

Imperial College London  
Department of Computing

# **Feature extraction in classification**

Raymond Liu

December 11, 2013

Supervised by Prof. Duncan F. Gillies

Submitted in part fulfilment of the requirements for the degree of Doctor of Philosophy in the  
Department of Computing of Imperial College London  
and the Diploma of Imperial College

# **Declaration**

## **Copyright**

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work

## **Originality**

I hereby certify that all material in this dissertation which is not my own work has been properly acknowledged.

Raymond Liu

# Acknowledgements

The wisdom imparted to me by my supervisor, Professor Duncan F. Gillies (Imperial College London), is more profound than he will ever know. His open-mindedness, scientific insight, and stereotypically British sense of humility, have been some of the most important inspirations in my life.

I have been very privileged in having the opportunity to discuss mathematical ideas and to seek mathematical insights from my friends and colleagues Dr. Richard Hayden, Dr. Anton Stefanek, and Dr. Gareth L. Jones.

I would like to thank my examiners Professor Steve R. Gunn (University of Southampton) and Dr. William Knottenbelt (Imperial College London) for their invaluable constructive comments and recommendations on this thesis. They have been extremely generous with their time and effort in examining my research and presentation in such fine detail.

None of this research would have been possible without the scholarship and funding awarded to me by Imperial College London and the Engineering and Physical Sciences Research Council (EPSRC), UK. I would like to thank these institutions for providing this lifetime opportunity to carry out the research that I love.

## Abstract

Feature extraction, or dimensionality reduction, is an essential part of many machine learning applications. The necessity for feature extraction stems from the curse of dimensionality and the high computational cost of manipulating high-dimensional data. In this thesis we focus on feature extraction for classification. There are several approaches, and we will focus on two such: the increasingly popular information-theoretic approach, and the classical distance-based, or variance-based approach.

Current algorithms for information-theoretic feature extraction are usually iterative. In contrast, PCA and LDA are popular examples of feature extraction techniques that can be solved by eigendecomposition, and do not require an iterative procedure. We study the behaviour of an example of iterative algorithm that maximises Kapur’s quadratic mutual information by gradient ascent, and propose a new estimate of mutual information that can be maximised by closed-form eigendecomposition. This new technique is more computationally efficient than iterative algorithms, and its behaviour is more reliable and predictable than gradient ascent. Using a general framework of eigendecomposition-based feature extraction, we show a connection between information-theoretic and distance-based feature extraction.

Using the distance-based approach, we study the effects of high input dimensionality and over-fitting on feature extraction, and propose a family of eigendecomposition-based algorithms that can solve this problem. We investigate the relationship between class-discrimination and over-fitting, and show why the advantages of information-theoretic feature extraction become less relevant in high-dimensional spaces.

# Contents

|                                                                             |           |
|-----------------------------------------------------------------------------|-----------|
| <b>1. Introduction</b>                                                      | <b>11</b> |
| 1.1. A primer . . . . .                                                     | 11        |
| 1.2. The problem . . . . .                                                  | 12        |
| 1.3. Curse of dimensionality, over-fitting, and small sample size . . . . . | 13        |
| 1.4. Scope . . . . .                                                        | 16        |
| 1.4.1. Feature selection and feature extraction . . . . .                   | 16        |
| 1.4.2. Linear and non-linear feature extraction . . . . .                   | 17        |
| 1.4.3. Supervised and unsupervised feature extraction . . . . .             | 18        |
| 1.4.4. Classification and regression . . . . .                              | 19        |
| 1.4.5. Classifiers . . . . .                                                | 19        |
| 1.5. Contributions . . . . .                                                | 20        |
| 1.5.1. Publications . . . . .                                               | 20        |
| 1.6. Notation . . . . .                                                     | 21        |
| 1.7. Distance-based approach . . . . .                                      | 21        |
| 1.8. Information-theoretic approach . . . . .                               | 25        |
| 1.8.1. Quadratic mutual information (QMI) . . . . .                         | 28        |
| 1.8.2. Gaussian kernel density estimation . . . . .                         | 29        |
| 1.9. Summary and structure of the thesis . . . . .                          | 31        |
| <b>2. Information-theoretic feature extraction</b>                          | <b>33</b> |
| 2.1. Gradient ascent for QMI maximisation . . . . .                         | 33        |
| 2.1.1. Whitening . . . . .                                                  | 37        |
| 2.1.2. Bandwidth selection . . . . .                                        | 38        |
| 2.1.3. Feature initialisation . . . . .                                     | 40        |
| 2.1.4. Learning rate and stopping criterion . . . . .                       | 42        |
| 2.2. Eigenvalue-based mutual information . . . . .                          | 43        |
| 2.2.1. Relationship with Laplacian maps . . . . .                           | 47        |
| 2.2.2. Computational complexity . . . . .                                   | 48        |
| 2.3. Summary . . . . .                                                      | 49        |
| <b>3. Over-fitting and distance-based feature extraction</b>                | <b>51</b> |
| 3.1. Limitations of information-theoretic feature extraction . . . . .      | 52        |
| 3.2. A measure of class-discrimination . . . . .                            | 56        |

|                                                                                        |           |
|----------------------------------------------------------------------------------------|-----------|
| 3.3. Soft discrimination . . . . .                                                     | 58        |
| 3.3.1. Other similar algorithms . . . . .                                              | 61        |
| 3.3.2. Pre-processing by PCA . . . . .                                                 | 63        |
| 3.3.3. Controlling the level of class-discrimination . . . . .                         | 63        |
| 3.3.4. Highly representative datasets . . . . .                                        | 65        |
| 3.4. Summary . . . . .                                                                 | 66        |
| <b>4. Conclusions</b>                                                                  | <b>68</b> |
| 4.1. Future research . . . . .                                                         | 69        |
| 4.1.1. Mutual information . . . . .                                                    | 69        |
| 4.1.2. Relevance vector selection and extraction . . . . .                             | 69        |
| 4.1.3. Relationship between feature extraction, classification and regression. . . . . | 72        |
| 4.1.4. Maximum-spread-zero-overlap and the maximum entropy principle . . . . .         | 72        |
| 4.1.5. Unification of feature extraction techniques . . . . .                          | 74        |
| <b>Appendices</b>                                                                      | <b>75</b> |
| <b>A. Supplementary experimental results</b>                                           | <b>75</b> |
| A.1. EMI maximisation . . . . .                                                        | 75        |
| A.1.1. Relatively low-dimensional datasets . . . . .                                   | 75        |
| A.1.2. High-dimensional datasets . . . . .                                             | 83        |
| A.2. Soft discriminant maps . . . . .                                                  | 86        |
| A.2.1. Relationship between $\beta$ and class-discrimination . . . . .                 | 93        |

# List of Figures

|       |                                                                                                                                                                                                                                                   |    |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 1.1.  | Illustration of small sample size using an artificial dataset . . . . .                                                                                                                                                                           | 14 |
| 1.2.  | Illustration of over-fitting using the AT&T Dataset of Faces . . . . .                                                                                                                                                                            | 15 |
| 1.3.  | An advantage of information-theoretic feature extraction . . . . .                                                                                                                                                                                | 26 |
| 1.4.  | Example distributions of a unimodal class and a bimodal class . . . . .                                                                                                                                                                           | 27 |
| 2.1.  | Plot of the QMI between the class label and each of the first 10 features, computed by various methods . . . . .                                                                                                                                  | 35 |
| 2.2.  | A demonstration that QMI ‘prefers’ good class-discrimination . . . . .                                                                                                                                                                            | 35 |
| 2.3.  | 2D projections of the AT&T Dataset of Faces, computed using LDA, EMI, Torkkola’s iterative QMI method, and PCA for comparison . . . . .                                                                                                           | 36 |
| 2.4.  | 2D projection of the iterative QMI algorithm where the data (AT&T Dataset of Faces) is not whitened as a pre-processing step. The initial features of the algorithm are set to random vectors. . . . .                                            | 38 |
| 2.5.  | Illustration of the behaviour of QMI with variable and fixed bandwidths. . . . .                                                                                                                                                                  | 39 |
| 2.6.  | 2D projection of the (pre-whitened) AT&T Dataset of Faces using the iterative QMI algorithm with a simplified version of Torkkola’s heuristic for bandwidth selection . . . . .                                                                   | 40 |
| 2.7.  | Values of QMI measured along the first 10 features extracted by LDA, EMI and the iterative QMI algorithm respectively, where the initial features for the QMI algorithms are LDA features. The dataset used is the AT&T Dataset of Faces. . . . . | 41 |
| 2.8.  | Graphs of $e_{nm}$ as in equation (2.3), and $g_{nm}$ as in equation (1.31), where we view $(y_n - y_m)$ as the abscissa . . . . .                                                                                                                | 45 |
| 2.9.  | The values of the various measures of mutual information along a feature, as the feature is rotated $\pi$ radians . . . . .                                                                                                                       | 45 |
| 2.10. | 5-fold cross-validation on the Pima dataset . . . . .                                                                                                                                                                                             | 46 |
| 2.11. | 2D projections of the Pima dataset computed by three feature extraction techniques. . . . .                                                                                                                                                       | 47 |
| 3.1.  | 2-fold cross-validation on the AT&T dataset of faces, showing error rates for EMI maximisation . . . . .                                                                                                                                          | 51 |
| 3.2.  | 2D projection of the AT&T Dataset of Faces where 10 superclasses are formed from the 40 original classes by regarding 4 as 1 . . . . .                                                                                                            | 52 |
| 3.3.  | A demonstration that QMI ‘prefers’ unimodal classes . . . . .                                                                                                                                                                                     | 55 |

|                                                                                                                                                                                                                                   |    |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 3.4. Values of $\Delta(\{\underline{y}_n, c_n\}_{n=1}^N)$ in the subspaces computed by PCA, LDA, and EMI respectively . . . . .                                                                                                   | 57 |
| 3.5. 2D projections of the AT&T Dataset of Faces computed by PCA, LDA and SDMs for various values of $\beta$ . The data is pre-processed using PCA, but not whitened. . . . .                                                     | 59 |
| 3.6. Values of the $\Delta$ -measure for the feature subspaces computed by SDMs for various values of $\beta$ , on the AT&T Dataset of Faces. The data is pre-processed by PCA but not whitened. . . . .                          | 60 |
| 3.7. 2-fold cross-validation on the AT&T Dataset of Faces, showing error rates for SDMs . . . . .                                                                                                                                 | 60 |
| 3.8. 2D projection of the AT&T Dataset of Faces computed by SDM ( $\beta = 50$ ), where the projection is computed on 80% of the data. The rest 20% are marked by grey circles. . . . .                                           | 61 |
| 3.9. A plot of $\beta$ against the $\Delta$ -measure in 5-dimensional feature subspaces computed by SDMs for various values of $\beta$ , on the AT&T Dataset of Faces . . . . .                                                   | 64 |
| 3.10. 2D projections of the JAFFE face dataset computed by PCA and LDA respectively, where the projections are computed using 80% of the dataset . . . . .                                                                        | 65 |
| 3.11. 2-fold cross-validation on the JAFFE face dataset . . . . .                                                                                                                                                                 | 66 |
| 4.1. Illustration of how a data distribution can be well-represented by only a small subset of the whole dataset . . . . .                                                                                                        | 70 |
| A.1. 2-fold cross-validation on the Breast Cancer dataset . . . . .                                                                                                                                                               | 76 |
| A.2. 2D projections of the pre-whitened Breast Cancer data. . . . .                                                                                                                                                               | 76 |
| A.3. 5-fold cross-validation on the Image Segmentation dataset . . . . .                                                                                                                                                          | 77 |
| A.4. A direct test on the pre-whitened Image Segmentation dataset, where the respective feature extractors are trained on the training dataset of 210 data points, and tested on the testing dataset of 2100 data points. . . . . | 77 |
| A.5. 2D projections of the pre-whitened training subset of the Image Segmentation dataset. . . . .                                                                                                                                | 78 |
| A.6. 2-fold cross-validation on the Parkinsons dataset . . . . .                                                                                                                                                                  | 79 |
| A.7. 2D projections of the pre-whitened Parkinsons dataset. . . . .                                                                                                                                                               | 79 |
| A.8. 2-fold cross-validation on the Silhouette dataset . . . . .                                                                                                                                                                  | 80 |
| A.9. 2D projections of the pre-whitened Silhouette dataset. . . . .                                                                                                                                                               | 81 |
| A.10. 2-fold cross-validation on the Yeast dataset . . . . .                                                                                                                                                                      | 82 |
| A.11. 2D projections of the pre-whitened Yeast dataset. . . . .                                                                                                                                                                   | 82 |
| A.12. 2-fold cross-validation on the Essex95 dataset . . . . .                                                                                                                                                                    | 83 |
| A.13. 2D projections of the pre-whitened Essex95 dataset. . . . .                                                                                                                                                                 | 84 |
| A.14. 2-fold cross-validation on the Essex Grimace dataset . . . . .                                                                                                                                                              | 84 |
| A.15. 2D projections of the pre-whitened Essex Grimace dataset. . . . .                                                                                                                                                           | 84 |
| A.16. 2-fold cross-validation on the JAFFE dataset . . . . .                                                                                                                                                                      | 85 |

|                                                                                                                                                                                                                 |    |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| A.17.2D projection of the pre-whitened JAFFE face dataset, computed by EMI, where<br>part of the projection is zoomed in . . . . .                                                                              | 86 |
| A.18.2-fold cross-validation on the Sheffield Face Dataset . . . . .                                                                                                                                            | 87 |
| A.19.2-fold cross-validation on the FVC1 fingerprint dataset . . . . .                                                                                                                                          | 87 |
| A.20.Classification error rates for the four Essex face datasets. . . . .                                                                                                                                       | 88 |
| A.21.2D projections of the Essex95 face dataset. Notice the inverse relationship be-<br>tween $\beta$ and the level of class-discrimination in the projection. . . . .                                          | 90 |
| A.22.Classification error rates for the four FVC fingerprint datasets. . . . .                                                                                                                                  | 91 |
| A.23.2D projections of the FVC1 dataset computed by SDMs for various values of $\beta$ .<br>Notice the inverse relationship between $\beta$ and the level of class-discrimination in<br>the projection. . . . . | 91 |
| A.24.Classification error rates for the Sheffield dataset. . . . .                                                                                                                                              | 92 |
| A.25.2D projections of the Sheffield dataset, computed by SDMs for various values of $\beta$ . . . . .                                                                                                          | 93 |
| A.26.Plots of $\beta$ against the <i>Delta</i> -measure in 5-dimensional feature subspaces com-<br>puted by SDMs for various values of $\beta$ . . . . .                                                        | 94 |

# List of Tables

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |    |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 2.1. | Times taken, in seconds, for algorithms PCA, LDA, eigenvalue-based MI, and iterative QMI to extract one feature and 39 features respectively, from the AT&T Dataset of Faces. The computing times for eigenvalue-based algorithms (PCA, LDA, EMI) are independent of the number of extracted features, due to the closed-form nature of eigendecomposition. This was a Python 2.7 implementation with packages Numpy, Matplotlib, and PIL (Python Imaging Library), executed on an Intel Core i7-2600K processor at 3.40 GHz with 16GB of RAM. . | 34 |
| A.1. | Relatively low-dimensional datasets and their characteristics. $N$ = total number of data points; $K$ = number of classes; $D$ = input dimensionality. . . . .                                                                                                                                                                                                                                                                                                                                                                                   | 80 |
| A.2. | High-dimensional datasets and their characteristics. $N$ = total number of data points; $K$ = number of classes; $D$ = input dimensionality. . . . .                                                                                                                                                                                                                                                                                                                                                                                             | 89 |

# 1. Introduction

## 1.1. A primer

Intelligent systems, including our own brains, learn to successfully categorise objects by effectively using their characteristics. Biological systems, like animal brains, can do this effortlessly but subconsciously. The translation of this ability to electronic systems requires a conscious and rigorous understanding of the underlying mechanisms, and it constitutes a significant part of machine learning and artificial intelligence.

The key to successfully categorising or classifying objects is effectively using their *characteristics*, or *features*. Suppose for example that we wish to categorise cars in terms of their top speed: fast, medium, or slow. Suppose also that we have no direct information about a car's speed. We may look at the size of its exhaust pipe, since it may give some information about how fast a car might go, and this is intuitively reasonable. The size of a car's exhaust pipe, in the language of machine learning, is a feature. Arguably, a better feature may be the retail price of a car, since this may be more closely related to a car's top speed. Neither of these two features are perfect however, since there are expensive cars (limousines) and cars with large exhaust pipes (off-road vehicles) that drive relatively slowly. However, cars that are both expensive and have large exhaust pipes may be more likely to be fast. In this view, we would like to keep both of these features in mind. So far we have identified two possibly useful features.

A feature that is not very useful, at least intuitively, is the colour of a car. Now this may be obvious to us, but imagine a child who has not seen many cars before. Imagine further that, by coincidence, the only fast cars the child has ever seen tend to be close to red in colour, and the slower cars tend to be blue or black. In the child's mind it is perfectly reasonable to use colour as a useful feature in deciding the top speed of a car, since in his (insufficient) experience, the colour of a car is a very good indicator of its speed. In the language of machine learning, the cars that the child has seen constitute the *training data*; and colour, as a feature, *over-fits* the training data, in the sense that it is useful in categorising the training data but not useful in general. Perhaps the child will adjust his choice of features as he grows older and sees more cars, and in the context of machine learning this is called *online learning*. (Though in this thesis we will not focus on the dichotomy of online and offline learning.) However, if the child does not adjust his views and insists on using a car's colour as a 'useful' feature, then there may be detrimental consequences in his ability to predict the top speed of a new car. For example, a car that is expensive and has large exhaust pipes, but is blue, might be declared

to be a slow car by the child, when it is in fact fast. This illustrates that, as well as retaining useful features, it is important to discard useless features.

So far we have identified three features about a car: two that are possibly useful and one that is probably not. In general, we may have an abundance of information about a car: manufacturer, country of origin, year of production, mass, height, length, wheel diameter, weight distribution, fuel consumption, and so on. More generally, we may have a huge amount of information about any type of object that we wish to categorise. Among all this information, only a small subset may be useful. Among this useful subset, an even smaller subset may be sufficient in adequately categorising the type of object in question. The behaviour of knowledge, when it is presented with huge amounts of information, can be strange and counterintuitive. In the language of machine learning, this is known as the *curse of dimensionality*. Identifying and constructing a subset of information that is small but sufficiently informative for the task in hand comes quite naturally to biological systems, but poses a challenge to computers. The first step in this challenge is to precisely define our intuition about information and knowledge, and for this we turn to mathematics.

## 1.2. The problem

The general form of the classification problem is as follows. We have a set of *training* data points, represented by vectors in a typically high-dimensional *input space*. These data points are allocated to different classes, and we know this allocation for these training data points. The classification task is to be able to classify new data points into their correct classes. We want to project the new data points from the high-dimensional input space onto a lower-dimensional *feature subspace*, in order that efficient and reliable classification can be executed. The basis vectors spanning the lower-dimensional feature subspace are the *features* that we seek. A feature is therefore represented by a unit vector living in the input space. The search for these features is the problem of *feature extraction*, also known as *dimensionality reduction* in the context of data representation and visualisation.

As an example, a learning machine may be tasked with classifying face images to the correct persons they represent. Each face image is represented by a vector, and each element of these is a number indicating the colour intensity of the corresponding pixel in the image. A set of training images is presented to the learning machine with their correct class labels. The machine's task is then to be able to recognise a person from a new image which it has not seen before. This is the task of face recognition [58, 21].

