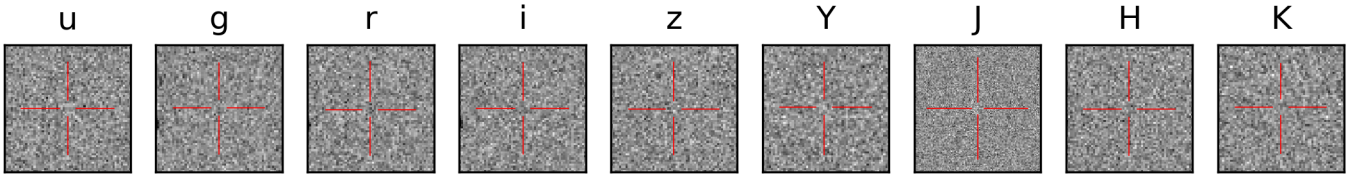


(b) Residual using the best-fit quasar SED

Figure 8.15: Candidate J105208.71−005847.0 has been rejected by visual inspection due to a diffraction spike but has been incorrectly accepted by the DT and RF. a) Original image data in SDSS and UKIDSS LAS, b) Residual using the best-fit quasar SED.



(a) Original

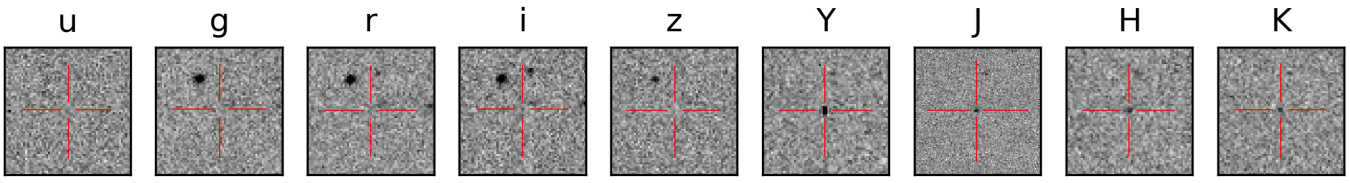


(b) Residual using the best-fit quasar SED

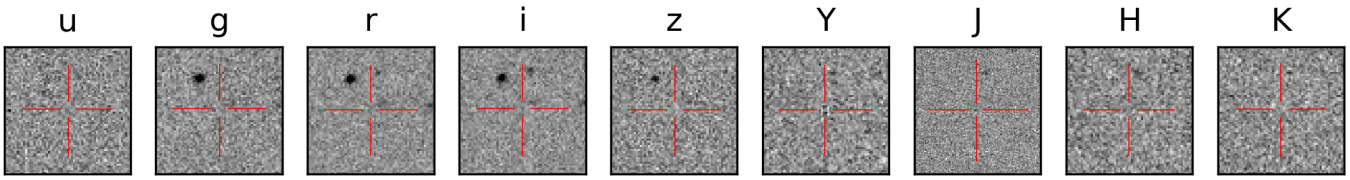
Figure 8.16: Candidate J134452.65+093505.3 has been rejected by visual inspection since the image consists mostly of noise. It has been accepted by the RF due to a large quasar probability of $P_{\mathrm{q}} = 0.96$ and a low residual chi-squared value of $\tilde{\chi}^2 = 1.19$. a) Original image data in SDSS and UKIDSS LAS, b) Residual using the best-fit quasar SED.

values of $\tilde{\chi}^2 = 1.28$ and $\tilde{\chi}^2_{\mathrm{max}} = 2.18$. This source might have been rejected due to non-zero flux in r, which indicates a redshift smaller than 6.5 if it was a quasar. However, since the initial visual classification we have updated the SED templates to be able to handle high-redshift quasars down to $\hat{z} = 5.2$. Hence, this might actually be a quasar at redshift $z \sim 5.2$. We could instead label all possible lower-redshift quasars $z = 5.2 - 6.5$ as accepted and filter out the desired predicted redshift range later on.

Candidate J105208.71−005847.0 in Figure 8.15 is close to a diffraction spike and hence has been rejected via visual inspection. Our metrics and ML classifiers are not able to pick this up and they incorrectly accept this candidate.



(a) Original



(b) Residual using the best-fit quasar SED

Figure 8.17: Candidate J152920.26+015436.4 has been rejected by visual inspection since the source appears to have moved between SDSS and UKIDSS. It is accepted by the RF. a) Original image data in SDSS and UKIDSS LAS, b) Residual using the best-fit quasar SED. The movement in SDSS is shown outside the cross-hair and hence our metrics do not pick this up. Instead, it gives a large quasar probability of $P_{\rm q} = 1.00$ and relatively low chi-squared values of $\tilde{\chi}^2 = 1.39$ and $\tilde{\chi}^2_{\rm max} = 3.75$.

Candidate J134452.65+093505.3 in Figure 8.16 essentially just consists of noise. This easily leads to low chi-squared values of $\tilde{\chi}^2 = 1.19$ and $\tilde{\chi}^2_{\rm max} = 1.62$ as the source is modelled with zero flux in most bands. Due to the small non-zero flux in H and K the quasar model fits the observed photometry best and gets a quasar probability of $P_{\rm q} = 0.96$ with a predicted redshift of $\hat{z} = 6.90$.

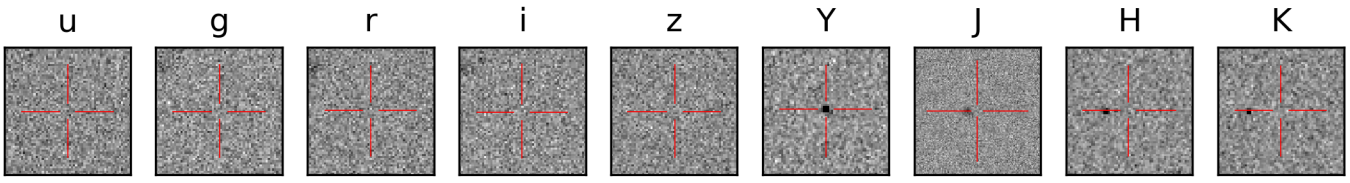
Figure 8.17 shows a candidate that seems to have moved between SDSS and UKIDSS and therefore was rejected via visual classification. However, the movement in SDSS is outside the cross-hair and hence is not captured by our metrics. A quasar probability of $P_{\rm q} = 1.00$ at estimated redshift $\hat{z} = 7.31$ and relatively low chi-squared values of $\tilde{\chi}^2 = 1.39$ and $\tilde{\chi}^2_{\rm max} = 3.75$ leads the machine learning classifiers to sometimes accept this candidate.

8.3.4.2 Classification cleaning tool

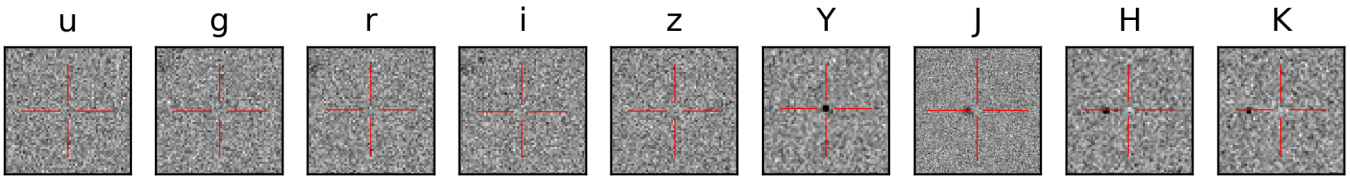
Our classifier further acts as a data cleaning tool by identifying candidates that have been labelled incorrectly during the original visual inspection.

Examples of candidates that have been flagged both by our two-metric threshold selection classifier and our machine learning classifiers are shown in Chapter 7 in Figures 7.22 and 7.23.

Other moving candidates are shown in Figures 8.18 - 8.20. Candidate J003556.65+002931.9 in Figure 8.18 is clearly moving in the NIR bands. In the initial visual classification, this source was classified as accepted. Our classifiers correctly rejected this candidate. Looking at the original image at the top, we can see that the source begins to move in the Y-band. This is also where the reduced chi-squared value is the largest with $\tilde{\chi}^2_{\rm max} = 22.68$. This can also be seen in the residual image at the bottom, where

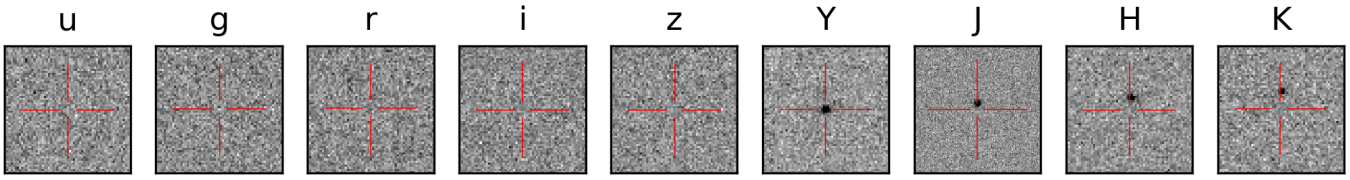


(a) Original

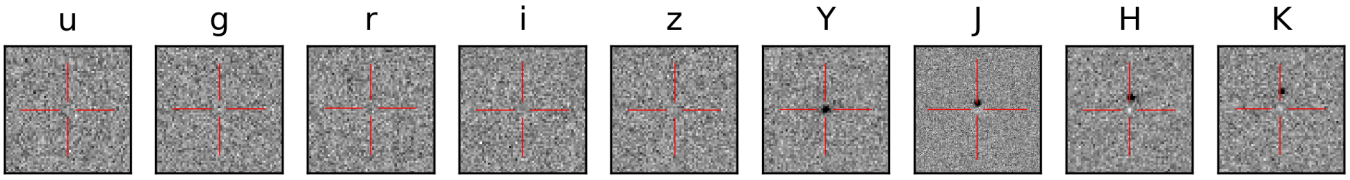


(b) Residual using the best-fit quasar SED

Figure 8.18: Candidate J003556.65+002931.9 is a moving source. It had originally been visually classified as accepted, but was correctly rejected by our RF, DT and two-metric threshold classifier. a) Original image data in SDSS and UKIDSS LAS, b) Residual using the best-fit quasar SED. Although the source has a quasar probability of 1, it has a large maximum chi-squared value $\tilde{\chi}^2_{\rm max} = 22.68$ in Y. In filter J, the source has already moved outside the cross hair and hence the chi-squared returns to smaller values.



(a) Original

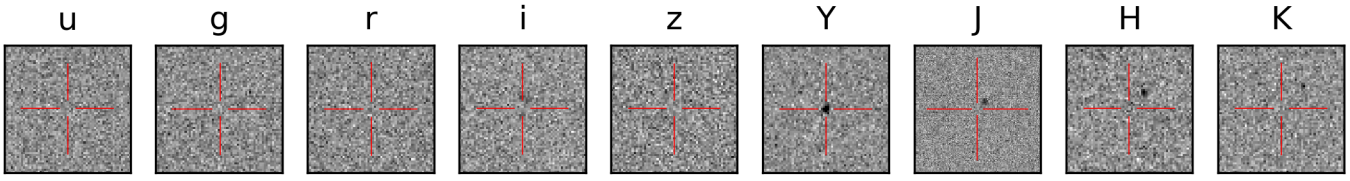


(b) Residual using the best-fit quasar SED

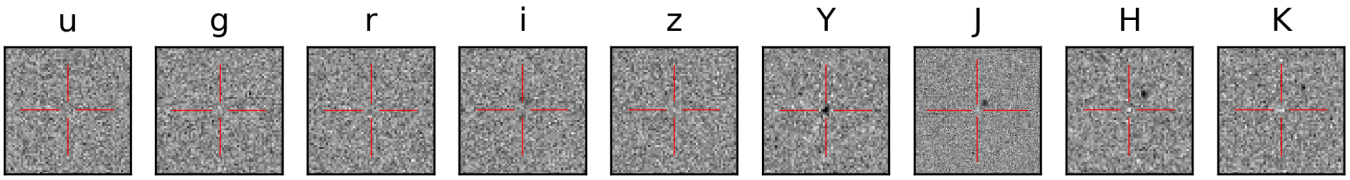
Figure 8.19: Candidate J003648.23+001851.3 is a moving source. It had originally been visually classified as accepted, but was correctly rejected by our RF, DT and two-metric threshold classifier. a) Original image data in SDSS and UKIDSS LAS, b) Residual using the best-fit quasar SED. The maximum residual chi-squared value is $\tilde{\chi}^2_{\rm max} = 22.68$ in Y due to a lower predicted flux compared to the actual one.

there is a signicant source flux left in Y.e In Figure 8.19, the maximum reduced chi-squared value of J003648.23+001851.3 is also in Y with $\tilde{\chi}^2_{\rm max} = 18.43$. This time the source does not appear to be moving in Y, but due to the zero flux in every band except for Y, the predicted flux in Y is much lower than the actual flux. Figure 8.20 shows another such candidate, with a maximum residual chi-squared value of $\tilde{\chi}^2_{\rm max} = 5.60$ in Y.

Examples of sources that were originally rejected by the visual classification but are actual good

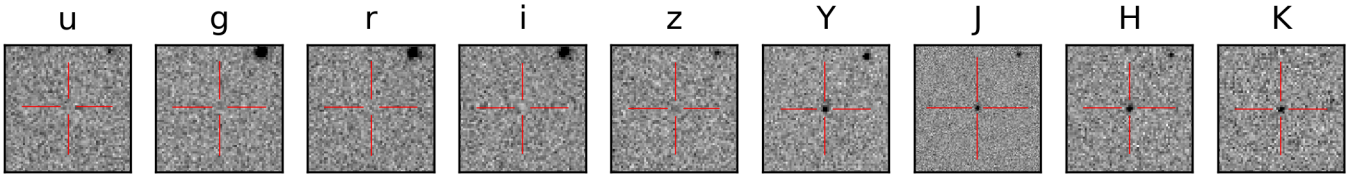


(a) Original

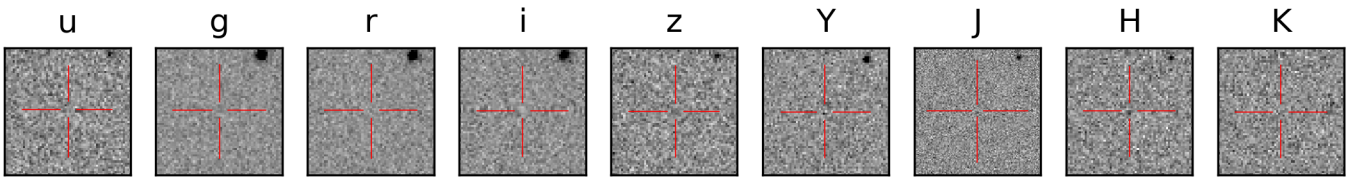


(b) Residual using the best-fit quasar SED

Figure 8.20: Candidate J003137.03+001232.8 is a moving source. It had originally been visually classified as accepted, but was correctly rejected by our machine learning classifiers. a) Original image data in SDSS and UKIDSS LAS, b) Residual using the best-fit quasar SED. The maximum chi-squared value can again be found in Y with $\tilde{\chi}^2_{\rm max} = 5.60$, which is again due to a large difference in predicted best fit quasar SED flux and the actual observed flux.



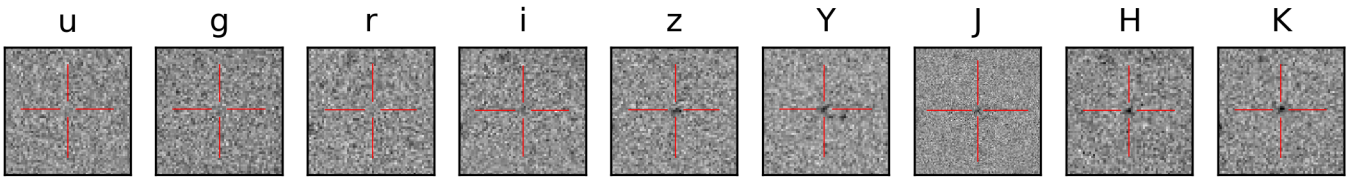
(a) Original



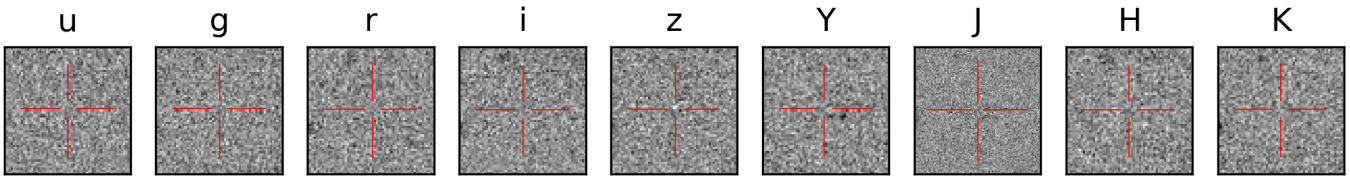
(b) Residual using the best-fit quasar SED

Figure 8.21: Candidate J115132.21+075108.8 is a possible quasar candidate. It had originally been visually rejected, but our RF, DT and two-metric threshold classifier accepted this source. a) Original image data in SDSS and UKIDSS LAS, b) Residual using the best-fit quasar SED. The low chi-squared values of $\tilde{\chi}^2 = 1.22$ and $\tilde{\chi}^2_{\rm max} = 1.83$ indicate that this source is a possible high-redshift quasar with predicted redshift of $\hat{z} = 7.5$.

quasar candidates can be seen in Figures 8.21 - 8.22. The source seen in Figure 8.21 appears to have a Gunn-Peterson trough, and fits well to one of our quasar SEDs with $\tilde{\chi}^2 = 1.22$ and $\tilde{\chi}^2_{\rm max} = 1.83$. Its predicted redshift is $\hat{z} = 7.5$ with a quasar probability of $P_{\rm q} = 0.01$. Candidate J140823.78+075826.9 in Figure 8.22 has a quasar probability of $P_{\rm q} = 0.89$ at predicted redshift of $\hat{z} = 5.8$ and therefore its classification was changed to 'accepted'. The largest residual chi-squared value is in band z with $\tilde{\chi}^2_{\rm max} = 2.84$ due to a difference between the calculated and predicted flux, and due to a non-point-source-like appearance possibly due to some data processing issues.



(a) Original



(b) Residual using the best-fit quasar SED

Figure 8.22: Candidate J140823.78+075826.9 is a possible quasar candidate. It had originally been visually rejected, but our RF, DT and two-metric threshold classifier accepted this source. a) Original image data in SDSS and UKIDSS LAS, b) Residual using the best-fit quasar SED. This candidate has with $P_{\mathrm{q}} = 0.89$ a high quasar probability. Its predicted redshift is at $\hat{z} = 5.8$.

8.4 Conclusions

Starting from sky locations of a list of $\sim$ 4300 cross-matched SDSS and UKIDSS LAS sources, selected and labelled by Mortlock et al. (2012), we apply the pipeline from Section 6 that first processes the candidates and extracts information such as database and PSF photometry, chi-squared goodness of fit $\tilde{\chi}^2$ and quasar probability P_{q}. Next, we discard candidates with $P_{\mathrm{q}} \leq 0.01$ in order to get rid of the majority of contaminants. The feature set from the remaining pool of candidates is fed into two different classifiers: a decision tree of depth $d = 4$ for interpretability and a random forest with balanced weights. The best results are obtained with the random forest with an AUC score of 0.94 ± 0.04. Using a survey- and redshift-independent feature set, we achieve AUC $= 0.83 \pm 0.06$, which approximately doubles the number of false positives and false negatives compared to training set B (but from a low baseline).

This can be adapted for use for upcoming surveys such as LSST and Euclid. The random forest will have to be trained on a new training set of similar size to the one discussed here. However, cuts in the quasar probability and/or redshift estimate can be applied either before or after going through the machine learning classifiers. This allows to sift out candidates which are not worth reviewing such that only interesting candidates will have to be visually inspected.

Chapter 9

CNN classification on raw image data

A convolutional neural network is a type of ANN that uses convolutions on pixel data, where the convolved kernels represent the learnt features. CNNs are described in detail in Section 5.7. They have been widely used in astronomy, for example to detect gravitational lenses (Schaefer et al., 2018; Davies et al., 2019a) and radio galaxies (Aniyan & Thorat, 2017), or to differentiate between stars and galaxies (Kim & Brunner, 2017). For quasar research they have been successfully applied to e.g. detect the photometric redshift of quasars in SDSS stripe 82 (Pasquet-Itam & Pasquet, 2018b), to classify damped Lyα systems from spectral data (Parks et al., 2018b), and to identify BAL quasars (Guo & Martini, 2019b).

Our data processing pipeline (see Chapter 6) obtains all the available image data in SDSS and UKIDSS. This image data then gets summarised to be used in e.g. the two-metric classifier (see Chapter 7). However, this does not make use of the complete image data. We therefore implement a convolutional neural network that classifies high-redshift quasars and contaminants, and uses the postage stamp images as its input. Compared to tree-based algorithms such as the decision tree and the random forest, the CNN learns relevant features in a purely data-driven way rather than having to rely on specific pre-defined features. However, the CNN has to be retrained and adapted for each individual survey, and surveys with different image properties cannot be easily combined.

9.1 Overview

Well-known CNN models such as AlexNet (Krizhevsky et al., 2017) and ResNet (He et al., 2015) require large datasets that we do not have. For instance, AlexNet has been trained on image data of order $> 10^6$. This is much larger than our dataset of a couple of thousand images. In addition, typical CNNs make use of pooling layers which give some translational invariance of a feature. This is something we are not interested in, and actually specifically do not want, since we want to detect if

a source has moved between the filters.

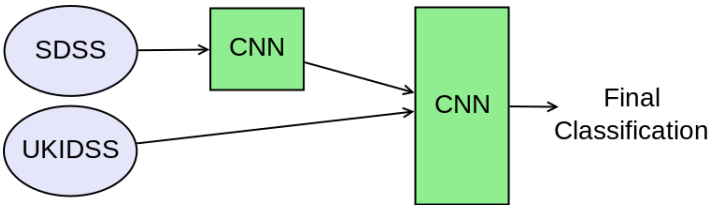


Figure 9.1: Hierarchical architecture for the convolutional neural network. The first CNN only takes image data of SDSS, whereas the second CNN takes the output of the first CNN together with the UKIDSS image data as an input. The output of the second CNN then corresponds to the final classification.

Therefore, we develop our own architecture that is not very deep and does not consist of any pooling layers in order to differentiate high-redshift quasars from contaminants. We build it as a hierarchical CNN classifier consisting of two CNNs - one for SDSS and one for UKIDSS. The input for a given source in each survey's CNN is the postage stamp data in all passbands of the survey. For SDSS, this is the set of postage stamps in u, g, r, i, z. The CNN for UKIDSS has the survey images in Y, J, H, K as its input, and in addition is fed the classification output (*high-redshift quasar* or *contaminant*) from the CNN of SDSS. This hierarchy represents an iterative refinement on the classification, which is illustrated in Figure 9.1. It is similar to the visual classification labels by Mortlock et al. (2012), who labelled the sources first in SDSS and subsequently in UKIDSS. Details on the visual classification labels are given in appendix A.

One of the advantages of the hierarchical CNN is that if in future we ever wanted to include a third survey, then the CNNs for SDSS and UKIDSS would not have to be re-trained. If instead we had one single CNN that included the images of all surveys, then it would have to be re-trained for each new survey that is added. One could have also implemented one CNN for each survey individually and combine the results at the end. However, then the results are only combined at the very end at the deepest layer where not a lot of training occurs anymore. This would likely lead to incorrect results. An even better solution would be to have individual CNNs that are very shallow with a multidimensional output that then could be combined with a few more dense layers. This would then allow for information to be passed between the different surveys. However, it is not clear what this multidimensional output should be trained against (i.e. what the output labels/images would be).

In order to handle an imbalanced training set, we artificially increase the number of positive samples by augmenting the data - for instance, rotating/flipping the image and injecting noise.

The training set is described in Section 9.2. In Section 9.3 we outline the architecture choices of the CNN that forms the building block of the hierarchy. The data augmentation techniques for the training set are reported in Section 9.4. Finally, the results are presented in Section 9.5.

9.2 Training dataset

We lay out the dataset that we are using for training and testing our CNN classifier.

9.2.1 Survey data

As with the dataset in Sections 7.4 and 8.1, we use optical data from the SDSS DR12 (Alam et al., 2015) and NIR data from the UKIDSS DR10PLUS LAS survey (Lawrence et al., 2007).

9.2.2 Training data

There are two classes: contaminant (class 0) and high-redshift quasar (class 1). A candidate is represented by the combination of one image per band of a given survey. So if a survey has 4 photometric bands, then an input sample source is represented by 4 images. This then acts as the input for the CNN (see Section 9.3). The output label is either 0 for a contaminant of 1 for a high-redshift quasar.

Instead of the validation set in Section 7.4 and the training set in Section 8.1, here we make use of the image data only, and do not query the surveys' databases in any way other than to retrieve the images and the locations of nearby sources. The initial dataset consists of $\sim$ 4300 candidates that have been inspected and labelled by Mortlock et al. (2012), who have used broad colour cuts as a pre-processing step (see Figure 3.11) to filter out most of the contaminants. While Mortlock et al. (2012) have used SDSS DR7 (Abazajian et al., 2009) and UKIDSS DR7 as the available survey data at the time, here we use the updated images from SDSS DR12 (Alam et al., 2015) and UKIDSS DR10PLUS (Dye et al., 2006).

The labels exist for both surveys separately, such that a source has a classification of either being a probable quasar candidate (accepted) or being a contaminant (rejected) in every survey. A candidate is only labelled as accepted if it is visually accepted in both SDSS and UKIDSS. Otherwise, it is labelled as rejected.

9.2.3 Pre-processing of the training set

We remove candidates from our training set that have missing data or postage stamps filled with NaN values. Further, we remove the PSF flux of nearby sources from the postage stamp. By removing nearby sources from the image data, they cannot be interpreted as important to the classification by the feature detectors of the CNN. In total, we remove 211 visually accepted sources and 1628 visually rejected sources from our data set. This leaves us with a training set of 2664 candidates.

9.2.3.1 Discard sources with missing images

An astronomical source that has not been observed in all of the passbands results in missing data. Since the CNN cannot handle missing data, we cannot use sources that are unobserved in at least one of the survey's photometric bands. This drastically reduces the initial dataset. The main reason for this is that we do not download images with data processing issues, as is described in Section 6.1.1.

9.2.3.2 Discard sources with incomplete postage stamps

Some sources lie on the edges of the observing area. As discussed in Section 6.1.2, we fill up such postage stamps with NaN values. The CNN, however, cannot deal with NaN values or missing data. We could swap all NaN values with e.g. with the mean of the background flux, but then the CNN would try to find patterns in the data where there should be none. Hence, we discard such candidates. In total, we remove 130 candidates from our input list due to incomplete postage stamps.

9.2.3.3 Removing nearby sources from the image

We want the CNN to disregard any source other than the central one. Exceptions are sources that are too close to the central source of interest such that the photometry cannot be extracted. Other exceptions are data processing issues within the image.

We therefore do not use the original postage stamp data, but rather use the original postage stamps minus our reconstructed postage stamp of nearby sources (see Section 6.1.7). That is, we make use of our reconstructed postage stamps before the central source is reconstructed, as is illustrated in Figure 6.9. We then subtract this reconstructed image from the original data in each band $b \in B$ and run r_b,

$$\boldsymbol{I}_{b,r_b} = \boldsymbol{O}_{b,r_b} - \boldsymbol{R}_{b,r_b}, \tag{9.1}$$

where $\boldsymbol{O}$ the original postage stamp, and $\boldsymbol{R}$ the reconstructed postage stamp without the central source.