Pictorially, we can think of a feature as a line in the input space. A set of such features will span a subspace which we call a *feature subspace*, onto which we wish to project our data points before any classification algorithm is executed. Now these features may simply be a subset of the basis vectors of the input space, or they may be linear combinations thereof. The former constitutes a simpler problem of *feature selection* [15], while the latter constitutes the more general problem of *linear feature extraction*, which is the main subject of this thesis. Features

may also be non-linear functions of the basis vectors of the input space, and this constitutes *non-linear feature extraction*. The dichotomy of linear and non-linear feature extraction will be briefly discussed in section 1.4.2. Both feature selection and feature extraction aim to reduce the dimensionality of the input space, before a classification algorithm is applied.

The need for dimensionality reduction is two-fold, and it stems from two practical difficulties associated with classification. The first is computational efficiency. A classification algorithm typically manipulates data points and carries out calculations in a vector space [16]. If that vector space is a high-dimensional input space, these calculations may become computationally intensive or infeasible. In the face recognition example, even an extremely modest  $128 \times 128$ -pixel, greyscale image lives in 16,384-dimensional space, and a typical use-case for a face recognition algorithm may be for a hand-held device to recognise faces within a fraction of a second.

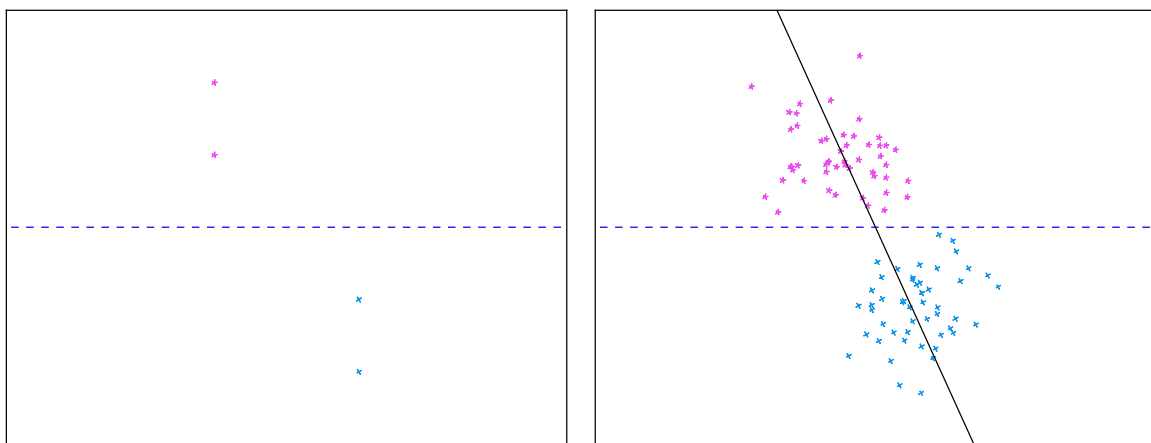
The second difficulty associated with classification is the famous problem known as *curse of dimensionality* [16, 3, sec.1.4], which will discuss in the next section.

### 1.3. Curse of dimensionality, over-fitting, and small sample size

It would be reasonable to hold the intuition that the more information a learning machine has, the more accurate it would be at making predictions. However the curse of dimensionality teaches us that this view is too simple. Now information comes in different forms, and in the context of curse of dimensionality we focus on two such forms. The first is information about each data point, and mathematically this translates to input dimensionality. For example a million-pixel image carries more information than a 256-pixel thumbnail. The second is global information about the underlying distribution of all the data points, and this is given by the abundance of training data points. For example, a dataset with 100 data points per class is more informative (about the underlying distribution of the data points) than a dataset with only 10 data points per class. We will call the former *specific information*, and the latter *global information*. A simple but accurate summary of the curse of dimensionality is this. Too much *specific information* is bad, and the more *global information* the better.

Trunk [51] and Bishop [3, sec.1.4] provided a mathematical example and intuitive, visual illustrations of curse of dimensionality, which we will not repeat here. We note that these approaches to understanding the curse of dimensionality are from the perspective of classification, but we want to understand it both from this and from another perspective — one of dimensionality reduction.

Curse of dimensionality is closely related to the *small sample size problem* and the problem of *over-fitting*, and in fact in many cases the three are just different names for the same problem. Curse of dimensionality occurs not just when the input dimensionality is high, but when it is high compared to the number of training data points, also called the sample size. When this happens, curse of dimensionality leads to over-fitting and other problems. We will illustrate



(a) Small sample size and over-fitting.

(b) Adequate sample size.

Figure 1.1.: The datasets in (a) and (b) come from the same underlying distribution. In (a) where the true distribution of the data is not well represented due to the small number of training data points (sample size), the feature (blue dashed line) produced by LDA discriminates the training classes well. However in (b) we can see that when more data points are presented, that feature performs poorly compared to the feature trained on the bigger data set (black line) shown in (b). Note that these are lines of *projection*, and not decision boundaries. We look for good separation between the two classes when the data points are *projected* onto the respective lines.

this with an artificial and a real-world example. Here for the sake of illustration, we use linear discriminant analysis (LDA, also known as Fisher’s discriminant) [3, sec.4.1.4] as our feature extraction algorithm. Consider the two class problem shown in figure 1.1, where the true distribution of each class is spherical Gaussian with the identity covariance matrix. LDA seeks features along which the variance of each class is minimised but the distances between classes are maximised. In figure 1.1a, due to the small number of training data points, the within-class variances in the horizontal direction are not faithfully represented by the training data, and consequently LDA produces a feature (blue dashed line) that is severely sub-optimal for classification. Note that the dashed line in the figure is a line of *projection*, and not a decision boundary. On the other hand, in figure 1.1b where we have adequately many training data points, the within-class variances in all directions are well-represented, and a good feature (solid black line) can be extracted. Note again that the lines are lines of *projection*, and not decision boundaries. The black line in the case of figure 1.1b is a better line of projection because when the data points are projected onto this line, they are better separated than if they were projected onto the blue dashed line. This illustrates the small sample size problem, but also illustrates the problem of over-fitting. In figure 1.1a, the LDA feature does a very good job of discriminating between the two classes for the training data points, and so it fits the training data points well. However, when more data points are produced from the true distribution of the data, we can see in figure 1.1b that the original feature would produce large classification errors. In this way, the original LDA feature is said to *over-fit* the training data presented in figure 1.1a.

In the above example, the artificial training data is 2D, and our task was to extract a single feature from an input dimensionality of 2. In general, as the input dimensionality increases, the problems of small sample size and over-fitting are exponentially exacerbated. This is because the search space, in which the optimal feature lives, increases exponentially, from  $\mathbb{R}^2$  to  $\mathbb{R}^3$  to  $\dots$  to  $\mathbb{R}^D$ ,  $D \gg 1$ . Intuitively, we can imagine that when the input dimensionality is large, there are ‘too many features to choose from’ and while there is only one optimal feature, most ‘others’ will lead to severe over-fitting due to small sample size. This is the curse of dimensionality. Again, when we say the input dimensionality is large, we mean that it is large compared to the sample size.

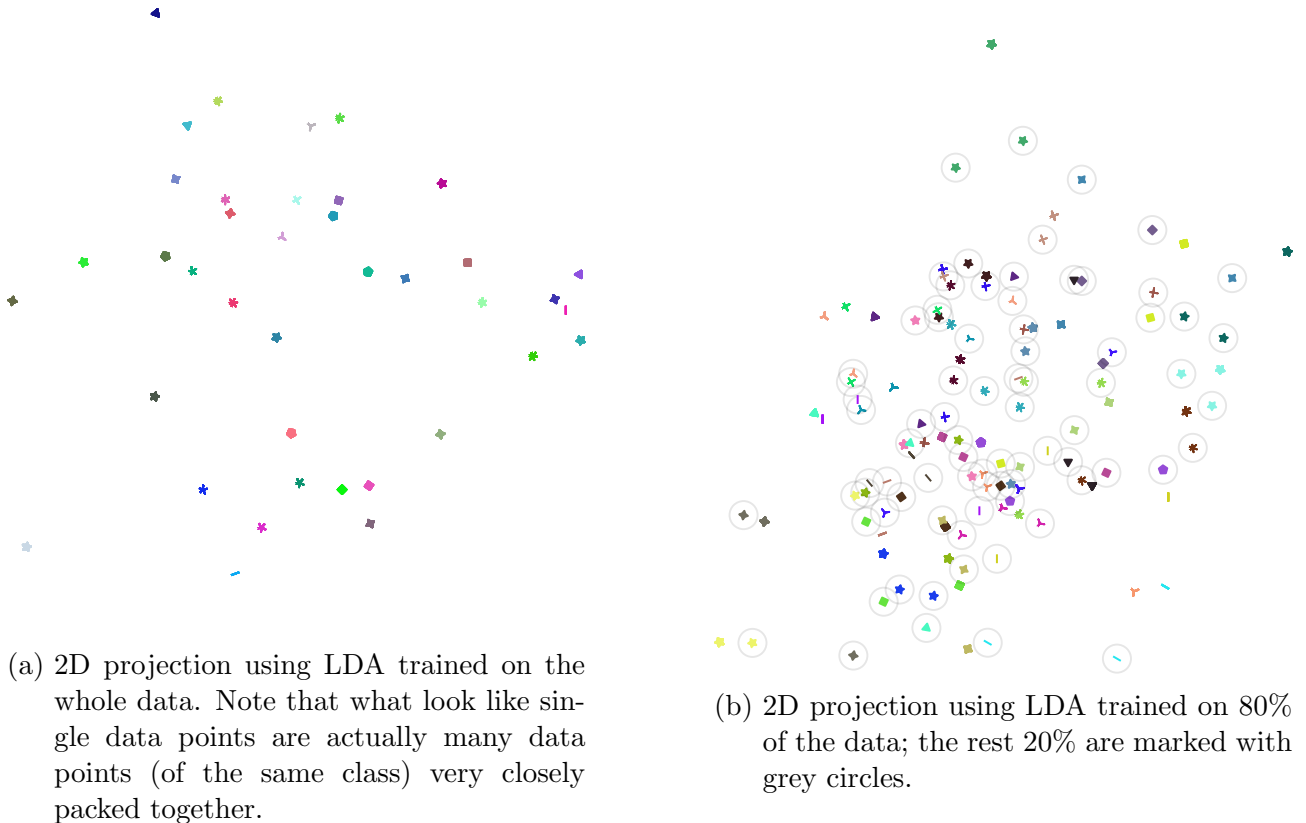


Figure 1.2.: The dataset is the AT&T dataset of faces, formerly the ORL face dataset, with an input dimensionality of 10,304, 40 classes, 10 data points per class. In (a) we can see that the two best LDA features fit the data very well, because each class forms a very compact cluster, well-separated from the rest. However, if we train LDA only on 80% of the data but project the whole dataset onto its best two features, as in (b), then we see that in each class, the 2 *testing* data points stray far away from the 8 training data points, which form a compact cluster that looks like a singleton point at this resolution.

The example in figure 1.1 may seem artificial and contrived, but its simplicity elucidates the problems that are commonplace in real-world, high-dimensional datasets. As a real-world example, let us consider the AT&T dataset of faces, formerly the ORL face dataset. It has 40 classes, 10 data points per class, and an input dimensionality of 10,304. More concretely, this means that there are 40 persons, 10 (greyscale) face images per person, and each face image has 10,304 pixels. We will extract 2 features from the 10,304 input features using LDA, and project

the data onto them. In figure 1.2a, we trained LDA using the whole data set, and projected the whole dataset onto its 2 best features. We can see the class discrimination very clearly. In fact, each class is so compact that it appears as a singleton point at this image resolution, and well-separated from the other classes. In this sense, the LDA features fit this data very well. In fact too well, as we see in figure 1.2b. In figure 1.2b, we split the data into an 80% *training set* and a 20% *testing set*. LDA is trained using just the training set to produce 2 features, but both the training set and the testing set are projected onto the features. The projected testing data points are marked with grey circles. Each class has 8 training data points that are tightly packed together much like figure 1.2a, but also 2 testing data points that are far from the 8 training data points. See for example the class on top of figure 1.2b marked by green stars. This illustrates that the feature extractor, LDA in this case, over-fits the training data, and fits the testing data poorly. In this sense, the extractor is said to give poor *generalisation performance*. For many feature extractors, the higher the input dimensionality of a dataset, the more likely this kind of over-fitting will occur. This is the curse of dimensionality as it manifests in dimensionality reduction for real-world applications.

The curse of dimensionality can be alleviated if we have an abundance of training data points that are highly representative of the true data distribution, as illustrated in figure 1.1b. However, for many real-world applications, this can be unrealistic in practice. The AT&T dataset of faces, for example, has 400 training data points, which is quite typical of a real-world scenario. Jain, Duin and Mao [16, sec.3] suggested that as a rough guideline, ideally we should have at least 10 times as many training data points as input dimensions. Now given that the input dimensionality of the AT&T dataset of faces (for example) is 10,304, such abundance of training data points is unrealistic for any real-world application. Hence the need for dimensionality reduction. However, as shown above, dimensionality reduction itself can suffer severely from curse of dimensionality. A detailed discussion of the effects of high input dimensionality, and the prevention and circumvention thereof, will be presented in chapter 3 of this thesis, and it constitutes one of the main contributions.

## 1.4. Scope

The scope of this thesis is *supervised linear feature extraction in classification*, and the use of *information-theoretic* techniques therein. In this section we put the scope of this thesis in the broader context of feature extraction, pattern recognition and machine learning.

### 1.4.1. Feature selection and feature extraction

A *feature* can be thought of as a line that lives in the typically high-dimensional input space, and it can be concretely represented by a unit vector. We wish to project the training data points onto a feature, or a subspace spanned by a set of features. Trivially, each of the basis vectors that span the input space is also itself a feature. *Feature selection* (or *variable selection*) is the process of selecting a subset of these original basis vectors that span the input space [15].

*Linear feature extraction*, on the other hand, is the process of constructing new features that are linear combinations of the original basis vectors. The dichotomy of linear and non-linear feature extraction will be explained in section 1.4.2.

In a sense, feature extraction generalises feature selection. However, feature selection has important applications in certain areas. For example, in classification of diseases using gene expression data [45, 39], where each feature corresponds to a gene (in contrast with face recognition where each feature corresponds to a pixel), it is often desirable to know which individual genes are related to the disease in question. This problem of gene selection is a natural application of feature *selection* [9, 39]. Feature *extraction*, on the other hand, would construct new features that are linear or non-linear combinations of genes, and so does not directly reveal the level of relevance of individual genes to a particular disease. Another application of feature selection is text classification [12, 8].

### **Wrapper, embedded, and filter methods of feature selection**

Feature selection can be subdivided into three kinds of techniques: *wrapper*, *embedded*, and *filter*. The *wrapper* technique [18] uses the classification performance of a particular classifier to assess the quality of a feature, and selects features based on this assessment. It treats the classifier as a black box, and so is classifier-agnostic. By its formulation, wrappers rely on training the classifier repetitively to perform feature selection, and so may have a comparatively large computation overhead. *Embedded* feature selectors are formulated as part of a classifier. Decision trees are examples of classifiers in which feature (variable) selection is an essential part [5]. *Filter* methods perform feature selection as a pre-processing step, completely independent of the classifier. Like wrappers, filters are classifier-agnostic, but unlike wrappers, filters do not require training the classifier, and so are often more computationally efficient.

The trichotomy of the wrapper, embedded, and filter techniques are not so emphatic in the feature *extraction* literature. However, it is possible to conceive of their analogy in feature extraction. For example, a feature extraction algorithm that iteratively improves the quality of a feature by incrementally changing its orientation, where quality is assessed by cross-validation using a classifier, could be regarded as a wrapper method, since it requires the classifier to assess the quality of a feature but is agnostic as to which classifier is used. In this view, all the feature extraction techniques considered in this thesis are *filter* techniques; they are completely independent of the classifier, do not require training the classifier, and should be used as a preprocessing step before classification.

### **1.4.2. Linear and non-linear feature extraction**

There is a degree of ambiguity in the feature extraction literature, regarding the words “linear” and “non-linear”. Here we will focus on two different meanings of *non-linear* feature extraction.

The first meaning of *non-linear* is that it qualifies the kind of transformation (projection) of data points from the high-dimensional input space to the lower-dimensional feature subspace. A *linear* transformation can always be written as multiplying a data point (which is a vector

in the input space) by a matrix, where the resulting vector is the corresponding data point in the *feature subspace*. Thus, to say that a feature extraction technique is *linear* is equivalent to saying that the extracted features are straight lines in the input space. Examples of linear feature extraction include PCA, LDA, and the two new techniques developed in this thesis, EMI and SDM. We will introduce them in chapters 2 and 3. An example of *non-linear* feature extraction, in the sense described above, is Kernel PCA [3, sec.12.3]. In Kernel PCA, the projection of data points from the input space to the feature space cannot be formulated as multiplication by a matrix.

Another meaning of *non-linear* is more of an interpretation rather than actual (mathematical) non-linearity. For example, Isomap is widely interpreted as a non-linear dimensionality reduction technique [47, 27]. Indeed, its objective is to find a lower-dimensional subspace that preserves the (estimated) *geodesic distances* between pairs of data points in the input space. It assumes that the data points live on a lower-dimensional *manifold* embedded in the high-dimensional input space. This manifold may be, for example, a curvy surface. The non-linear *geodesic distance* is the distance between two points along this manifold. This is in contrast with the linear Euclidean distance, which is simply the distance between two points along a straight line. Isomap tries to map the data points embedded in this non-linear manifold onto a (linear) feature subspace with the same *intrinsic dimensionality* as the manifold. In this view, Isomap clearly has a non-linear interpretation. However, the ultimate optimisation of its objective function can be formulated as an eigendecomposition problem, and the solutions, which are the features to extract, are vectors (straight lines) that live in the input space [55]. Therefore, although Isomap has a non-linear interpretation, the formulation of the essential optimisation is still one of *linear* feature extraction. That is to say, the projection of data points onto the lower-dimensional feature space is still essentially projection onto straight lines. The authors in [55] present a unifying framework that generalises dimensionality techniques including Laplacian Eigenmaps [2], Locally Linear Embedding [38], Isomap [47], Hessian LLE [10], and Local Tangent Space Alignment [56]. All of these techniques have non-linear interpretations, but as shown in [55], the essential formulations of these techniques are ones of *linear* feature extraction.

The scope of this thesis is *linear* feature extraction. We use *linear* to mean the kind of projection from input space to feature space. That is, all the techniques considered in [55] will be considered linear. Kernel PCA, however, is still considered non-linear.

### 1.4.3. Supervised and unsupervised feature extraction

*Supervised* feature extraction means that the class labels of the training data points are taken into account in formulating the feature extraction technique. The resulting features will depend on the class labels of the training data points. *Unsupervised* feature extraction techniques are independent of the class labels of the data points, and usually consider the relations between pairs of data points in the input space. An example of *unsupervised* feature extraction technique is PCA. More examples of unsupervised feature extraction include all but one of the techniques

considered in [55] (excluding LDA), and all those considered in [27]. Notice from [55] and [27] that in the unsupervised context, the term *dimensionality reduction* is often used instead of *feature extraction*. In this thesis, we will use the two terms interchangeably.

Examples of *supervised* feature extraction include LDA, and the two new techniques developed in this thesis, EMI and SDM. We will focus mainly on supervised feature extraction. However, we will use PCA in experimental comparisons, since it is popular even in supervised contexts, and often performs better than supervised techniques in classification. In fact, in chapter 3 we will study why PCA out-performs some supervised techniques for some datasets, and propose a new technique that can out-perform PCA under these settings.

#### 1.4.4. Classification and regression

In one sense, the problem of regression can be regarded as more general than the problem of classification. In both classification and regression, each input data point is associated with an *output value*. For training data points, these output values are given; the task of the learner (classification or regression algorithm) is to predict the output value for new data points whose output values are not known in advance. In classification, these output values are discrete, and must be one of a finite number of values known as *classes*. For example, in face recognition, each *class* corresponds to an individual person, and each data point corresponds to a single face image. In regression on the other hand, the output values can live in a continuous set, for example  $\mathbb{R}$ .