Next, we average over all available runs n_b in each passband,

$$\boldsymbol{I}_b = \frac{1}{n_b} \sum_{r_b=1}^{n_b} \boldsymbol{I}_{b,r_b}. \tag{9.2}$$

Apart from at least one run of each survey that has images in all bands, additional runs sometimes only contain image data in a few photometric bands. For instance, we might have additional runs that only have image data available in z, Y, H, and the image data in u, g, r, i, J, K is missing. We could not use such a runs on its own as the input for the CNN, since the CNN does not handle missing data. What we can do, however, is average over all runs for each band. This makes use of all available

image data. If we had only used runs that have image data in all bands, we still would have missed out on incomplete runs that are potentially important to the classification.

For instance, there will be visually rejected candidates that look like quasars in one run and look like contaminants in at least one other run. It is therefore important to include all available data. The disadvantage is that the average smoothes out any extreme image data which could lead to the acceptance of otherwise visually rejected candidates. In the example above, the run that has led to the rejected visual classification label is the most important one. However, we rather accept a contaminant than to miss out on a possible quasar candidate.

We also note that we can only reconstruct nearby sources that resemble point sources. Any other source shape is not captured by our pipeline and hence a nearby galaxy will not be properly removed. Since most of the candidates that are in the presence of a galaxy are visually rejected, the CNN will likely learn to detect galaxies and automatically reject such candidates. In general this is fine but it might lead to some misclassifications.

9.2.3.4 Down-sample images in J

The CNN requires the same image size for all channels. However, the pixel size in J is double that of the other bands, namely 0.2 arcseconds per pixels compared to 0.4 arcseconds per pixel. Therefore, the same number of pixels in J as that in the other bands corresponds to half the sky area in arcseconds compared to the area of the other bands. Instead of using a smaller sky area in J, we down-sample the double-sized J-band image that covers the same area by half, down to the image size of the other bands. This is so that the same-sized image captures the same area as the other bands even though this means a loss of information.

9.3 Architecture

We explain the building blocks of our CNN. The input is different depending on if a single survey is used or if multiple surveys are combined. For instance, the input varies between the CNN of SDSS and the CNN of SDSS+UKIDSS. One CNN is trained on SDSS images only. Another CNN has as its input the output class probability of the CNN of SDSS and the images of UKIDSS. The output of the CNN represents the probability P (based on the training data) that the input sample is a high-redshift quasar.

Categorical cross-entropy is used as our loss function (see Appendix C.1.1.1), and the Adam optimizer is used as our optimization procedure (see Appendix C.1.2.2).

9.3.1 Input and output of the CNN

Let N be the total number of input samples and N_c be the number of classes for these inputs. The class labels are given by y, where each element can take on any value in $\{0, .., N_c - 1\}$. Further, let $\boldsymbol{d_n}$ be the input to the CNN for sample $n \leq N$.

The input for a source is given by the set of postage stamp images $\{\boldsymbol{I}_{n,b}\}_{b \in B}$ that contains each passband $b \in B$. Each band image $\boldsymbol{I}_{n,b}$ has dimensions of $l \times l$. So if a survey has $n_{channels}$ number of passbands, then the input $\boldsymbol{d_n} = \{\boldsymbol{I}_{n,b}\}_b$ for the CNN has dimensions of $(n_{channels} \times l \times l)$. For any subsequent CNN the input has dimensions of $(n_{channels} \times l \times l, n_{channels})$, since the output of the previous CNN is included in the input. This can be written as

$$f_1 \quad : \quad \mathbb{R}^{n_{channels} \times l \times l} \longrightarrow \mathbb{R}^{N_c} \tag{9.3}$$

$$f_{k>1} \quad : \quad \left(\mathbb{R}^{n_{channels} \times l \times l}, \mathbb{R}^{N_c}\right) \longrightarrow \mathbb{R}^{N_c}, \tag{9.4}$$

where f_1 corresponds to the (unknown) function that is learned by the first CNN, and $f_{k>1}$ the (unknown) function that is learned by any subsequent CNN. For the k^{th} CNN the input is thus given by $(\{\boldsymbol{I}_{n,b}\}_{b \in B}, f_{k-1}(\{\boldsymbol{I}_{n,b}\}_{b \in B}))$. If we have K CNNs, then the individual CNNs are combined together in a hierarchy to form the final learned function f_K.

The class representations are given by

$$i \quad \in \quad \{0, .., N_c - 1\} \tag{9.5}$$

$$i_n \quad = \quad \text{true class of } \boldsymbol{d_n} \tag{9.6}$$

Let $\boldsymbol{a_n}$ define the output of the CNN before it has been fed through the final activation function,

$$\boldsymbol{a_n} = f(\boldsymbol{d_n}), \quad \forall n = 1, .., N. \tag{9.7}$$

The final result, after the activation function in the last layer of the CNN has been applied to a_n, then corresponds to the probability $P(y = i|\boldsymbol{a_n})$ that input $\boldsymbol{d_n}$ is classified as class i. During training, $P(y = i_n|\boldsymbol{a_n})$ will be calculated to see if the CNN has correctly classified input $\boldsymbol{d_n}$ as being part of class i_n. Calculating this probability as part of the loss function and backpropagating its result will help the CNN to "learn" and update its weights. After training, a input $\boldsymbol{d}$ with $\boldsymbol{a} = f(\boldsymbol{d})$ will be classified to class i for which the probability $P(y = i|\boldsymbol{a})$ is the greatest.

In this case we have two classes ($N_c = 2$), where class 0 corresponds to the candidate not being a quasar and class 1 represents a possible high-redshift quasar. Our hierarchical classifier consists of two connected CNNs from SDSS and UKIDSS. For SDSS, we have $n_{channels} = 5$ with filters u, g, r, i, z, and for UKIDSS we have $n_{channels} = 4$ using the filters Y, J, H, K. Their respective architectures are shown in Figures 9.2 and 9.3.

9.3.2 Standardisation

First, we standardise the initial input in each filter. That is, postage stamps of passband u will be normalized given all the postage stamps in that passband, and so on. That is, for an image I_f of filter f, the normalized image is given by

$$I_f = \frac{I_f - \mu_f}{\sigma_f}, \forall f \in \{1, 2, .., N_f\}, \tag{9.8}$$

where μ_f the mean and σ_f the standard deviation of the postage stamps of filter f. The mean and standard deviation are calculated over the whole dataset for each separate filter.

9.3.3 No pooling layers

Pooling layers (see Section 5.7.2) increase the region of the translational invariance within an input image. Any information on the location of that feature within that region is thus lost. Thus, using pooling layers would prevent us from detecting if a source has moved between the bands, and therefore we refrain from using any pooling layers. We can do this as our input image size is small, and hence a few convolutional layers can down-sample the input enough such that the final fully connected layers do not have too many parameters.

The kernels further lead to regional translational invariance. The larger the kernel size, the smaller is the translational invariance. If the kernel is the same size as the image then this does not give any translational invariance as it is not possible for the feature kernel to move over the image. However, a feature kernel as large as the image creates a large amount of parameters. It further drastically down-scales the image to one pixel unless non-zero padding is used. If the kernel is only one pixel wide, then this gives translational invariance over the whole image. However, a kernel of one pixel width will not be able to pick up any useful features. Our image data is of size 19×19 in pixel size, and our kernel sizes are 5×5 at higher layers and 3×3 at lower layers.

9.3.4 Channels

Initially, we wanted the CNN to find a relation between two successive filters since these are related. We therefore first tried to "force" the convolutional neural network to learn features between two subsequent filters for all channels. That is, to learn features between the u and g band, g and r band, r and i band and i and z band. The input was given as image pairs, and these image pairs were then convolved multiple times before being merged with the output features of the other channel pairs. This gave similar results compared to convolving over all input channels. It also drastically increased the amount of parameters and training time. For this reason, and because our dataset containing fewer

than 5000 candidates is quite small, we changed the architecture to use as many input channels in the first layer as there are photometric filters in the data. That is, 5 channels in SDSS and 4 channels in UKIDSS. This drastically decreased the amount of parameters and sped up execution time.

9.3.5 Overfitting

In order to reduce overfitting, we apply dropout in every second layer, and batch normalization before every convolutional layer. This is explained in detail in Appendix C.2.3.

9.4 Data augmentation

The number of positive candidates is small. We only have ~ 300 positive samples in our training set, of which only 14 are spectroscopically confirmed quasars. We augment the given dataset artificially as to create an equal number of positive and negative candidates.

The following augmentation techniques are implemented:

- **Eigenvector noise**

  Before training, we calculate the principal components of the dataset for each band. We can use these to reduce the dimensionality of the ADU data by projecting the images onto their eigenvectors. The projected images should contain enough information such that the central source is still visible. This has the effect of noise reduction and signal filtering. However, we have found that it does not help in the performance of our CNN, and thus decided to not include this step. What we do instead is to use the eigenvectors to inject some noise. These are scaled according to a normal distribution $\mathcal{N}(0, 0.1)$ and then added to the image.

- **Gaussian noise**

  Gaussian noise $\mathcal{N}(0, a^2\,\sigma^2)$ is added to each image, where σ the standard deviation of the normalized image. Setting the noise level too high makes the central source indiscernible, and setting it too low does not augment the data enough. We have found that using values $1 \leq a \leq 2$ gave good results.

- **Horizontal and vertical flips**

  Horizontal and vertical flips are applied as these do not change any of the pixel ADU values.

- **Rotate by multiples of 90 degrees**

  Rotations that are multiples of 90 degrees are used. The reason for this is that rotations at any other angle other than multiples of 90 degrees require the interpolation of the image data. For

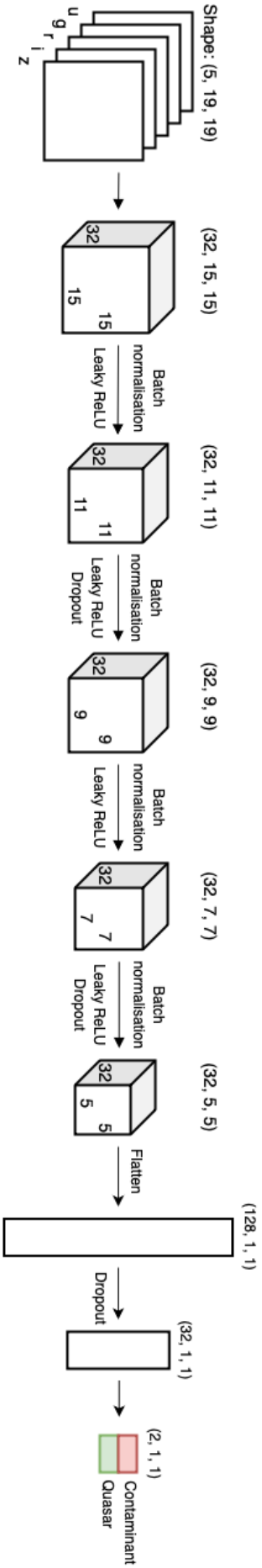


Figure 9.2: Architecture of the convolutional neural network for SDSS.

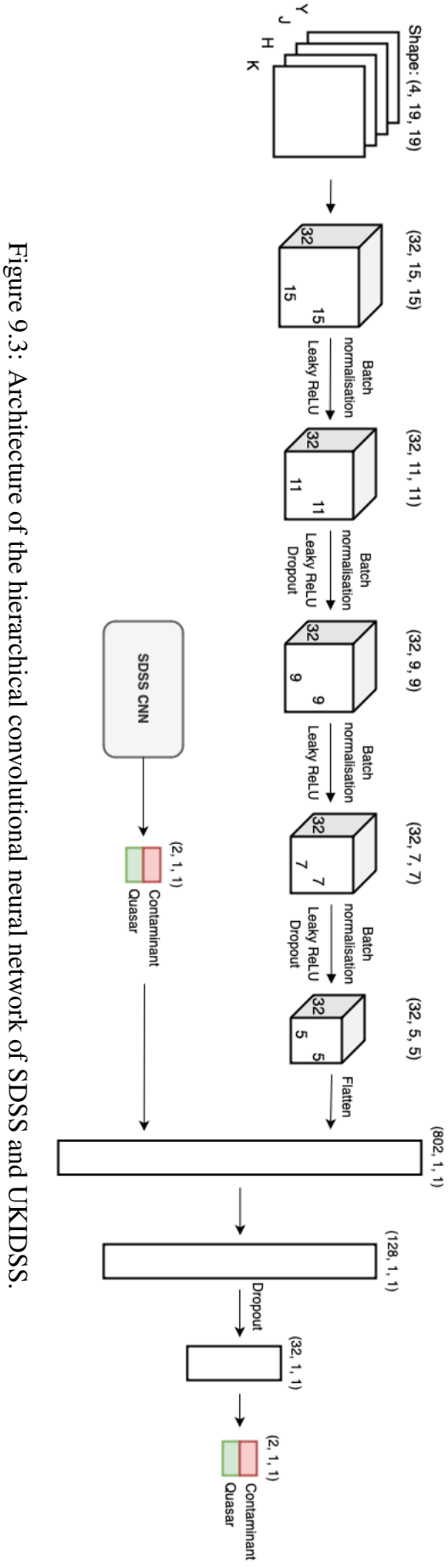


Figure 9.3: Architecture of the hierarchical convolutional neural network of SDSS and UKIDSS.

the central source there are only a handful of pixels that carry most of the information, and hence an interpolated image might change the shape of the central source and/or reduce the measured flux.

9.5 Results

We describe how well our CNN classifier can reproduce the results by Mortlock et al. (2012).

Analogously to the applications of the decision tree and the random forest, we use the ROC AUC score as our performance measure. The overall AUC score was obtained to be 0.90 ± 0.05. This has been calculated by successively selecting a random subset of our complete data set as our training set. The leftover samples of each iteration were then split up between the test set and the validation set. The AUC score was then averaged over 10 successions. We plot the ROC curve for an instance of our hierarchical CNN in Figure 9.4, where the diamond-shaped threshold in the ROC curve corresponds to a point in the ROC curve that results in the greatest distance from the diagonal. The corresponding confusion matrix can be seen in Figure 9.5. We can see that the CNN performs worse than the RF, but better than the DT and the two-metric threshold classifier.

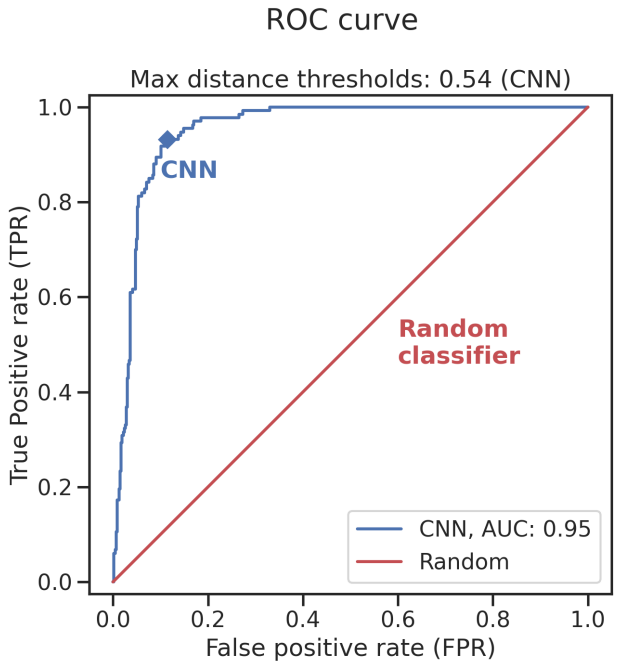


Figure 9.4: ROC curve and AUC score for an instance of the CNN. The threshold indicated by the diamond represents the largest distance from the diagonal. The red line shows the base performance of random classifications (AUC = 0.5).

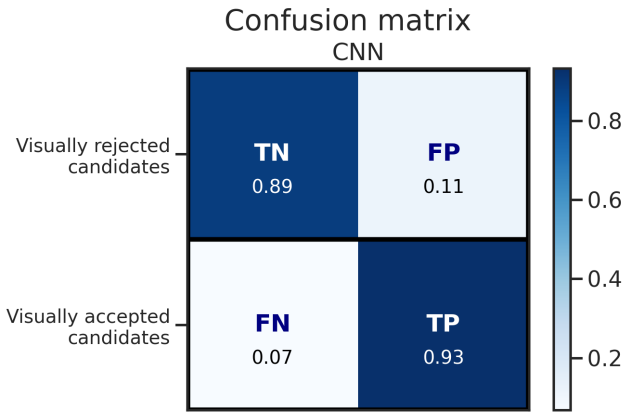


Figure 9.5: Confusion matrix of the CNN, corresponding to the threshold as shown in Figure 9.4. The number of FN and FP is higher in the CNN than in the RF, which indicates that the RF performs better.

9.5.1 Misclassifications

We show a few examples of sources where the CNN's classification is different to the visual classification labels. They can be seen in Figures 9.7 - 9.10. The predicted quasar redshift distribution of a set of false positives and false negatives can be seen in Figure 9.6. We can see no obvious peak in the predicted redshifts for the false negative candidates, which indicates that there is no specific redshift bin in which the CNN performs much better or worse. The least amount of false negatives are in the highest redshift bins which is a good sign, though that is likely just because there are fewer candidates at those predicted redshifts. The predicted quasar redshift is not meaningful for false positives, though we can see that these would be predicted at the lower redshift end which is good as that means that there are less contaminants at the more interesting higher redshift end.

Since the dataset we are using has been visually inspected and hand-labelled, it is clear that at least some candidates will have been classified incorrectly. Other candidate classifications have been correctly classified by the CNN, but the initial visual classification was incorrect. These include candidates that have moved or that were accepted in SDSS and rejected in UKIDSS but should have been rejected in both surveys. Even though the overall classification stays the same, an incorrect classification in one of the surveys makes a difference to the overall CNN as the label error not only affects the current CNN but also any CNN that follows.

Looking at the false negatives shows us the pattern with which the model classifies a candidate incorrectly. This includes cases where there is a detector flaw or some data processing issues that were not flagged by the survey's database. Before we removed nearby point sources from the input postage stamp images, the CNN would reject many candidates that were in the immediate vicinity of another source.

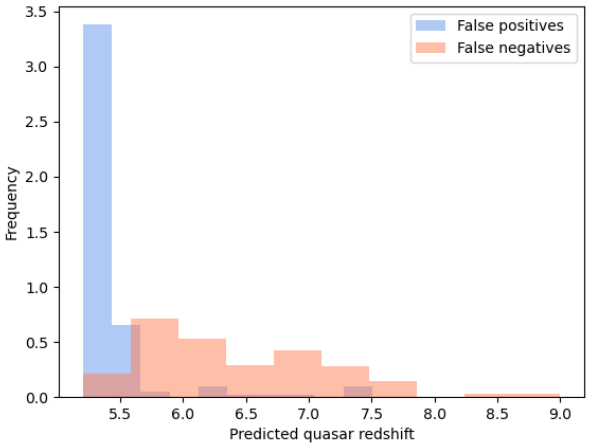


Figure 9.6: Predicted quasar redshifts for a set of $\sim$ 200 false positives and $\sim$ 200 false negatives. These have been obtained by running a test set through the trained CNN. Note that the predicted quasar redshift is not meaningful for the false positive sources.

9.5.1.1 False negatives

Looking at the false negatives has shown us the pattern with which the model classifies a candidate as a contaminant. It can also show us candidates where upon visual re-inspection the CNN actually correctly rejected a candidate and the visual classification label is updated.

Examples of false negatives produced by the CNN can be seen in Figures 9.7 - 9.14. These typically include candidates that have properties that are rare in the training set, such as having several additional sources of varying shapes near the central candidate. Upon re-inspecting the candidate seen in Figure 9.7, the visual classification label is updated to 'rejected' based on the significant flux in g. Therefore, the CNN's classification is actually correct.

Figure 9.9 shows a candidate that likely was visually observed in only one of the runs. Non-zero flux in u, g would have otherwise led to the visual rejection of this source.

The reason for the rejection of the source in Figure 9.10 is likely a combination of a very bright non-point source in the vicinity of the central source, and in addition several other faint extended sources that are nearby. Our pipeline is not able to properly reconstruct all of these sources since there is at least one source that does not resemble a point source. Therefore, the residual leaves a remaining non-zero flux apart from the central source. Input images like these are rare for the CNN as it will not encounter this specific issue very often. Therefore it is possible to deal with such issues manually. However, it could be solved by fitting for extended and/or elliptical sources in our data processing pipeline. Fitting extended sources could be done by our two-Gaussian PSF, with one of the Gaussians having a large spread. For elliptical sources, another source function would have to be added to the pipeline.

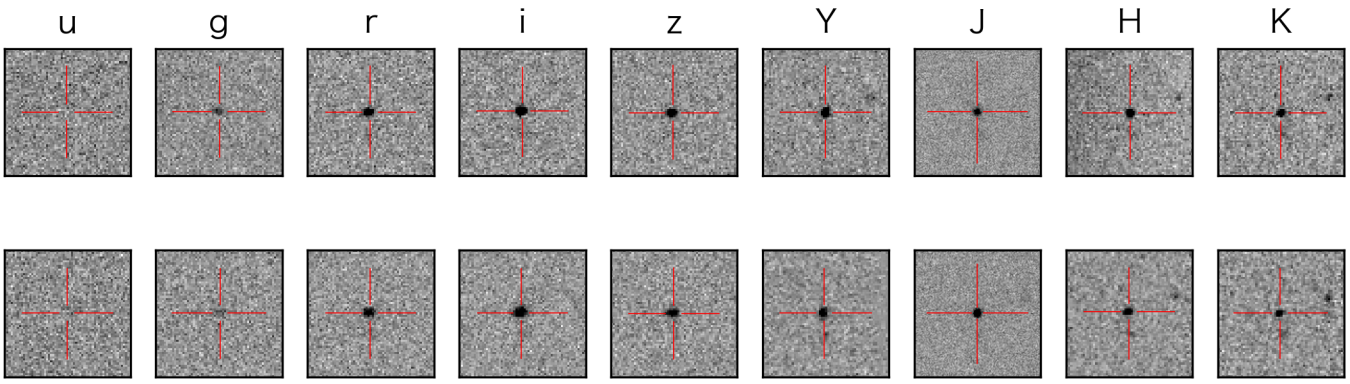


Figure 9.7: Candidate J035849.86+003702.4 was accepted based on visual inspection but has been rejected by the CNN. Upon re-inspection, the visual classification label is changed to 'rejected' due to the significant flux in *g*.

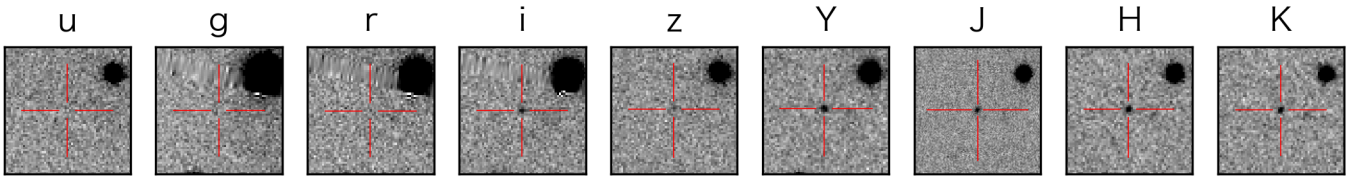


Figure 9.8: Candidate J083428.80+082752.3 was accepted based on visual inspection but has been rejected by the CNN. Upon re-inspection, the visual classification label is changed to 'rejected' due to the data processing issues in the optical bands.

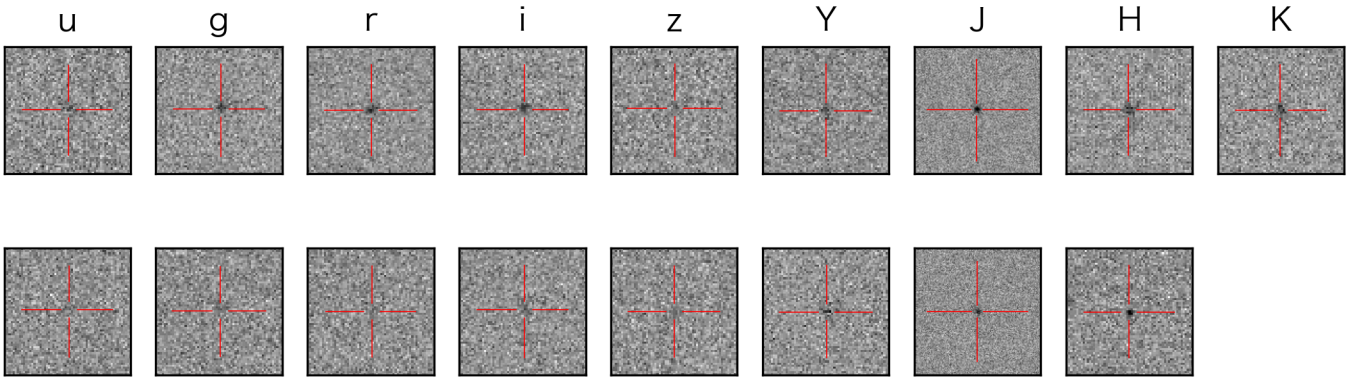


Figure 9.9: Candidate J145622.42+065221.3 was accepted based on visual inspection but has been rejected by the CNN. Upon re-inspection, the visual classification label is changed to 'rejected' due to the variance in flux.

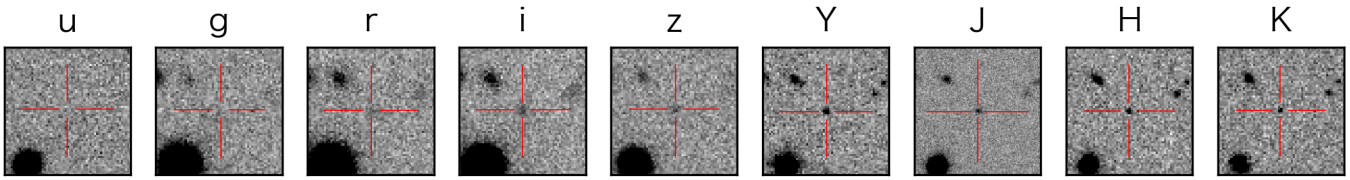


Figure 9.10: Candidate J224943.96+052432.5 was accepted based on visual inspection but has been rejected by the CNN. It was likely rejected by the CNN due to the fact that there are not many training samples that include several sources.

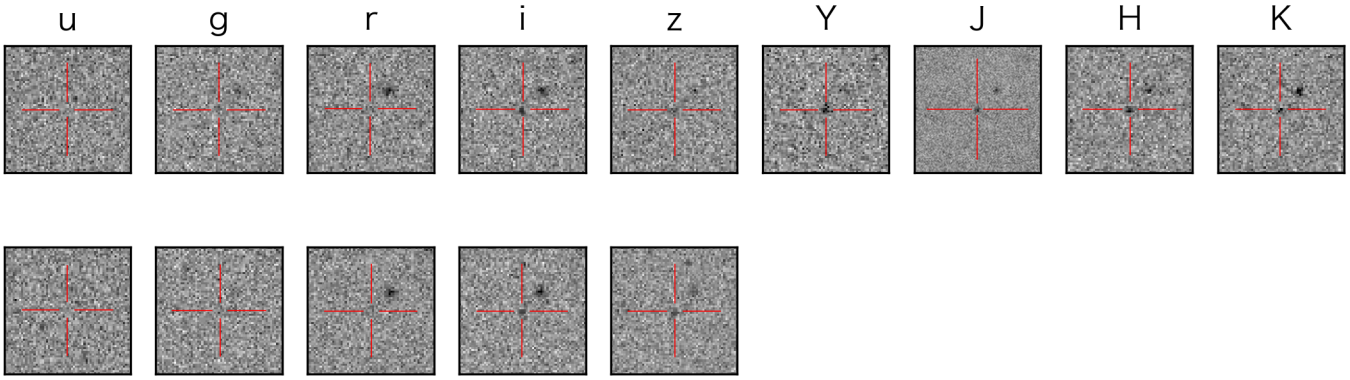


Figure 9.11: Candidate J010404.13−010741.7 that was visually rejected but has been falsely accepted by the CNN.

9.5.1.2 False positives

Figures 9.11 - 9.14 show a couple of candidates that have been flagged as false positives repeatedly. These are sources that were rejected based on visual inspection, but that the CNN labelled as accepted.

These plots show the drawback of taking the average image data over all runs. While one of the runs has a visible source in Y (and J), the second run does not. This therefore cannot be a quasar. The average, however, still shows some source flux in Y (and J). Based on this information alone it makes sense that these source are accepted by the CNN.

9.5.2 Limitations of this method

Overall, for the purpose of this project, the CNN has the following limitations

- Cannot handle varying pixel sizes between channels
- Cannot handle varying orientations between channels

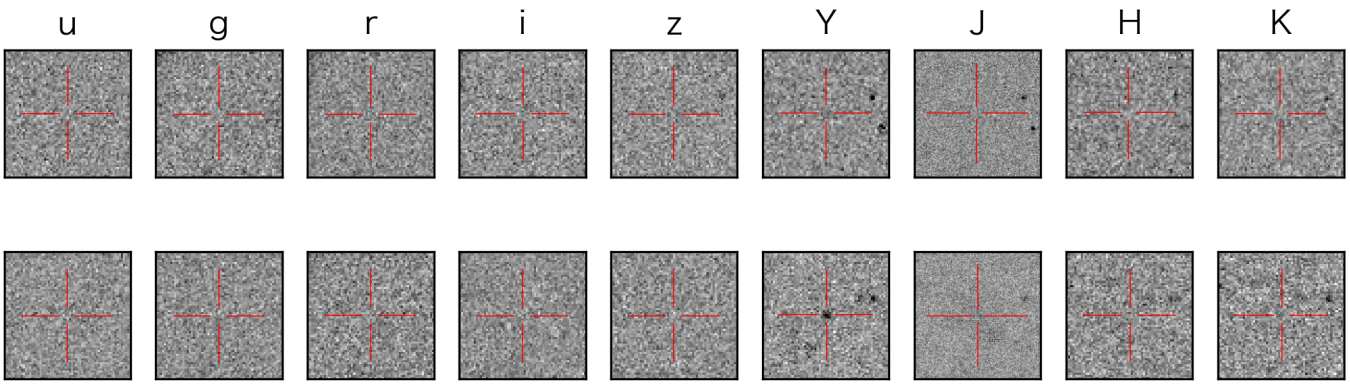


Figure 9.12: Candidate J090458.51+054423.8 that was visually rejected but has been falsely accepted by the CNN.

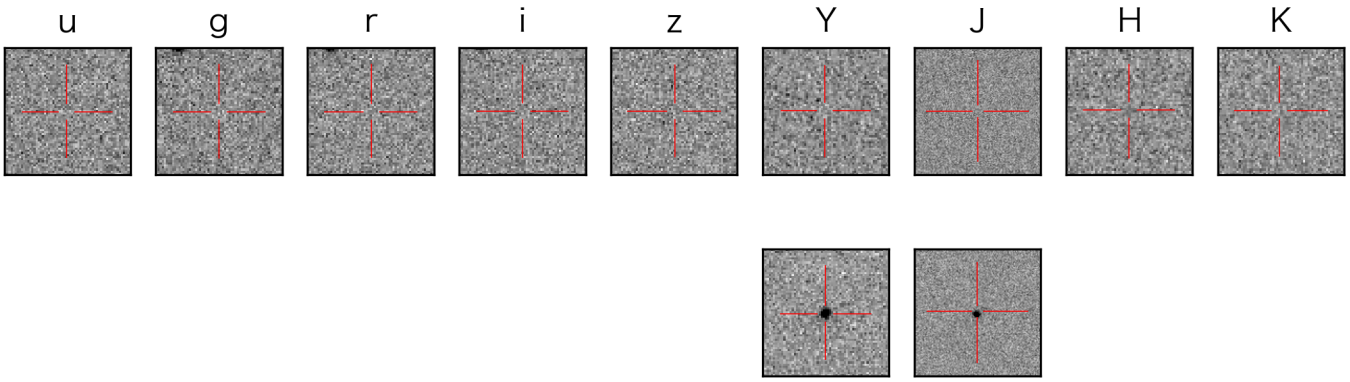


Figure 9.13: Candidate J131517.90−002125.0 that was visually rejected but has been falsely accepted by the CNN.

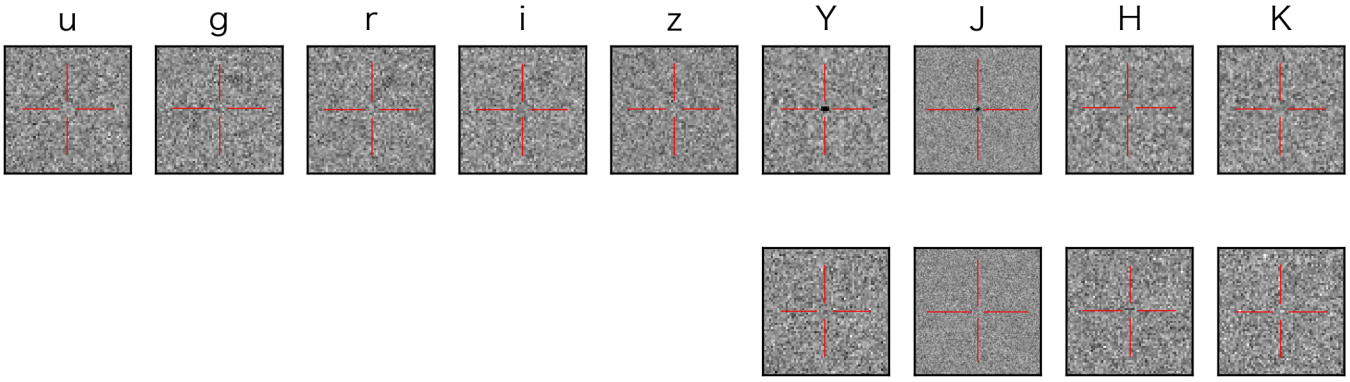


Figure 9.14: Candidate J142241.22+063553.8 that was visually rejected but has been falsely accepted by the CNN.

- Cannot handle missing data
- No prior information or likelihood included

One of the drawbacks of using image data as the input is that the CNN cannot handle varying pixel scales very well. This is in contrast to the DT and RF from chapter 8. Further, as we have explained in Section 6.1.3.1, the survey images are oriented in different ways which makes it more difficult for the CNN to find meaningful features. For future applications it might therefore be a better solution to align the input images, even if that means that some information is lost. Another drawback of the CNN is that it cannot handle missing data well such that any incomplete data has to be discarded. Finally, the classification probability is entirely based on the training set, so that neither any prior information nor likelihood are included. In this case, the classification is dependent on the survey (pixel size, ADU values) and the redshift range of the sources in the set. Therefore, if we wanted to use the CNN on either a different surveys or at a different redshift range, then the entire network would have to be retrained. This would have to be done on a new training set, which means it would have to be visually inspected and labelled. This can become time consuming very quickly.

9.6 Conclusions

We have implemented a hierarchical convolutional neural network based on postage stamp data of SDSS and UKIDSS in order to detect possible high-redshift quasar candidates. The result of AUC = 0.90 ± 0.05 gives a good classification performance that is better than the DT and the two-metric threshold classifier, but worse than the RF (see Sections 7.5 and 8.3).

The good performance comes at the cost of having to re-train the hierarchical CNN for any new survey. However, most of the computing time will come from the data processing pipeline since we remove any nearby sources. In general, the fact that the output classification is entirely dependent on the training set makes the CNN a less useful classifier compared to the RF.

Improvements can be made in the way candidates with multiple runs are summarized. Instead of averaging over several runs, it would be better to include all available data in some way. This would also create a larger set of training samples with which the CNN might even outperform the RF. In future, one could include the artificially generated postage stamp images of quasars and contaminants (as explained in Section 7.3) in the training set.

Chapter 10

Conclusion

The aim of this project was to automate the selection of high-redshift quasars for the era of Euclid and LSST. Automating this process enables us to reduce the amount of visual inspection required in the classification process, which is needed for upcoming surveys that will produce terabytes of data each day. Our methods were adapted and tested by attempting to recreate the visual classification labels of quasars at $z \gtrsim 5.2$ from Mortlock et al. (2012).

We first establish a good understanding of AGN in Chapter 1. In Chapter 2, we explain quasars as a subclass of AGN, and why finding quasars at high-redshifts is useful. Chapter 3 then gives details on what distinguishes high-redshift quasars from other astronomical sources. It also explains our approach on finding high-redshift quasars based on Bayesian model comparison and machine learning. Chapters 4 and 5 give background knowledge on these two areas.

Our methods for processing and analysing image data is laid out in Chapter 6. It explains the following two pipelines that we have built:

1. **Data processing.** Pipeline 1 assesses image quality. It fits point sources to our candidates of interest, from which their magnitudes and fluxes can be extracted.

2. **Classifies probable $z \gtrsim 5.2$ QSOs.** Pipeline 2 is used for the classification task of finding high-redshift quasars, and makes use of the results from pipeline 1.

Our first classifier is based on Bayesian model comparison and is described in Chapter 7. The next two classifiers are based on machine learning techniques: a random forest (Chapter 8) and a CNN based on the raw image data (Chapter 9). The RF uses the summarised attributes from pipeline 2, together with corresponding photometry values from the survey catalogues, as their input. The CNN uses the postage stamp images in each filter as its input instead.

Our dataset consists of cross-matched data of $\sim$ 4300 candidates from SDSS and UKIDSS that have been visually classified by Mortlock et al. (2012). We have seen that all of our classifiers perform well at reproducing the results. We have also seen that they can act as a cleaning tool, highlighting candidates that should have been classified differently. The AUC scores of the two-metric threshold classifier, the RF and the CNN are summarized in Table 10.1.

Classifier	**AUC score**
Two-metric	0.73
RF	0.94 ± 0.04
CNN	0.90 ± 0.05

Table 10.1: AUC scores of the RF and the CNN.

The RF achieved the best overall results with an AUC score of 0.94 ± 0.04. For a threshold of 0.58, a FN rate of 1% and a FP rate of 10% were obtained. The two-metric technique gave the lowest results with AUC = 0.73. However, the manual classifier is reproducible and interpretable in the sense that re-running the classifier will always give the same result. This is not the case for the ML classifiers, where re-training them on the same dataset can give different results.

The advantage of all of these classifiers is that they can be applied in an automated way, requiring less visual inspection than with previous methods. With the launch of Euclid scheduled for the end of 2023, we are looking forward to its first data release in order to test our methods of detecting high-z quasars. Having a magnitude limit of 24.0 in Y, J, H, Euclid will allow us to look deeper into the past, enabling us to detect quasars at $z \gtrsim 8$. Including optical data from LSST, which is scheduled to launch in 2024, will make the process of finding quasars at $7 < z < 7.5$ more efficient, as it will grant us to sift out more false positives. All in all, we are very excited about the next era of space telescopes and high-redshift quasar discoveries.

10.1 Future Prospects

For the next generation of photometric surveys, one can use our pipeline to detect high-redshift quasars at $z \gtrsim 5.2$. The advantage of this work is that we take photometry, morphology and stationarity into account when deciding whether a source is a likely high-redshift candidate or not. This is the first time that this combination of properties has been tested on the pixel level. In addition, the pipeline codebase can easily be extended to other surveys as it has been written in a modular fashion. One only needs to write the code for the database access and database queries, and some service-layer functions to address any particularities specific to that survey. For example, one needs to add the conversion between flux and magnitude if that is different from the existing conversions in the code.

A disadvantage is that the thresholds for the chi-squared values will need to be re-calibrated for future surveys. This is because their photometry will be much more accurate, and hence the errors will be lower. This means that the chi-squared values will increase for all model populations. To take this into account, one would have to collect a small sample set of quasar candidates to re-adjust the thresholds. Other options include trying to offset the new chi-squared values instead of collecting a whole new quasar sample. However, this would need to be tested to see if it gives good results, which again would require collecting small sample populations. The same goes for the machine learning classifiers. One could try to use the random forest with the more survey independent features without retraining it on the new dataset, though again it would be better to retrain the classifier on a new sample set.

In terms of computational requirements, one can run the pipeline in parallel using one process per candidate. The bottleneck in the computation time is the database access. That is, the majority of time is taken accessing the database for downloading images and requesting queries. However, the number of parallel processes we can use is limited by the number of simultaneous queries a database server can handle before timing out. For example, the UKIDSS database can handle far fewer simultaneous database queries compared to the SDSS server. We therefore have to scale back the number of parallel processes we use. However, our pipeline also stores the images and query results locally, meaning it is not necessary to access the database again if not needed.

All in all, applying our pipeline to upcoming surveys will require a sample set of similar size to the one used here. In addition, some code will also be needed to handle the database access, PSF form and magnitude-flux conversions.

Bibliography

Abazajian K. N., et al., 2009, ApJS, 182, 543

Abraham S., Philip N. S., 2010, Photometric Determination of Quasar Candidates. p. 147

Akhlaghi M., Ichikawa T., 2015, ApJS, 220, 1

Alam S., et al., 2015, ApJS, 219, 12

Alister Seguel P. J., Schleicher D. R. G., Boekholt T. C. N., Fellhauer M., Klessen R. S., 2020, MNRAS, 493, 2352

Allard F., 2014, in Booth M., Matthews B. C., Graham J. R., eds, Vol. 299, Exploring the Formation and Evolution of Planetary Systems. pp 271–272, doi:10.1017/S1743921313008545

Allard F., Baraffe I., Chabrier G., Barman T. S., Hauschildt P. H., 2003, in Deming D., Seager S., eds, Astronomical Society of the Pacific Conference Series Vol. 294, Scientific Frontiers in Research on Extrasolar Planets. pp 483–490

Almeida C., Baugh C. M., Lacey C. G., 2007, MNRAS, 376, 1711

Altman D., Bland J., 1995, BMJ, 311, 485

Alvarez M. A., Abel T., 2007, MNRAS, 380, L30

Anand G. S., Tully R. B., Rizzi L., Shaya E. J., Karachentsev I. D., 2019, ApJ, 880, 52

Anders P., Gieles M., de Grijs R., 2006, A&A, 451, 375

Anderson J., 2016, Empirical Models for the WFC3/IR PSF, Instrument Science Report WFC3 2016-12, 42 pages

Anguiano B., et al., 2020, AJ, 160, 43

Aniyan A. K., Thorat K., 2017, ApJS, 230, 20

Arons J., Wingert D. W., 1972, ApJ, 177, 1

Artigau É., 2018, in Deeg H. J., Belmonte J. A., eds, , Handbook of Exoplanets. p. 94, doi:10.1007/978-3-319-55333-7_94

Artigau É., Bouchard S., Doyon R., Lafrenière D., 2009, ApJ, 701, 1534

Bañados E., et al., 2016, ApJS, 227, 11

Bañados E., et al., 2018, Nature, 553, 473

Bach K., Lee H.-W., 2015, MNRAS, 446, 264

Bachchan R. K., Hobbs D., Lindegren L., 2016, A&A, 589, A71

Baeza-Yates R., 2011, Modern information retrieval : the concepts and technology behind search, 2nd ed. edn. Addison-Wesley, Harlow

Bailer-Jones C. A. L., Smith K. W., 2008, in Bailer-Jones C. A. L., ed., American Institute of Physics Conference Series Vol. 1082, American Institute of Physics Conference Series. pp 3–8 (arXiv:0809.3373), doi:10.1063/1.3059079

Bailer-Jones C. A. L., Smith K. W., Tiede C., Sordo R., Vallenari A., 2008, MNRAS, 391, 1838

Bailer-Jones C. A. L., Fouesneau M., Andrae R., 2019, MNRAS, 490, 5615

Ball N. M., Brunner R. J., 2010, International Journal of Modern Physics D, 19, 1049

Bambi C., 2018, Annalen der Physik, 530, 1700430

Barnett R., Warren S. J., Cross N. J. G., Mortlock D. J., Fan X., Wang F., Hewett P. C., 2021, MNRAS, 501, 1663

Baron D., 2019, arXiv e-prints, p. arXiv:1904.07248

Barthel P. D., 1989, ApJ, 336, 606

Barvainis R., 1987, ApJ, 320, 537

Bayes T., Price R., 1763, Philosophical Transactions of the Royal Society of London, 53, 370

Becker R. H., et al., 2001, The Astronomical Journal, 122, 2850–2857

Becker G. D., Bolton J. S., Lidz A., 2015, PASA, 32, e045

Beckmann V., Shrader C. R., 2012, Active Galactic Nuclei. John Wiley & Sons, Ltd

Bengio Y., Courville A., Vincent P., 2012, arXiv e-prints, p. arXiv:1206.5538

Berger J. O., 1985, Statistical decision theory and Bayesian analysis, second edition. edn. Springer Series in Statistics, Springer, New York, [New York

Bertin E., 2011, in Evans I. N., Accomazzi A., Mink D. J., Rots A. H., eds, Astronomical Society of the Pacific Conference Series Vol. 442, Astronomical Data Analysis Software and Systems XX. p. 435

Bhandari N., Leonard C. D., Rau M. M., Mandelbaum R., 2021, arXiv e-prints, p. arXiv:2101.00298

Blagus R., Lusa L., 2015, BMC Bioinformatics, 16

Blandford R., Netzer H., Woltjer L., Courvoisier T., Mayor M., 1990, Active galactic nuclei. Springer-Verlag

Bochanski J. J., Hawley S. L., Covey K. R., West A. A., Reid I. N., Golimowski D. A., Ivezić Ž., 2010, AJ, 139, 2679

Bolstad W. M., 2017, Introduction to Bayesian statistics, third edition. edn. Wiley, Hoboken, New Jersey

Born M., 1999, Principles of optics : electromagnetic theory of propagation, interference and diffraction of light, 7th (expanded) ed. edn. Cambridge University Press, Cambridge

Bottou L., Curtis F. E., Nocedal J., 2016, arXiv e-prints, p. arXiv:1606.04838

Bovy J., et al., 2011, ApJ, 729, 141

Bovy J., et al., 2012, ApJ, 749, 41

Box G. E. P., Tiao G. C., 1992 - 1973, Bayesian inference in statistical analysis, wiley classics library ed. edn. Wiley, New York

Breiman L., 2001, Machine Learning, 45, 5–32

Bruzual G., Charlot S., 2003, MNRAS, 344, 1000

Burgasser A. J., 2014, in Astronomical Society of India Conference Series. pp 7–16 (arXiv:1406.4887), doi:10.48550/arXiv.1406.4887

Burnham K. P., 2002, Model selection and multimodel inference : a practical information-theoretic approach, 2nd ed. edn. Springer, New York

Carballo R., González-Serrano J. I., Benn C. R., Jiménez-Luján F., 2008, MNRAS, 391, 369

Carilli C. L., et al., 2010, ApJ, 714, 834

Carlberg R. G., Yee H. K. C., Morris S. L., Lin H., Hall P. B., Patton D. R., Sawicki M., Shepherd C. W., 2001, ApJ, 552, 427

Carlin B. P., 2008, Bayesian methods for data analysis, 3rd ed. edn. Chapman & Hall/CRC texts in statistical science series, Chapman and Hall/CRC

Carrasco D., et al., 2015, A&A, 584, A44

Cartwright H., 2015, Artificial Neural Networks. Vol. 1260, doi:10.1007/978-1-4939-2239-0,

Casali M., et al., 2001, in Clowes R., Adamson A., Bromage G., eds, Astronomical Society of the Pacific Conference Series Vol. 232, The New Era of Wide Field Astronomy. p. 357

Casali M., et al., 2007, A&A, 467, 777

Cen R., Haiman Z., 2000, ApJ, 542, L75

Chawla N. V., Bowyer K. W., Hall L. O., Kegelmeyer W. P., 2011, arXiv e-prints, p. arXiv:1106.1813

Chiu K., Bunker A., Glazebrook K., Jiang L., Fan X., Liu M., Allers K., 2007, in Afonso J., Ferguson H. C., Mobasher B., Norris R., eds, Astronomical Society of the Pacific Conference Series Vol. 380, Deepest Astronomical Surveys. p. 333

Ciardi B., Ferrara A., 2005, Space Sci. Rev., 116, 625

Cireşan D. C., Meier U., Masci J., Gambardella L. M., Schmidhuber J., 2011, arXiv e-prints, p. arXiv:1102.0183

Cireşan D., Meier U., Schmidhuber J., 2012, arXiv e-prints, p. arXiv:1202.2745

Claeskens J. F., Smette A., Vandenbulcke L., Surdej J., 2006, MNRAS, 367, 879

Clarke A. O., Scaife A. M. M., Greenhalgh R., Griguta V., 2020, A&A, 639, A84

Clevert D.-A., Unterthiner T., Hochreiter S., 2015, arXiv e-prints, p. arXiv:1511.07289

Collin S., Boisson C., Mouchet M., Dumont A. M., Coupé S., Porquet D., Rokaki E., 2002, A&A, 388, 771

Colquhoun D., 2017, Royal Society open science

Cross S., Harrison R., Kennedy R., 1995, The Lancet (British edition), 346, 1075

Croton D. J., 2013, PASA, 30, e052

Cruz K. L., et al., 2007, AJ, 133, 439

D'Abrusco R., Longo G., Walton N. A., 2009, MNRAS, 396, 223

Davies F. B., et al., 2018, ApJ, 864, 143

Davies A., Serjeant S., Bromley J. M., 2019a, MNRAS, 487, 5263

Davies F. B., Hennawi J. F., Eilers A.-C., 2019b, ApJ, 884, L19

Davis S. W., Laor A., 2011, ApJ, 728, 98

Di Matteo T., Springel V., Hernquist L., 2005, Nature, 433, 604

Doert M., Errando M., 2014, ApJ, 782, 41

Doi M., et al., 2010, AJ, 139, 1628

Domingos P., 2012, Commun. ACM, 55, 78

Dundar M., Krishnapuram B., Bi J., Rao R., 2007. pp 756–761

Dye S., et al., 2006, MNRAS, 372, 1227

Eckart A., Genzel R., 1996, Nature, 383, 415

Eilers A.-C., et al., 2020, ApJ, 900, 37

Elvis M., 2000, ApJ, 545, 63

Elvis M., 2010, in Peterson B. M., Somerville R. S., Storchi-Bergmann T., eds, IAU Symposium Vol. 267, Co-Evolution of Central Black Holes and Galaxies. pp 55–64, doi:10.1017/S1743921310005570

Etz A., 2017, Advances in Methods and Practices in Psychological Science, 1

Euclid Collaboration 2019, A&A, 631, A85

Fan J.-H., 2002, PASJ, 54, L55

Fan X., et al., 2000, AJ, 120, 1167

Fan X., et al., 2001, AJ, 122, 2833

Fan X., et al., 2004, AJ, 128, 515

Fan X., Carilli C. L., Keating B., 2006, ARA&A, 44, 415

Fan X., et al., 2019, BAAS, 51, 121

Fey A. L., Eubanks M., Kingham K. A., 1997, AJ, 114, 2284

Fey A. L., et al., 2015, AJ, 150, 58

Fisher R. A., 1970, Statistical methods for research workers, 14th ed., revised and enlarged. edn. Oliver & Boyd, Edinburgh

Fornacon-Wood I., Mistry H., Johnson-Hart C., Faivre-Finn C., O'Connor J., Price G., 2022, International Journal of Radiation Oncology*Biology*Physics, 112, 1076

Francis P. J., Hewett P. C., Foltz C. B., Chaffee F. H., Weymann R. J., Morris S. L., 1991, ApJ, 373, 465

Fukushima K., 1980, Biological cybernetics, 36, 193

Fukushima K., Miyake S., 1982, Pattern recognition, 15, 455

Gaia Collaboration et al., 2016, A&A, 595, A1

Galamhos C., Matas J., Kittler J., 1999, in Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149). pp 554–560 Vol. 1, doi:10.1109/CVPR.1999.786993

Gao D., Zhang Y.-X., Zhao Y.-H., 2008, MNRAS, 386, 1417

Gelman A., 2007, Data analysis using regression and multilevel/hierarchical models. Analytical methods for social research, Cambridge University Press, Cambridge

Gelman A., 2014, Bayesian data analysis, third edition. edn. Chapman & Hall/CRC texts in statistical science, Chapman & Hall/CRC, Boca Raton, Fla ;

Géron A., 2017, Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems, first edition. edn. O'Reilly Media, Inc., Sebastopol, California

Ghez A. M., Klein B. L., Morris M., Becklin E. E., 1998, The Astrophysical Journal, 509, 678–686

Giancoli D. C., 2000 - 2000, Physics for scientists engineers, third edition. edn. Prentice Hall, Upper Saddle River, N.J

Gibson E., 2020, Statistics in Biopharmaceutical Research, 13, 1

Girardi M., Biviano A., Giuricin G., Mardirossian F., Mezzetti M., 1993, ApJ, 404, 38

Gonzalez R. C., 2018, Digital image processing, fourth edition, global edition. edn. Pearson, New York

Goodfellow I. J., Warde-Farley D., Mirza M., Courville A., Bengio Y., 2013, arXiv e-prints, p. arXiv:1302.4389

Goodfellow I., Bengio Y., Courville A., 2016, Deep Learning. MIT Press

Goodman S., 2008, Seminars in hematology, 45, 135

Goodrich R. W., et al., 2001, ApJ, 561, L23

Goto T., 2006, MNRAS, 371, 769

Goulding N. T., et al., 2012, MNRAS, 427, 3358

Graham M. J., Djorgovski S. G., Drake A. J., Mahabal A. A., Chang M., Stern D., Donalek C., Glikman E., 2014, MNRAS, 439, 703

Granato G. L., De Zotti G., Silva L., Bressan A., Danese L., 2004, ApJ, 600, 580

Groves B., 2007, in Ho L. C., Wang J. W., eds, Astronomical Society of the Pacific Conference Series Vol. 373, The Central Engine of Active Galactic Nuclei. p. 511 (arXiv:astro-ph/0612309)

Gunn J. E., Peterson B. A., 1965, ApJ, 142, 1633

Gunn J. E., et al., 1998, AJ, 116, 3040

Guo Z., Martini P., 2019a, ApJ, 879, 72

Guo Z., Martini P., 2019b, ApJ, 879, 72

Haardt F., Maraschi L., 1993, ApJ, 413, 507

Habouzit M., Volonteri M., Dubois Y., 2017, MNRAS, 468, 3935

Hackenberger B., 2019, Croatian Medical Journal, 60, 50

Haiman Z., 2002, ApJ, 576, L1

Hambly N. C., et al., 2008, MNRAS, 384, 637

Hamilton W., Turnbull R., 1853, Discussions on Philosophy and Literature, Education and University Reform: Chiefly from the Edinburgh Review. Harper & brothers, https://books.google.co.uk/books?id=D6atDXwP_J0C

Hazard C., Mackey M. B., Shimmins A. J., 1963, Nature, 197, 1037

He K., Zhang X., Ren S., Sun J., 2015, arXiv e-prints, p. arXiv:1512.03385

Heintz K. E., Fynbo J. P. U., Høg E., Møller P., Krogager J. K., Geier S., Jakobsson P., Christensen L., 2018, A&A, 615, L8

Hernitschek N., et al., 2016, ApJ, 817, 73

Herrera Ruiz N., Middelberg E., Norris R. P., Maini A., 2016, A&A, 589, L2

Hewett P., Warren S. J., Leggett S. K., Hodgkin S., 2006, MNRAS, 367, 454

Hinton G. E., et al., 1986, in Proceedings of the eighth annual conference of the cognitive science society. p. 12

Hinton G. E., Srivastava N., Krizhevsky A., Sutskever I., Salakhutdinov R. R., 2012, arXiv e-prints, p. arXiv:1207.0580

Ho T. K., 1995, in Proceedings of 3rd International Conference on Document Analysis and Recognition. pp 278–282 vol.1, doi:10.1109/ICDAR.1995.598994

Hogg D. W., 1999, arXiv e-prints, pp astro–ph/9905116

Hothorn T., Bühlmann P., Dudoit S., Molinaro A., Laan M., 2006, Biostatistics (Oxford, England), 7, 355