The techniques for solving classification and regression, respectively, may vary significantly. In particular, we may exploit the discreteness of the output value in classification. For a comprehensive introduction to regression and classification, we refer the reader to [3], and in particular chapters 3 and 4 therein. The scope of this thesis is feature extraction for *classification*.

#### 1.4.5. Classifiers

A complete classification package usually consists of at least two parts: feature extractor and classifier. In some cases the feature extractor may be a part of the classifier (as mentioned in subsection 1.4.1), and in other cases the classifier may be separate from the feature extractor.

Different types of classifiers abound in the machine learning literature [3], and some are more ‘sophisticated’ than others. Among the simplest classifiers are the Naive Bayes Classifier [37] and the nearest neighbour classifier [28]; among the more ‘sophisticated’ classifiers are the Support Vector Machine, Neural Networks, and the Relevance Vector Machine [3]. Some ‘sophisticated’ classifiers map the input data points to another space non-linearly, as part of the classification procedure. The 2-class Support Vector Machine, for example, maps the input data points onto an infinite-dimensional space (if we use the Gaussian kernel for example), before mapping it back down to a one-dimensional space (a line) for classification. As such, support vector machines may not be suitable for a study of feature extraction algorithms. Indeed, support vector machines may themselves be regarded as feature extraction algorithms in their

own right, and we will discuss this in section 4.1.3.

In this thesis we want to focus on studying the behaviour of *feature extractors*, in the context of classification but independently of classifiers. As such we will use one of the simplest classification algorithms for evaluation: the nearest neighbour classifier, which classifies a new testing data point according to the training data point nearest to it in the feature subspace.

## 1.5. Contributions

The main contributions of this thesis will be presented in chapters 2 and 3.

In chapter 2 we will study the behaviour of a current information-theoretic feature extraction algorithm that is based on gradient ascent, and propose a new information-theoretic feature extractor that has a closed-form solution based on eigendecomposition. To the author’s knowledge, no other information-theoretic feature extraction algorithm has so far been proposed, that has an eigenvalue solution. The fact that our new technique is based on eigendecomposition makes it significantly more computationally efficient, and enables us to study information-theoretic feature extraction in high-dimensional input spaces.

In chapter 3 we will study the behaviour of feature extraction algorithms in high-dimensional input spaces, and investigate why PCA often out-performs more ‘sophisticated’ feature extractors, including LDA. We will propose a new algorithm that is able to consistently out-perform PCA in the high-dimensional setting.

### 1.5.1. Publications

An algorithm for feature selection using order statistics is published as the following. In this algorithm, the data points in the input space are pre-processed such that only their orderings along each input feature is retained. This reveals a measure of overlap between classes, based on which we can select features with minimal overlap.

- Raymond Liu and Duncan F. Gillies. “Feature selection using order statistics”. In: *Proceedings of the Eleventh International Conference on Pattern Recognition and Information Processing*. 2011

A feature extraction algorithm using eigenvalue-based mutual information (which we will see in chapter 2 of this thesis) is introduced in the following publication.

- Raymond Liu and Duncan F. Gillies. “An eigenvalue-problem formulation for non-parametric mutual information maximisation for linear dimensionality reduction”. In: *International conference on image processing, computer vision, and pattern recognition*. Vol. 2. 2012, pp. 905–910

A more detailed theoretical account of the underlying estimate of mutual information employed in [22] (above) is presented in the following.

- Raymond Liu and Duncan F. Gillies. “An Estimate of Mutual Information that Permits Closed-Form Optimisation”. In: *Entropy* 15.5 (2013), pp. 1690–1704. ISSN: 1099-4300. DOI: 10.3390/e15051690. URL: <http://www.mdpi.com/1099-4300/15/5/1690>

## 1.6. Notation

Here we provide a brief description of parts of our notational convention that may depart from established mathematical conventions, and present some of the most commonly used symbols and their usual meanings.

Vectors and matrices of real numbers (as opposed to random variables) will have single and double underlines respectively, like  $\underline{x}$  and  $\underline{\underline{x}}$ . Other notational conventions may use capital letters for matrices, but we want to reserve capital letters for random variables, for example a random vector and a random matrix may look like  $\underline{X}$  and  $\underline{\underline{X}}$  respectively. The probability density function of a random variable  $X$ , evaluated at a point  $x$ , will be denoted by  $p_X(x)$ . For the sake of visual clarity and brevity, we will sometimes omit the subscript on the  $p$  at the slight risk of confusion, but meanings should be clear from context. So for example  $p(x, y)$  and  $p(x|y)$  will mean  $p_{X,Y}(x, y)$  and  $p_{X|Y}(x|y)$  respectively.

The basic notational design of our problem is as follows. We are given  $N$  data points  $\underline{x}_1, \dots, \underline{x}_N$  that live in  $D$ -dimensional *input space*. Usually  $D \gg 1$ . Sometimes it is convenient to use the  $N \times D$  *design matrix*  $\underline{\underline{x}}$ , whose  $n^{\text{th}}$  row is the  $n^{\text{th}}$  data point  $\underline{x}_n^T$ . In *supervised* problems, each training data point  $\underline{x}_n$  will have an associated class label  $c_n$ ,  $1 \leq c_n \leq K$ . We wish to find  $P$  unit vectors  $\underline{w}_1, \dots, \underline{w}_P$  in the input space, where  $P < D$ , onto which we wish to project our data points. The projected data points are usually denoted by  $\underline{y}_1, \dots, \underline{y}_N$ , and the space they live in is called the *feature space* or *feature subspace*. Note that  $\underline{x}_1, \dots, \underline{x}_N, \underline{w}_1, \dots, \underline{w}_P \in \mathbb{R}^D$  and  $\underline{y}_1, \dots, \underline{y}_N \in \mathbb{R}^P$ . Again, sometimes it is convenient to use the design matrix  $\underline{\underline{y}}$  in the feature space, whose  $n^{\text{th}}$  row is the  $n^{\text{th}}$  projected data point  $\underline{y}_n^T$ . The column vectors  $\underline{w}_1, \dots, \underline{w}_P$  can be collected together to form the *projection matrix*  $\underline{\underline{w}}$ , whose  $p^{\text{th}}$  column is  $\underline{w}_p$ . Thus we have  $\underline{\underline{y}} = \underline{\underline{x}}\underline{\underline{w}}$  and in particular, for each  $n = 1, \dots, N$  we have  $\underline{y}_n = \underline{\underline{w}}^T \underline{x}_n$ . With this notation, the challenge of linear feature extraction therefore is to find a ‘good’ projection matrix  $\underline{\underline{w}}$ .

## 1.7. Distance-based approach

The classic feature extraction techniques can be understood from two equivalent perspectives, one in terms of the variance of the data distribution, and another in terms of the distances between data points. We will use as an example, perhaps the most common feature extraction technique, principle component analysis (PCA) [3, sec.12.1]. PCA finds a set of features, or equivalently a feature subspace, in which the total variance of the data is maximised. Thus the

quantity to maximise is (refer to section 1.6 for notation)

$$\frac{1}{N} \sum_{n=1}^N \|\underline{y}_n - \bar{\underline{y}}\|^2 \quad (1.1)$$

where  $\bar{\underline{y}} := \frac{1}{N} \sum_{n=1}^N \underline{y}_n$  is the sample mean of the projected data points. Note that  $(\underline{y}_n - \bar{\underline{y}}) = \underline{\underline{w}}^T(\underline{x}_n - \bar{\underline{x}})$ , and equation (1.1) can be written as follows.

$$\begin{aligned} \frac{1}{N} \sum_{n=1}^N \|\underline{y}_n - \bar{\underline{y}}\|^2 &= \frac{1}{N} \sum_{n=1}^N \text{tr} \left[ (\underline{y}_n - \bar{\underline{y}})(\underline{y}_n - \bar{\underline{y}})^T \right] \\ &= \frac{1}{N} \sum_{n=1}^N \text{tr} \left[ \underline{\underline{w}}^T (\underline{x}_n - \bar{\underline{x}}) (\underline{x}_n - \bar{\underline{x}})^T \underline{\underline{w}} \right] \\ &= \text{tr} \left[ \underline{\underline{w}}^T \left( \frac{1}{N} \sum_{n=1}^N (\underline{x}_n - \bar{\underline{x}}) (\underline{x}_n - \bar{\underline{x}})^T \right) \underline{\underline{w}} \right]. \end{aligned} \quad (1.2)$$

It is easy to show [3, sec.12.1] that the optimal  $\underline{\underline{w}}$  is given by the largest eigenvectors of the total sample covariance matrix

$$\underline{\underline{s}}_t := \frac{1}{N} \sum_{n=1}^N (\underline{x}_n - \bar{\underline{x}}) (\underline{x}_n - \bar{\underline{x}})^T. \quad (1.3)$$

By “largest eigenvectors”, we mean the eigenvectors with the largest eigenvalues. (Of course the eigenvectors are assumed to be given as unit vectors.) These largest eigenvectors are called *principal components*. The above is the variance-based perspective of PCA. We will now derive the distance-based perspective of PCA. By noting that

$$\sum_{n=1}^N \sum_{m=1}^N \underline{x}_n \underline{x}_m^T = \sum_{n=1}^N \underline{x}_n \sum_{m=1}^N \underline{x}_m^T \quad \text{and} \quad \sum_{n=1}^N \underline{x}_n = N \bar{\underline{x}} \quad (1.4)$$

and some symmetries in  $n$  and  $m$  in the sums, we have the following.

$$\begin{aligned} \sum_{n=1}^N \sum_{m=1}^N (\underline{x}_n - \underline{x}_m) (\underline{x}_n - \underline{x}_m)^T &= 2N \sum_{n=1}^N \underline{x}_n \underline{x}_n^T - 2 \sum_{n=1}^N \sum_{m=1}^N \underline{x}_n \underline{x}_m^T \\ &= 2N \sum_{n=1}^N \underline{x}_n \underline{x}_n^T - 2N^2 \bar{\underline{x}} \bar{\underline{x}}^T = 2N \left[ \sum_{n=1}^N \underline{x}_n \underline{x}_n^T - 2N \bar{\underline{x}} \bar{\underline{x}}^T + N \bar{\underline{x}} \bar{\underline{x}}^T \right] \\ &= 2N \sum_{n=1}^N (\underline{x}_n - \bar{\underline{x}}) (\underline{x}_n - \bar{\underline{x}})^T = 2N^2 \underline{\underline{s}}_t. \end{aligned} \quad (1.5)$$

Thus we see that eigendecomposition of  $\underline{\underline{s}}_t$  is equivalent to eigendecomposition of  $\sum_{n=1}^N \sum_{m=1}^N (\underline{x}_n - \underline{x}_m)(\underline{x}_n - \underline{x}_m)^T$ . Similarly, we can show that

$$2N \sum_{n=1}^N \|\underline{y}_n - \underline{\bar{y}}\|^2 = \sum_{n=1}^N \sum_{m=1}^N \|\underline{y}_n - \underline{y}_m\|^2. \quad (1.6)$$

Now recall that PCA seeks a lower-dimensional projection of the data that maximises the quantity (1.1), or equivalently the LHS of equation (1.6). Then we see that equation (1.6) reveals a distance-based perspective of PCA, that it seeks a lower-dimensional projection of the data such that the squared pair-wise distances  $\|\underline{y}_n - \underline{y}_m\|^2$  between data points are maximised.

So far we have shown that PCA can be done either by seeking the largest eigenvectors of  $\underline{\underline{s}}_t$  or equivalently by seeking the largest eigenvectors of the matrix  $\sum_{n=1}^N \sum_{m=1}^N (\underline{x}_n - \underline{x}_m)(\underline{x}_n - \underline{x}_m)^T$ . One would not usually choose to do the latter because the matrix  $\underline{\underline{s}}_t$  is much more computationally efficient to construct, and furthermore, it lends itself to further computation optimisation in the case when  $D \geq N$  [3, sec.12.1.4]. However, the distance-based perspective of PCA lends itself to a useful, more general family of distance-based feature extraction techniques. Several names have been given to members or subsets of this family, including weighted PCA [44, 54, 29], Laplacian PCA [57], and Laplacian Eigenmaps [2]. We will adopt the name *Laplacian maps* for the general family of feature extraction methods, which we will define below. It is worth mentioning that there is another view of PCA, in which the subspace is one that minimises the *reconstruction error* [32, 3, sec.12.1.2].

Note that the LHS of (1.5) is a sum of (rank 1) matrices, each of which is constructed by a pair of data points  $(\underline{x}_n, \underline{x}_m)$ . Now consider a weight  $\omega_{nm} \in \mathbb{R}$  on every pair  $(\underline{x}_n, \underline{x}_m)$  of data points, and define  $\underline{\underline{\omega}} \in \mathbb{R}^{N \times N}$  to be the *weight matrix*, whose  $(n, m)^{\text{th}}$  element is the weight  $\omega_{nm}$ . (Not to be confused with the projection matrix  $\underline{\underline{w}}$ . Throughout this thesis we assume that each  $\omega_{nm}$  is symmetric in  $n$  and  $m$ , that is,  $\omega_{nm} = \omega_{mn}$ . Recall that  $\underline{\underline{x}}$  is the design matrix, and define an *objective matrix*

$$\underline{\underline{m}}(\underline{\underline{\omega}}, \underline{\underline{x}}) := \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N \omega_{nm} (\underline{x}_n - \underline{x}_m)(\underline{x}_n - \underline{x}_m)^T. \quad (1.7)$$

By a similar analysis to that in PCA, it is easy to show that the largest (or smallest) eigenvalues of  $\underline{\underline{m}}(\underline{\underline{\omega}}, \underline{\underline{x}})$  span a subspace in which the following is maximised (or minimised).

$$f(\underline{\underline{\omega}}, \underline{\underline{y}}) := \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N \omega_{nm} \|\underline{y}_n - \underline{y}_m\|^2. \quad (1.8)$$

Given weights  $\underline{\underline{\omega}}$  and data  $\underline{\underline{x}}$ , let  $\underline{\underline{w}}(\underline{\underline{\omega}}, \underline{\underline{x}})$  be the projection matrix whose columns comprise the eigenvectors of  $\underline{\underline{m}}(\underline{\underline{\omega}}, \underline{\underline{x}})$ , then we say that  $\underline{\underline{w}}$  is a *Laplacian map*; its columns are features and it maps a high-dimensional data point  $\underline{x}$  onto a lower-dimensional subspace via  $\underline{x} \rightarrow \underline{\underline{w}}^T \underline{x}$ . For the sake of compactness we will sometimes omit the parentheses after  $\underline{\underline{m}}$ ,  $f$ , and  $\underline{\underline{w}}$ . Notice from equations (1.5) and (1.6) that PCA can be formulated as just a special Laplacian map

where all the weights  $\omega_{nm}$  are 1. The reason we call these *Laplacian* maps is that equations (1.7) and (1.8) can be rewritten, in matrix form, as

$$\underline{\underline{m}} = \underline{\underline{x}}^T \underline{\underline{l}} \underline{\underline{x}} \quad \text{and} \quad f = \text{tr}[\underline{\underline{y}}^T \underline{\underline{l}} \underline{\underline{y}}], \quad (1.9)$$

where the *Laplacian matrix*  $\underline{\underline{l}}$  is defined to be

$$\underline{\underline{l}} := \text{diag}[\underline{\underline{\omega}} \underline{\underline{1}}] - \underline{\underline{\omega}}, \quad (1.10)$$

where we define the 1-vector  $\underline{\underline{1}} := (1, \dots, 1)^T$ .

Laplacian matrices stem from graph theory. Given an *undirected weighted graph*, the  $(i, j)^{\text{th}}$  off-diagonal element of the Laplacian matrix is the negative weight of the edge between the  $i^{\text{th}}$  and the  $j^{\text{th}}$  vertices, or 0 if those vertices are not adjacent to each other. Each  $(i, i)^{\text{th}}$  diagonal element is  $(-1) \times (\text{sum of the off-diagonal elements in the } i^{\text{th}} \text{ row})$ . Assuming that all the weights between vertices are non-negative, then the Laplacian matrix has the following properties [52].

- It is symmetric.
- Each row sums to 0.
- It is positive semi-definite.

So in this view, the training dataset is a graph where each data point is a vertex, and each unordered pair  $\{\underline{\underline{x}}_n, \underline{\underline{x}}_m\}$  of data points has an edge with weight  $\omega_{nm}$  which is symmetric in  $n$  and  $m$  and could be 0.

The methodology of finding eigenvectors of  $\underline{\underline{m}}$  in order to maximise the objective function  $f$ , as in equations (1.9), is in fact very general. If we relax the requirement that  $\underline{\underline{l}}$  be a Laplacian, then it is shown in [55] that this methodology generalises many popular dimensionality reduction techniques including Laplacian Eigenmaps [2], Locally Linear Embedding [38], Isomap [47], Hessian LLE [10], and Local Tangent Space Alignment [56]. This generalisation stems from the observation that all of these methods are eigendecomposition problems, and that the constraints of the underlying optimisation in each method can be encoded in the matrix  $\underline{\underline{l}}$ .

So far we have considered weighting each pair of data points. This generalises weighting each individual data point as in, for example, the maximisation of  $\sum_{n=1}^N \omega_n \|\underline{\underline{y}}_n - \underline{\underline{y}}\|^2$ , where the equivalent pairwise weights are  $\frac{\omega_n \omega_m}{\sum_n \omega_n}$ . A localised version of this can be found in [57]. A different approach to a weighted generalisation of PCA is to consider weights on the input features, as opposed to data points. The name *spatial weighting* has been given to this approach in [44]. In image recognition for example, spatial weighting corresponds to weighting each pixel (or pairs thereof), and data point weighting (so called *temporal weighting* in [44]) corresponds to weighting each image (or pairs thereof). One can of course use both weighting techniques simultaneously, at the expense of some increase in computational complexity.

To end this section, we briefly mention LDA [3, sec.4.1.4], which seeks a subspace that maximises between-class variance and minimises within-class variance. This subspace is spanned by the largest eigenvectors of the objective matrix  $\underline{\underline{s}}_w^{-1} \underline{\underline{s}}_b$ , where  $\underline{\underline{s}}_w$  and  $\underline{\underline{s}}_b$  are the within-class and the between-class covariances respectively. For a class  $c$  ( $1 \leq c \leq K$ ), let  $N_c$  be its size,  $I_c$  be its index set, and  $\bar{\underline{x}}_c$  be its mean. Let  $\bar{\underline{x}}$  be the over-all mean. Then  $\underline{\underline{s}}_w$  and  $\underline{\underline{s}}_b$  are defined by

$$\underline{\underline{s}}_w := \frac{1}{N} \sum_{c=1}^K \sum_{n \in I_c} (\underline{x}_n - \bar{\underline{x}}_c)(\underline{x}_n - \bar{\underline{x}}_c)^T \quad \text{and} \quad \underline{\underline{s}}_b := \sum_{c=1}^K \frac{N_c}{N} (\bar{\underline{x}}_c - \bar{\underline{x}})(\bar{\underline{x}}_c - \bar{\underline{x}})^T. \quad (1.11)$$

One problem with LDA as it is formulated above is that the matrix  $\underline{\underline{s}}_w$  has rank at most  $N - K$ , and so is not invertible for datasets whose input dimensionality is larger than that. Even if we apply PCA first and reduce the dimensionality of the data to  $N - 1$ ,  $\underline{\underline{s}}_w$  is still invertible. However, by using  $\underline{\underline{s}}_t = \underline{\underline{s}}_b + \underline{\underline{s}}_w$ , we can formulate LDA equivalently as seeking the largest eigenvectors of the objective matrix  $\underline{\underline{s}}_t^{-1} \underline{\underline{s}}_b$ . Now  $\underline{\underline{s}}_t$  has rank  $N - 1$  (as long as all the data points are linearly independent), which can still be smaller than the input dimensionality  $D$ . This is solved by the popular technique often known as PCA+LDA, where the dimensionality of the data is reduced to  $N - 1$  by first applying PCA. It can be shown [53] that preprocessing the data by PCA loses no discriminative information with respect to LDA. Therefore in this thesis, whenever we say LDA, we mean PCA+LDA.