Hovatta T., et al., 2016, A&A, 596, A78

IBM Cloud Education 2021, `https://www.ibm.com/cloud/learn/underfitting`

Inoue Y., Khangulyan D., Doi A., 2021, Galaxies, 9, 36

Ioffe S., Szegedy C., 2015, arXiv e-prints, p. arXiv:1502.03167

Janitza S., Strobl C., Boulesteix A.-L., 2013, BMC bioinformatics, 14, 119

Jeffreys H., 1946, Proceedings of the Royal Society. A, Mathematical, physical, and engineering sciences, 186, 453

Jeffreys H., 1955, The British journal for the philosophy of science, 5, 275

Jeffreys H., 1961 - 1983, Theory of probability, 3rd ed. edn. Oxford science publications, Clarendon, Oxford

Jeon M., Besla G., Bromm V., 2017, ApJ, 848, 85

Jiang L., et al., 2016, ApJ, 833, 222

Jiang L., et al., 2022, Nature Astronomy, 6, 850

Jin X., Zhang Y., Zhang J., Zhao Y., Wu X.-b., Fan D., 2019, MNRAS, 485, 4539

Johnson J. L., Haardt F., 2016, PASA, 33, e007

Jolliffe I., 1986, Principal Component Analysis. Springer series in statistics, Springer-Verlag

Jumper J., et al., 2021, Nature, 596, 583

Kang S.-J., Fan J.-H., Mao W., Wu Q., Feng J., Yin Y., 2019, ApJ, 872, 189

Karachentsev I. D., Kudrya Y. N., Karachentseva V. E., Mitronova S. N., 2006, Astrophysics, 49, 450

Kashikawa N., et al., 2015, ApJ, 798, 28

Kass R. E., Wasserman L., 1996, Journal of the American Statistical Association, 91, 1343

Keel W. C., 2002a, Optical spectra of various kinds of active galactic nuclei, https://pages.astronomy.ua.edu/keel/agn/spectra.html

Keel B., 2002b, Quasars as lighthouses: the Lyman-alpha forest at low and high redshift, http://pages.astronomy.ua.edu/keel/agn/forest.html

Kellermann K. I., Sramek R., Schmidt M., Shaffer D. B., Green R., 1989, AJ, 98, 1195

Kembhavi A. K., Narlikar J. V., 1999, Quasars and active galactic nuclei : an introduction

Kim E. J., Brunner R. J., 2017, MNRAS, 464, 4463

Kimball A. E., Ivezic Z., 2007, in American Astronomical Society Meeting Abstracts. p. 132.19

Kingma D. P., Ba J., 2014, arXiv e-prints, p. arXiv:1412.6980

Kirkpatrick J. A., Schlegel D. J., Ross N. P., Myers A. D., Hennawi J. F., Sheldon E. S., Schneider D. P., Weaver B. A., 2011, ApJ, 743, 125

Kolykhalov P., Sunyaev R., 1984, Advances in space research, 3, 249

Koo D. C., Kron R. G., 1982, A&A, 105, 107

Koratkar A., Blaes O., 1999, PASP, 111, 1

Krizhevsky A., Sutskever I., Hinton G. E., 2017, Commun. ACM, 60, 84–90

Krolik J. H., 1999, Active galactic nuclei : from the central black holes to the galactic environment. Princeton series in astrophysics, Princton University Press, Princeton, NJ

Kuhn M., Johnson K., 2013 - 2013, Applied predictive modeling. Springer, New York

Laigle C., et al., 2016, ApJS, 224, 24

Lambert B., 2018, A student's guide to Bayesian statistics. SAGE, Los Angeles

Lang K. J., Waibel A. H., Hinton G. E., 1990, Neural Networks, 3, 23

Lang D., Hogg D. W., Jester S., Rix H.-W., 2009, AJ, 137, 4400

Laor A., Behar E., 2008, MNRAS, 390, 847

Laplace P. S., 1841, Memoire de l'Academie Royale des Sciences de Paris (Savants étrangers), pp 27–65

Laplace P. S., 1995, Philosophical essay on probabilities. Sources in the history of mathematics and physical sciences ; v.13, Springer, New York

Laplace P. S., 1878 - 1912, ŒUvres complètes de Laplace. Gauthier-Villars, Paris

Laureijs R., et al., 2011, arXiv e-prints, p. arXiv:1110.3193

Lawrence A., et al., 2007, MNRAS, 379, 1599

LeCun Y., Boser B., Denker J. S., Henderson D., Howard R. E., Hubbard W., Jackel L. D., 1989, Neural Computation, 1, 541

LeCun Y., Bengio Y., Hinton G., 2015, Nature (London), 521, 436

Lecun Y., Bottou L., Bengio Y., Haffner P., 1998, Proceedings of the IEEE, 86, 2278

Lemaitre G., Nogueira F., Aridas C. K., 2016, arXiv e-prints, p. arXiv:1609.06570

Lépine S., Rich R. M., Shara M. M., 2003, AJ, 125, 1598

Lewis G., 1926, Nature, 118, 874

Liu J., Jiang G., Bai Y., Chen T., Wang H., 2020, arXiv e-prints, p. arXiv:2001.07384

Lodato G., Natarajan P., 2006, MNRAS, 371, 1813

Loeb A., Barkana R., 2001, ARA&A, 39, 19

Loshchilov I., Hutter F., 2017, arXiv e-prints, p. arXiv:1711.05101

Lupton R. H., Gunn J. E., Szalay A. S., 1999, AJ, 118, 1406

Lynden-Bell D., 1969, Nature, 223, 690

Ma C., et al., 1998, AJ, 116, 516

Ma C., et al., 2009, IERS Technical Note, 35, 1

Ma Z., et al., 2019, ApJS, 240, 34

MacLeod C. L., et al., 2012, ApJ, 753, 106

MacMillan D. S., 2005, in Romney J., Reid M., eds, Astronomical Society of the Pacific Conference Series Vol. 340, Future Directions in High Resolution Astronomy. p. 477 (arXiv:astro-ph/0309826)

Madau P., Rees M. J., 2000, ApJ, 542, L69

Massari D., Helmi A., 2018, A&A, 620, A155

Matsuoka Y., et al., 2016, ApJ, 828, 26

Matsuoka Y., et al., 2018, PASJ, 70, S35

Matsuoka Y., et al., 2019, ApJ, 872, L2

Matthews T. A., Sandage A. R., 1963, ApJ, 138, 30

Matthews T. A., Morgan W. W., Schmidt M., 1964, ApJ, 140, 35

McCulloch W. S., Pitts W., 1990, Bulletin of mathematical biology, 52, 99

McHardy I. M., 1989, in Hunt J., Battrick B., eds, ESA Special Publication Vol. 296, Two Topics in X-Ray Astronomy, Volume 1: X Ray Binaries. Volume 2: AGN and the X Ray Background. pp 1111–1124

McLure R. J., Dunlop J. S., 2004, MNRAS, 352, 1390

Mehlig B., 2019, arXiv e-prints, p. arXiv:1901.05639

Meshcheryakov A. V., Glazkova V. V., Gerasimov S. V., Mashechkin I. V., 2018, Astronomy Letters, 44, 735

Mesinger A., Furlanetto S. R., 2008, MNRAS, 385, 1348

Messerli F. H., 2012, The New England journal of medicine, 367, 1562

Meusinger H., Hinze A., de Hoon A., 2011, A&A, 525, A37

Mezcua M., 2017, International Journal of Modern Physics D, 26, 1730021

Mihos C., 2018, http://burro.case.edu/Academics/Astr222/Cosmo/Models/age2.html

Miralda-Escudé J., 1998, ApJ, 501, 15

Mitchell T. M., 1997, Machine Learning, 1 edn. McGraw-Hill, Inc., New York, NY, USA

Mitra S., Choudhury T. R., Ferrara A., 2018, MNRAS, 473, 1416

Moffat A. F. J., 1969, A&A, 3, 455

Mortlock D. J., 2014, arXiv e-prints, p. arXiv:1405.4701

Mortlock D. J., 2016, in Mesinger A., ed., Astrophysics and Space Science Library Vol. 423, Understanding the Epoch of Cosmic Reionization: Challenges and Progress. p. 187 (arXiv:1511.01107), doi:10.1007/978-3-319-21957-8_7

Mortlock D. J., et al., 2009, A&A, 505, 97

Mortlock D. J., et al., 2011, Nature, 474, 616

Mortlock D. J., Patel M., Warren S. J., Hewett P. C., Venemans B. P., McMahon R. G., Simpson C., 2012, MNRAS, 419, 390

Murphy K. P., 2012, Machine learning: a probabilistic perspective. Adaptive computation and machine learning series, MIT Press, Cambridge, MA

Myers A. D., Richards G. T., Brunner R. J., Schneider D. P., Strand N. E., Hall P. B., Blomquist J. A., York D. G., 2008, ApJ, 678, 635

NASA 2021, Roman Reference Information, `https://roman.gsfc.nasa.gov/science/Roman_Reference_Information.html`

Nandra K., Pounds K. A., Stewart G. C., George I. M., Hayashida K., Makino F., Ohashi T., 1991, MNRAS, 248, 760

Nanni R., Hennawi J. F., Wang F., Yang J., Schindler J.-T., Fan X., 2021, p. arXiv:2111.03073

Neath A., Flores J., Cavanaugh J., 2017, Wiley Interdisciplinary Reviews: Computational Statistics, 10, e1420

Novikov I. D., Thorne K. S., 1973, in Black Holes (Les Astres Occlus). pp 343–450

Nyland K., et al., 2020, ApJ, 905, 74

O'Hagan A., 1994, Kendall's advanced theory of statistics. Vol.2B, Bayesian inference. Edward Arnold., London

Oke J. B., Gunn J. E., 1983, ApJ, 266, 713

Osterbrock D. E., 2006, Astrophysics of gaseous nebulae and active galactic nuclei, second edition. edn. University Science Books, Sausalito, CA

Pacucci F., Volonteri M., Ferrara A., 2015, MNRAS, 452, 1922

Padovani P., 2011, MNRAS, 411, 1547

Padovani P., Miller N., Kellermann K. I., Mainieri V., Rosati P., Tozzi P., 2011, ApJ, 740, 20

Padovani P., et al., 2017, A&A Rev., 25, 2

Pâris I., et al., 2011, A&A, 530, A50

Pâris I., et al., 2018, A&A, 613, A51

Parks D., Prochaska J. X., Dong S., Cai Z., 2018a, MNRAS, 476, 1151

Parks D., Prochaska J. X., Dong S., Cai Z., 2018b, MNRAS, 476, 1151

Parr T., Turgutlu K., Csiszar C., Howard J., 2018, Beware Default Random Forest Importances, `https://explained.ai/rf-importance/`

Pasquet-Itam J., Pasquet J., 2018a, A&A, 611, A97

Pasquet-Itam J., Pasquet J., 2018b, A&A, 611, A97

Peirani S., de Freitas Pacheco J. A., 2008, Phys. Rev. D, 77, 064023

Peng N., Zhang Y., Zhao Y., 2013, Science China Physics, Mechanics, and Astronomy, 56, 1227

Peters C. M., Richards G., 2015, in IAU General Assembly. p. 2257231

Peterson B. M. B. M., 1997, An introduction to active galactic nuclei. Cambridge University Press, Cambridge

Piatek S., Pryor C., Olszewski E. W., Harris H. C., Mateo M., Minniti D., Tinney C. G., 2003, AJ, 126, 2346

Pipien S., Cuby J. G., Basa S., Willott C. J., Cuillandre J. C., Arnouts S., Hudelot P., 2018, A&A, 617, A127

Pishro-Nik H., 2014, Introduction to probability, statistics, and random processes. Kappa Research, LLC, Blue Bell, PA

Planck M., 1991, The Theory of Heat Radiation. Dover Books on Physics Series, Dover Publications, https://books.google.co.uk/books?id=UnGPVwLybcEC

Pogson N., 1856, MNRAS, 17, 12

Prijono B., 2018, https://indoml.com/2018/03/07/student-notes-convolutional-neural-networks

Quinlan J. R., 1986, Mach. Learn., 1, 81–106

Raiffa H., Schlaifer R., 2000 - 1961, Applied statistical decision theory. Wiley classics library, Wiley, New York, Chichester

Raileanu L. E., Stoffel K., 2004, Annals of Mathematics and Artificial Intelligence, 41, 77–93

Ratnatunga K. U., Newell E. B., 1984, AJ, 89, 176

Rauch M., 1998, ARA&A, 36, 267

Rauch M., 2000, Lyman Alpha Forest. p. 2140, doi:10.1888/0333750888/2140

Reed S. L., et al., 2017, MNRAS, 468, 4702

Reed S. L., et al., 2019, MNRAS, 487, 1874

Rees M. J., 1984, ARA&A, 22, 471

Reynolds C. S., 1999, in Poutanen J., Svensson R., eds, Astronomical Society of the Pacific Conference Series Vol. 161, High Energy Processes in Accreting Black Holes. p. 178 (arXiv:astro-ph/9810018)

Richards G. T., et al., 2002, AJ, 123, 2945

Richards G. T., et al., 2004, ApJS, 155, 257

Richards G. T., et al., 2009, ApJS, 180, 67

Richards G. T., et al., 2015, ApJS, 219, 39

Rieke G. H., Rieke M. J., 1988, ApJ, 330, L33

Robbins H., Monro S., 1951, The Annals of Mathematical Statistics, 22, 400

Robertson B. E., Ellis R. S., Dunlop J. S., McLure R. J., Stark D. P., 2010, Nature, 468, 49

Robson I., 1996, Active galactic nuclei. Wiley-Praxis series in astronomy and astrophysics, Wiley in association with Praxis, Chichester

Rohrmann R. D., Rueda M. V., 2022, A&A, 667, A3

Rumelhart D. E., Hinton G. E., Williams R. J., 1986, Nature, 323, 533

Russakovsky O., et al., 2014, arXiv e-prints, p. arXiv:1409.0575

Rybicki G. B., Lightman A. P., 1979, Radiative processes in astrophysics

STScI Development Team 2018, synphot: Synthetic photometry using Astropy, Astrophysics Source Code Library, record ascl:1811.001 (ascl:1811.001)

Saha S., 2018, https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neura

Salpeter E. E., 1964, ApJ, 140, 796

Sandage A., Wyndham J. D., 1965, ApJ, 141, 328

Santos M., Soares J., Henriques Abreu P., Araujo H., Santos J., 2018, IEEE Computational Intelligence Magazine, 13, 59

Sany Laber E., Murtinho L., 2018, arXiv e-prints, p. arXiv:1810.00029

Schaefer C., Geiger M., Kuntzer T., Kneib J. P., 2018, A&A, 611, A2

Scherer D., Müller A., Behnke S., 2010. pp 92–101, doi:10.1007/978-3-642-15825-4_10

Schindler J.-T., et al., 2019, ApJS, 243, 5

Schmidt M., 1963, Nature, 197, 1040

Schmidt M., 1969, ARA&A, 7, 527

Schmidt M., Green R. F., 1983, ApJ, 269, 352

Schneider D. P., et al., 2010, AJ, 139, 2360

Shakura N. I., Sunyaev R. A., 1973, A&A, 500, 33

Shapiro P. R., Giroux M. L., 1987, ApJ, 321, L107

Sheth R. K., Diaferio A., 2001, MNRAS, 322, 901

Shin S., Im M., Kim Y., 2022, arXiv e-prints, p. arXiv:2208.00570

Shu Y., Koposov S. E., Evans N. W., Belokurov V., McMahon R. G., Auger M. W., Lemon C. A., 2019, MNRAS, 489, 4741

Siegel A. F., Wagner M. R., 2022, in Siegel A. F., Wagner M. R., eds, , Practical Business Statistics (Eighth Edition), eighth edition edn, Academic Press, pp 137–170, doi:https://doi.org/10.1016/B978-0-12-820025-4.00006-3, https://www.sciencedirect.com/science/article/pii/B9780128200254000063

Silber J. H., Rosenbaum P. R., 1997, Medical Care, 35, OS77

Silver D., et al., 2016, Nature, 529, 484

Silver D., et al., 2017, Nature, 550, 354

Simard P., Steinkraus D., Platt J., 2003, in Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.. pp 958–963, doi:10.1109/ICDAR.2003.1227801

Simon J. D., 2018, ApJ, 863, 89

Sivia D., Skilling J., 2006, Data Analysis: A Bayesian Tutorial. Oxford science publications, Oxford University Press, Incorporated, Oxford

Skrzypek N., Warren S. J., Faherty J. K., 2016, A&A, 589, A49

Smith H. J., Hoffleit D., 1963, AJ, 68, 292

Sobel I., 2014, Presentation at Stanford A.I. Project 1968

Soltan A., 1982, MNRAS, 200, 115

Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R., 2014, Journal of Machine Learning Research, 15, 1929

Stern D., et al., 2007, ApJ, 663, 677

Stewart M., 2019, https://towardsdatascience.com/simple-introduction-to-convolutional-neur

Stoughton C., et al., 2002, AJ, 123, 485

Strobel N., 2014, Kirchoff's Laws and Spectroscopy, https://www.e-education.psu.edu/astro801/book/export/html/1549

Strobl C., Boulesteix A.-L., Zeileis A., Hothorn T., 2007, BMC bioinformatics, 8, 25

Strobl C., Boulesteix A.-L., Kneib T., Augustin T., Zeileis A., 2008, BMC bioinformatics, 9, 307

Strobl C., Hothorn T., Zeileis A., 2009, The R Journal, 1, 14

Suchkov A. A., Hanisch R. J., Voges W., Heckman T. M., 2006, AJ, 132, 1475

Suzuki N., 2006, ApJS, 163, 110

Swinburne Astronomy Online 2008, COSMOS - The SAO Encyclopedia of Astronomy: Absorption Lines, http://astronomy.swin.edu.au/cosmos/A/Absorption+Line

Syversveen A., 1998

Szegedy C., et al., 2014, arXiv e-prints, p. arXiv:1409.4842

Tantalo R., Chinellato S., Merlin E., Piovan L., Chiosi C., 2010, A&A, 518, A43

Tee W. L., Fan X., Wang F., Yang J., Malhotra S., Rhoads J. E., 2023, ApJ, 956, 52

Temple M. J., Hewett P. C., Banerji M., 2021, MNRAS, 508, 737

Tinney C., Da Costa G., Zinnecker H., 1996, The Messenger, 83, 29

Totani T., et al., 2014, PASJ, 66, 63

Trakhtenbrot B., Volonteri M., Natarajan P., 2017, ApJ, 836, L1

Tuccillo D., González-Serrano J. I., Benn C. R., 2015, MNRAS, 449, 2818

Tversky A., Kahneman D., 1974, Science (American Association for the Advancement of Science), 185, 1124

Urry C. M., Padovani P., 1995, PASP, 107, 803

Vagabond J., 2013, Black Body Radidation, https://esfsciencenew.wordpress.com/2013/10/29/black-body-radiation/

Vanden Berk D. E., et al., 2001, AJ, 122, 549

Vanden Berk D. E., et al., 2004, ApJ, 601, 692

Venemans B. P., McMahon R. G., Warren S. J., Gonzalez-Solares E. A., Hewett P. C., Mortlock D. J., Dye S., Sharp R. G., 2007, MNRAS, 376, L76

Venemans B. P., et al., 2013, ApJ, 779, 24

Venugopal V., Bhagdikar P. S., 2012, arXiv e-prints, p. arXiv:1202.4572

Vestergaard M., Osmer P. S., 2009, ApJ, 699, 800

Volonteri M., 2010, A&A Rev., 18, 279

Wagenveld J. D., Saxena A., Duncan K. J., Röttgering H. J. A., Zhang M., 2022, A&A, 660, A22

Walker M. G., Mateo M., Olszewski E. W., 2009, AJ, 137, 3100

Wang R., et al., 2008, ApJ, 687, 848

Wang F., et al., 2016, ApJ, 819, 24

Wang F., et al., 2018a, arXiv e-prints, p. arXiv:1810.11926

Wang F., et al., 2018b, ApJ, 869, L9

Wang F., et al., 2021, ApJ, 907, L1

Wang H., et al., 2023, Nature, 620, 47

Warren S., Hewett P., 2002, in Metcalfe N., Shanks T., eds, Astronomical Society of the Pacific Conference Series Vol. 283, A New Era in Cosmology. p. 369 (arXiv:astro-ph/0201216)

Warren S. J., Hewett P. C., Osmer P. S., 1994, ApJ, 421, 412

Weber R., Schek H.-J., Blott S., 1998, in Proceedings of the 24rd International Conference on Very Large Data Bases. VLDB '98. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, p. 194–205

Weedman D. W., 1986, Quasar astronomy. Cambridge Astrophysics, Cambridge University Press

Weinberg S., 1972, Gravitation and Cosmology: Principles and Applications of the General Theory of Relativity

Wenzl L., et al., 2021, AJ, 162, 72

White R. L., et al., 2000, ApJS, 126, 133

Wikipedia contributors, Lyman series, https://en.wikipedia.org/wiki/Lyman_series

Williams P. K. G., 2018, in Deeg H. J., Belmonte J. A., eds, , Handbook of Exoplanets. p. 171, doi:10.1007/978-3-319-55333-7_171

Willott C., 2011, Nature, 474, 583

Willott C. J., et al., 2007, AJ, 134, 2435

Willott C. J., et al., 2010, AJ, 139, 906

Wilson A. S., Colbert E. J. M., 1995, ApJ, 438, 62

Wise J. H., 2012, arXiv e-prints, p. arXiv:1201.4820

Wittman D., , Fisher Matrix for Beginners, https://wittman.physics.ucdavis.edu/Fisher-matrix-guide.pdf

Wright E. L., 2004, Lyman Alpha Forest, http://www.astro.ucla.edu/~wright/Lyman-alpha-forest.html

Wright E. L., et al., 2010, AJ, 140, 1868

Xie G. Z., Zhang Y. H., Fan J. H., Liu F. K., 1993, A&A, 278, 6

Xin B., Ivezić Ž., Lupton R. H., Peterson J. R., Yoachim P., Jones R. L., Claver C. F., Angeli G., 2018, AJ, 156, 222

Yang J., et al., 2019, AJ, 157, 236

Yang J., et al., 2020, ApJ, 897, L14

Yèche C., et al., 2010, A&A, 523, A14

Yip C. W., et al., 2004, AJ, 128, 2603

York D. G., et al., 2000, AJ, 120, 1579

Yu Q., Tremaine S., 2002, MNRAS, 335, 965

Zaroubi S., 2013, in Wiklind T., Mobasher B., Bromm V., eds, Astrophysics and Space Science Library Vol. 396, The First Galaxies. p. 45 (arXiv:1206.0267), doi:10.1007/978-3-642-32362-1_2

Zellem R. T., et al., 2022, arXiv e-prints, p. arXiv:2202.05923

Zhang C., Bengio S., Hardt M., Recht B., Vinyals O., 2016, arXiv e-prints, p. arXiv:1611.03530

Zhang A., Lipton Z. C., Li M., Smola A. J., 2021, arXiv e-prints, p. arXiv:2106.11342

Zhao Y., Zhang Y., 2008, Advances in Space Research, 41, 1955

Zheng W., Kriss G. A., Telfer R. C., Grimes J. P., Davidsen A. F., 1997, ApJ, 475, 469

de L'Hospital G., 1696, Analyse des infiniment petits pour l'intelligence des lignes courbes. de L'Imprimerie royale

IBM Research, https://web.archive.org/web/20081216010431/http://www.research.ibm.com/deepblue/home/html/b.shtml

van den Bergh S., Herbst E., Pritchet C., 1973, AJ, 78, 375

van der Wel A., et al., 2014, ApJ, 788, 28

Appendix A

Visual classification labels

SDSS	UKIDSS	Summary	Explanation
A	a	acceptable	candidate is acceptable
B	b	bright-star	candidate is affected by a nearby bright star
C	c	cross-talk	candidate is (UKDISS) cross-talk
D	d		
E	e	extended	candidate is extended (hence possible mis-match)
F	f	flaw	candidate is coincident with a detector flaw
G	g	galaxy	candidate is (or is in/near) a galaxy
H	h	halo	candidate is in the halo of a bright star
I	i	image	candidate is visible (expected or not)
J	j	repeat *J*	candidate has repeat *J* measurements
K	k		
L	l	low-noise	candidate has unrealistically low noise
M	m	moving	candidate is moving (e.g., an asteroid)
N	n	noise	candidate is noise
O	o	outside	candidate is outside the survey's area
P	p	pair	candidate is one of a close pair
Q	q	quasar	candidate is a confirmed quasar
R	r	rejected	candidate is rejected
S	s	spike	candidate is close to a diffraction spike
T	t	trail	candidate is close to a satellite or meteor trail
U	u	unexamined	candidate is yet to be examined
V	v	variable	candidate is time-variable (e.g., a supernova)
W	w	wrong	candidate photometry is wrong (too faint)
X	x		
Y	y		
Z	z	persistence	candidate is persistence image

Table A.1: Visual classification labels

The candidates have been classified into different categories by (Mortlock et al., 2012). These categories are summarised in Table A.1. We show postage stamp plots for each category that is present

in our dataset. The most important classification is *R* for *rejection* and *A* for *acceptance*. Lower-case characters represent the classification for the UKIDSS images, and upper-case letters represent the classification of the images in SDSS. A candidate is accepted if it is accepted in both SDSS and UKIDSS, and therefore contains both 'A' and 'a' in its label. Otherwise, the candidate is rejected.

Candidates were first visually inspected in the SDSS dataset. If the candidate was already rejected in the SDSS dataset (R), then the corresponding images in the UKIDSS dataset were often not examined (u).

A / a The candidate is accepted. An example of a candidate that has been accepted in both SDSS and UKIDSS is shown in Figure A.1.

B / b The candidate is affected by a nearby bright star. An example of such a classification can be seen in Figure A.2. Here, the candidate has classifications *IB u*. This means that the candidate is visible in the cross hairs and is affected by a bright star. Moreover, the location has not been examined in UKIDSS yet. The nearby star overlaps with the central source of interest and therefore the photometry cannot be properly calculated. This gives rise to large reduced chi-squared values of $\tilde{\chi}^2 = 174.58$ and $\tilde{\chi}^2_{\rm max} = 588.55$.

C / c The candidate is cross-talk. An example is shown in Figure A.3, where the cross-talk is clearly visible in band *Y*, which also has $\tilde{\chi}^2_{\rm max} = 5.37$. Another example is shown in Figure A.4, where cross-talk is clearly visible in the UKIDSS bands.

E / e The candidate is extended. An example of such visual classification can be seen in Figure A.5. Since the candidate had been rejected in SDSS, the UKIDSS images were not observed. However, this could be with $P_{\rm q} = 0.999$ a good quasar candidate.

F / f The candidate is coincident with a detector flaw. This can be seen in Figure *A*.6, where a line is visible in UKIDSS. This is especially important as the flaw seems to occur within the cross hairs where the source is located. The classification of J084758.34+020504.1 is given by *Afr*, which means that the source had been visually accepted in SDSS, but was rejected due to the detector flaw in UKIDSS.

G / g The candidate is in/near a galaxy. A galaxy is clearly visible in Figure A.7.

H / h The candidate is in the halo of a bright star. An example can be seen in Figure A.8. Candidate J081959.57+064822.4 is clearly in the halo of a near bright star. Our data processing pipeline seems to be able to deal better with the influence of the bright nearby star and therefore seems to obtain more accurate photometry than the database does (e.g. see band *z*). Another example is shown in Figure A.9, which is also in the halo of a bright star and one of a close pair.

I / i The candidate is visible, as is exemplified in Figure *A*.1.

J / j The candidate has more than one measurement in *J*. This is shown in Figure A.10, which shows candidate J083700.81+241219.2 with classification *A jr*.

L / l The candidate has unrealistically low noise. An example of this classification can be seen in Figure *A*.11.

M / m The candidate is moving, e.g. it is an asteroid or a star. This is illustrated in Figure A.12, where candidate J112442.42+003517.8 is moving in UKIDSS.

N / n The candidate is noise. This is exemplified in Figure A.13.

O / o The candidate is outside the survey's area. This can again be seen in Figure A.13.

P / p The candidate is one of a close pair. An example candidate with this classification can be seen in Figure *A*.14, where a nearby star is visible in UKIDSS.

R / r The candidate is rejected. Some examples are shown in Figures A.8, A.9 and A.14.

S / s The candidate is close to a diffraction spike. Candidate J132600.21+082335.1 has such a visual classification label and can be seen in Figure A.15.

T / t The candidate is close to a satellite or meteor trail. Candidate J154642.41−014756.6 in Figure *A*.16 is an example of that.

V / v The candidate is time-variable, e.g. a supernova. This is illustrated in Figure A.17.

W / w The photometry of the candidate is wrong. This can occur if e.g. the object is too faint or if nearby sources falsify the photometry. This is illustrated in Figure *A*.18 which shows candidate J141902.93+001740.5 with classification *IWR ai*.

Z / z Candidate is persistence image. This is illustrated in Figure A.19.

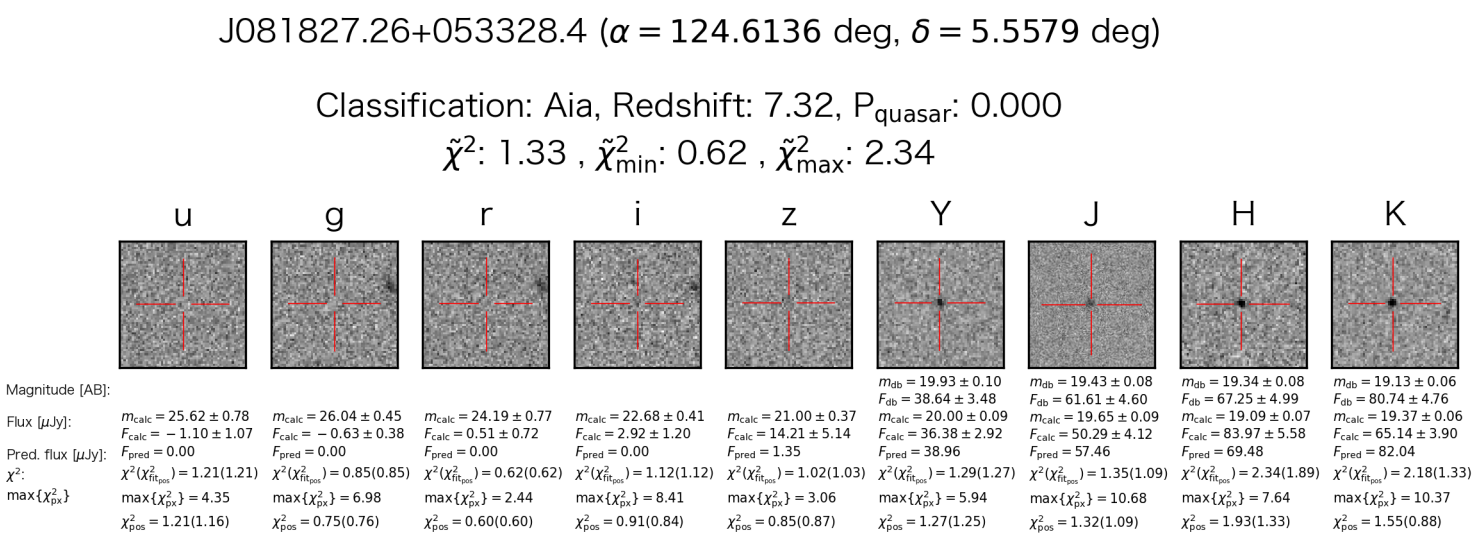


Figure A.1: Candidate J081827.26+053328.4 with classification A ia. The candidate is: (A) accepted in SDSS, (i) visible, (a) accepted in UKIDSS.

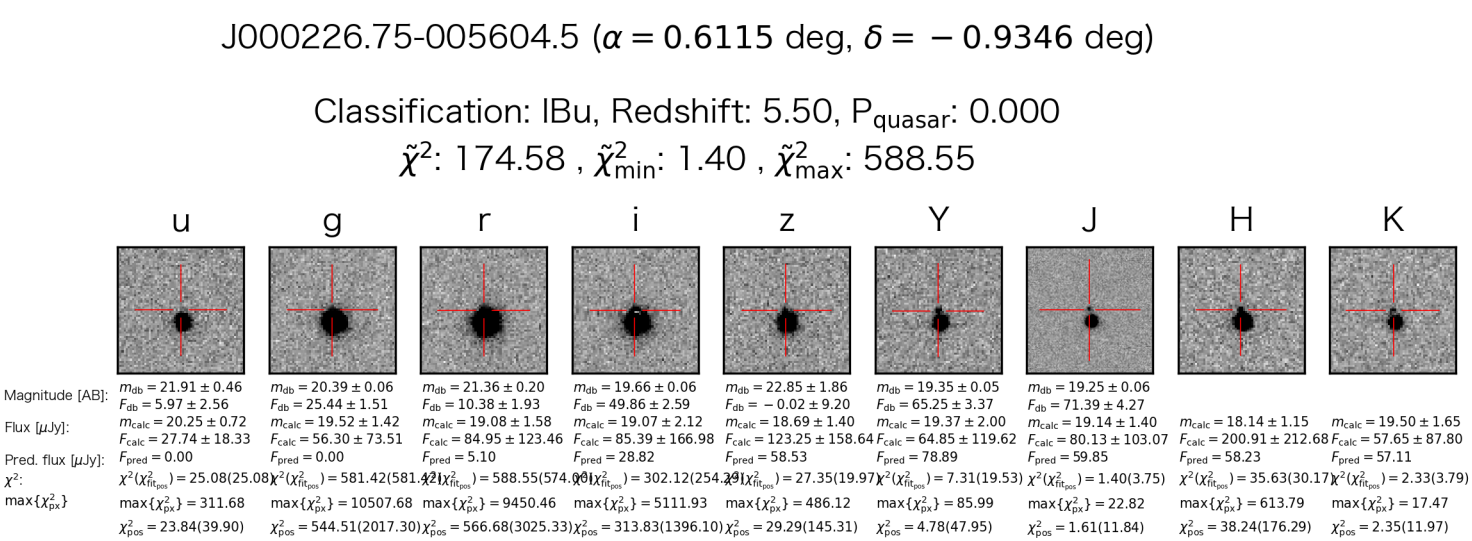


Figure A.2: Candidate J000226.75−005604.5 with classification IB u. The candidate is: (I) visible, (B) affected by a nearby bright star, (u) unobserved in UKIDSS.

J013035.08-002823.8 ($\alpha = 22.6462$ deg, $\delta = -0.4733$ deg)

Classification: Aicr, Redshift: 7.10, P_{quasar}: 0.021
$\tilde{\chi}^2$: 1.38 , $\tilde{\chi}^2_{min}$: 0.75 , $\tilde{\chi}^2_{max}$: 5.37

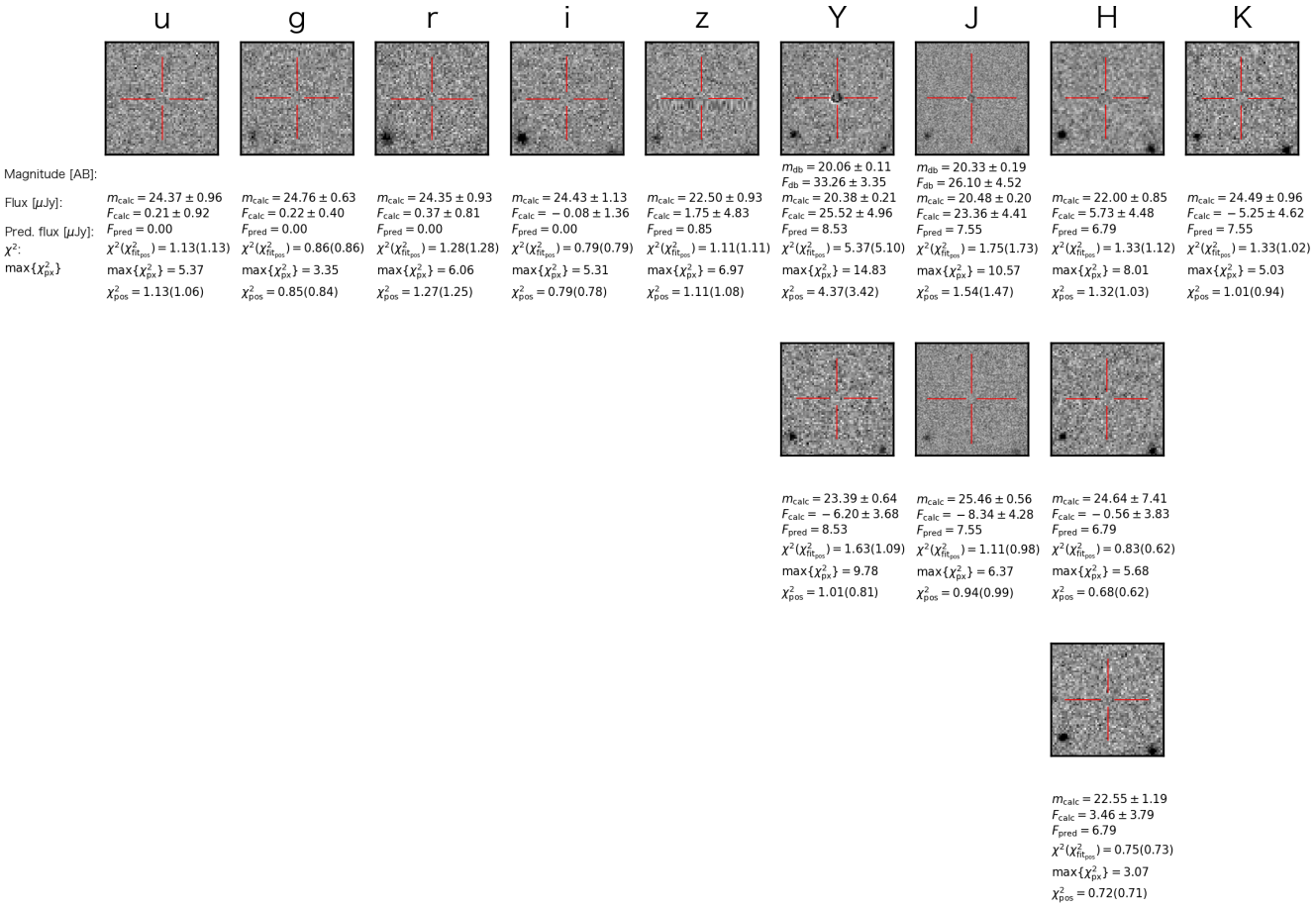


Figure A.3: Candidate J013035.08−002823.8 with classification A icr. The candidate is: (A) accepted in SDSS, (i) visible, (c) cross-talk, (r) rejected in UKIDSS.

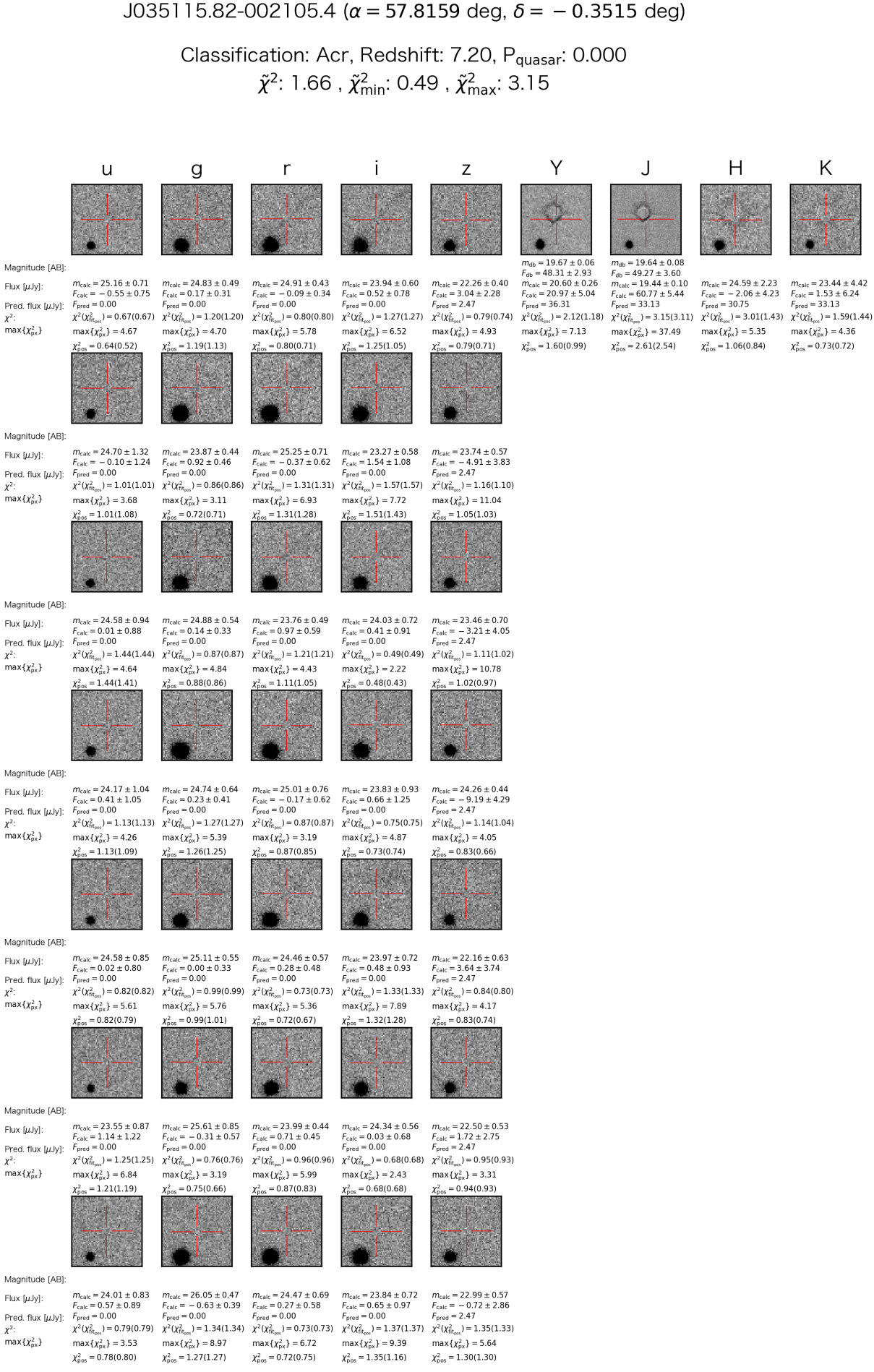


Figure A.4: Candidate J035115.82−002105.4 with classification A cr. The candidate is: (A) accepted in SDSS, (c) cross-talk, (r) rejected in UKIDSS.

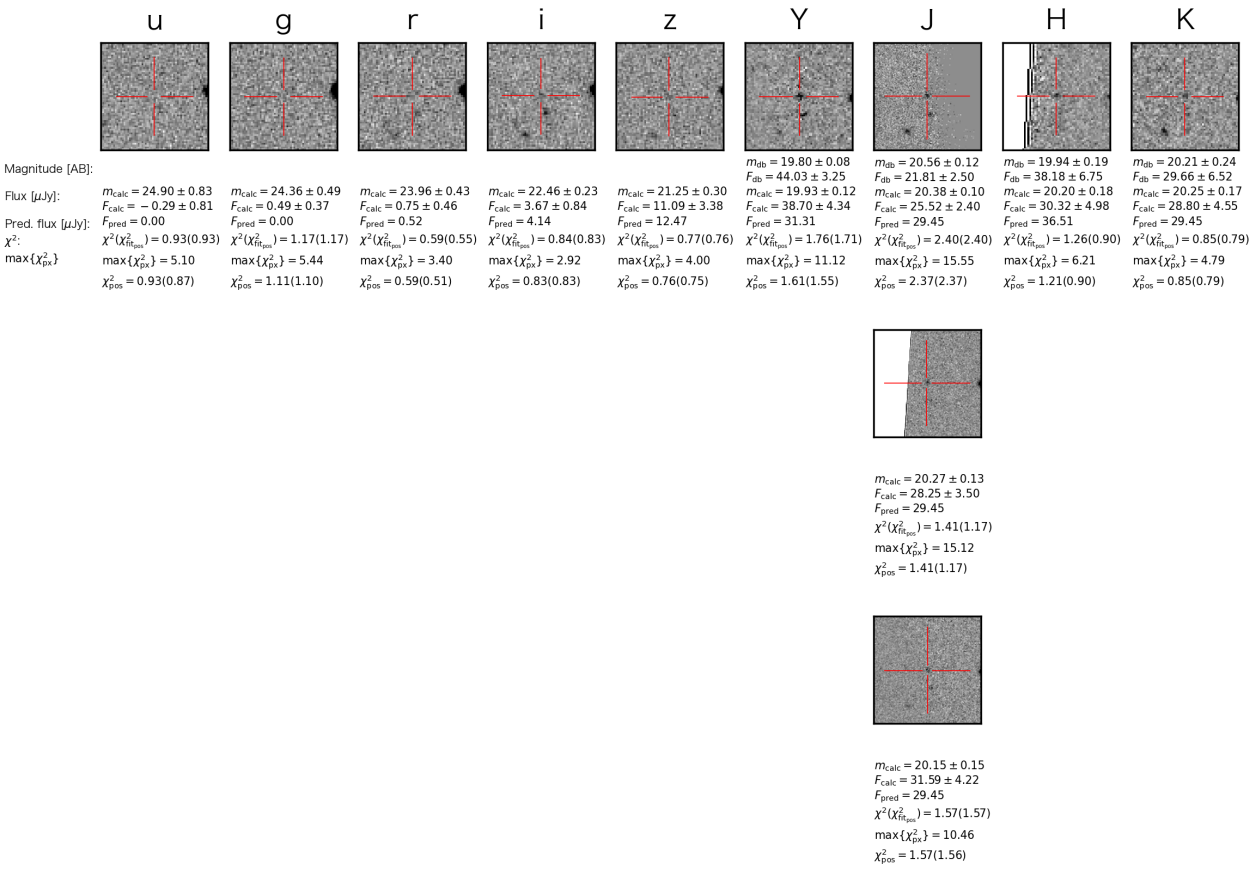


Figure A.5: Candidate J084119.73+272754.6 with classification ER. The candidate is: (E) extended, (R) rejected in SDSS.

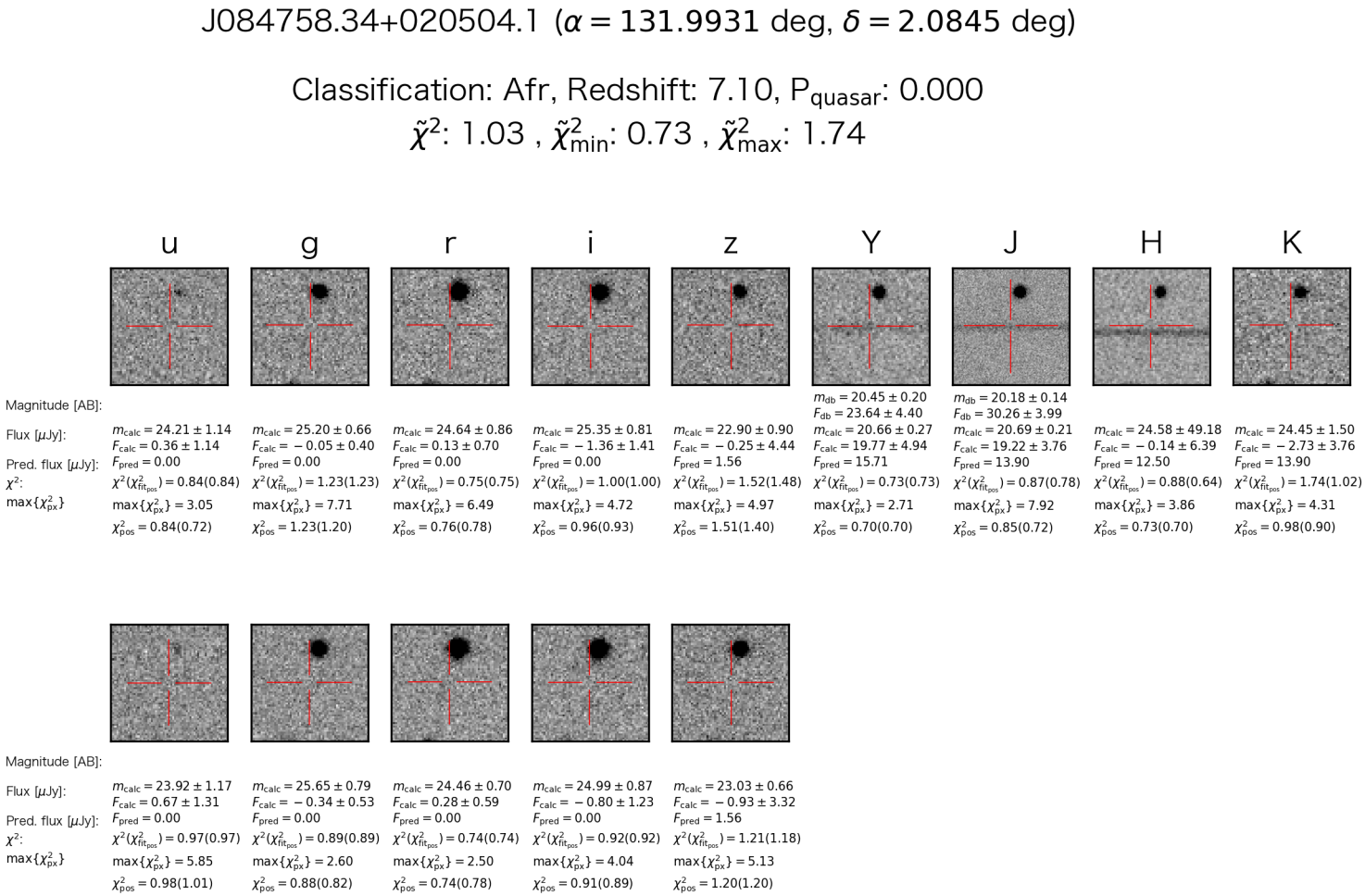


Figure A.6: Candidate J084758.34+020504.1 with classification A fr. The candidate is: (A) accepted in UKIDSS, (f) coincident with a detector flaw, (r) rejected in UKIDSS.

J002754.23-010736.9 ($\alpha = 6.9760$ deg, $\delta = -1.1269$ deg)

Classification: GRiprgr, Redshift: 5.50, P_{quasar}: 1.000
$\tilde{\chi}^2$: 3.87 , $\tilde{\chi}^2_{min}$: 0.43 , $\tilde{\chi}^2_{max}$: 15.03

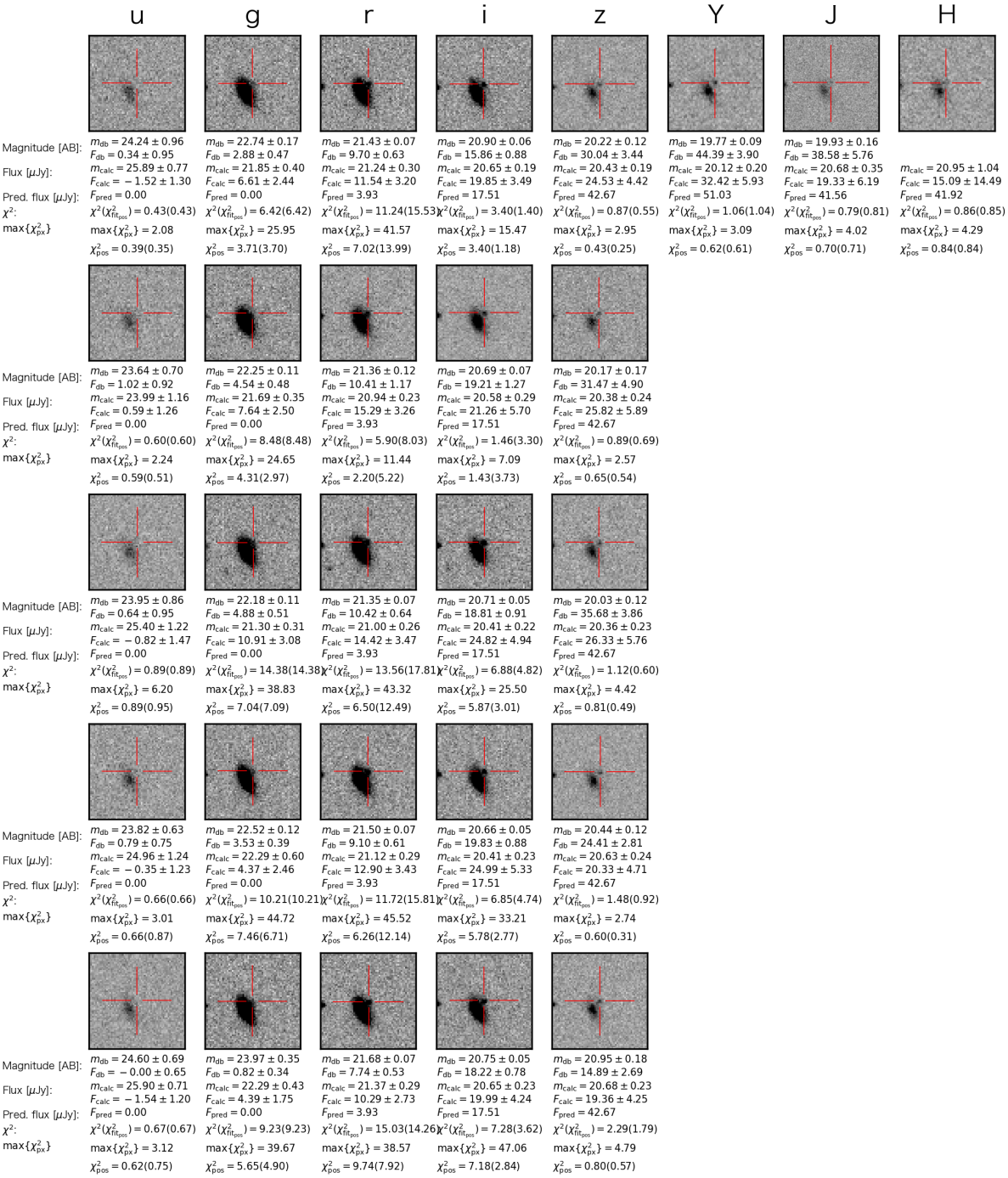


Figure A.7: Candidate J002754.23−010736.9 with classification GR iprgr. The candidate is: (G) in/near a galaxy, (R) rejected in SDSS, (i) visible, (p) one of a close pair, (r) rejected in UKIDSS.