PCA, LLE, Laplacian Eigenmaps, LTSA, Isomap are examples of *unsupervised* dimensionality reduction techniques, which means that these techniques do not take into account the categorical information of the data. LDA is an example of *supervised* feature extraction, where categorical information of the data is used as part of the algorithm. The term *dimensionality reduction* is often used in the unsupervised setting, for example data visualisation [27]; while the term *feature extraction* is often used in the supervised setting, for example classification. For our purposes, the two terms are interchangeable, and we will focus on supervised feature extraction in this thesis.

## 1.8. Information-theoretic approach

In section 1.7, all the dimensionality reduction techniques are based on distances between pairs of data points  $\|\underline{y}_n - \underline{y}_m\|$ , or distances between a data point and some centroid (mean)  $\|\underline{y}_n - \bar{\underline{y}}\|$ . These distances may be weighted (Laplacian Eigenmaps), or estimated along a manifold (for example in LLE), or be confined to local neighbourhoods (LLE, Laplacian Eigenmaps, LTSA, Isomap, and so on), but the interpretations thereof are based on distance or variance. In this section we study a view of feature extraction that is based on information.

The information-theoretic approach to feature extraction is motivated by the fact that classical distance-based techniques fail for some data distributions, as illustrated in figure 1.3 where the aim is to extract a single feature from 2D data. Here the two classes have almost equal means, so the between-class variance is almost zero. LDA therefore will try to find a direction

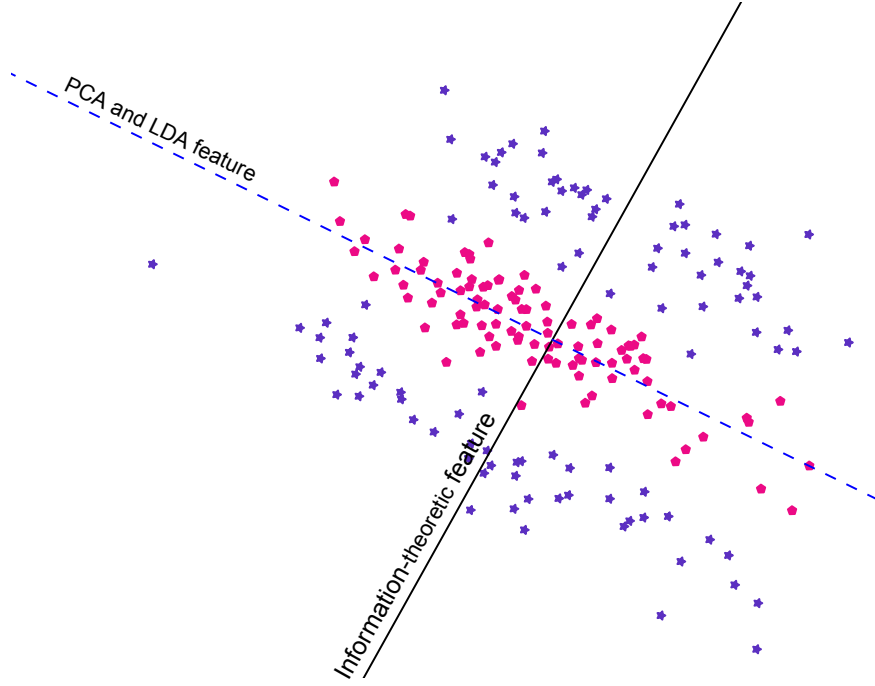


Figure 1.3.: This artificial 2D data is categorised into two classes: pink pentagons and blue stars. We can see that the black line is a better line of projection for the data, in terms of class-discrimination, than the blue dashed line. The blue dashed line is the feature computed using LDA; PCA produces a very similar feature in this case. The black line is computed using an information-theoretic technique (in fact our eigenvalue-based mutual information method [22, 23]).

along which the average within-class variance is minimised. This happens to be the blue dotted line shown, because all the other directions will produce larger class variances in the blue star class, due to the heavily bimodal nature of its class-conditional distribution. This is why LDA fails in maximising class-discrimination in this case. From this intuition we see that LDA is liable to fail whenever the class-conditional distributions are not unimodal Gaussian. PCA will seek a direction along which the variance of the whole data is maximised, which in this case is very similar to the LDA feature.

An information-theoretic approach, on the other hand considers the *probability distribution* of each class (the class-conditional distribution  $p_{Y|C}(y|c)$ ), and maximises the *divergence* between the distributions of different classes. The term *divergence* can be intuitively understood as the difference, or lack of overlap, between probability distributions. Along the blue dashed line in figure 1.3, we can see that the distributions of the two classes overlap completely. But along the information-theoretic feature (black line), the class distributions exhibit much better class-discrimination, and look like figure 1.4. Such features are found by maximising an estimate of the *mutual information* between the data  $Y$  along the feature and the class variable  $C$ .

Mutual information and *entropy* are closely related. In his seminal work on information theory [40], Shannon defined the entropy of a discrete random variable  $X$  to be

$$H_S(X) := - \sum_x p_X(x) \log_2 p_X(x), \quad (1.12)$$

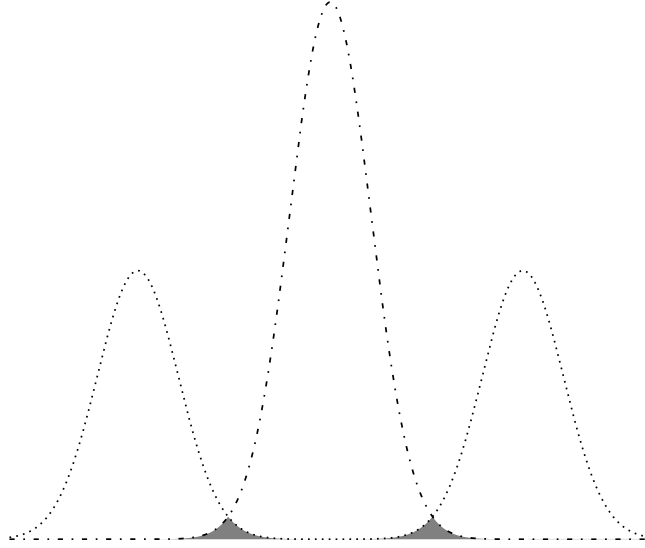


Figure 1.4.: Example distributions of a unimodal class and a bimodal class. The grey-shaded areas are where the distributions overlap.

where the sum is over all possible realisations of  $X$ . The subscript S on the  $H$  stands for Shannon. The entropy of a random variable is a measure of how unsure we are about its value, and for discrete random variables it is always at least 0, with equality iff it is in fact deterministic. The mutual information between two discrete random variables is defined as

$$I_S(X; Y) := \sum_x \sum_y p_{X,Y}(x, y) \log_2 \frac{p_{X,Y}(x, y)}{p_X(x)p_Y(y)}, \quad (1.13)$$

where the sums  $\sum_x$  and  $\sum_y$  run over all possible values of  $X$  and  $Y$  respectively. The mutual information between two random variables is the amount of information one random variable contains about the other. That is, how much less uncertain we are about  $X$  if we knew the value of  $Y$ . This is expressed in the following identity.

$$I_S(X; Y) = H_S(X) - H_S(X|Y), \quad (1.14)$$

where  $H_S(X|Y)$  is the *conditional entropy* of  $X$  given  $Y$ , that is, the average amount of uncertainty we have about  $X$  given that we know  $Y$ .

$$\begin{aligned} H_S(X|Y) &:= \mathbb{E}_Y \left[ - \sum_x p_{X|Y}(x|y) \log_2 p_{X|Y}(x|y) \right] \\ &= - \sum_y \sum_x p_{X,Y}(x, y) \log_2 p_{X|Y}(x|y). \end{aligned} \quad (1.15)$$

Mutual information is shown to be always non-negative [7, chap.2], with equality iff  $X$  and  $Y$  are independent.

If one or both of the random variables in (1.13) is continuous, we replace the corresponding sum with an integral. As a side-note, if  $X$  is a continuous variable, then we can replace the sum in (1.12) with an integral, to form the *differential entropy* of  $X$ . Note that while the entropy

of a discrete random variable is always non-negative, the differential entropy of a continuous random variable can be negative. However, whether the random variables  $X$  and  $Y$  are discrete or continuous, the mutual information  $I_S(X; Y)$  is always non-negative, and is 0 iff they are independent [7, chap.8].

Some feature extraction techniques try to maximise estimates of Shannon's mutual information as defined in equation (1.13) [41, 19, 30]. However, a popular alternative is to use Kapur's *quadratic mutual information* (QMI) [50], which we will study shortly. First note that Shannon's mutual information is closely related to *Kullback-Leibler divergence*, also known as *relative entropy*, defined as follows

$$D_{\text{KL}}(p \parallel q) := \sum_x p(x) \log_2 \frac{p(x)}{q(x)}, \quad (1.16)$$

where  $p$  and  $q$  are two probability mass functions of random variable  $X$  and the sum runs over all possible values of  $X$ .  $D_{\text{KL}}(p \parallel q)$  is called the divergence of  $q$  from  $p$ . Again, if  $X$  is a continuous random variable, the sum is replaced by an integral. By comparing (1.16) and (1.13) we can see that Shannon mutual information can easily be expressed as the KL divergence of the independent joint distribution  $p_X \cdot p_Y$  from the true joint distribution  $p_{X,Y}$ .

$$I_S(X; Y) = D_{\text{KL}}(p_{X,Y} \parallel p_X \cdot p_Y). \quad (1.17)$$

### 1.8.1. Quadratic mutual information (QMI)

Shannon's formulation of entropy and mutual information can be characterised by a set of intuitively evident axioms [40, 36]. Kapur [17, p.178] argued that "if the aim is not to calculate an absolute value of divergence, but rather to find a distribution that minimizes/maximizes the divergence, the axioms in deriving the measure can be relaxed and yet the result of the optimization is the same distribution" [50]. A number of families of alternative measures of divergence is proposed in [17], and one that is particularly mathematically and computationally convenient is the following.

$$D_Q(p, q) := \int \left( p(x) - q(x) \right)^2 dx. \quad (1.18)$$

Note that the integrand is a quadratic expression of the PDFs  $p$  and  $q$ , hence the subscript Q on the  $D$ . We can see that  $D_Q(p, q)$  is a measure of the total squared difference between  $p$  and  $q$ , and when  $p(x) = q(x) \forall x$ , then  $D_Q(p, q) = 0$ . With this definition of divergence between two PDFs, and with the identity (1.17) in mind, we can define the *quadratic mutual information* between two continuous random variables  $X$  and  $Y$  [50], as

$$I_Q(X; Y) := \int_X \int_Y \left( p_{X,Y}(x, y) - p_X(x)p_Y(y) \right)^2 dx dy. \quad (1.19)$$

Notice that when  $X$  and  $Y$  are independent, we have  $p_{X,Y}(x, y) = p_X(x)p_Y(y)$ , and so  $I_Q(X; Y) = 0$ . This confirms our intuition that when two random variables are independent, their mutual information should be 0. When one or both of  $X$  and  $Y$  is a discrete random variable, the respective integral is replaced by a sum. Now if we recall the notation  $Y = \underline{w}^T \underline{X}$  (section 1.6), we can write the QMI of (or along) a feature  $\underline{w}$  as

$$I_Q(Y; C) = \sum_{c=1}^K \int_Y \left( p(y, c) - p(y)p(c) \right)^2 dy. \quad (1.20)$$

Torkkola's QMI algorithm [50] find features along which  $I_Q(Y; C)$  is maximised. The features are found sequentially, where the  $(i + 1)^{\text{th}}$  feature is found in the space orthogonal to the space spanned by the first  $i$  features. Each feature is found by gradient ascent where the  $(t + 1)^{\text{th}}$  iteration is given by

$$\underline{w}_{t+1} = \frac{\underline{v}_{t+1}}{\|\underline{v}_{t+1}\|}, \quad \underline{v}_{t+1} = \underline{w}_t + \eta \nabla_{\underline{w}_t} I_Q(Y; C), \quad (1.21)$$

where  $\eta$  is the learning rate, which can be variable. The initial unit vector  $\underline{w}_0$  may be random or a feature computed by another feature extraction algorithm such as LDA.

### 1.8.2. Gaussian kernel density estimation

The QMI  $I_Q(Y; C)$  in the form of equation (1.20) involves PDFs of the data, which are not given. In order to apply gradient ascent, a concrete estimate of  $I_Q(Y; C)$  is needed. This can be done using *Gaussian kernel density estimation* [31, 3, sec.2.5.1], which is particularly convenient when applied to  $I_Q(Y; C)$  in the form of (1.20). We take advantage of the property that the convolution of two Gaussians is another Gaussian.

$$\int_{\mathbb{R}} G(z - x, \sigma_x^2) G(z - y, \sigma_y^2) dz = G(x - y, \sigma_x^2 + \sigma_y^2), \quad (1.22)$$

where the Gaussian function  $G(x - \mu, \sigma^2)$  is defined by

$$G(x - \mu, \sigma^2) := \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right). \quad (1.23)$$

This *Gaussian convolution property* will enable us to avoid numerical integration.

For a one-dimensional random variable  $X$  with continuous distribution, and given  $N$  realisations (data points)  $x_1, \dots, x_N$ , the PDF of  $X$  can be estimated by

$$p_X(x) \approx \frac{1}{N} \sum_{n=1}^N k(x \mid x_n, h), \quad (1.24)$$

where  $k(\cdot \mid x_n, h)$  is a *kernel function* corresponding to the data point  $x_n$ , with *kernel width*  $h$ , hence the name *kernel density estimation* (KDE). Intuitively, each  $\frac{k(\cdot \mid x_n, h)}{N}$  is a 'hat' put on

top of  $x_n$ , and KDE is the sum of the hats of the data points. The kernel can be chosen to be Gaussian.

$$k(x \mid x_n, h) := G(x - x_n, h^2), \quad \forall n. \quad (1.25)$$

By Bayes' theorem equation (1.20) can be re-written as

$$I_Q(Y; C) = \sum_{c=1}^K p(c)^2 \int_Y \left( p(y|c) - p(y) \right)^2 dy. \quad (1.26)$$

With KDE (1.24) the Gaussian kernel (1.25), the over-all probability distribution  $p_Y$  and the class-conditional distribution  $p_{Y|C}$  can be estimated as

$$\begin{aligned} p_Y(y) &= \frac{1}{N} \sum_{n=1}^N G\left(y - y_n, \frac{\sigma^2}{2}\right), \\ p_{Y|C}(y|c) &= \frac{1}{N_c} \sum_{n \in I_c} G\left(y - y_n, \frac{\sigma^2}{2}\right). \end{aligned} \quad (1.27)$$

The prior class probabilities  $p_C$  are simply estimated by its proportion in the training set,  $p_C(c) = \frac{N_c}{N}$ . We can unpack the parentheses in equation (1.26) to obtain three integrands,

$$I_Q(Y; C) = \sum_{c=1}^K p(c)^2 \int_Y \left[ p(y|c)^2 + p(y)^2 - 2p(y)p(y|c) \right] dy. \quad (1.28)$$

Now if we substitute in the estimates (1.27), the integrands can be written as

$$\begin{aligned} p(y)^2 &= \frac{1}{N^2} \sum_{n=1}^N \sum_{m=1}^N G\left(y - y_n, \frac{\sigma^2}{2}\right) G\left(y - y_m, \frac{\sigma^2}{2}\right), \\ p(y|c)^2 &= \frac{1}{N_c^2} \sum_{n \in I_c} \sum_{m \in I_c} G\left(y - y_n, \frac{\sigma^2}{2}\right) G\left(y - y_m, \frac{\sigma^2}{2}\right), \\ p(y)p(y|c) &= \frac{1}{NN_c} \sum_{n \in I_c} \sum_{m=1}^N G\left(y - y_n, \frac{\sigma^2}{2}\right) G\left(y - y_m, \frac{\sigma^2}{2}\right). \end{aligned} \quad (1.29)$$

Finally, if we substitute (1.29) into (1.28), and use the Gaussian convolution property (1.22), we arrive at a concrete estimate

$$I_Q(Y; C) = \frac{1}{N^2} \left[ \left( \sum_{c=1}^K \frac{N_c^2}{N^2} \right) \left( \sum_{n=1}^N \sum_{m=1}^N \right) + \left( \sum_{c=1}^K \sum_{n \in I_c} \sum_{m \in I_c} \right) - 2 \left( \sum_{c=1}^K \frac{N_c}{N} \sum_{n \in I_c} \sum_{m=1}^N \right) \right] g_{nm}, \quad (1.30)$$

where we use the short-hand notation

$$g_{nm} := G(y_n - y_m, \sigma^2). \quad (1.31)$$

The same derivation can be found in slightly more detail in [50]. Note that while the sums in (1.30) may look like triple sums, they're actually at most double sums over the whole dataset.

Now let us derive a more notationally compact form of  $I_Q(Y; C)$ , which will be useful later. If we define

$$\rho_{nm} := \frac{1}{N^2} \left( \mathbb{I}[c_n = c_m] + \left( \sum_{c=1}^K \frac{N_c^2}{N^2} \right) - \frac{2N_{c_n}}{N} \right), \quad (1.32)$$

then we claim that equation (1.30) can be simply written as

$$I_Q(Y; C) = \sum_{n=1}^N \sum_{m=1}^N \rho_{nm} g_{nm}. \quad (1.33)$$

The reason is as follows. Equation (1.30) has three sums: the total sum  $\sum_{n=1}^N \sum_{m=1}^N$ , the within-class sum  $\sum_{c=1}^K \sum_{n \in I_c} \sum_{m \in I_c}$ , and the mixed sum  $\sum_{c=1}^K \sum_{n \in I_c} \sum_{m=1}^N$ . Pick any *ordered* pair of indices  $(n, m)$  (as opposed to an unordered pair  $\{n, m\}$ ), and consider where  $g_{nm}$  appears in each of those three sums. It will appear exactly once in the total sum, with coefficient  $\left( \sum_{c=1}^K \frac{N_c^2}{N^2} \right)$ ; it will appear in the within-class sum, with coefficient 1, exactly when  $\underline{x}_n$  and  $\underline{x}_m$  are in the same class; finally, it will appear in the mixed sum exactly once (since we're considering *ordered* pairs  $(n, m)$ ), with coefficient  $\frac{2N_{c_n}}{N}$ .

## 1.9. Summary and structure of the thesis

We have introduced two broad approaches to feature extraction: distance-based and information-theoretic.

Under the distance-based paradigm, we have reviewed the classic techniques PCA and LDA. For PCA we noted the pairwise distance perspective, and have formulated a generalisation which we call Laplacian maps, of which PCA and Laplacian Eigenmaps are special cases. In chapter 2 we will see that this generalisation forms a link between information-theoretic and distance-based feature extraction. A slightly further generalisation of it (the relaxation of the Laplacian condition) encapsulates LLE, Isomap, HLLE, and LTSA [55]. Distance-based techniques are mathematically elegant and computationally efficient because the algorithms boil down to the eigendecomposition of some objective matrix.