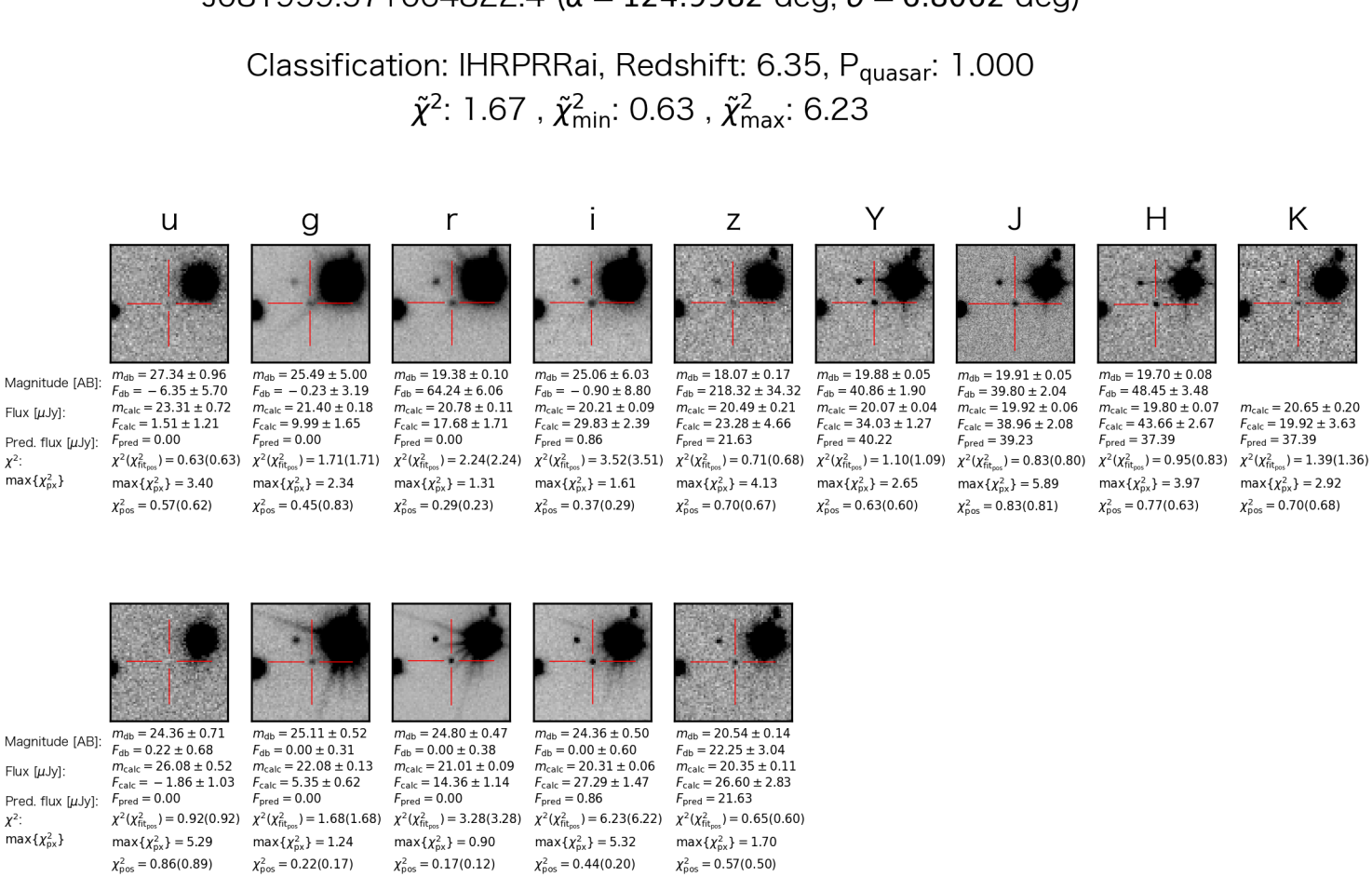
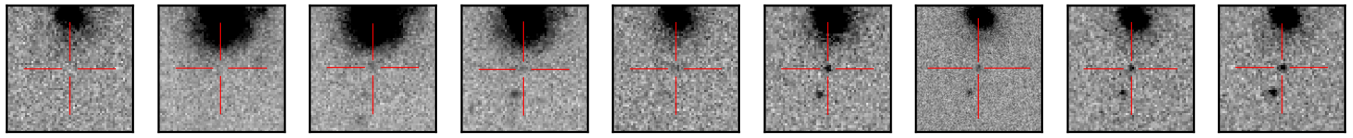


Figure A.8: Candidate J081959.57+064822.4 with classification IHRPRR ai. The candidate is: (I) visible, (H) in the halo of a bright star, (P) one of a close pair, (R) rejected in SDSS, (a) accepted in UKIDSS.

J124934.81+134759.9 ($\alpha = 192.3951$ deg, $\delta = 13.8000$ deg)

Classification: RPRHRirh, Redshift: 5.50, P_{quasar}: 0.000

$\tilde{\chi}^2$: 0.88 , $\tilde{\chi}^2_{min}$: 0.38 , $\tilde{\chi}^2_{max}$: 1.38



	u	g	r	i	z	Y	J	H	K
Magnitude [AB]:	$m_{db} = 25.56 \pm 0.76$	$m_{db} = 23.56 \pm 0.35$	$m_{db} = 22.22 \pm 0.16$	$m_{db} = 21.91 \pm 0.20$	$m_{db} = 20.46 \pm 0.21$	$m_{db} = 20.24 \pm 0.11$	$m_{db} = 20.28 \pm 0.14$	$m_{db} = 19.58 \pm 0.14$	$m_{db} = 19.28 \pm 0.09$
Flux [μJy]:	$F_{db} = -1.02 \pm 1.00$ $m_{calc} = 26.74 \pm 0.41$	$F_{db} = 1.29 \pm 0.47$ $m_{calc} = 26.38 \pm 0.52$	$F_{db} = 4.66 \pm 0.68$ $m_{calc} = 23.04 \pm 0.31$	$F_{db} = 6.19 \pm 1.19$ $m_{calc} = 22.77 \pm 0.47$	$F_{db} = 23.90 \pm 4.75$ $m_{calc} = 20.73 \pm 0.22$	$F_{db} = 29.62 \pm 3.14$ $m_{calc} = 20.37 \pm 0.11$	$F_{db} = 28.77 \pm 3.81$ $m_{calc} = 20.42 \pm 0.12$	$F_{db} = 54.47 \pm 7.07$ $m_{calc} = 19.62 \pm 0.12$	$F_{db} = 71.80 \pm 6.21$ $m_{calc} = 19.49 \pm 0.09$
	$F_{calc} = -3.60 \pm 1.42$	$F_{calc} = -0.94 \pm 0.55$	$F_{calc} = 2.12 \pm 0.65$	$F_{calc} = 2.68 \pm 1.28$	$F_{calc} = 18.57 \pm 3.95$	$F_{calc} = 25.74 \pm 2.61$	$F_{calc} = 24.71 \pm 2.82$	$F_{calc} = 51.37 \pm 5.81$	$F_{calc} = 58.02 \pm 4.99$
Pred. flux [μJy]:	$F_{pred} = 0.00$	$F_{pred} = 0.00$	$F_{pred} = 1.92$	$F_{pred} = 7.33$	$F_{pred} = 27.56$	$F_{pred} = 32.11$	$F_{pred} = 35.73$	$F_{pred} = 49.17$	$F_{pred} = 59.42$
χ^2:	$\chi^2(\chi^2_{fit_{pos}}) = 1.30(1.30)$	$\chi^2(\chi^2_{fit_{pos}}) = 0.38(0.38)$	$\chi^2(\chi^2_{fit_{pos}}) = 0.44(0.34)$	$\chi^2(\chi^2_{fit_{pos}}) = 0.68(0.58)$	$\chi^2(\chi^2_{fit_{pos}}) = 0.72(0.69)$	$\chi^2(\chi^2_{fit_{pos}}) = 1.38(1.31)$	$\chi^2(\chi^2_{fit_{pos}}) = 0.99(0.95)$	$\chi^2(\chi^2_{fit_{pos}}) = 0.85(0.85)$	$\chi^2(\chi^2_{fit_{pos}}) = 1.20(1.07)$
max$\{\chi^2_{px}\}$	max$\{\chi^2_{px}\} = 5.85$	max$\{\chi^2_{px}\} = 2.15$	max$\{\chi^2_{px}\} = 2.19$	max$\{\chi^2_{px}\} = 1.90$	max$\{\chi^2_{px}\} = 2.44$	max$\{\chi^2_{px}\} = 7.22$	max$\{\chi^2_{px}\} = 7.15$	max$\{\chi^2_{px}\} = 2.72$	max$\{\chi^2_{px}\} = 7.33$
	$\chi^2_{pos} = 1.08(1.02)$	$\chi^2_{pos} = 0.35(0.25)$	$\chi^2_{pos} = 0.44(0.28)$	$\chi^2_{pos} = 0.42(0.38)$	$\chi^2_{pos} = 0.59(0.58)$	$\chi^2_{pos} = 1.11(1.06)$	$\chi^2_{pos} = 0.85(0.82)$	$\chi^2_{pos} = 0.84(0.84)$	$\chi^2_{pos} = 1.20(1.07)$

Figure A.9: Candidate J124934.81+134759.9 with classification RPRHR irh. The candidate is: (R) rejected in SDSS, (P) one of a close pair, (H) in the halo of a bright star, (i) visible, (r) rejected in UKIDSS.

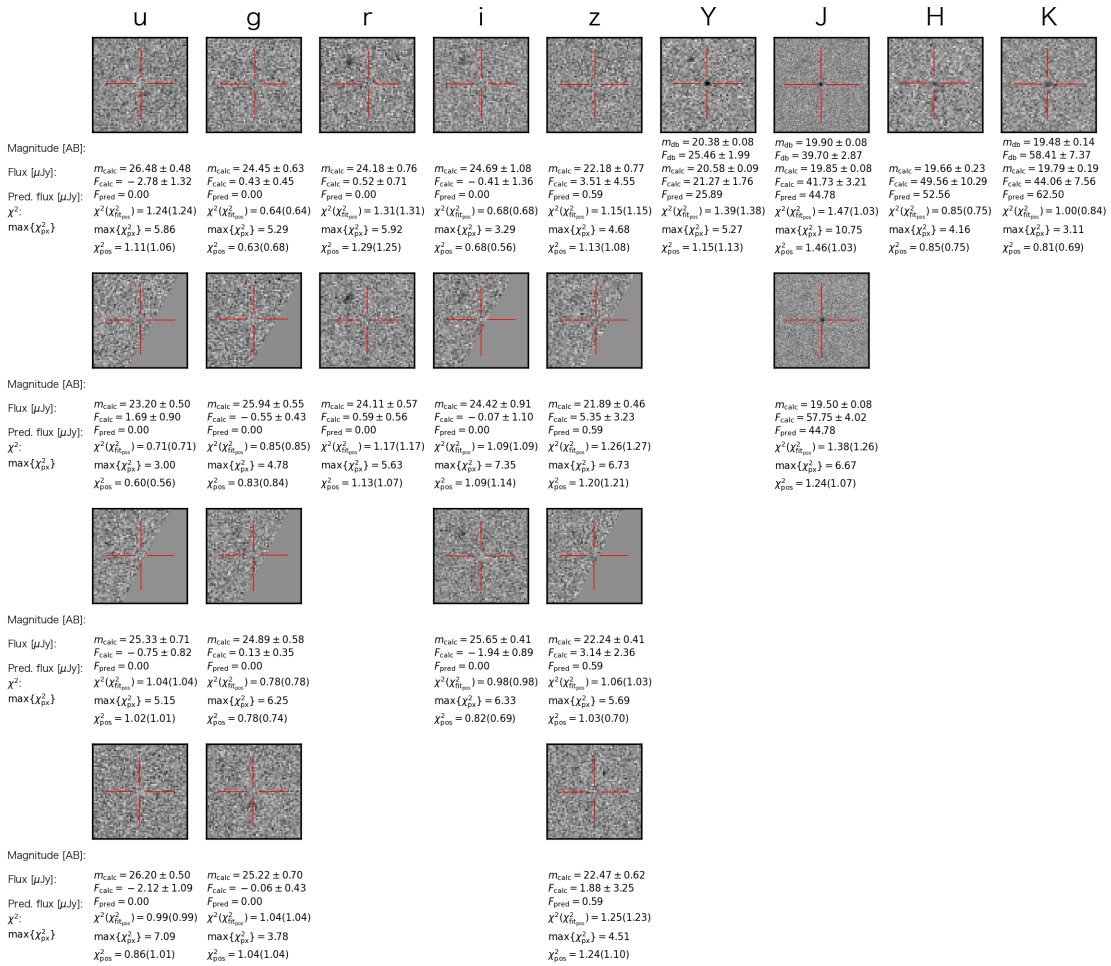
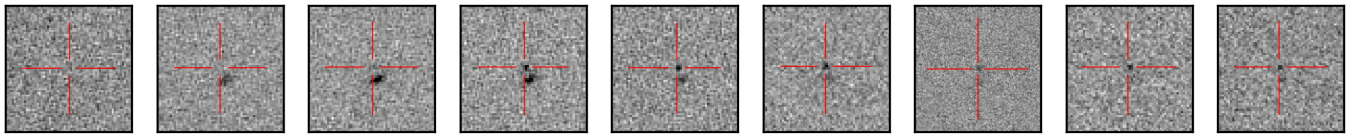


Figure A.10: Candidate J083700.81+241219.2 with classification A jr. The candidate: (A) is accepted in SDSS, (j) has a repeat J measurement, (r) is rejected in UKIDSS.

J115252.60+013045.7 ($\alpha = 178.2192$ deg, $\delta = 1.5127$ deg)

Classification: LRu, Redshift: 5.52, P_{quasar}: 0.003

$\tilde{\chi}^2$: 1.00 , $\tilde{\chi}^2_{min}$: 0.58 , $\tilde{\chi}^2_{max}$: 1.83



	u	g	r	i	z	Y	J	H	K
Magnitude [AB]:	$m_{db} = 24.17 \pm 0.53$	$m_{db} = 25.01 \pm 0.46$	$m_{db} = 23.54 \pm 0.27$	$m_{db} = 21.78 \pm 0.08$	$m_{db} = 20.96 \pm 0.14$	$m_{db} = 20.42 \pm 0.13$	$m_{db} = 20.24 \pm 0.14$	$m_{db} = 20.19 \pm 0.18$	
Flux [μJy]:	$F_{db} = 0.41 \pm 0.54$ $m_{calc} = 23.52 \pm 0.48$	$F_{db} = 0.06 \pm 0.28$ $m_{calc} = 25.39 \pm 0.59$	$F_{db} = 1.26 \pm 0.38$ $m_{calc} = 23.75 \pm 0.43$	$F_{db} = 7.01 \pm 0.53$ $m_{calc} = 21.86 \pm 0.14$	$F_{db} = 14.77 \pm 1.99$ $m_{calc} = 20.96 \pm 0.12$	$F_{db} = 24.51 \pm 3.01$ $m_{calc} = 20.63 \pm 0.13$	$F_{db} = 28.78 \pm 3.63$ $m_{calc} = 20.38 \pm 0.13$	$F_{db} = 30.00 \pm 4.96$ $m_{calc} = 20.35 \pm 0.19$	$m_{calc} = 20.55 \pm 0.20$
Pred. flux [μJy]:	$F_{calc} = 1.18 \pm 0.69$ $F_{pred} = 0.00$	$F_{calc} = -0.17 \pm 0.37$ $F_{pred} = 0.00$	$F_{calc} = 0.99 \pm 0.52$ $F_{pred} = 1.29$	$F_{calc} = 6.48 \pm 0.88$ $F_{pred} = 6.55$	$F_{calc} = 14.82 \pm 1.79$ $F_{pred} = 18.18$	$F_{calc} = 20.25 \pm 2.44$ $F_{pred} = 23.23$	$F_{calc} = 25.69 \pm 3.13$ $F_{pred} = 20.96$	$F_{calc} = 26.38 \pm 4.64$ $F_{pred} = 24.08$	$F_{calc} = 21.93 \pm 4.01$ $F_{pred} = 26.45$
χ^2:	$\chi^2(\chi^2_{fit_{pos}}) = 0.88(0.88)$	$\chi^2(\chi^2_{fit_{pos}}) = 0.58(0.58)$	$\chi^2(\chi^2_{fit_{pos}}) = 0.58(0.53)$	$\chi^2(\chi^2_{fit_{pos}}) = 1.83(1.04)$	$\chi^2(\chi^2_{fit_{pos}}) = 1.51(0.75)$	$\chi^2(\chi^2_{fit_{pos}}) = 0.87(0.85)$	$\chi^2(\chi^2_{fit_{pos}}) = 1.01(0.92)$	$\chi^2(\chi^2_{fit_{pos}}) = 1.22(0.88)$	$\chi^2(\chi^2_{fit_{pos}}) = 1.25(0.90)$
$\max\{\chi^2_{px}\}$	$\max\{\chi^2_{px}\} = 3.27$	$\max\{\chi^2_{px}\} = 3.44$	$\max\{\chi^2_{px}\} = 3.05$	$\max\{\chi^2_{px}\} = 22.12$	$\max\{\chi^2_{px}\} = 6.32$	$\max\{\chi^2_{px}\} = 4.80$	$\max\{\chi^2_{px}\} = 7.30$	$\max\{\chi^2_{px}\} = 6.51$	$\max\{\chi^2_{px}\} = 3.66$
	$\chi^2_{pos} = 0.79(0.70)$	$\chi^2_{pos} = 0.58(0.47)$	$\chi^2_{pos} = 0.56(0.52)$	$\chi^2_{pos} = 1.83(1.00)$	$\chi^2_{pos} = 1.41(0.74)$	$\chi^2_{pos} = 0.82(0.81)$	$\chi^2_{pos} = 0.98(0.87)$	$\chi^2_{pos} = 1.20(0.79)$	$\chi^2_{pos} = 1.20(0.89)$

	u	g	r	i	z
Magnitude [AB]:	$m_{db} = 25.80 \pm 0.62$	$m_{db} = 24.38 \pm 0.62$	$m_{db} = 23.45 \pm 0.41$	$m_{db} = 22.15 \pm 0.21$	$m_{db} = 20.62 \pm 0.22$
Flux [μJy]:	$F_{db} = -1.38 \pm 0.98$ $m_{calc} = 23.01 \pm 0.66$	$F_{db} = 0.48 \pm 0.46$ $m_{calc} = 25.96 \pm 0.76$	$F_{db} = 1.39 \pm 0.62$ $m_{calc} = 23.95 \pm 0.64$	$F_{db} = 4.91 \pm 0.97$ $m_{calc} = 22.29 \pm 0.31$	$F_{db} = 20.53 \pm 4.21$ $m_{calc} = 20.81 \pm 0.25$
Pred. flux [μJy]:	$F_{calc} = 2.06 \pm 1.39$ $F_{pred} = 0.00$	$F_{calc} = -0.56 \pm 0.61$ $F_{pred} = 0.00$	$F_{calc} = 0.76 \pm 0.68$ $F_{pred} = 1.29$	$F_{calc} = 4.31 \pm 1.28$ $F_{pred} = 6.55$	$F_{calc} = 17.16 \pm 4.16$ $F_{pred} = 18.18$
χ^2:	$\chi^2(\chi^2_{fit_{pos}}) = 1.11(1.11)$	$\chi^2(\chi^2_{fit_{pos}}) = 0.72(0.72)$	$\chi^2(\chi^2_{fit_{pos}}) = 0.61(0.57)$	$\chi^2(\chi^2_{fit_{pos}}) = 0.70(0.64)$	$\chi^2(\chi^2_{fit_{pos}}) = 0.76(0.76)$
$\max\{\chi^2_{px}\}$	$\max\{\chi^2_{px}\} = 3.55$	$\max\{\chi^2_{px}\} = 7.19$	$\max\{\chi^2_{px}\} = 3.14$	$\max\{\chi^2_{px}\} = 3.27$	$\max\{\chi^2_{px}\} = 6.07$
	$\chi^2_{pos} = 1.07(1.04)$	$\chi^2_{pos} = 0.69(0.65)$	$\chi^2_{pos} = 0.59(0.56)$	$\chi^2_{pos} = 0.64(0.60)$	$\chi^2_{pos} = 0.75(0.76)$

Figure A.11: Candidate J115252.60+013045.7 with classification LR u. The candidate: (L) has unrealistically low noise, (R) is rejected in SDSS, (u) has not been observed in UKIDSS.

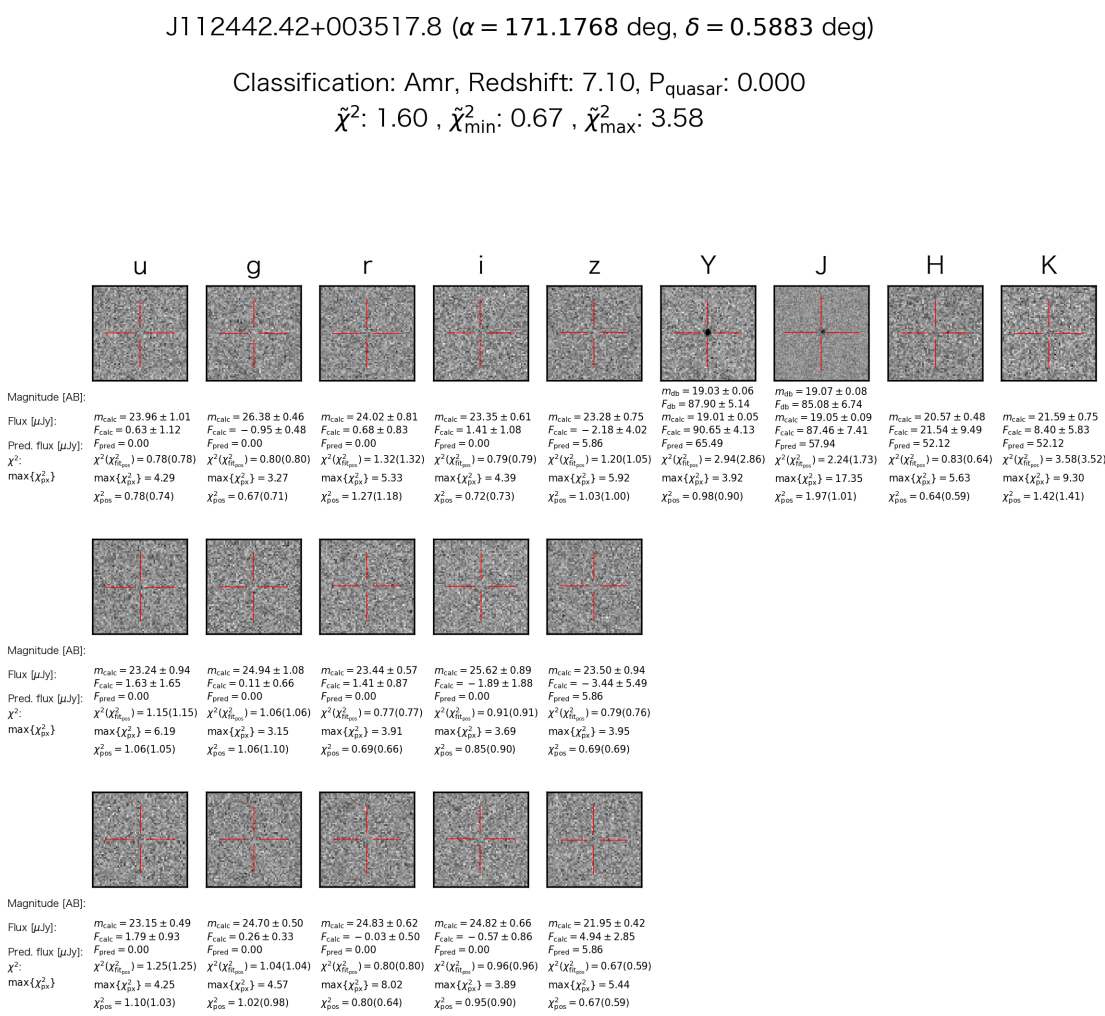


Figure A.12: Candidate J112442.42+003517.8 with classification A mr. The candidate is: (A) accepted in SDSS, (m) moving, (r) rejected in UKIDSS.

J035848.16+010102.7 ($\alpha = 59.7007$ deg, $\delta = 1.0174$ deg)

Classification: NORi, Redshift: 5.50, P_{quasar}: 0.000
$\tilde{\chi}^2$: 1267.21 , $\tilde{\chi}^2_{min}$: 8.88 , $\tilde{\chi}^2_{max}$: 9509.41

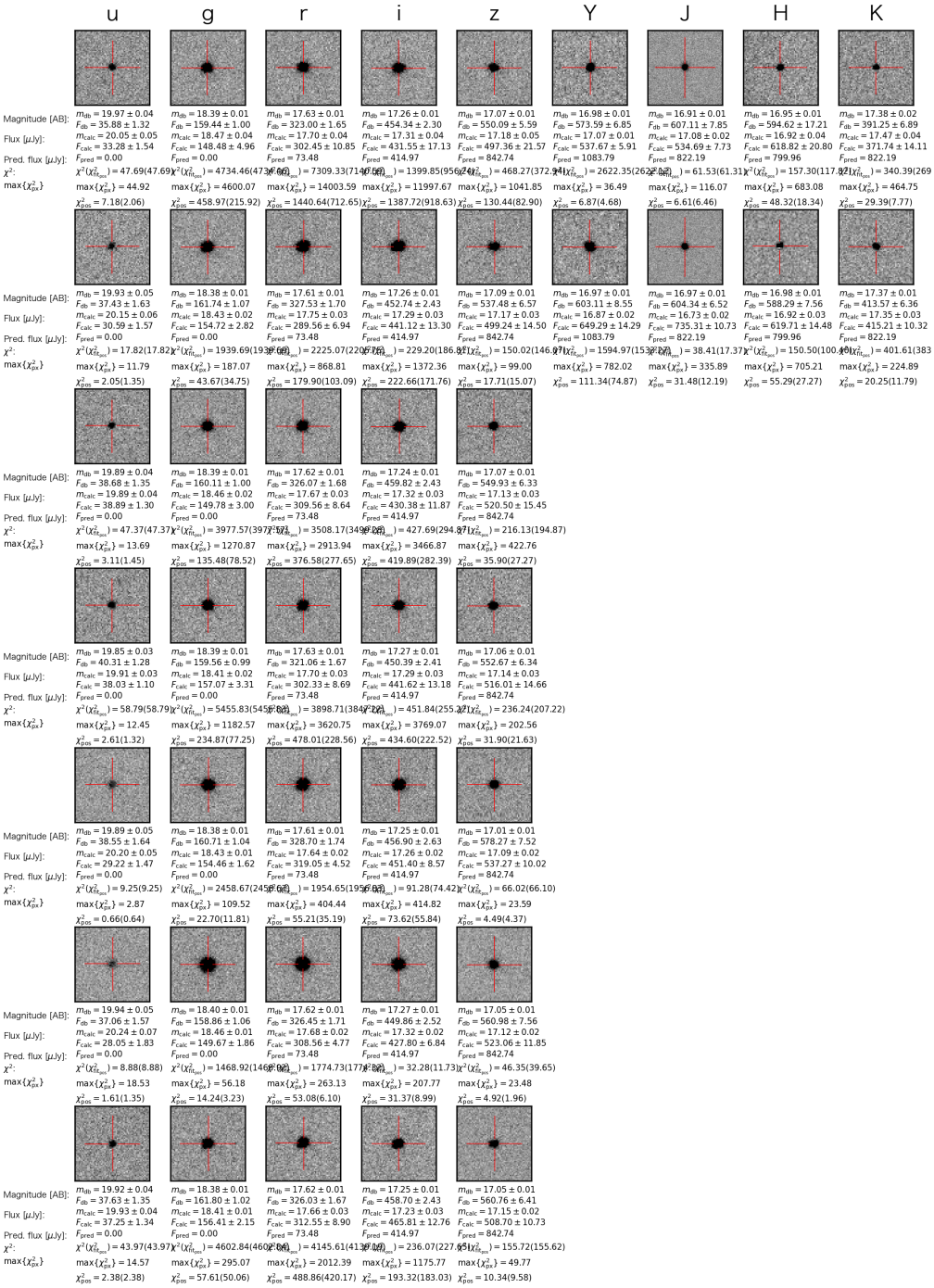


Figure A.13: Candidate J035848.16+010102.7 with classification NOR i. The candidate is: (N) noise, (O) outside the survey's area, (R) rejected in SDSS (u) unobserved in UKIDSS.