Under the information-theoretic paradigm, we focused mainly on the maximisation of QMI [50]. So far we have seen that QMI can be maximised using gradient ascent. However, the iterative nature of gradient ascent makes it less stable and more computationally intensive than the eigendecomposition problem that underpins distance-based feature extraction. Furthermore, gradient ascent induces free variables in the algorithm which are difficult to estimate in a principled manner. One of the fundamental problems with using gradient ascent for mutual information maximisation is that the underlying optimisation problem is not convex. We will study these points in more detail in chapter 2, where we will also introduce a new information-theoretic feature extraction technique that has an eigendecompositional solution [22].

A very important concept to keep in mind when developing feature extraction or machine learning algorithms is the curse of dimensionality, which we have discussed in section 1.3. Experiments in [27] have shown that for real-world datasets, often PCA out-performs more recent,

more sophisticated dimensionality reduction algorithms (Laplacian Eigenmaps, LLE, Isomap, LTSA and so forth). Our experiments show similar results. The under-performance of a feature extraction algorithm is often interpreted in terms of data distribution. In particular, a common suggestion for improving the classification performance of a feature extraction technique is to *increase the discriminating power* [33, 29], or to *maximise the separation or margins* (of sorts) between classes [34, 20, 25]. In chapter 3 we will present a geometric view of feature extraction in terms of class-separation, from which we derive the argument that for high-dimensional datasets, the wisdom of maximising class-separation should be tamed with over-fitting in mind. Specifically, we find through experiments that under the high-dimensional setting, the naive approach to maximising class-separation often leads to bad classification performance, and that contrary to the intuition of maximising class-separation, lower class-separation entails better classification accuracy. We will explain this phenomenon in terms of curse of dimensionality. This insight will lead to a simple but effective family of feature extraction algorithms that consistently out-performs PCA in terms of classification accuracy.

Supervised and unsupervised feature extraction is still a very active area of research, and due to the plethora of different techniques and approaches, there is little agreement as to the standardisation of the field [14]. We will conclude our particular exposition of this subject, together with recommendations for future research, in chapter 4.

## 2. Information-theoretic feature extraction

In section 1.8 we introduced Shannon’s mutual information and quadratic mutual information (QMI). For continuous data, precise estimates of Shannon’s mutual information are extremely computationally intensive, since they involve numerical integration. For an algorithm to be feasible in maximising Shannon’s mutual information, one can resort to discretisation as an estimate. There are other estimates of mutual information from data, of which *squared-loss mutual information* is an example [46]. So far in the field of feature extraction, these estimates of mutual information can be optimised only through an iterative, gradient-ascent-type procedure. In this chapter we will introduce a new estimate of mutual information which we will call *eigenvalue-based mutual information* (EMI), that can be maximised via closed-form eigendecomposition, thus significantly reducing computational complexity while avoiding other difficulties associated with iterative optimisation. But first, we will study iterative procedures for maximising mutual information using QMI and Torkkola’s algorithm [50] as a representative example.

### 2.1. Gradient ascent for QMI maximisation

It was shown that Torkkola’s gradient ascent algorithm in maximising QMI can give superior classification results for some low-dimensional datasets, while the conventional feature extraction methods PCA and LDA perform better on others. No high-dimensional datasets were tested in [50] however, possibly due to the high computational complexity of iterative algorithms. QMI is a theoretically elegant and practically applicable estimate of mutual information. Regarding its maximisation by gradient ascent however, there are practical drawbacks, which can be summarised as follows.

- The computational complexity (cost) of iterative optimisation algorithms can be very high, and QMI in its current form, equations (1.30) or (1.33), can only be maximised iteratively. Table 2.1 shows a comparison of the computing times of PCA, LDA, Torkkola’s iterative QMI algorithm, and our proposed eigenvalue-based MI method.
- The current iterative algorithms have many free parameters, including the learning rate and the stopping criterion, for which there is not yet a principled method of estimating.

- Our experiments show that a straight-forward application of gradient ascent can have unstable and unpredictable convergence in high-dimensional spaces. In particular, it does not always maximise QMI since the shape of the objective function is not convex. Figure 2.1 illustrates this, for the AT&T Dataset of Faces. Note that this is not an intrinsic deficiency of QMI, but rather of gradient ascent as applied to maximising QMI.

|     | Time to extract |                  |
|-----|-----------------|------------------|
|     | first feature   | 39 features      |
| PCA | 0.406           | (0.406)          |
| LDA | 1.185           | (1.185)          |
| EMI | 4.992           | (4.992)          |
| QMI | <b>265.034</b>  | <b>10330.358</b> |

Table 2.1.: Times taken, in seconds, for algorithms PCA, LDA, eigenvalue-based MI, and iterative QMI to extract one feature and 39 features respectively, from the AT&T Dataset of Faces. The computing times for eigenvalue-based algorithms (PCA, LDA, EMI) are independent of the number of extracted features, due to the closed-form nature of eigendecomposition. This was a Python 2.7 implementation with packages Numpy, Matplotlib, and PIL (Python Imaging Library), executed on an Intel Core i7-2600K processor at 3.40 GHz with 16GB of RAM.

In figure 2.1, the reader may ask why LDA produces high values of QMI, similar to that produced by the EMI method, despite the fact that LDA is not designed to maximise QMI. To answer this question, we recall equation (1.26). Let us consider a simple 2-class example, where both classes have equal prior probabilities  $P_C(c) = \frac{1}{2}$ . In this case, equation (1.26) tells us that  $I_Q(Y; C)$  is proportional to a sum of two summands, each of which is the square of the area between the graphs of the class-conditional distribution  $p_{Y|C}$  and the overall distribution  $p_Y$ . Let us suppose further that the classes are normally distributed with the same variance but different means. Then figure 2.2 shows us that maximising QMI is equivalent to minimising the overlap between the 2 class-conditional distributions, or equivalently maximising the separation between the classes.

The intuition shown in figure 2.2 can be validly extrapolated to higher dimensions, general class-conditional distributions, and multi-class datasets. LDA maximises the class-separation of the data in the sense that it maximises the between-class distances and minimises within-class distances between data points. Now depending on the class-conditional distributions and input dimensionality, LDA will not always succeed, as illustrated in figure 1.3. However, where LDA does succeed, different classes will be well-separated and so the QMI along the features produced by LDA will be high. The AT&T Dataset of Faces is one in which LDA does succeed in maximising the separation between classes, as shown in figure 2.3a (and figure 1.2a). The similarity in the values of QMI obtained by LDA and our EMI method is explained by the similarity in their respective 2D projections shown in figures 2.3a and 2.3b. Figure 2.3c, in contrast, does not exhibit good separation between classes, which explains the relatively low values of QMI obtained by the iterative method as shown in figure 2.1. Furthermore, figure

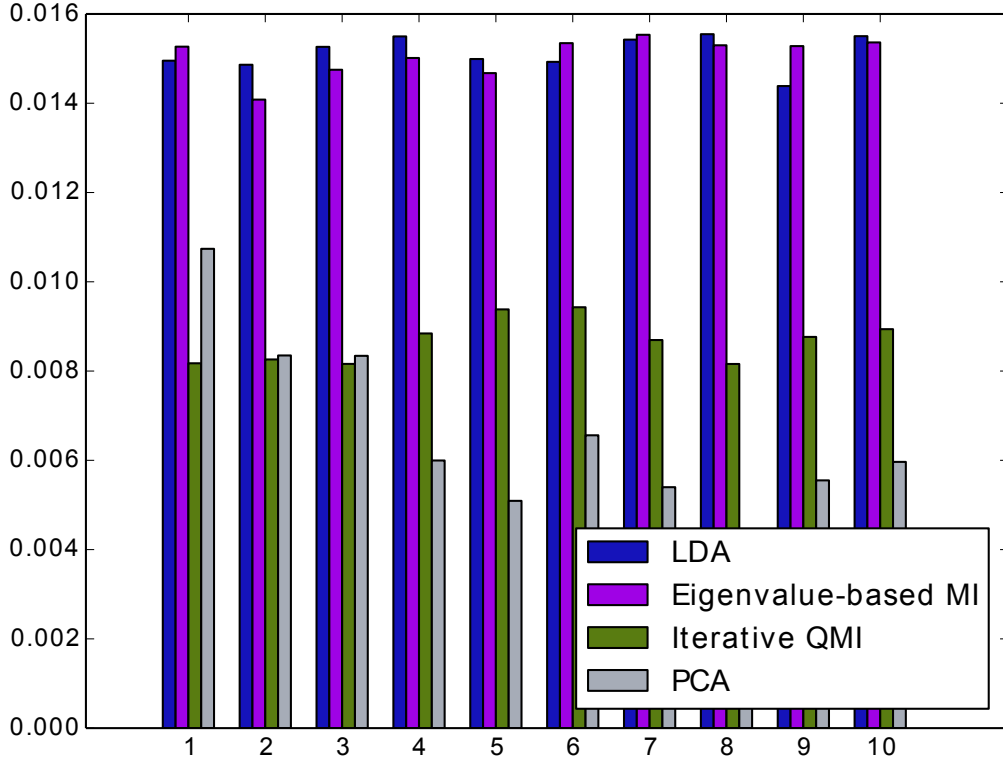


Figure 2.1.: Plot of the QMI between the class label and each of the first 10 features, computed by various methods. The horizontal axis indexes the features, while the vertical axis measures the QMI along each feature. We see that Torkkola’s iterative QMI algorithm gives lower values of QMI than the EMI algorithm, which gives values similar to LDA. Note that LDA is not designed to maximise QMI. PCA is also included for the sake of comparison. In the experiment, the data was whitened and the initial features for the iterative QMI algorithm were set to random features.

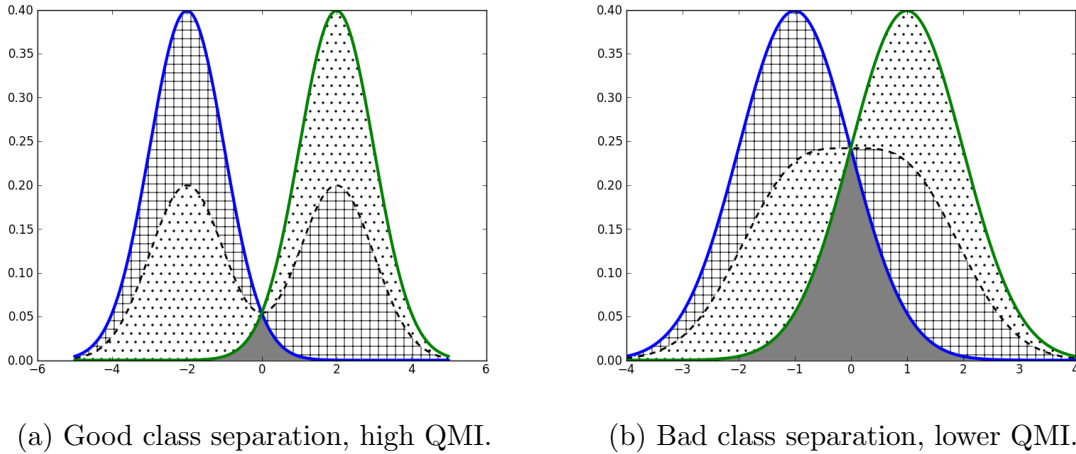


Figure 2.2.: The solid lines are the class-conditional distributions for the 2 classes, and the dashed line is the overall distribution  $p_Y$ . The checkered and the dotted areas correspond to the value of  $|p_{Y|C}(y|c) - p_Y(y)|$  for each of the 2 classes respectively, and the squares of only these areas contribute to the value of QMI, not the grey-shaded area. The grey-shaded area is the overlap between the two class-conditional distributions. We see that the smaller the overlap, the larger the value of the dotted and checkered areas, and therefore the larger the value of QMI.

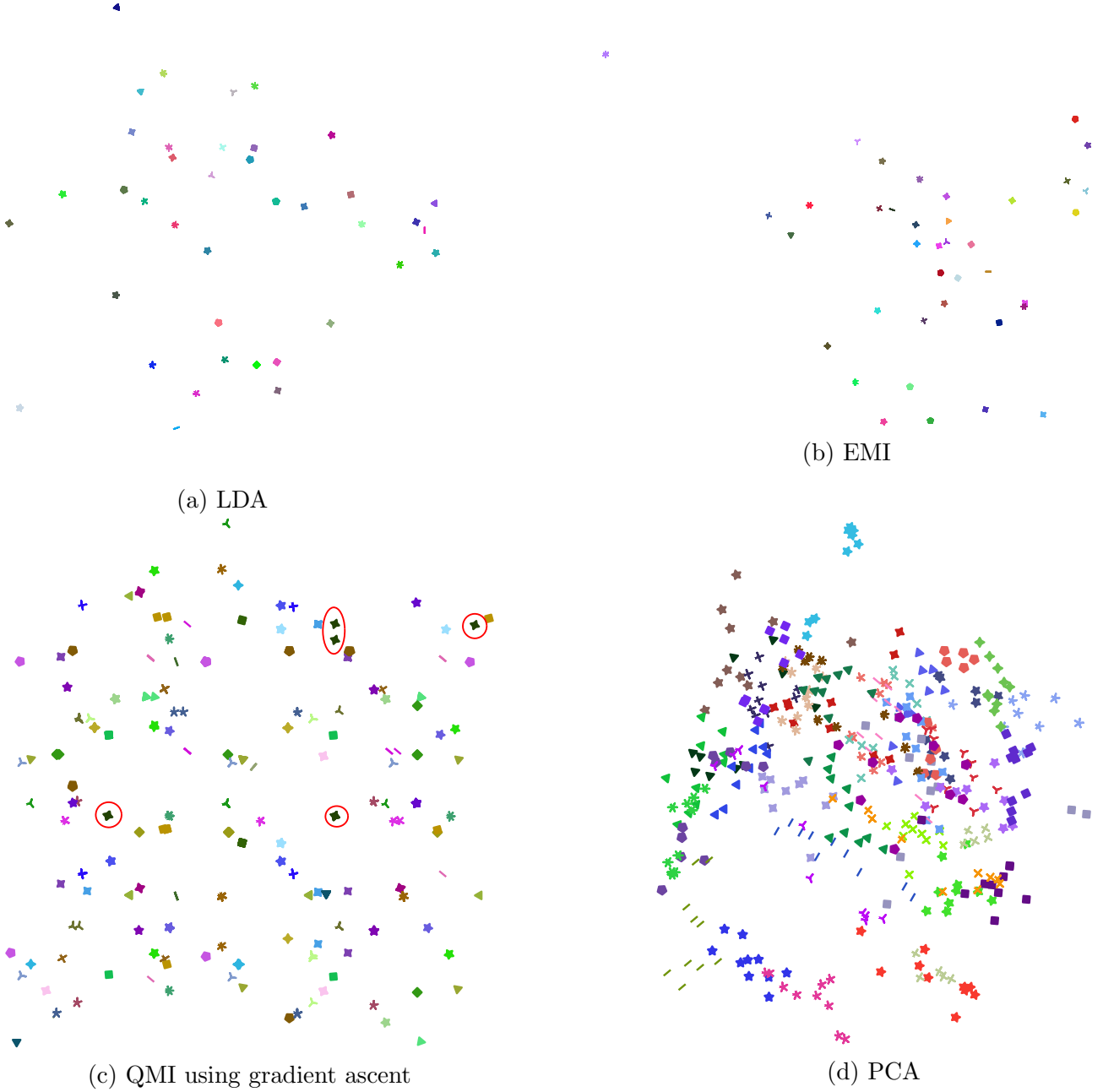


Figure 2.3.: 2D projections of the AT&T Dataset of Faces, computed using (a) LDA, (b) our EMI method, (c) Torkkola’s iterative QMI method, and (d) PCA for comparison. In (a) we see that each of the 40 classes in the dataset is in a compact cluster, well separated from others. In fact, each class is so tightly packed that at this resolution it looks like a singleton point. Similar observations can be made from (b). In (c) however, each class seems to be scattered into 2 or more clusters, as exemplified by the class marked by red circles. Note that each class of the dataset has 10 data points, and the class marked by red circles seems to be in 4 clusters. For all of (a), (b) and (c), the data was whitened as a pre-processing step.

2.3c suggests that the iterative QMI method in this case might have sought a local optimum, due to the scattering nature of individual classes into more than one cluster.

In what follows in this section we will discuss some of the free parameters and other aspects of maximising QMI by gradient ascent. Some of these also apply to other iterative algorithms.

### 2.1.1. Whitening

By *whitening*, we mean pre-processing the data by normalised PCA such that the resulting data has a variance of 1 in any direction. Specifically, the data is projected onto its principal component space where in addition, each component of the data is divided by the square root of the corresponding eigenvalue of the total sample covariance matrix. The resulting data after this pre-processing has dimensionality at most  $\min\{D, (N - 1)\}$ .

In producing figures 2.3 and 2.1, the AT&T Dataset of Faces was whitened as a pre-processing step. In practical terms, whitening makes little difference to LDA in real-world high-dimensional datasets. The LDA projection of the AT&T Dataset of Faces, for example, looks like figure 2.3a whether or not the data is whitened. Our EMI algorithm that we will introduce later requires that whitening be applied as a pre-processing step, for reasons that will become clear.

On maximising QMI via gradient ascent however, whitening seems to have a significant effect for high-dimensional datasets, as shown in figures 2.3c and 2.4. In figure 2.3c the data was whitened, whereas in figure 2.4 the same data was not. With the exception of whitening, all other parameters of the algorithms used to produce the two figures are kept identical. We see that figure 2.4 exhibits weak class discrimination and a high degree of overlap between classes, and that while figure 2.3c may show less overlap between classes, it nevertheless suggests that the algorithm might have sought a local maximum. In fact, figure 2.4 produces a lower value of QMI than the PCA projection shown in figure 2.3d.

The following is a possible explanation for the under-performance of the iterative QMI algorithm without whitening. We will show that without whitening, the algorithm may be data-dependent and the convergence may be unstable. Recall from equation (1.33) that  $I_Q(Y; C)$  is a weighted sum of  $g_{nm}$  terms, where  $g_{nm}$  is defined in equation (1.31) (and equation (1.23)). In a gradient ascent algorithm, according to equation (1.21), we need to consider the derivative  $\nabla_{\underline{w}} I_Q(Y; C)$ , which is a sum of  $\nabla_{\underline{w}} g_{nm}$  terms. Each  $\nabla_{\underline{w}} g_{nm}$  term is given by

$$\nabla_{\underline{w}} g_{nm} = -\frac{(y_n - y_m)}{\sigma^2} \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(y_n - y_m)^2}{2\sigma^2}\right] (\underline{x}_n - \underline{x}_m). \quad (2.1)$$

We will see in the next section that the estimation of  $\sigma$  depends on the variance of the data along the direction  $\underline{w}$ . If we use Silverman's rule of thumb for example [43],  $\sigma$  would be directly proportional to the standard deviation of the data along  $\underline{w}$ , which in turn scales like each  $(y_n - y_m)$ . In equation (2.1) we can see that there is a  $\sigma^{-1}$  term outside the exponential that is not 'cancelled out' by any  $(y_n - y_m)$  term or  $(\underline{x}_n - \underline{x}_m)$  term. This means that each  $\nabla_{\underline{w}} g_{nm}$  term, and therefore  $\nabla_{\underline{w}} I_Q(Y; C)$ , scales inversely to the spread of the data along  $\underline{w}$ . Since each gradient step depends on  $\nabla_{\underline{w}} I_Q(Y; C)$ , we observe the following.

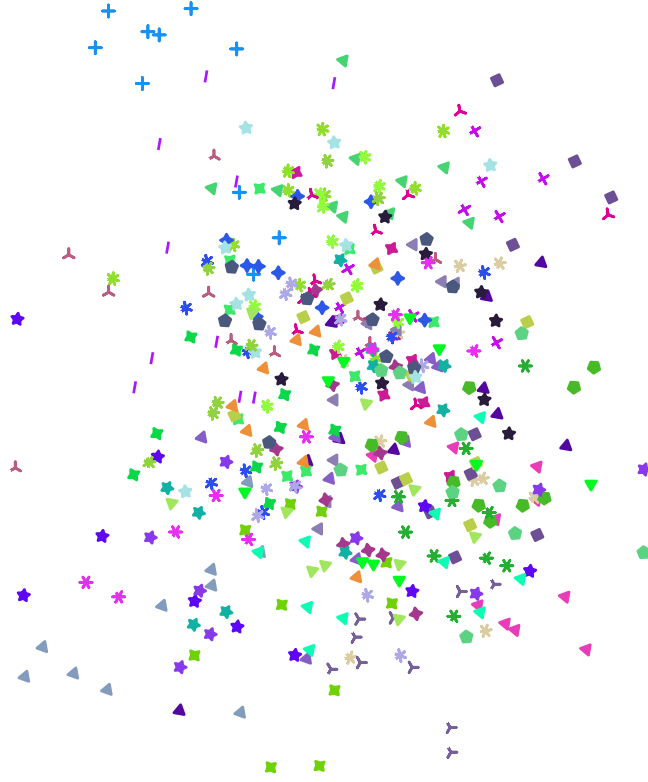


Figure 2.4.: 2D projection of the iterative QMI algorithm where the data (AT&T Dataset of Faces) is not whitened as a pre-processing step. The initial features of the algorithm are set to random vectors.

- When  $\underline{w}$  is a direction along which the data has large variance,  $\nabla_{\underline{w}} I_Q(Y; C)$  will be small and so the convergence will be slow.
- Conversely, when  $\underline{w}$  is a direction along which the data has small variance,  $\nabla_{\underline{w}} I_Q(Y; C)$  can be very large and convergence may be erratic.

If a dataset is whitened, then the variance of the data along any  $\underline{w}$  will be constant, and so  $\nabla_{\underline{w}} I_Q(Y; C)$  will not depend directly on the variance along  $\underline{w}$  but only on the value of  $I_Q(Y; C)$ , as it ought to since it is only  $I_Q(Y; C)$  that we are trying to maximise. On the other hand if the data is not whitened, then  $\nabla_{\underline{w}} I_Q(Y; C)$  will depend on the variance along  $\underline{w}$ , which for the reasons listed above, may make the algorithm unstable or unpredictable. Furthermore, different datasets will in general have very different variances along any  $\underline{w}$ , and whitening will ensure that any pre-processed dataset will have variance 1 along any  $\underline{w}$ , so we see that whitening is also a way of making the convergence of the algorithm less data-dependent.

Note that in contrast, whitening makes little difference to the behaviour of LDA for high-dimensional datasets because LDA does not depend on any kernel width in the way that QMI depends on  $\sigma$ .

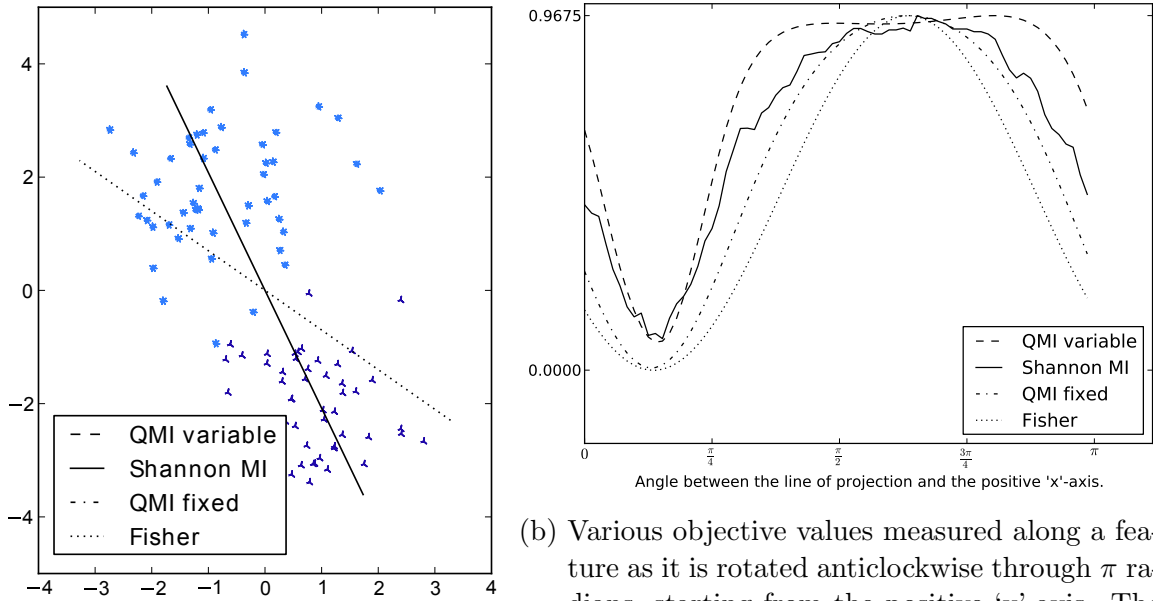
### 2.1.2. Bandwidth selection

Bandwidth selection refers to the choice of the kernel width which we denoted by  $h$  in equation (1.25). Note that in our notation,  $\sigma$  is directly related to the kernel width by  $\sigma = h\sqrt{2}$ . A

principled method for determining the kernel width is Silverman’s rule of thumb [43], according to which the kernel width is estimated by

$$h = \left( \frac{4}{3N} \right)^{\frac{1}{5}} \hat{s}, \quad (2.2)$$

where  $\hat{s}$  is the sample standard deviation. By this, the kernel width will be constant throughout the gradient ascent iterations if the data is whitened, and variable (not constant) in general if the data is not whitened. In this way we see that bandwidth selection is related to whitening. Torkkola proposed a different way of setting the kernel width [50, p.1426], according to which the kernel width is set to approximately half of the distance between the two furthest data points at the beginning of the iterations, and gradually decreasing to about half of the average distances between all pairs of data points towards the end of the iterations. This is a very heuristic approach but seems to work as claimed. In this approach, the kernel width is variable whether the data is whitened or not. We can alternatively take the slightly simpler approach, one in which the kernel width is always set to half of the average pairwise distance. The kernel width would still be variable in this approach, whether the data is whitened or not.



(a) Example 2-class data. Solid line: optimal feature with respect to LDA, Bayes error and QMI with fixed bandwidth. Dotted line: optimal feature with respect to QMI with variable bandwidth (Silverman).

(b) Various objective values measured along a feature as it is rotated anticlockwise through  $\pi$  radians, starting from the positive 'x'-axis. The maxima of Fisher’s discriminant (LDA) and QMI with fixed bandwidth agree, and both agree approximately with the estimate of Shannon’s MI. However, QMI with variable bandwidth has two local maxima at the ‘wrong’ places.

Figure 2.5.: Illustration of the behaviour of QMI with variable and fixed bandwidths.

Let us study the behaviour of the gradient ascent algorithm with respect to variable and fixed bandwidths. First consider the example artificial dataset in figure 2.5a. This dataset is generated from two homoscedastic Gaussians with the identity covariance matrix centred at  $(-1, 2)$  and  $(1, -2)$ . Since the classes are homoscedastic Gaussian, LDA is guaranteed to

produce the optimal feature with respect to Bayes error. In figure 2.5a, the solid black line is the LDA-optimal feature, which is (almost) the same as the optimal feature with respect to QMI with fixed bandwidth. The dotted line marks the feature obtained by QMI with variable bandwidth which is clearly suboptimal, and in figure 2.5b we can see why. For comparison, the values of Shannon’s MI along the features are also included, these are estimated using discretisation.

Regarding the dichotomy of using Torkkola’s heuristic or Silverman’s rule of thumb for estimating the bandwidth, our experiments so far suggest that it makes no qualitative difference in the behaviour of the algorithm with respect to high-dimensional real-world datasets. Figure 2.6 illustrates observation. We can see that it looks qualitatively similar to figure 2.3c. Except for the method used for bandwidth selection, all other parameters of the algorithms used to produced the two figures are identical.

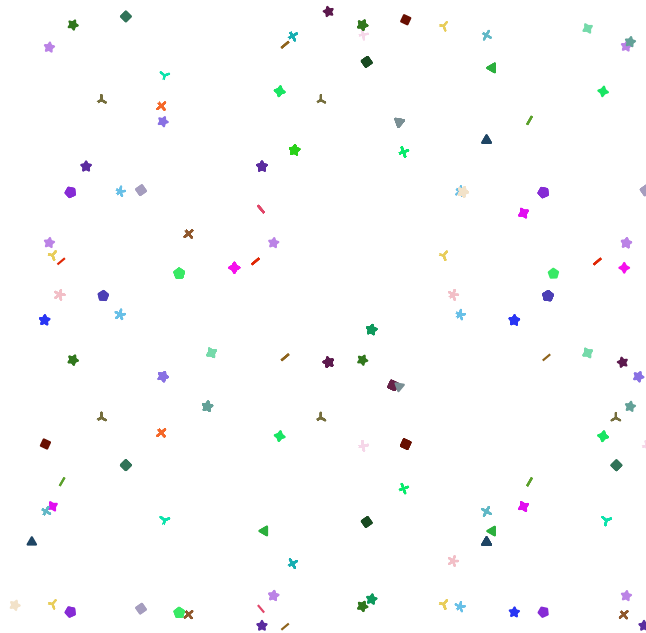


Figure 2.6.: 2D projection of the (pre-whitened) AT&T Dataset of Faces using the iterative QMI algorithm with a simplified version of Torkkola’s heuristic for bandwidth selection, whereby the bandwidth is always set to half of the average pairwise distances of the data.

Based on our observations so far of the effects of whitening and bandwidth selection, perhaps a good recommendation for using gradient ascent for maximising QMI for high-dimensional datasets is to pre-whiten the data and then to use Silverman’s rule of thumb for bandwidth selection. Whitening could make the algorithm less data-dependent and more stable, and Silverman’s rule of thumb will produce a fixed bandwidth for whitened data.

### 2.1.3. Feature initialisation

Before the gradient ascent iterations begin, how should we initialise the features? This is an important question. They could be initialised as random features, or as LDA features, as Torkkola did for some of his experiments [50]. However, there are two noteworthy points about

initialising the features by LDA. First, for some datasets, LDA cannot produce enough features to begin with; the Pima Indian Diabetes dataset for example, available from the UCI database, is a 2-class problem, and so LDA can only extract one single feature. Second, while initialising the features by LDA can improve the performance of gradient ascent, it may arguably mask the effects of the gradient ascent iterations. We will explain this point presently.

The iterative QMI algorithm used to produce figures 2.1 and 2.3 were initialised by random features. Recall that figure 2.1 shows the value of QMI along the features computed by LDA, PCA, QMI maximisation and EMI maximisation respectively. It shows that the iterative QMI algorithm is not effective in extracting features with high QMI for this dataset, compared with LDA and our EMI algorithm. In particular, the QMI algorithm seems to have sought a local maximum. Figure 2.7 shows a plot that is analogous to figure 2.1, but here the QMI algorithm is initialised with LDA features. We see that with LDA features as initial features, the iterative QMI algorithm is able to (slightly but visibly) improve on the quality of the LDA features, as judged by QMI. In particular, using LDA features to initialise the QMI algorithm significantly improves its performance compared with using random features.

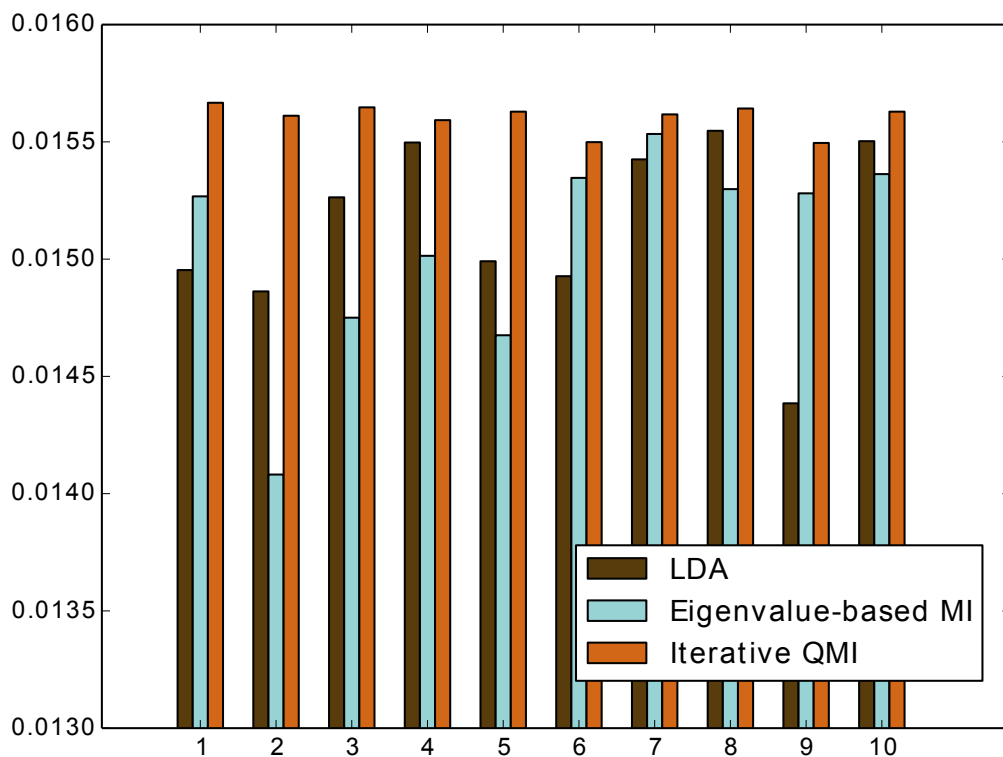


Figure 2.7.: Values of QMI measured along the first 10 features extracted by LDA, EMI and the iterative QMI algorithm respectively, where the initial features for the QMI algorithms are LDA features. The dataset used is the AT&T Dataset of Faces.

With this observation it would be reasonable to say that whenever possible, use LDA to initialise the features, and use random features only when LDA cannot extract enough features (in 2-class or few-class problems). At the same time however, a comparison between figures 2.7 and 2.1 suggests that LDA is *necessary* for the proper execution of the iterative QMI algorithm, and that the algorithm produces improvements on the initial features computed by LDA. Given

that one of the main advantages of information-theoretic feature extraction is that it is robust in situations where LDA would fail (figure 1.3), it would therefore be reasonable to require that an information-theoretic feature extraction algorithm be able to execute properly (without getting stuck in local maxima) without the help of LDA.

The LDA and EMI algorithms used in figure 2.7 are identical to those used in figure 2.1, and their 2D projections are shown in figures 2.3a and 2.3b. The projection of the QMI algorithm initialised by LDA would be similar to that of LDA, but with even tighter compactness of the classes, and we can understand this intuitively by considering figure 2.2 and explanations thereof.

#### 2.1.4. Learning rate and stopping criterion

The learning rate of gradient ascent is denoted  $\eta$  in equation (1.21). It governs the amount by which the feature is updated at each gradient ascent step. If it is too large, the iterations may never converge to the optimum but only swing either side of it. If it is too small, convergence be slow. One intuitively attractive heuristic is to have a large learning rate in the beginning of the iterations, and gradually decrease it toward the end, as recommended in [50]. However, in our experiments we have not found this method of varying the learning rate to be effective. In general, there is not yet a principled procedure to determine the learning rate, and likewise, the stopping criterion.

We could choose the stopping criterion to be when  $\|\nabla_{\underline{w}} I_Q(Y; C)\|$  falls below a certain threshold, which means that the update becomes sufficiently small. This makes intuitive sense if we hope that the objective function  $I_Q(Y; C)$  obeys certain smoothness conditions even in high-dimensional spaces. In particular, we hope that the gradient  $\nabla_{\underline{w}} I_Q(Y; C)$  is near zero in a neighbourhood of the optimum. On a related note, we also want to make the learning rate small enough such that the iteration does not jump out of this neighbourhood in one step. However, even if these smoothness conditions are met, it is still very plausible that there be areas where  $I_Q(Y; C)$  is far from optimal and yet  $\|\nabla_{\underline{w}} I_Q(Y; C)\|$  is small. This would cause the iterations to stop prematurely. A simpler heuristic for choosing the stopping criterion is to stop when  $I_Q(Y; C)$  starts decreasing (as opposed to continuing to increase), hoping that this is an indication that we have reached and have slightly overtaken the maximum. Again, this depends heavily on the smoothness of the objective function and the learning rate, and could suffer from local maxima. In our experiments, we have tested a range of learning rates and numbers of iterations, and a set of parameters that seems to do best for the whitened AT&T Dataset of Faces comprises a constant learning rate between 2 to 5, and a number of iterations between 50 and 100. Figures 2.1, 2.3c, 2.4, 2.6, and 2.7 were produced with these parameters. For non-whitened high-dimensional data, our experiments suggest that the performance of the algorithm is somewhat unaffected by what parameters we choose, and is consistently poor. Note however, that these parameters are somewhat data dependent, and that our experiments have shown that the behaviour of the iterative QMI algorithm in high-dimensional datasets and low-dimensional datasets can be very different. The fundamental problem is that

the underlying optimisation is not convex.

In summary, there is considerable variety in the behaviour of QMI maximisation by gradient ascent, depending on the dataset, input dimensionality, bandwidth selection, feature initialisation, learning rate and stopping criterion. A comprehensive evaluation of the behaviour and performance of QMI maximisation by gradient ascent, for all types of datasets and all representative values of free parameters, is not the aim of this thesis. Indeed it would be very time-consuming. Instead, we will now propose a novel method that circumvents the difficulties associated with gradient ascent and iterative algorithms in general.

## 2.2. Eigenvalue-based mutual information

In section 2.1 we have studied some properties of QMI maximisation by gradient ascent, and have seen how they may lead to various implementational challenges. In this section we will propose a novel technique of information-theoretic feature extraction that can be solved by eigendecomposition. This technique makes use of a new estimate of mutual information that is slightly different from QMI, and we will call it *eigenvalue-based mutual information* (EMI) [23]. (*Spectral mutual information* may also be a suitable name.) In summary, EMI maximisation has the following advantages over QMI maximisation by gradient ascent.

- Its computational complexity is much lower (see table 2.1). Specifically, for high-dimensional datasets it is of the same order as that of PCA.
- EMI maximisation has no free parameters, and its formulation and implementation is not data-dependent.
- Because of its closed-form nature, the behaviour of EMI maximisation is consistent, predictable, and is guaranteed to find global maxima.

We derive EMI from a pairwise interaction view of QMI. Mutual information is often interpreted as the *difference* or *similarity* between two probability distributions: the true joint distribution and the joint distribution under the independence assumption. In the context of estimating mutual information *from data* however, equation (1.33) uncovers an alternative view of mutual information, one in terms of *pairwise interactions*  $g_{nm}$  between each pair  $\{\underline{x}_n, \underline{x}_m\}$  of training data points, weighted by the  $\rho_{nm}$ . This view is especially applicable and intuitive in classification problems. Each pairwise interaction  $g_{nm}$  is monotonically decreasing in the distance  $|\underline{y}_n - \underline{y}_m|$  between two data points along the feature  $\underline{w}$ , as is clear from equation (1.31). So for example if we simply wanted to maximise the sum  $\sum_{n=1}^N \sum_{m=1}^N g_{nm}$  (rather than equation (1.33)), then we will obtain a feature along which the data points are as close to each other as possible, which is obviously not desirable from a classification perspective. However the weights  $\rho_{nm}$  can be negative, in which case the corresponding pairwise distance is maximised. Let us conceive of a simple example in which there are 2 classes ( $c = 2$ ), and each class has 5 training data points ( $N_1 = N_2 = 5$  and  $N = 10$ ). Then the reader may verify from equation

(1.32) that if two data points  $\underline{x}_n$  and  $\underline{x}_m$  are in the same class, then  $\rho_{nm} = \frac{1}{200}$ ; and if they are in different classes, then  $\rho_{nm} = -\frac{1}{200}$ . This means that in the maximisation of  $I_Q(Y; C)$  as in equation (1.33), the within-class distances are minimised while the between-class distances are maximised. On a side-note, we see that in this pairwise interaction view of mutual information maximisation, there is significant resemblance to LDA. However, unlike LDA, QMI has advantageous information-theoretic properties as illustrated by figure 1.3.