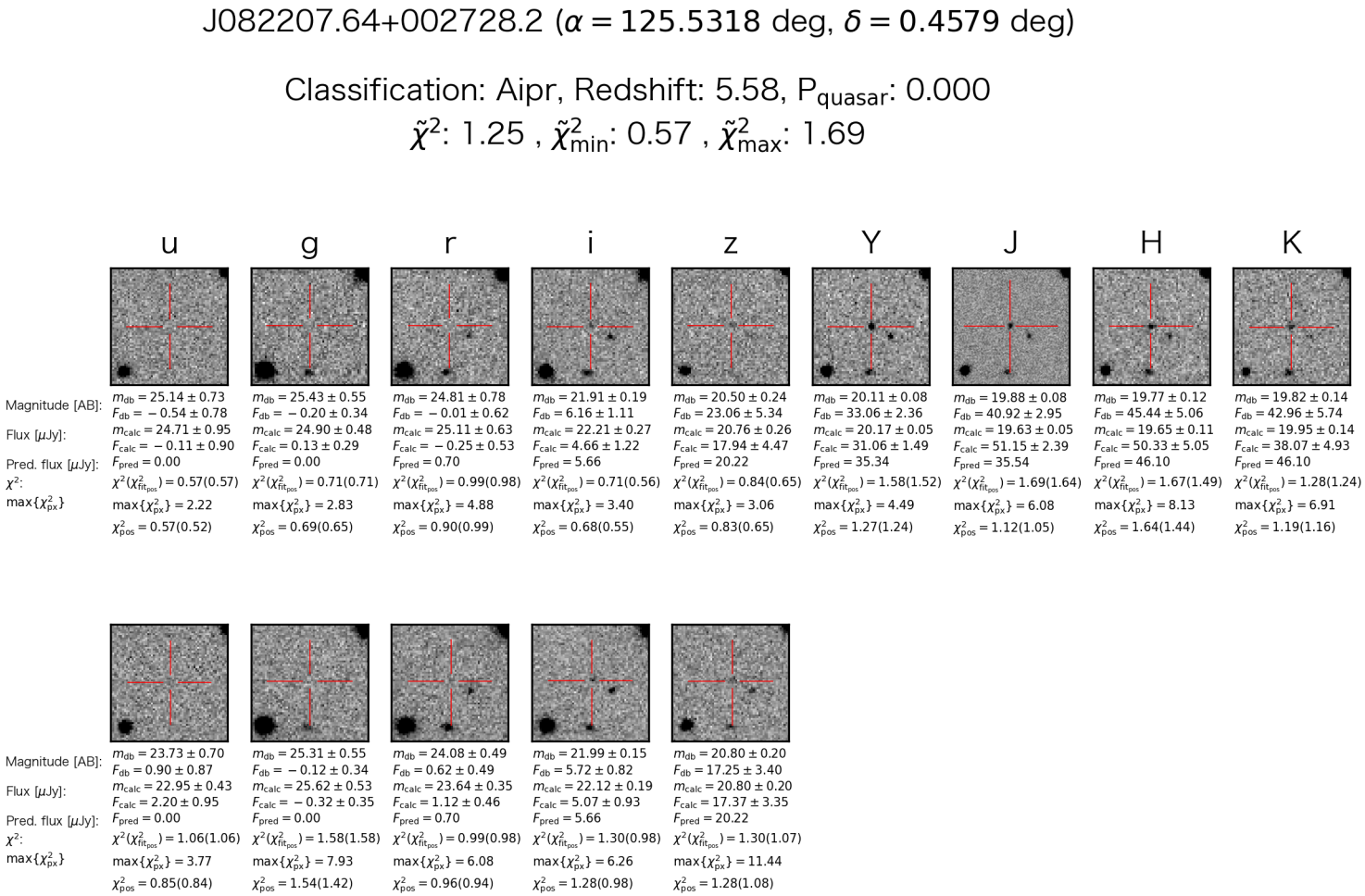


Figure A.14: Candidate J082207.64+002728.2 with classification A ipr. The candidate is: (A) accepted in SDSS, (i) visible, (p) one of a close pair, (r) rejected in UKIDSS.



Figure A.15: Candidate J132600.21+082335.1 with classification IA rsi. The candidate is: (I) visible, (A) accepted in SDSS, (s) close to a diffraction spike, (r) rejected in UKIDSS.

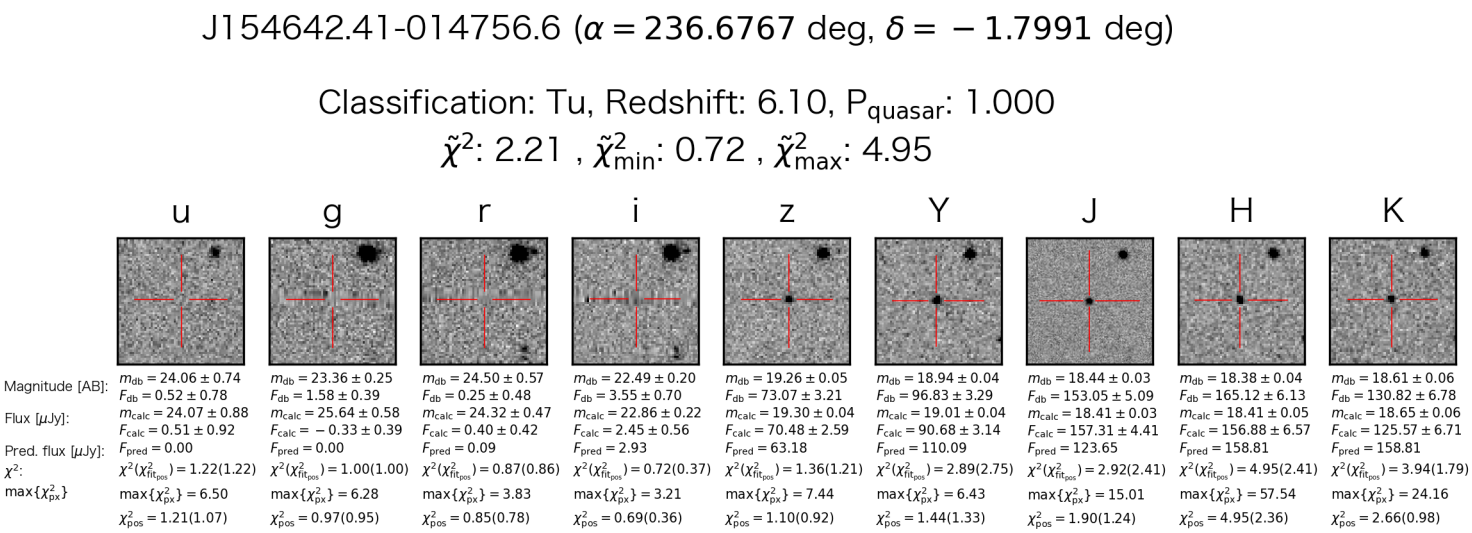
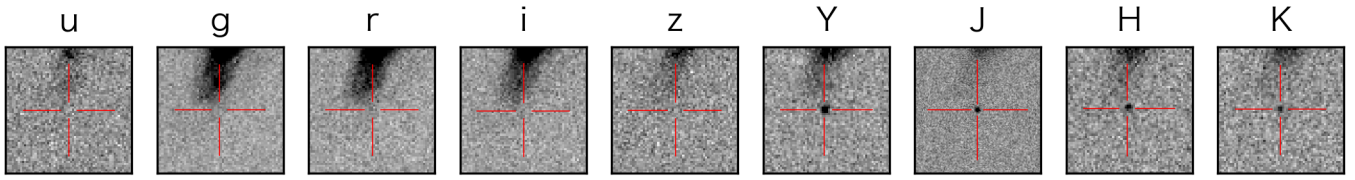


Figure A.16: Candidate J154642.41−014756.6 with classification T u. The candidate is: (T) close to a satellite or meteor trail, (u) unobserved in UKIDSS.

J143135.16+102210.9 ($\alpha = 217.8965$ deg, $\delta = 10.3697$ deg)

Classification: Avr, Redshift: 7.15, P_{quasar}: 1.000

$\tilde{\chi}^2$: 1.02 , $\tilde{\chi}^2_{min}$: 0.38 , $\tilde{\chi}^2_{max}$: 1.58



	u	g	r	i	z	Y	J	H	K
Magnitude [AB]:						$m_{db} = 19.41 \pm 0.04$	$m_{db} = 19.70 \pm 0.06$	$m_{db} = 19.71 \pm 0.09$	$m_{db} = 20.10 \pm 0.14$
Flux [μJy]:	$m_{calc} = 25.90 \pm 0.53$ $F_{calc} = -1.53 \pm 0.90$	$m_{calc} = 24.86 \pm 0.63$ $F_{calc} = 0.15 \pm 0.39$	$m_{calc} = 24.35 \pm 0.58$ $F_{calc} = 0.37 \pm 0.50$	$m_{calc} = 24.65 \pm 0.64$ $F_{calc} = -0.35 \pm 0.80$	$m_{calc} = 23.38 \pm 0.37$ $F_{calc} = -2.72 \pm 2.07$	$F_{db} = 62.47 \pm 2.53$ $m_{calc} = 19.48 \pm 0.04$ $F_{calc} = 58.61 \pm 2.10$	$F_{db} = 47.74 \pm 2.92$ $m_{calc} = 19.55 \pm 0.05$ $F_{calc} = 55.15 \pm 2.60$	$F_{db} = 47.54 \pm 4.05$ $m_{calc} = 19.69 \pm 0.09$ $F_{calc} = 48.26 \pm 3.86$	$F_{db} = 33.16 \pm 4.34$ $m_{calc} = 20.26 \pm 0.12$ $F_{calc} = 28.59 \pm 3.17$
Pred. flux [μJy]:	$F_{pred} = 0.00$	$F_{pred} = 0.00$	$F_{pred} = 0.00$	$F_{pred} = 0.00$	$F_{pred} = 4.24$	$F_{pred} = 56.41$	$F_{pred} = 50.69$	$F_{pred} = 46.31$	$F_{pred} = 46.31$
χ^2:	$\chi^2(\chi^2_{fit_{pos}}) = 1.58(1.58)$	$\chi^2(\chi^2_{fit_{pos}}) = 0.38(0.38)$	$\chi^2(\chi^2_{fit_{pos}}) = 0.59(0.59)$	$\chi^2(\chi^2_{fit_{pos}}) = 0.42(0.42)$	$\chi^2(\chi^2_{fit_{pos}}) = 1.23(0.83)$	$\chi^2(\chi^2_{fit_{pos}}) = 1.07(1.07)$	$\chi^2(\chi^2_{fit_{pos}}) = 1.36(1.33)$	$\chi^2(\chi^2_{fit_{pos}}) = 1.18(0.82)$	$\chi^2(\chi^2_{fit_{pos}}) = 1.35(1.30)$
max{χ^2_{px}}	max{χ^2_{px}} = 14.28	max{χ^2_{px}} = 1.56	max{χ^2_{px}} = 3.39	max{χ^2_{px}} = 3.63	max{χ^2_{px}} = 6.28	max{χ^2_{px}} = 4.72	max{χ^2_{px}} = 10.41	max{χ^2_{px}} = 4.11	max{χ^2_{px}} = 2.80
	$\chi^2_{pos} = 1.46(1.46)$	$\chi^2_{pos} = 0.37(0.43)$	$\chi^2_{pos} = 0.57(0.55)$	$\chi^2_{pos} = 0.42(0.39)$	$\chi^2_{pos} = 0.87(0.83)$	$\chi^2_{pos} = 1.03(1.02)$	$\chi^2_{pos} = 1.31(1.28)$	$\chi^2_{pos} = 1.16(0.77)$	$\chi^2_{pos} = 0.50(0.45)$

Figure A.17: Candidate J143135.16+102210.9 with classification A vr. The candidate is: (A) accepted in SDSS, (v) time-variable, (r) rejected in UKIDSS.

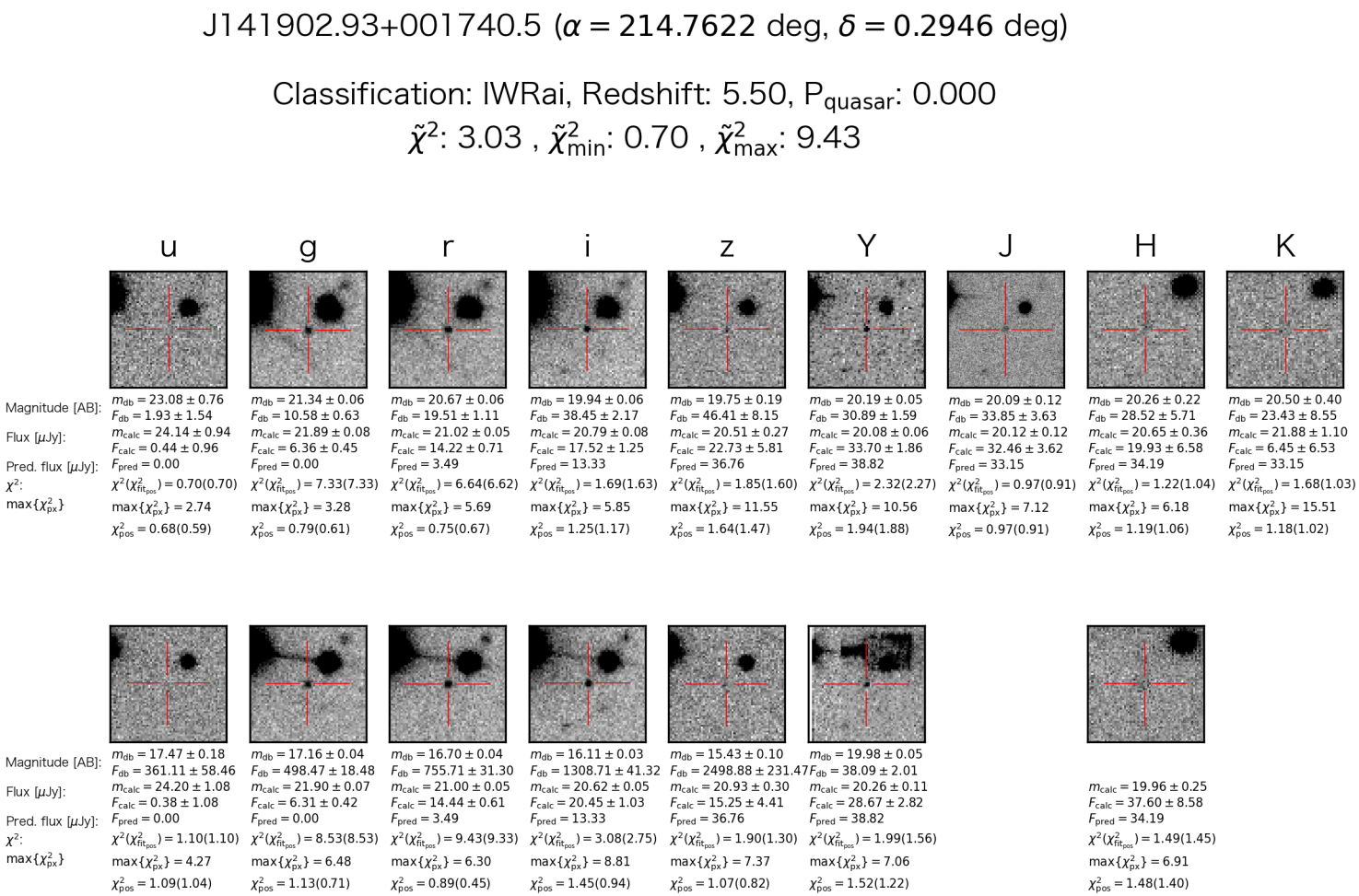


Figure A.18: Candidate J141902.93+001740.5 with classification IWR ai. The candidate is: (I) visible, (W) with wrong photometry, (R) rejected in SDSS, (i) visible, (a) accepted in UKIDSS.

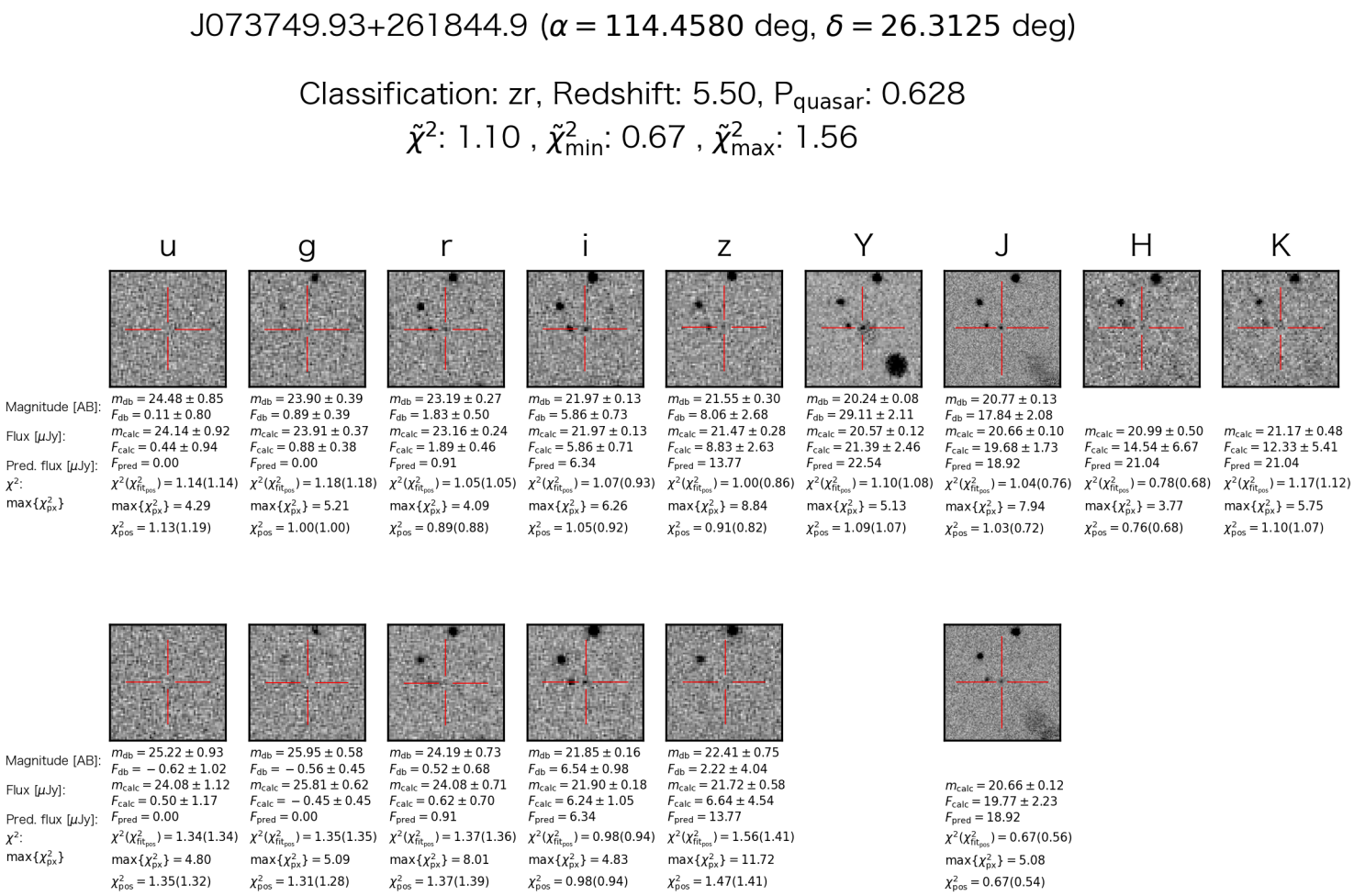


Figure A.19: Candidate J073749.93+261844.9 with classification zr. The candidate is: (z) persistence image, (r) rejected in UKIDSS.

Appendix B

Bayesian inference vs frequentist statistics

B.1 Frequentist statistics

The main difference between frequentist and Bayesian statistics is that in classical statistics the parameters $\boldsymbol{\theta}$ are considered fixed. In Bayesian statistics, however, the parameters $\boldsymbol{\theta}$ are considered random variables themselves and the observed data realisations thereof. Bayes' theorem gives us back a probability density function of the model parameters conditional on the observed data, and hence includes a description of uncertainty. With Bayes' rule, once the model has been set up one updates one's belief about the model parameters with each new observed data sample.

Compared to Bayes' statistics, frequentist inference does not assign probabilities to parameters. It also does not take into account any prior belief. It gives the probability of an outcome of an experiment based on the relative frequency of that outcome (Fornacon-Wood et al., 2022). This experiment has to be repeatable over and over again, and the outcomes have to be independent (Gelman, 2014). The uncertainty depends on the number of observed events. For example, let us take a fair coin again. If we flip the coin multiple times, the probability will get closer and closer to 0.5 for both 'heads' and 'tails'. This is based on the central limit theorem, which states that as one collects more and more data samples from a population distribution, the sampling distribution will take on a normal distribution with mean the mean of the collected samples getting closer and closer to the mean of the population. In the case of flipping a fair coin, we know that the probability is 0.5 for both 'heads' and 'tails'. By flipping the coin a large amount of times, the average ratio of showing 'heads' (or 'tails') should get closer and closer to 0.5. However, if we flipped the coin only a couple of times, we could not rule out that the coin is unfair, e.g. that it is weighted so that one side shows up more times than the other one.

B.1.1 Null hypothesis and p-values

B.1.1.1 Null hypothesis

Frequentist statistics makes use of a *null hypothesis*, which represents a hypothesis before the data is collected (Fornacon-Wood et al., 2022). For example, one might test whether a certain medicine helps to reduce deaths from Covid. The null hypothesis would then state that the new medicine has no effect on the outcome and hence that it does not alleviate Covid deaths. One would then do an experiment with a group A of Covid patients that receives that medicine and a control group B that does not receive that particular medicine. Then, one compares the Covid deaths between those two groups.

B.1.1.2 P-value

The *p-value* gives the probability of the observed outcome plus any more extreme / less likely cases combined if the null hypothesis was true (Fisher, 1970). For example, in a Gaussian probability distribution, if we were to observe $x > 0$, then the p-value would be the integral of the Gaussian from x to its tails up to infinity (Goodman, 2008). More formally, for a symmetric probability distribution centered around zero, the single-tailed p-value is given by

$$\text{one-tailed p-value} : \begin{cases} P(X \geq x|\mathcal{H}_0) & \text{for } x > 0 \\ P(X \leq x|\mathcal{H}_0) & \text{for } x \leq 0 \end{cases} \tag{B.1}$$

where $\mathcal{H}_0$ represents the null hypothesis, X a random variable (e.g., mean) and x is the observed outcome. A two-tailed p-value is given by the integral over the extremes on both sides of the tail,

$$\text{two-tailed p-value} : \begin{cases} P(|X| \geq |x|\,|\mathcal{H}_0) & \text{for } |x| > 0 \\ P(|X| \leq |x|\,|\mathcal{H}_0) & \text{for } |x| \leq 0 \end{cases} \tag{B.2}$$

In our case, the one-tailed p-value tells us the probability that group A could have at least the observed amount of deaths if the null hypothesis was true. That is, it gives the probability of observed deaths or more deaths given that the medicine cocktail does not have any effect on the amount of deaths. The larger the p-value, the more likely one would observe the observed data given that the null hypothesis is true. Here, the larger the p-value, the more likely the medicine does not have any effect on the observed deaths. The sample size should be taken into account as well. A large sample size reduces the variance. For example, if there was a large variance in the number of deaths then one could have a large p-value even if the medicine was working. Hence, a large p-value is no proof of anything in it of itself (Goodman, 2008). Further, note that the p-value is conditional on the null hypothesis $\mathcal{H}_0$ to be true (see equation (B.1)). It does not say anything about observing data x if the null hypothesis

$\mathcal{H}_0$ is not true. To get the probability $P(\mathcal{H}_0)$ that the null hypothesis is true one would need Bayes' theorem.

This shows one of the shortcomings of frequentist inference. It also shows that a high p-value, or significant p-value, is not equivalent to clinical importance. As Altman & Bland (1995) puts it, "Absence of evidence is not evidence of absence". In our case, this means that a high p-value does not necessarily imply that the medicine does not help in reducing Covid deaths.

B.1.1.3 Hypothesis testing

In *hypothesis testing* one tests the null hypothesis against an *alternative hypothesis*. One can then reject or accept a certain hypothesis. One speaks of 'non-significant' results if $p > 0.05$. Similarly, a p-value of $p \leq 0.05$ is called statistically 'significant'. In those cases one typically rejects the null hypothesis. One can also test against multiple hypothesis. However, one should be vary about comparing the resulting p-values. If one was to compare different treatments for Covid and the different hypothesis turn out to have similar p-values, one should not imply that there is no difference between the treatments. Rather, one should note that there is an absence of evidence of a difference (Altman & Bland, 1995).

Unfortunately, results are often published that either knowingly or un-knowingly overstate the evidence. This has lead to the 'reproducibility crisis', which occurs when an experiment is reproduced but reaches contradicting outcomes (Gibson, 2020). In that case, it is impossible to say which or if any of the two experiments come to the correct conclusion. Further, any work that depends on the outcomes of the initial experiment cannot be trusted anymore either. Generally, a small p-value does not correspond to significant evidence, and one can still have a large probability that the result is a false positive (Colquhoun, 2017). For the reasons mentioned above, Colquhoun (2017) recommends to never use the terms 'significant' or 'non-significant'.

B.2 Advantages and disadvantages of Bayesian and frequentist statistics

In Bayesian statistics, we are able to e.g. predict whether patient A has Covid under the condition that the test was positive for any set of disease prevalence in the population (see Section 4.2.3 for the full calculation). In frequentist statistics, we would not be able to infer the probability for cases we have not actually observed. That is, if we only collected data at a prevalence of 1%, then this is the only case where we could give a good estimate of the probability that patient A had Covid given the positive test result. For any other population prevalence, we would have to collect data first before we

could give any predictions. This shows the power of Bayesian statistics. We can infer probabilities of events that have not occured (yet).

A disadvantage for the Bayesian method occurs when we do not have any initial beliefs and do not know the conditional data distribution, and also cannot make educated guesses about them. Let us say we had developed a new Covid test and did not know the specificity of sensitivity yet. We can only find out about those numbers by taking measurements. That is, letting people take the test and separately analyse whether they have Covid or not. If the number of people taking part in that trial is large, then our initial belief is not that important as the data will notch us into the right direction. If the number is small, then the choice of prior belief is crucial.

A disadvantage for frequentist statistics is that one can have opposing results with the same data set (Hackenberger, 2019). This is because the evidence in p-values is often overstated. The p-value is conditional on the null hypothesis $\mathcal{H}_0$, but does not say anything about the probability that the null hypothesis is true. Further, experimental design has to be clearly stated in advance, since if this is not the case then no clear conclusions or p-values can be drawn from the experiment.

Appendix C

Additional information on machine learning

C.1 Loss functions and optimization techniques

C.1.1 Loss functions

A loss function measures how well our estimator performs. It compares a set of predicted outputs with that of the true outputs. We separate the loss functions depending on whether we have a classification task or a regression task. For each of the loss functions below, we set m to be the number of samples , $\{x_i\}_{i=1,...,m}$ to be the set of input features and $\{y_i\}_{i=1,...,m}$ to be the set of input class labels. We further set $\{\hat{y}_i\}_{i=1,...,m}$ to be the outcomes as predicted by our estimator. For a classification task, $\hat{y}_i$ represents a predicted class, while for a regression task it represent a numerical value.

For classification tasks, our output consists of probabilities that a sample belongs to the possible output classes.

C.1.1.1 Multi-class cross-entropy

Multi-class (or categorical) cross-entropy is defined as (Murphy, 2012)

$$H_{N_c} = -\frac{1}{m} \sum_{i=1}^{m} \sum_{c=1}^{N_c} y_{ic} \log \hat{y}_{ic}, \tag{C.1}$$

where y_{ic} the actual label of sample i, and $\hat{y}_{ic}$ the predicted probability by our classifier that sample i belongs to class c. Typically, the actual labels y_{ic} are either 0 or 1 ($y_{ic} = 1$ for exactly one class only), whereas the predicted probabilities can range between those two values. For both y_i and $\hat{y}_i$ it holds

true that the sum of their elements is equal to one,

$$\forall i = 1, \ldots, m : \tag{C.2}$$

$$\sum_{c=1}^{N_c} y_{ic} = 1, \tag{C.3}$$

$$\text{and} \sum_{c=1}^{N_c} \hat{y}_{ic} = 1. \tag{C.4}$$

C.1.1.2 Binary cross-entropy

In a binary classification problem we only have two classes, 0 and 1, and therefore $N_c = 2$. Binary cross-entropy is then defined as (Murphy, 2012)

$$H_2 = -\frac{1}{m} \sum_{i=1}^{m} \sum_{c=1}^{2} y_{ic} \log \hat{y}_{ic}) \tag{C.5}$$

$$= -\frac{1}{m} \sum_{i=1}^{m} y_{i0} \log \hat{y}_{i0} + (1 - y_{i0}) \log (1 - \hat{y}_{i0}), \tag{C.6}$$

where y_i the N_c-dimensional vector of the actual label of sample i, and $\hat{y}_{i0}$ the predicted probability by our classifier that sample i belongs to class 0. The probability that sample i belongs to class 1 is $\hat{y}_{i1}) = (1 - \hat{y}_{i0})$.