Another consequence of this view of QMI is that we can now generalise the pairwise interactions between data points. In QMI with Gaussian kernel density estimation, each pairwise interaction  $g_{nm}$  is a Gaussian function of  $(y_n - y_m)$ . The salient characteristics of  $g_{nm}$  that give  $I_Q(Y; C)$  its information-theoretic properties are as follows. Note that since  $y_n = \underline{w}^T \underline{x}_n$  by definition and that  $\|\underline{w}\| = 1$ , we always have  $0 \leq |y_n - y_m| \leq \|\underline{x}_n - \underline{x}_m\|$ .

- It is symmetric in  $(y_n - y_m)$  and monotonically decreasing in  $|y_n - y_m|$ .
- It reaches its maximum when  $y_n - y_m = 0$ , where the maximum is  $\frac{1}{\sigma\sqrt{2\pi}}$ .
- It reaches its minimum when  $|y_n - y_m| = \|\underline{x}_n - \underline{x}_m\|$ , where the minimum is  $\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\|\underline{x}_n - \underline{x}_m\|^2}{2\sigma^2}\right)$ .

All of these properties can be preserved by using an alternative, *negative-parabolic* (as opposed to Gaussian) pairwise interaction, in the form of  $a - b(y_n - y_m)^2$ . More precisely, define

$$e_{nm} := \frac{1}{\sigma\sqrt{2\pi}} \left(1 - \frac{1 - e^{-\frac{\|\underline{x}_n - \underline{x}_m\|^2}{2\sigma^2}}}{\|\underline{x}_n - \underline{x}_m\|^2} (y_n - y_m)^2\right). \quad (2.3)$$

We see that if we view  $(y_n - y_m)$  as the abscissa, then the graph of  $e_{nm}$  in equation (2.3) is a negative parabola, hence the name *negative parabolic* pairwise interaction. Figure 2.8 illustrates the differences and similarities between  $e_{nm}$  and  $g_{nm}$ .

Now we can measure the mutual information between the data and the class label by using the following, instead of QMI as in equation (1.33).

$$I_E(Y; C) := \sum_{n=1}^N \sum_{m=1}^N \rho_{nm} e_{nm}, \quad (2.4)$$

where  $\rho_{nm}$  is the same as in equation (1.32). For the example dataset in figure 1.3, the black line is the feature along which the maximum value of  $I_E(Y; C)$  is obtained. We call  $I_E(Y; C)$  *eigenvalue-based mutual information* (EMI), for reasons that will become clear shortly. Figure 2.9 demonstrates the similarities between EMI, QMI and Shannon's MI (equation (1.13)).

The real advantage of using EMI instead of QMI is that it can be optimised analytically, obviating the need for any iterative procedure.  $e_{nm}$  can be written as  $\underline{w}^T \underline{e}_{nm} \underline{w}$  where

$$\begin{aligned} \underline{e}_{nm} &:= \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(\underline{x}_n - \underline{x}_m)(\underline{x}_n - \underline{x}_m)^T}{2\sigma^2}\right] \\ &= \frac{1}{\sigma\sqrt{2\pi}} \left[I - \frac{(1 - e^{-\frac{\|\underline{x}_n - \underline{x}_m\|^2}{2\sigma^2}})}{\|\underline{x}_n - \underline{x}_m\|^2} (\underline{x}_n - \underline{x}_m)(\underline{x}_n - \underline{x}_m)^T\right], \end{aligned} \quad (2.5)$$

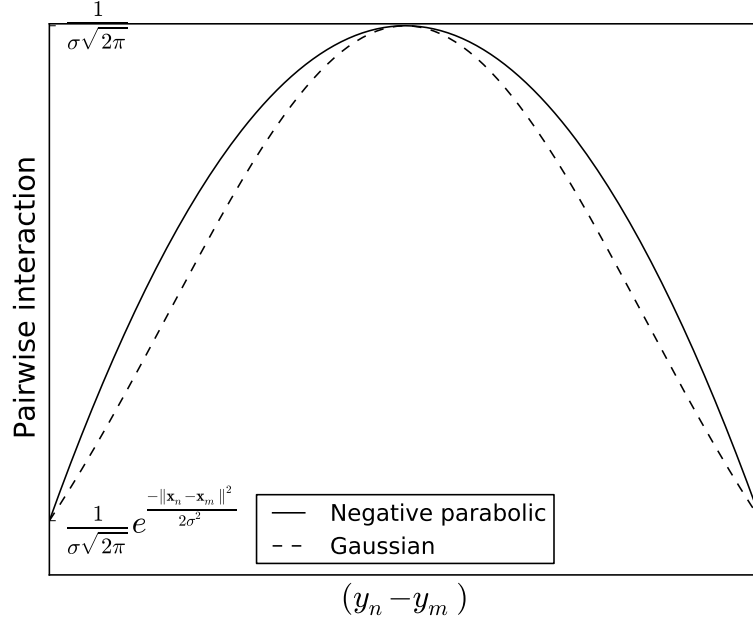


Figure 2.8.: Graphs of  $e_{nm}$  as in equation (2.3), and  $g_{nm}$  as in equation (1.31), where we view  $(y_n - y_m)$  as the abscissa. The two pairwise interactions agree at their maximum (in the middle) and at their minima (two sides).

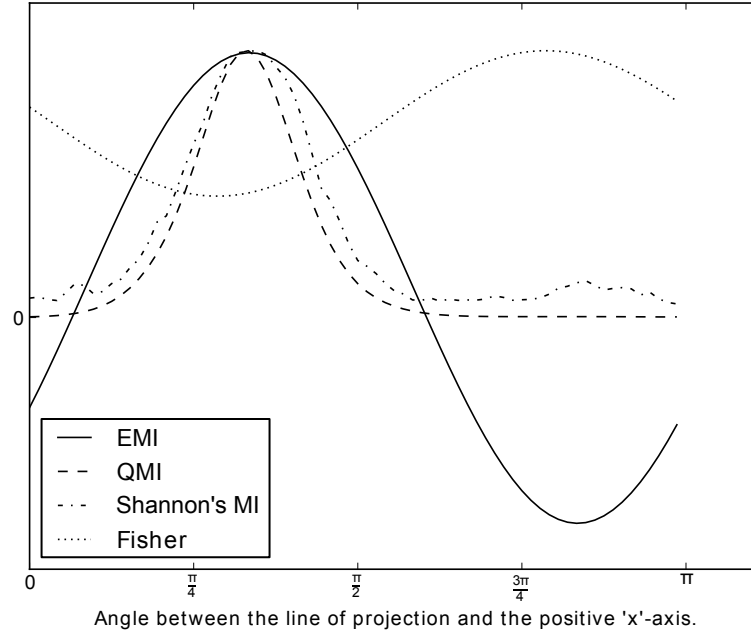


Figure 2.9.: The values of the various measures of mutual information along a feature, as the feature is rotated  $\pi$  radians. The dataset used is the one shown in figure 1.3. Note that some of the measures are rescaled so that all measures are visually comparable, however it is not the actual value of the measure that matters in the context of optimisation, but rather the shape of the graph. We see that EMI, QMI and Shannon's MI peak at almost the same place. For the sake of comparison, Fisher's discriminant (LDA) is also included. We see that Fisher's discriminant measure does not peak at the 'right' place. The Fisher-optimal feature (computed by LDA) is shown as the blue dashed line in figure 1.3.

where the second line follows from evaluating the matrix exponential in the first line using Taylor expansion. Thus, if we define a matrix  $\underline{\underline{e}}$  by

$$\underline{\underline{e}} := \sum_{n=1}^N \sum_{m=1}^N \rho_{nm} \underline{\underline{e}}_{nm}, \quad (2.6)$$

then we see that EMI can be written as  $I_E(Y; C) = \underline{w}^T E \underline{w}$ . Now finding a feature that maximises EMI is equivalent to maximising  $I_E(Y; C)$  in  $\underline{w}$ , and we see that the maximising  $\underline{w}$  are just the largest eigenvectors of the matrix  $\underline{\underline{e}}$ .

The reader may notice some similarities between  $e_{nm}$  and the *Epanechnikov kernel*, which is defined as follows.

$$h_{nm} := \frac{3}{4} (1 - (y_n - y_m)^2) \mathbb{I}[|y_n - y_m| \leq 1]. \quad (2.7)$$

This also has a negative parabolic shape, but there are several fundamental differences between this and our pairwise interaction  $e_{nm}$ . First,  $e_{nm}$  has a variable width that depends on  $\|\underline{x}_n - \underline{x}_m\|$ , while in contrast  $h_{nm}$  does not, and is fixed-width. In particular,  $h_{nm}$  does not take into account any pairs of points for which  $|y_n - y_m| > 1$ . Moreover, the indicator function  $\mathbb{I}[|y_n - y_m| \leq 1]$  cannot be encoded in a matrix in the way that  $e_{nm}$  can be encoded in the matrix  $\underline{\underline{e}}_{nm}$  (equation (2.5)) via  $e_{nm} = \underline{w}^T \underline{\underline{e}}_{nm} \underline{w}$ . It is the ability of a pairwise interaction (or kernel) to be encoded in a matrix that allows the associated dimensionality reduction algorithm to be formulated as an eigenvalue-problem.

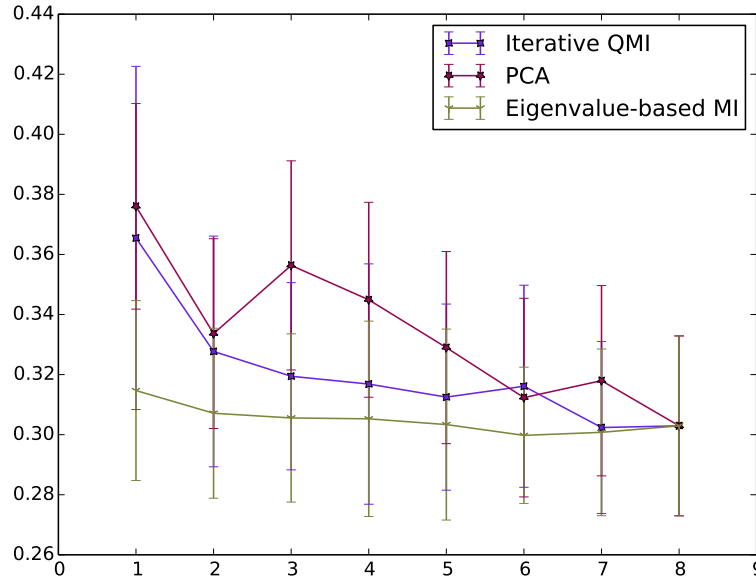


Figure 2.10.: Average classification error rates of the nearest-neighbour classifier through 10 repeats of 5-fold cross-validation, for the subspaces computed by PCA, EMI and the iterative QMI method respectively. The dataset used is the Pima Indians Diabetes Dataset, available from the UCI Machine Learning Repository. LDA was not included in this evaluation because the dataset only has 2 classes and LDA would only be able to extract 1 feature. We see that EMI has lower (average) error rates than PCA and Torkkola’s iterative QMI method.

Figure 2.10 illustrates a practical application of EMI. It shows the average classification error rates of the nearest-neighbour classifier through 10 repeats of 5-fold cross-validation, for the subspaces computed by PCA, the EMI method, and the iterative QMI method respectively, using the Pima Indians Diabetes Dataset available from the UCI machine learning repository. The error bars (not to be confused with classification error rates) on the respective graphs show one sample standard deviation either side of the average classification error rates, averaged over the numbers of repeats and folds. Note that in this experiment there are a total of 50 evaluations (per dimensionality of the feature subspace), since there are 10 repeats and 5 folds per repeat. Throughout this thesis, an error bar will always show one sample standard deviation either side of the average classification error rate.

Figure 2.11 shows the 2D projections computed by the three methods. Note the similarity between the projections computed by the two information-theoretic methods, in contrast to that of PCA. Note also that a 2D projection cannot be computed using LDA since this is a 2-class problem and LDA would only be able to compute one feature. Recall in section 2.1.3 that initial features for the iterative QMI algorithm can be either random or LDA features. This is a case where the initial features cannot be LDA features (except the first one). In this experiment the parameters of the iterative QMI algorithm were set as follows. The data is pre-whitened, the learning rate was set to 1, the bandwidth was estimated using Silverman’s rule of thumb (and therefore constant), and the initial features were random.

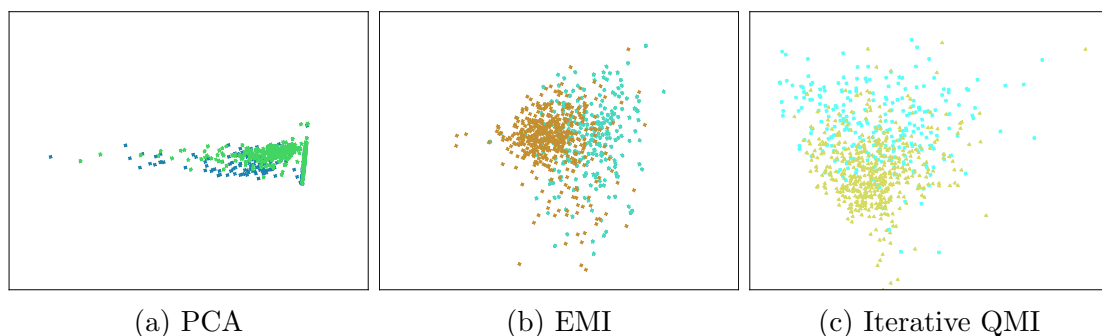


Figure 2.11.: 2D projections of the Pima dataset computed by three feature extraction techniques.

### 2.2.1. Relationship with Laplacian maps

Recall that in section 1.7 we defined a generalisation of PCA which we call *Laplacian maps* (equations (1.7) and (1.8)). Laplacian maps also generalise an unsupervised dimensionality reduction technique called Laplacian Eigenmaps [2]. The objective of the Laplacian Eigenmaps technique is to minimise the distances between nearby data points in the lower-dimensional feature subspace, and it does so by eigendecomposition. For each training data point  $\underline{x}_n$ , only its nearest neighbours are considered. Therefore in the formulation of Laplacian Eigenmaps, most of the weights  $\omega_{nm}$  are zero, and they are non-zero only when  $\underline{x}_n$  and  $\underline{x}_m$  are close neighbours. Consequently, the corresponding Laplacian matrices of Laplacian Eigenmaps (equation (1.10)) are sparse, and so the Laplacian Eigenmaps technique is very computationally efficient.

Laplacian Eigenmaps is a *local* technique as opposed to a *global* technique like PCA. It is worth mentioning that experiments in [27] have shown that local techniques generally have lower performance than global techniques for real-world datasets in terms of classification accuracy. Our experiments also support this observation.

Now we show that Laplacian maps also generalises EMI. From equations (2.3) and (2.4), we can re-write the EMI along a feature  $\underline{w}$  as the following.

$$I_E(Y; C) = \left( \sum_{n=1}^N \sum_{m=1}^N \frac{\rho_{nm}}{\sigma\sqrt{2\pi}} \right) - \left( \sum_{n=1}^N \sum_{m=1}^N \rho_{nm} \frac{(1 - e^{-\frac{\|\underline{x}_n - \underline{x}_m\|^2}{2\sigma^2}})}{\sigma\sqrt{2\pi}\|\underline{x}_n - \underline{x}_m\|^2} (y_n - y_m)^2 \right). \quad (2.8)$$

Note that the first term on the right-hand-side of equation (2.8) does not involve  $\underline{w}$ , and so is irrelevant in the maximisation. If we want a low-dimensional subspace spanned by  $P$  features  $\underline{w}_1, \dots, \underline{w}_P$  that maximises the EMI along each feature, then the quantity we aim to maximise can be written as follows.

$$I_E(\underline{Y}; C) = \left( M \sum_{n=1}^N \sum_{m=1}^N \frac{\rho_{nm}}{\sigma\sqrt{2\pi}} \right) - \left( \sum_{n=1}^N \sum_{m=1}^N \rho_{nm} \frac{(1 - e^{-\frac{\|\underline{x}_n - \underline{x}_m\|^2}{2\sigma^2}})}{\sigma\sqrt{2\pi}\|\underline{x}_n - \underline{x}_m\|^2} \|\underline{y}_n - \underline{y}_m\|^2 \right). \quad (2.9)$$

Now if we define the pairwise weights

$$\omega_{nm} := \rho_{nm} \frac{(1 - e^{-\frac{\|\underline{x}_n - \underline{x}_m\|^2}{2\sigma^2}})}{\sigma\sqrt{2\pi}\|\underline{x}_n - \underline{x}_m\|^2}, \quad (2.10)$$

noting that the first term on the right-hand-side of equation (2.9) does not involve  $\underline{w}_1, \dots, \underline{w}_P$ , we see that maximising EMI as in equation (2.9) is equivalent to minimising equation (1.8), whose solution is given by the smallest eigenvectors of the matrix  $\underline{\underline{x}}^T \underline{\underline{L}} \underline{\underline{x}}$  as shown in equations 1.9. Thus we see that EMI can be seen as a special case of Laplacian maps.

So far we have shown two perspectives of EMI. First, it is a modification of QMI where, instead of using Gaussian pairwise interactions  $g_{nm}$  (equation (1.31)), we use negative parabolic pairwise interactions  $e_{nm}$  (equation (2.5)). Second, it is a special case of Laplacian maps where the eigenvectors of the objective matrix  $\underline{\underline{m}}$  maximise an estimate of mutual information. PCA and Laplacian Eigenmaps are both special cases of Laplacian maps, and both are examples of distance-based dimensionality reduction. Indeed, Laplacian maps were defined from a distance-based perspective, where each pairwise distance  $\|\underline{y}_n - \underline{y}_m\|$  is weighted by  $\omega_{nm}$ . It is interesting to see now that for particular choices of  $\omega_{nm}$ , we can formulate information-theoretic feature extraction from Laplacian maps. In this way, Laplacian maps serve as a connection between distance-based and information-theoretic feature extraction.

### 2.2.2. Computational complexity

The EMI technique requires that the data be whitened, which implies that PCA must be applied prior to EMI maximisation. For now let us assume that  $D \geq N$ . For each data point  $\underline{x}_n$ , define

the *centred* data point  $\hat{x}_n := x_n - \bar{x}$ , and let  $\hat{\underline{x}}$  denote the centred design matrix, whose  $n^{\text{th}}$  row is the  $n^{\text{th}}$  centred data point  $\hat{x}_n^T$ . Then the sample covariance matrix can be written as  $\frac{1}{N}\hat{\underline{x}}^T\hat{\underline{x}}$ , whose construction takes  $\mathcal{O}(ND^2)$  time. However in the case  $D \geq N$ , PCA can be done more efficiently by eigendecomposition of the  $N \times N$  Gram matrix  $\hat{\underline{x}}\hat{\underline{x}}^T$  [3, sec.12.1.4], whose construction takes  $\mathcal{O}(N^2D)$  time, and whose eigendecomposition takes  $\mathcal{O}(N^3)$  time. So in the case  $D \geq N$ , PCA takes  $\mathcal{O}(N^2D + N^3) = \mathcal{O}(N^2D)$  time.