C.1.1.3 Negative log-likelihood

Negative log-likelihood is explained in more detail in Section 5.2.1.1. It is given by

$$\text{NLL} = -\sum_{i=1}^{m} \log p_{\text{model}} (x_i, \boldsymbol{\theta}) = -\sum_{i=1}^{m} \log \hat{y}_i. \tag{C.7}$$

C.1.2 Optimization techniques

An optimizer updates the parameters of a machine learning classifier such that it enhances the classification performance.

C.1.2.1 Stochastic gradient descent

Gradient descent (GD) updates the weights of a network based on the average gradient of the loss function over all samples in the training set (Mitchell, 1997). However, if the data set is large it can be

relatively slow to find a local minimum, and if many of these local minima are present then it can be difficult to get closer to the global minimum. Stochastic gradient descent (SGD) approximates this by calculating gradient descent on only a subset of all samples (LeCun et al., 2015; Bottou et al., 2016). This gives an estimate of the average gradient of the complete data set, hence the name stochastic in gradient descent (Robbins & Monro, 1951). While GD updates the weights only after each complete iteration over the data set, SGD computes it after each small subset of the training set. This allows SGD to overcome the problems of GD. Namely, it is less computationally expensive per update, and because it looks at a larger proportion of the parameter space it is more likely to find the global minimum.

SGD takes $m < N$ amount of samples to calculate the gradient, where N is the total amount of training samples. The subset $\{\mathbf{d}_1, .., \mathbf{d}_m\}$ is called a minibatch. The gradient is then given by (Goodfellow et al., 2016)

$$g = \frac{1}{m}\nabla_{\boldsymbol{\theta}} \sum_{n=1}^{m} L\left(f(\mathbf{d}_n, \boldsymbol{\theta}), y_n\right) \tag{C.8}$$

The parameter update is then given by

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \alpha\, g \tag{C.9}$$

where α is the learning rate. Calculating the gradient over the total amount of samples is computationally inefficient if the objective function is not differentiable w.r.t its parameters (Kingma & Ba, 2014). Hence, SGD is especially valuable when the objective function is of stochastic nature, which is the case in many machine learning algorithms. It further can handle sources of noise such as data subsampling and dropout. Moreover, a small batch size can lead to a better generalization error and thus acts as a type of regularization. A small batch size gives an unbiased estimator of the exact gradient. This only holds for the first epoch, where samples are not yet repeated. After the first epoch, the estimator becomes biased. This also holds for gradient descent with N samples.

Furthermore, the learning rate α has to be decreased during learning when SGD is used, since the estimator introduces some noise. In comparison, gradient descent with $m = N$ can use a fixed learning rate α since the true gradient is always obtained and thus g decreases when near a minimum with $g = 0$ at the extremum.

In order to speed up the convergence rate of gradient descent, a so-called *momentum* can be included. It has the effect of taking the previous gradients into account. This helps to increase the step size when the direction of the descent does not change, and decreases the step size when the sign of the gradient changes.

The step size is largest when many successive gradients point towards the same direction.

C.1.2.2 Adam optimizer

The Adam optimizer is part of a class of optimizers that use adaptive learning rates α (Kingma & Ba, 2014). Hence, α no longer is a hyperparameter. These algorithms calculate α by scaling the momentum inversely proportional to the previous values of the gradient.

The name *Adam* comes from *adaptive moments*. As the name suggests, in this algorithm first- and second-order moments are estimated using the first-order gradient. The first moment corresponds to the mean and the second moment to the uncentred variance of the gradient.

Initially, the first and second moment estimates are set to the zero vector. This results in a bias towards zero especially in the first iteration steps and when the decay rates are small. Hence , the bias is corrected for in both estimates.
The update rules at time step t are given by

$$\mathbf{s} = \rho_1 \mathbf{s} + (1 - \rho_1)\, g \tag{C.10}$$

$$\mathbf{r} = \rho_2 \mathbf{r} + (1 - \rho_2)\, g \odot g \tag{C.11}$$

$$\hat{\mathbf{s}} = \frac{\mathbf{s}}{1 - \rho_1^t} \tag{C.12}$$

$$\hat{\mathbf{r}} = \frac{\mathbf{r}}{1 - \rho_2^t} \tag{C.13}$$

where $\rho_1, \rho_2 \in [0, 1)$ the exponential decay rates with typical values of 0.99. The first two lines are the estimates for the first and second moment, and the last two lines are their bias corrections. One can see that $\rho_1^t(\rho_2^t)$ gets smaller as t increases, and hence $1 - \rho_1^t(1 - \rho_2^t)$ gets bigger. Therefore, bias correction is especially valuable where $\rho_1(\rho_2)$ is small.
The parameters are then updated via

$$\Delta\boldsymbol{\theta} = \frac{\hat{\mathbf{s}}}{\sqrt{\hat{\mathbf{r}}} + \delta} \tag{C.14}$$

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \epsilon\, \Delta\boldsymbol{\theta} \tag{C.15}$$

where $\delta\boldsymbol{\theta}$ a small constant of order 10^{-8} used for numerical stability. For most cases, the step size is bounded by the learning rate, $|\Delta\boldsymbol{\theta}| \lessapprox \epsilon$.

A small step size $\Delta\boldsymbol{\theta}$ indicates a greater uncertainty in the direction of the mean estimate of the gradient, which is wanted around an extremum point. Kingma & Ba (2014) call this automatic adjustment of the step size a form of automatic annealing.

C.2 Regularisation

In order to reduce the test error, one can introduce regularisation techniques. This avoids overfitting the classifier while still performing well on the training set.

C.2.1 Weight decay

Regularisation in terms of weight decay tries to decrease the value of the weights of the machine learning classifier/regressor (Loshchilov & Hutter, 2017). This is useful if we have a large amount of features. Weight decay adds a regularisation term $\mathcal{R}$ to the loss function $\mathcal{L}$, which then becomes the cost function $\mathcal{J}$. Instead of optimizing the loss function as done previously, we now optimize the cost function,

$$\mathcal{J}(\hat{y}, y, w) = \mathcal{L}(\hat{y}, y, w) + \lambda \mathcal{R}(w), \tag{C.16}$$

where w the weights that need to be optimized. $\lambda \geq 0$ is a fixed user-defined value. If $\lambda = 0$, then no regularisation is applied and the cost function just becomes the loss function. If $\lambda > 0$, then the weights w are updated to become smaller. For $\lambda >> 0$, the weights go to zero and hence no features are used. This represents underfitting. Typically, one chooses λ to be between 0 and 0.1 (Kuhn & Johnson, 2013).

Overall, optimizing the cost function instead of the loss function now becomes a tradeoff between optimising the training data and minimising the weights.

C.2.1.1 $L1$ regularisation (Lasso regression)

$L1$ weight decay for N_w weights w is defined as the sum over the absolute values of the individual weights,

$$\mathcal{R}(w) = \sqrt{|w^T|}\,\sqrt{|w|} = \sum_i^{N_w} |w_i| = \|w\|_1^2 . \tag{C.17}$$

$L1$ regularisation therefore is defined as the L^1 norm. This has the effect of setting small weights to zero, which essentially gets rid of unimportant features. It also decreases the value of large weights to not rely on a few features only, which further reduces overfitting.

C.2.1.2 *L2* regularisation (Ridge regression)

L2 weight decay for weights w is defined as the sum of the individual weights squared,

$$\mathcal{R}(w) = w^T w = \sum_i^{N_w} w_i^2 = \|w\|_2^2 . \tag{C.18}$$

For elements w_i in the weight vector that are in the range $0 \leq |w_i| < 1$, squaring the element results in a smaller value, $\|w_i\|_2^2 < w_i$. On the other hand, weights larger than 1 will be transformed into a larger value, $\|w_i\|_2^2 > w_i$. Therefore, large weights will give a large regularisation term, which will drastically increase the cost function. It will increase it even more than if we took the L^1 norm. These weights will therefore be down-scaled by the optimizer. Small weights with absolute value below 1, however, are more likely to stay small. Unlike the *L*1 regularisation, the *L*2 regularisation does not completely get rid of less important features.

One can see that the maximum a posteriori estimation from Section 4.4.2 with a Gaussian prior on the parameters corresponds to L_2 regularisation (Goodfellow et al., 2016).

Note that even though this is called *L*2 regularisation, it does not correspond to the L^2 norm. The L^2 norm corresponds to the Euclidian distance, and is the square root of the *L*2 regularisation. The Euclidian distance is defined as the smallest distance between two points, therefore $\|w\|_2 \leq \|w\|_2^2$ and $\|w\|_2 \leq \|w\|_1^2$.Thus, the L^2 norm is always smaller or equal to the L^1 norm and to the *L*2 regularisation. The *L*2 regularisation, however, is only smaller than the *L*1 regularisation for $w_i < 1, \forall i \in \{1, \ldots, N_w\}$, where N_w the number of weights of the classifier.

C.2.2 Early stopping

As we have seen in Section 5.3, there is an optimal capacity that represents the sweet spot between underfitting and overfitting. Typically, we look at such a graph after training. The idea behind early stopping is to analyse the training and generalisation loss after each epoch and stop once the generalisation error starts to increase.

C.2.3 Regularisation specific for neural networks

Neural networks are explained in Section 5.6. Convolutional neural networks are explained in more detail in Section 5.7. It is advised to first read through the introductions of both classifiers before continuing to read these regularisation techniques.

C.2.3.1 Dropout

In a neural network or a fully connected layer of a CNN, one temporarily discards each neuron at a given layer with a certain probability at each step during training (Srivastava et al., 2014; Hinton et al., 2012; Goodfellow et al., 2013). This is called *dropout*, as one drops out neurons from the network. This is done to ensure that there is not one single neuron that decides the overall result of the network. Rather, we want to build some robustness to the network. This means that each neuron detects at least one useful feature.

For instance, if you want to detect faces and one neuron has developed a feature to find noses, then this feature alone might give a good accuracy in finding faces. However, if this neuron is removed from the network then the accuracy will drop drastically. Therefore, by randomly removing neurons from the classifier at each layer, the network forces the other neurons to find other features in a face such as eyes, mouth or ears. Together these feature detections will be far more accurate than a detection of noses alone.

If our goal is to make a classifier that distinguishes dogs from cats based on image data, then we want to have several neurons that can detect each feature. This makes the network more reliable. For example, we want to have several neurons that can detect a snout, and several neurons that are activated if eyes are present. If we only rely on one neuron for all of that, chances are that when this neuron fails or makes a bad prediction then it will drastically decrease the performance of our estimator. Note that dropout is not applied anymore after training is complete.

For hidden layers, a good choice of probability is 0.5. For the input layer, it is better to use a value close to 1 (Srivastava et al., 2014).

C.2.3.2 Batch normalization

The problem of deep networks with many layers is that the input of each layer depends on the output of the previous layer. That is, at each hidden layer of such a deep network, the inputs to that layer depend on the weights and input of the previous layer. Now during training the weights are constantly adapted such that the input distribution of each layer changes constantly (Ioffe & Szegedy, 2015). Therefore, any small change in one layer affects all layers that come after that. This slows down training as e.g. one has to use small learning rates.

The solution to this is batch normalisation, which essentially normalises the inputs of each layer. This allows us to use larger learning rates and generally speeds up training. If we think about it this makes sense, since one usually normalises or scales the training data for a classifier or regressor. One can think of a layer in a deep network as an individual classifier/regressor, and hence it makes sense that the input is normalised for each of them.

C.3 Dimensionality reduction

The goal of dimensionality reduction is to de-correlate the data and to project it to lower dimensions. The training data are often assumed to be independent and identically distributed. If we take the example of the maximum likelihood as described in Section 5.2.1, we use the i.i.d. assumption to obtain $\boldsymbol{\theta}_{ML}$ as a function of the product of the individual probabilities $P(d_i, \boldsymbol{\theta})$, which is far easier to compute than if the samples $\mathbf{d}$ were dependent or not identically distributed. However, this i.i.d. assumption is often wrong and we have to find a way to de-correlate the input data such that the i.i.d. assumption holds again. This can be done by either getting rid of redundant information or by finding a transformation into a new space in which the dataset is uncorrelated.

C.3.1 Curse of dimensionality

The more features we have, the more ways of configuring the feature vectors. In general, if we have d dimensions where each feature dimension can have v distinct values, then one has $O(v^d)$ possible ways of assembling that feature vector. So if we had 10 dimensions with 20 values each, we would have 10^{20} configurations. At 40 dimensions with 10 possible values each, we receive 10^{40} configurations. In order to build a strong classifier, one needs at least as many training samples as configurations. If this is not the case, we are left with sparse data, with which it is much more difficult to make accurate predictions on a new unseen sample. With sparse data it further becomes more difficult to separate samples into groups in a meaningful way. This is called the *curse of dimensionality*. We therefore need a way to reduce the dimension of the feature vectors.

C.3.2 PCA: Principal Component Analysis

Principal component analysis (PCA) is a linear orthogonal transformation that de-correlates any linear relationship between the dataset (Jolliffe, 1986). The input space is rotated such that it aligns with the principal axes of variance. These principal axes of variance then consist of the new basis vectors. PCA thus finds a linear combination of features with the highest variance. These features of highest variance are called *principal components*. After applying PCA, the input data has no linear relationship. Note that PCA cannot remove non-linear relationships. PCA is also used for dimensionality reduction (Goodfellow et al., 2016). It can be used to reduce noise and filter out the signal. Note that this is only true if the dimension of the new basis vectors is less than that for the original space, otherwise no data reduction occurs.

Let the data points $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\} \in \mathbb{R}^n$ be compressed to vectors $\{\mathbf{c}_1, \ldots, \mathbf{c}_m\} \in \mathbb{R}^l$, where we want $l < n$. The goal is to find a decoding function that approximates the original data from the compressed point

(Goodfellow et al., 2016),

$$f(\mathbf{x}) = \mathbf{c} \tag{C.19}$$

$$\mathbf{x} \approx g(f(\mathbf{x})) \tag{C.20}$$

That is, suppose we have an encoding function f that transforms data points $\mathbf{x}$ from the original data space to the compressed point $\mathbf{c}$. We want to find a decoding function that approximates the inverse function of f. Let us use a linear mapping which can be represented via

$$g(\mathbf{c}) = \mathbf{D}\,\mathbf{c},\ \mathbf{D} \in \mathbb{R}^{n\times l} \tag{C.21}$$

where we require the columns of $\mathbf{D}$ to be orthogonal to each other and to have unit norm. The optimal compressed point $\mathbf{c}^*$ will be the point that, after decoding it via the function g, will have the smallest distance to the original point $\mathbf{x}$ as measured via the squared L^2 norm,

$$\mathbf{c}^* = \underset{\mathbf{c}}{\operatorname{argmin}} \|\mathbf{x} - g(\mathbf{c})\|_2^2 \tag{C.22}$$

By the definition of the L^2 norm, the distributive property and by $\mathbf{x}^T g(\mathbf{c}) = g(\mathbf{c})^T \mathbf{x}$, one arrives at

$$\mathbf{c}^* = \underset{\mathbf{c}}{\operatorname{argmin}}\, \mathbf{x}^T\mathbf{x} - 2\mathbf{x}^T g(\mathbf{c}) + g(\mathbf{c})^T g(\mathbf{c}) \tag{C.23}$$

Now, only considering the terms that depend on $\mathbf{c}$ and inserting equation (C.21), one retrieves

$$\mathbf{c}^* = \underset{\mathbf{c}}{\operatorname{argmin}} -2\mathbf{x}^T\mathbf{D}\mathbf{c} + \mathbf{c}^T\mathbf{c}, \tag{C.24}$$

where we used that $\mathbf{D}^T\mathbf{D} = \mathbf{I}_l$. This can be solved by taking the gradient w.r.t. $\mathbf{c}$,

$$\nabla_{\mathbf{c}}\left(-2\mathbf{x}^T\mathbf{D}\mathbf{c} + \mathbf{c}^T\mathbf{c}\right) = \mathbf{0} \tag{C.25}$$

$$\Rightarrow \mathbf{c} = \mathbf{D}^T\mathbf{x} \tag{C.26}$$

This shows that the encoding function is represented via the transpose of the decoding matrix $\mathbf{D}$. Therefore,

$$\mathbf{x} \approx r(\mathbf{x}) = g(f(\mathbf{x})) = \mathbf{D}\mathbf{D}^T\mathbf{x} \tag{C.27}$$

Now we know the best encoding function to our decoding function g, we want to know the best encoding matrix $\mathbf{D}$. This can be done by minimizing the Frobenius norm of elements $\{x_i^j - r(x_i^j)\}_{ij}$.

$$\mathbf{D}^* = \underset{\mathbf{D}}{\operatorname{argmin}} \|\mathbf{X} - \mathbf{D}\mathbf{D}^T\mathbf{X}\|_F^2,\ \text{with}\ \mathbf{D}^T\mathbf{D} = \mathbf{I}_l \tag{C.28}$$

where $\mathbf{X} \in \mathbb{R}^{m\times n}$ the matrix of data points $\{\mathbf{x}_i\}$. Notice that while $\mathbf{D}^T\mathbf{D} = \mathbf{I}_l$ the identity matrix of

dimension $l \times l$, $\mathbf{DD}^T \in \mathbb{R}^{n \times n}$ a matrix of dimension $m \times m$. This is solved in the following,

$$\begin{aligned}
\mathbf{D}^* &= \underset{\mathbf{D}}{\operatorname{argmin}}\, Tr\Big[\left(\mathbf{X} - \mathbf{DD}^T\mathbf{X}\right)^T\left(\mathbf{X} - \mathbf{DD}^T\mathbf{X}\right)\Big] && \text{(C.29)}\\
&= \underset{\mathbf{D}}{\operatorname{argmin}}\, Tr\Big[-\mathbf{X}^T\mathbf{DD}^T\mathbf{X} + \mathbf{X}^T\mathbf{DD}^T\mathbf{DD}^T\mathbf{X} + \mathbf{X}^T\mathbf{X} - \mathbf{X}^T\mathbf{DD}^T\mathbf{X}\Big] && \text{(C.30)}\\
&\overset{\mathbf{D}^T\mathbf{D}=\mathbf{I}_l}{=} \underset{\mathbf{D}}{\operatorname{argmin}} -Tr\Big[\mathbf{X}^T\mathbf{DD}^T\mathbf{X}\Big] && \text{(C.31)}\\
&\overset{Tr[ABC]=Tr[\underline{BCA}]=Tr[CAB]}{=} \underset{\mathbf{D}}{\operatorname{argmax}}\, Tr\Big[\mathbf{D}^T\mathbf{XX}^T\mathbf{D}\Big] && \text{(C.32)}\\
& && \text{(C.33)}
\end{aligned}$$

This looks as if we can apply Eigendecomposition of the form $\mathbf{A} = \mathbf{Q\Lambda Q^T}$. Here,

$$\begin{aligned}
\mathbf{\Lambda} &= \mathbf{D}^{*T}\mathbf{XX}^T\mathbf{D}^* && \text{(C.34)}\\
\Leftrightarrow \mathbf{D}^*\mathbf{\Lambda} &= \mathbf{XX}^T\mathbf{D}^* && \text{(C.35)}\\
\Leftrightarrow \mathbf{D}^*\mathbf{\Lambda}\mathbf{D}^{*T} &= \mathbf{XX}^T && \text{(C.36)}
\end{aligned}$$

and hence $\mathbf{D}^*$ are the Eigenvectors of $\mathbf{XX}^T$ that corresponds to the largest Eigenvalues.

C.3.3 Non-linear PCA: Kernel trick

Linear PCA cannot remove non-linear relationships between the data. Instead, one can make use of the 'kernal trick' to try to remove non-linearities. The idea behind the kernel trick is that by adding extra dimensions one can (sometimes) make the dataset linear. This is done by applying a kernel function to the input which, if successful, will make the data linear in the transformed space. In this new space linear PCA can then be applied to find the eigenvectors corresponding to the largest eigenvalues of the transformed data.

C.3.3.1 Kernel function

The Kernel function k acts on two data points and outputs a single transformed variable that is a function of those two inputs. It is defined as

$$\begin{aligned}
k : \mathcal{X} \times \mathcal{X} &\longrightarrow \mathcal{Y} && \text{(C.37)}\\
(\mathbf{u}, \mathbf{v}) &\longmapsto \phi(\mathbf{u}) \cdot \phi(\mathbf{v}), && \text{(C.38)}
\end{aligned}$$

where $\mathbf{u}, \mathbf{v}$ vectors of the input space $\mathcal{X}$, and ϕ a transformation to a new space $\mathcal{Y}$, $\phi : \mathcal{X} \longmapsto \mathcal{Y}$.

Radial basis function The radial basis function (RBF) is defined as a Gaussian distribution,

$$k(\mathbf{u}, \mathbf{v}) = \exp\left(-\frac{\|\mathbf{u} - \mathbf{v}\|_2^2}{2\,\sigma^2}\right),$$

where $\|.\|_2^2$ the L^2 norm, $\|\mathbf{u} - \mathbf{v}\|_2^2$ the Euclidian distance between $\mathbf{u}$ and $\mathbf{v}$, and σ the standard deviation. This gives a maximum at $\mathbf{u} = \mathbf{v}$, as then the Euclidian distance is equal to zero.

C.3.3.2 Example: Elliptical to linear plane

Let us take an example of an ellipse in $\mathbb{R}^2$. This can be written as $ax^2 + bxy + cy^2 + d = 0$, which is not linear. However, we can make it linear by using a polynomial kernel,

$$k(x, y) = (x^2, xy, y^2) = (z_1, z_2, z_3) \tag{C.39}$$

Plugging this into our ellipse equation, we receive $az_1 + bz_2 + cz_3 + d = 0$, which is indeed linear and corresponds to a plane in $\mathbb{R}^3$. We have pre-processed our non-linear data by applying a polynomial kernel function, from which we obtained a linear model on a new space.

C.4 Backpropagation in artificial neural networks

Let us go through the algorithm step by step. First, we want to calculate the gradient of J w.r.t. the output $\mathbf{a}_l$ (Goodfellow et al., 2016),

$$\nabla_{\mathbf{a}_l} J = \nabla_{\mathbf{h}_l} J \odot v_l{}'(\mathbf{a}_l) \tag{C.40}$$

The first term $\nabla_{\mathbf{h}_l} J$ explains how fast the cost function J is changing w.r.t. the output $\mathbf{h}_l$ of layer l of the activation function. The second term, $v_l{}'(\mathbf{a}_l)$, corresponds to the change in the activation function v_l of layer l w.r.t. the output of the same layer. The symbol $\odot$ is called the Hadamard product, which represents elementwise multiplication. If J does not depend a lot on the output of layer l, then $\nabla_{\mathbf{h}_l} J$ will be small. In the case for ReLU as the output function, $v_l{}'(\mathbf{a}_l)$ will be constant (equal to 1) for $\mathbf{a}_l \geq 0$ and zero otherwise.

Next, from this and equation (5.23) we can calculate the gradient w.r.t the network parameters $\mathbf{W}_l$ and $\mathbf{b}_l$,

$$\nabla_{\mathbf{b}_l} J = \nabla_{\mathbf{a}_l} J \frac{\partial \mathbf{a}_l}{\partial \mathbf{b}_l} + \lambda \nabla_{\mathbf{b}_l} \Omega(\boldsymbol{\theta}) \overset{\left[\frac{\partial \mathbf{a}_l}{\partial \mathbf{b}_l} = 1\right]}{=} \nabla_{\mathbf{a}_l} J + \lambda \nabla_{\mathbf{b}_l} \Omega(\boldsymbol{\theta}) \tag{C.41}$$

$$\nabla_{\mathbf{W}_l} J = \nabla_{\mathbf{a}_l} J \frac{\partial \mathbf{a}_l}{\partial \mathbf{W}_l} + \lambda \nabla_{\mathbf{W}_l} \Omega(\boldsymbol{\theta}) \overset{\left[\frac{\partial \mathbf{a}_l}{\partial \mathbf{W}_l} = \mathbf{h}_{l-1}{}^T\right]}{=} \nabla_{\mathbf{a}_l} J\, \mathbf{h}_{l-1}{}^T + \lambda \nabla_{\mathbf{W}_l} \Omega(\boldsymbol{\theta}) \tag{C.42}$$

Here we have used the chain rule that can be applied to any vector $\mathbf{z}$ that is a function of $\mathbf{a}_l$,

$$\nabla_{\mathbf{z}} J = \nabla_{\mathbf{a}_l} J \, \frac{\partial \mathbf{a}_l}{\partial \mathbf{z}} \tag{C.43}$$

Finally, the gradient of J w.r.t. the activation output $\mathbf{h}_{l-1}$ of the layer above can be calculated via

$$\nabla_{\mathbf{h}_{l-1}} J = \nabla_{\mathbf{a}_l} J \, \frac{\partial \mathbf{a}_l}{\partial \mathbf{h}_{l-1}} \overset{\left[\frac{\partial \mathbf{a}_l}{\partial \mathbf{h}_{l-1}} = \mathbf{W}_l\right]}{=} \mathbf{W}_l{}^T \, \nabla_{\mathbf{a}_l} J \tag{C.44}$$

This allows us to iteratively go through the network. The gradient corresponding to the output of the last layer $\mathbf{x}_n$ for an input image set $\mathbf{d_n}$ is given by

$$\nabla_{\mathbf{h}_D} J = \nabla_{\mathbf{x}_n} J = \nabla_{\mathbf{x}_n} \left(-\log P(y = i_n | \mathbf{x_n}) \right) \tag{C.45}$$

Now let us summarize all the steps from above. Consider the case where $J = L$, i.e. $\lambda = 0$. The backpropagation algorithm is then given by

$$\forall l = D, .., 1 : \tag{C.46}$$

$$1.) \quad \nabla_{\mathbf{a}_l} J = \nabla_{\mathbf{h}_l} J \odot v_l{}' \left(\mathbf{a}_l \right) \tag{C.47}$$

$$2.) \quad \nabla_{\mathbf{b}_l} J = \nabla_{\mathbf{a}_l} J \tag{C.48}$$

$$3.) \quad \nabla_{\mathbf{W}_l} J = \nabla_{\mathbf{a}_l} J \, \mathbf{h}_{l-1}{}^T \tag{C.49}$$

$$4.) \quad \nabla_{\mathbf{h}_{l-1}} J = \mathbf{W}_l{}^T \, \nabla_{\mathbf{a}_l} J \tag{C.50}$$

The implementation of backpropagation follows in such a way that any gradient is calculated only once and then stored in a table for quick access. This is very efficient as a node within the network typically has multiple parents which in return are parents for many nodes. For example, take $u^{(1)}, .., u^{(N)}$ to be the outputs of nodes in a network, where $u^{(N)}$ the last output node. The derivative of $u^{(N)}$ w.r.t some other node $u^{(j)}$ is given by

$$\frac{\partial u^{(N)}}{\partial u^{(j)}} = \sum_{i:\, j \in Parent(u^{(i)})} \frac{\partial u^{(N)}}{\partial u^{(i)}} \frac{\partial u^{(j)}}{\partial u^{(i)}} \tag{C.51}$$

As can be seen, for $j = 1, .., N$ and $i : j \in Parent(u^{(i)})$, the partial derivative $\frac{\partial u^{(N)}}{\partial u^{(i)}}$ will be calculated multiple times if node i has multiple parents. This is were backpropagation comes in. The partial derivatives $\frac{\partial u^{(N)}}{\partial u^{(i)}}$ are stored in a table such that they can be accessed if needed for a calculation.