After PCA, the dimensionality of the data is reduced from  $D$  to at most  $N - 1$ . Given that the  $N_c$  and  $N$  are pre-computed, we see in equation (1.32) that each  $\rho_{nm}$  takes  $\mathcal{O}(1)$  time to compute. Since each  $\|x_n - x_m\|^2$  takes  $\mathcal{O}(N - 1)$  time to compute (after PCA has been applied), we see from equation (2.10) that each  $\omega_{nm}$  takes  $\mathcal{O}(N)$  time to compute, thus the  $N \times N$  weight matrix  $\underline{\omega}$  takes  $\mathcal{O}(N^3)$  time to construct. From equation (1.10) we see that the Laplacian matrix  $\underline{l}$  also takes  $\mathcal{O}(N^3)$  time to construct. The objective matrix of a Laplacian map is  $\underline{x}^T \underline{l} \underline{x}$ , whose construction and eigendecomposition both take time  $\mathcal{O}(N^3)$ . So any Laplacian map takes  $\mathcal{O}(N^3)$  to compute, after the application of PCA. Since EMI maximisation is a Laplacian map, it also takes  $\mathcal{O}(N^3)$  time after PCA.

Therefore, in the case  $D \geq N$ , the computational complexity of EMI maximisation (including the PCA step) is  $\mathcal{O}(N^2D + N^3) = \mathcal{O}(N^2D)$ , which is the same as that of PCA. In the case  $D < N$  however, a similar analysis shows that PCA takes  $\mathcal{O}(ND^2)$  time while EMI maximisation still takes  $\mathcal{O}(N^2D)$  time, which is higher by a factor of  $\frac{N}{D}$ . Note however that often, practical problems that require dimensionality reduction will have high input dimensionality  $D$  and typically low sample size  $N$ , such as image recognition or gene expression microarray data. Note also that the above analysis is theoretical. An actual implementation of EMI maximisation can be optimised to drive the running time much lower than to simply compute and eigendecompose  $\underline{x}^T \underline{l} \underline{x}$ .

For comparison let us calculate the computational complexity of QMI maximisation by gradient ascent. First let us suppose that PCA is not pre-applied to the data. Suppose we wish to extract  $P$  features (typically  $1 < P < 100$ ), and to use  $R$  iterations of gradient ascent per feature (typically  $50 \leq R \leq 200$ ). At each gradient ascent step,  $\nabla_{\underline{w}} I_Q(Y; C) = \sum_{n=1}^N \sum_{m=1}^N \rho_{nm} \nabla_{\underline{w}} g_{nm}$  needs to be computed. By equation (2.1), each  $\nabla_{\underline{w}} g_{nm}$  takes  $\mathcal{O}(D)$  time. Thus  $\nabla_{\underline{w}} I_Q(Y; C)$  takes  $\mathcal{O}(N^2D)$  time, the same as PCA or EMI maximisation. This is one gradient ascent iteration, and there will be  $PR$  such iterations in total, so the total computational complexity of QMI maximisation by gradient ascent is  $\mathcal{O}(PRN^2D)$ . If we required that the data be pre-whitened or pre-processed by PCA so that the dimensionality is reduced from  $D$  to  $N - 1$ , then the complexity will be  $\mathcal{O}(N^2D + PRN^3)$  if  $D \geq N$ , or  $\mathcal{O}(PRN^2D)$  if  $D < N$ .

## 2.3. Summary

In this chapter we have studied QMI maximisation by gradient ascent [50] in some detail. We have seen how the behaviour of the algorithm is affected by whitening, bandwidth selection, feature initialisation, learning rate and stopping criterion. The optimal values of the free pa-

rameters depend on the dataset, and consequently the formulation of QMI maximisation by gradient ascent is data-dependent. Because of the variety in the behaviour of the algorithm and its relatively high computational complexity, we have not carried out a full experimental investigation of the algorithm’s behaviour with respect to its parameters. Instead, we have proposed a new information-theoretic feature extraction technique that circumvents the implementational challenges facing QMI maximisation by gradient ascent.

The proposed technique uses EMI as an estimate of mutual information, which enables the feature extraction problem to be solved by eigendecomposition. This has several advantages. There are no free parameters; the computational complexity is of the same order as PCA, and much lower than that of QMI maximisation by gradient ascent; the formulation of EMI is not data-dependent; its behaviour is stable and predictable; and it is always guaranteed to find the global optimum.

### 3. Over-fitting and distance-based feature extraction

In this chapter we study the limitations of information-theoretic feature extraction when applied to high-dimensional datasets. From the perspective of over-fitting and the curse of dimensionality (section 1.3), we study why LDA and EMI under-perform on these datasets compared to PCA, even though LDA and EMI clearly give better class-discrimination. According to these insights, we prescribe a simple but effective technique to solve the over-fitting problem and achieve consistently better classification accuracy than PCA.

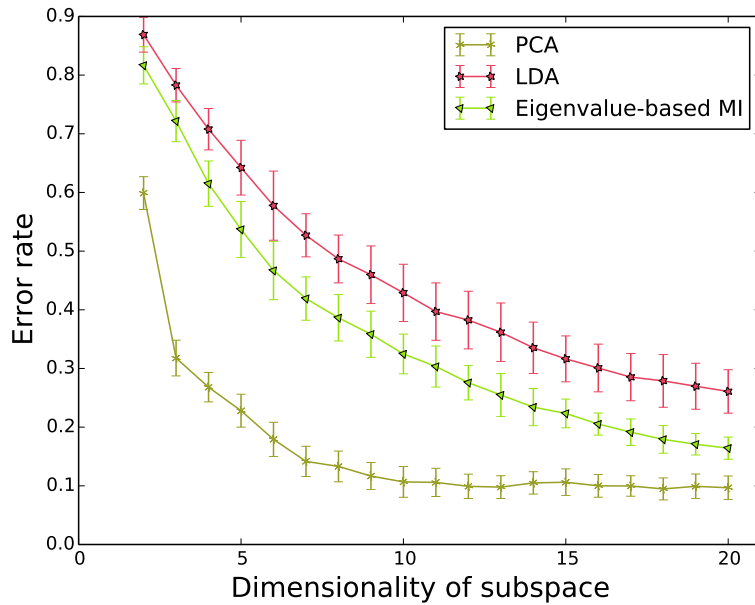


Figure 3.1.: Average classification error rates of the nearest-neighbour classifier through 12 repeats of 2-fold cross-validation, for the subspaces computed by PCA, LDA, and EMI respectively, on the AT&T Dataset of Faces. The data is pre-whitened. PCA achieves better classification accuracy even though it gives lower class-discrimination than LDA and EMI.

Figure 2.10 shows a dataset on which information-theoretic feature extraction out-performs PCA. From figure 2.11 we can see why: the information-theoretic projections clearly show better class-discrimination than the PCA projection. However, this intuitively attractive idea that the better the class-discrimination, the better the quality of feature extraction, seems inconsistent in figure 3.1. (Recall that throughout this thesis, an error bar on a graph shows one sample standard deviation either side of the average classification error rate, averaged over the numbers of repeats and folds.) Figure 3.1 shows that PCA achieves the highest classification

accuracy, while figure 2.3 clearly shows that the PCA achieves lower class-discrimination. In short, this is caused by over-fitting, as shown in figure 1.2. Over-fitting typically occurs in high-dimensional datasets, such as the AT&T Dataset of Faces or any image dataset. More experimental results similar to figure 3.1 are presented in appendix A.

### 3.1. Limitations of information-theoretic feature extraction

Figure 1.3 and extrapolations thereof constitute a common argument in favour of using information theory in feature extraction. One of the main benefits of taking the information-theoretic approach is that it is robust against non-Gaussian, and specifically multimodal class-conditional distributions, whereas LDA is not. For high-dimensional datasets however, we will now show that this advantage becomes less relevant.

Recall in figure 2.3a that LDA is able to compute feature subspaces that exhibit good class-discrimination for the AT&T Dataset of Faces. Since we know that LDA is optimal for datasets with homoscedastic Gaussian class-conditional distributions, we might suppose that the classes in this dataset have Gaussian distributions in the input space, and in particular that they are unimodal. For the sake of illustration let us re-group the 40 classes in the dataset into 10 classes, by mixing 4 classes into 1 larger superclass. The resulting dataset then clearly has multimodal class-conditional sample distributions. Indeed, figure 2.3a confirms that each superclass has at least 4 modes since it is composed of 4 original classes, all of which are well-separated from each other. By the intuition in figure 1.3, we might think that LDA might fail for this dataset and that an information-theoretic approach might give better class-discrimination. However, figure 3.2 shows us that this is not true.

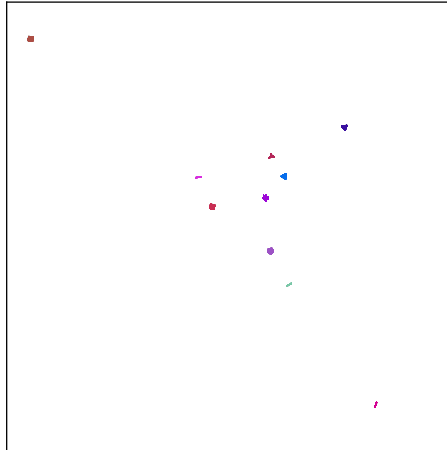


Figure 3.2.: 2D projection of the AT&T Dataset of Faces where 10 superclasses are formed from the 40 original classes by regarding 4 as 1. Each superclass then clearly has a multimodal distribution in the input space, and yet LDA is still able to find extremely good class-discrimination, in that each of the 10 superclasses are extremely compact and well separated from others.

The same result is obtained when we re-group the classes into different numbers of superclasses. This phenomenon may be explained in terms of the high dimensionality of the dataset

used, and we begin with three toy examples.

First let us consider an unsupervised, 3D dataset that is composed of two clearly separated spherical Gaussian clusters, say centred at coordinates  $(-2, -2, 0)$  and  $(2, 2, 0)$  respectively, each with the identity covariance matrix. Consider 2D projections of this dataset. If we choose the  $xy$ -plane as our projection, then the data would appear bimodal, with two spherical Gaussians centred at  $(-2, -2)$  and  $(2, 2)$  respectively. However, there is a 2D projection in which the data appears as a spherical *unimodal* Gaussian, namely the plane perpendicular to the vector  $(1, 1, 0)^T$ .

Next, consider a slightly modified dataset that contains an additional Gaussian cluster centred at coordinates  $(2, -2, 0)$ . Now there is no 2D projection in which this trimodal dataset appears unimodal, since there is no straight line in the 3D input space that could intersect the centres of all 3 Gaussian clusters.

Finally, consider the same dataset but with an additional redundant dimension. That is, a 4D dataset with 3 distinct spherical Gaussian clusters, centred at coordinates  $(2, 2, 0, 0)$ ,  $(-2, -2, 0, 0)$ , and  $(2, -2, 0, 0)$  respectively. Then we can easily check that the plane spanned by the vectors  $(0, 0, 1, 0)^T$  and  $(0, 0, 0, 1)^T$  is a 2D projection in which the dataset appears as a unimodal Gaussian.

In general, suppose we have an unsupervised dataset with  $D$  input dimensions and  $L$  distinct clusters whose *centres* (modes, means or medians) are at  $\underline{\mu}_1, \dots, \underline{\mu}_L$ . Suppose we aim to find a  $P$ -dimensional subspace in which the dataset appears to have one centre. Then we want to solve

$$\underline{\mu}_1^T \underline{w} = \underline{\mu}_2^T \underline{w} = \dots = \underline{\mu}_L^T \underline{w} \quad (3.1)$$

in  $\underline{w}$ . The degrees of freedom (DoF) of  $\underline{w}$  permitted by equation (3.1) will then be the maximum dimensionality of a subspace in which the dataset (which we assume to be *multicentral* in the input space with  $L > 1$  centres) appears *unicentral*. Any subspace with dimensionality higher than that will necessarily embed a multicentral dataset, since in such a subspace, equation (3.1) will not hold in general. On the other hand, we can always find a subspace with dimensionality at most the DoF of  $\underline{w}$  in equation (3.1) in which the dataset appears unicentral, since we can always choose the subspace to only contain those  $\underline{w}$  that satisfy equation (3.1). Note that the DoF of  $\underline{w}$  permitted by equation (3.1) is  $D - (L - 1)$ . Thus we have the following observation.

**Lemma 1.** *Suppose a  $D$ -dimensional dataset has  $L$  distinct clusters. There exists a  $P$ -dimensional projection such that the centres (means, medians or modes) of the clusters coincide if and only if  $P \leq D - L + 1$ .*

We can generalise this for supervised datasets. Suppose each class  $c$  ( $1 \leq c \leq K$ ) has  $L_c$  clusters, centred at  $\underline{\mu}_1^{(c)}, \dots, \underline{\mu}_{L_c}^{(c)}$ . Suppose we aim to find a  $P$ -dimensional subspace in which the projection of the dataset is such that each class appears to have one single cluster. Then we have  $K$  simultaneous sets of equations to solve in  $\underline{w}$ .

$$\underline{w}^T \underline{\mu}_1^{(c)} = \underline{w}^T \underline{\mu}_2^{(c)} = \dots = \underline{w}^T \underline{\mu}_{L_c}^{(c)}, \quad 1 \leq c \leq K. \quad (3.2)$$

For each class  $c$ , equation (3.2) has  $L_c - 1$  *degrees of restriction*. So the total number of degrees of restriction are  $\sum_{c=1}^K (L_c - 1) = L - K$ , where we let  $L := \sum_{c=1}^K L_c$ , the total number of clusters. Therefore the total number of DoF for  $\underline{w}$  is  $D - (L - K)$ . The following observation summarises this.

**Lemma 2.** *Suppose a  $D$ -dimensional supervised dataset has  $K$  classes and  $L$  separate clusters, where  $L \geq K$ . There exists a  $P$ -dimensional projection of the dataset such that the centres (means, medians or modes) of the clusters in each class coincide, if and only if  $P \leq D - L + K$ .*

A trivial case of lemma 2 is when each class is already a single cluster in the input space, that is,  $L = K$ . Then lemma 2 says that a subspace has unicentral classes if and only if the subspace has dimensionality at most  $D - L + K = D$ , obviously. A more interesting case of lemma 2 is when we regard each data point  $\underline{x}_n$  as its own singleton cluster, that is,  $L = N$ . Then lemma 2 says that we can find a subspace in which each class of the data appears as a singleton data point, if and only if the dimensionality of the subspace is at most  $D - N + K$ . When this happens, we say that the subspace is a *perfect fit*, or that it is *perfectly fitting* for the training dataset. For the AT&T Dataset of Faces, which has  $D = 10,304$ ,  $K = 40$ , and  $N = 400$ , we see that there is a perfectly fitting subspace as long as its dimensionality is at most  $10,304 - 400 + 40 = 9,944$ . This applies analogously to all high-dimensional datasets with a relatively low sample size, and datasets in image recognition and gene expression microarrays are typical examples. It is safe to say that in feature extraction, we do not want to extract 9,944 features from an input dimensionality of 10,304. Typically we want to extract between 2 and 40 features. Even if we pre-process the data using PCA, the dimensionality of a perfectly fitting feature subspace can still be up to  $(N - 1) - N + K = K - 1$ , which in the case of the AT&T Dataset of Faces, is 39. So in a typical feature extraction task, we can always extract perfectly fitting subspaces, or at least subspaces in which each class is tightly packed as one unimodal cluster. Now given that LDA is designed to minimise the average within-class covariance and maximise the between-class covariance, it is not surprising that figures 2.3a and 3.2 are obtained using the AT&T Dataset of Faces. Note that LDA does not necessarily produce perfectly fitting feature subspaces, but very well-fitting ones.

Another implication of lemma 2 is that claims pertaining to explaining the superior classification performance of a feature extraction algorithm in terms of its superior class-discrimination [34], should be taken with care. For low-dimensional data such as the Pima Indians Diabetes dataset, we have seen in figures 2.10 and 2.11 that superior class-discrimination does indeed lead to superior classification performance. However, for high-dimensional data such as the AT&T Dataset of Faces, we have seen that we can always find perfectly fitting subspaces, and that these are prone to over-fitting (figure 1.2) and will give low classification accuracy (figure 3.1). Therefore, when over-fitting is shown to occur, and when a feature extraction algorithm is shown to give superior classification accuracy, this superiority cannot be explained purely by its superior discriminative power compared with another feature extraction algorithm — there must be another explanation. An example of such a claim can be found in [34], where a feature

extraction algorithm called Infomargin is introduced. The authors have shown that Infomargin can give better class-discrimination than Torkkola’s QMI maximisation, and claim that because of this, Infomargin gives superior classification accuracy on several datasets, including the AT&T Dataset of Faces.

Now we can explain why the advantages of mutual information over LDA becomes less relevant in high-dimensional spaces. Recall equation (1.26) and the property of QMI shown in figure 2.2, in particular that QMI maximisation leads to minimising class-overlap, or equivalently maximising class-discrimination. Consider the two kinds of class-discrimination shown in figure 3.3 (figure 3.3a is in fact identical to figure 2.2a). To ensure a fair comparison, both diagrams are constructed such that the overall variances are equal, and that each Gaussian cluster in the bimodal class is half as wide as the unimodal class. Both figures 3.3a and 3.3b show clear discrimination between the two classes. However, in figure 3.3b where one of the classes is bimodal, the overlap between the two classes is slightly larger. Since QMI maximisation is equivalent to minimising overlap, we see that QMI prefers unimodal class discrimination, given that the overall variance remains unchanged. Indeed, this observation is confirmed by figure 2.3 where QMI maximisation by gradient ascent was shown to compute projections in which the classes are multimodal, and figure 2.1 where QMI maximisation by gradient ascent was shown to achieve lower QMI than LDA and EMI maximisation, which compute projections with unimodal classes.

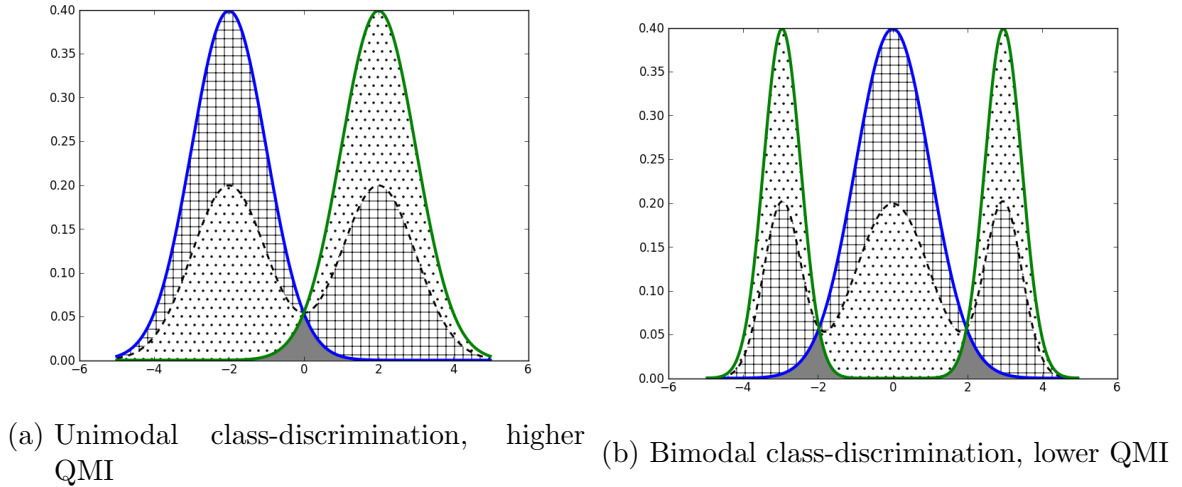


Figure 3.3.: The solid lines are the class-conditional distributions for the 2 classes, and the dashed line is the overall distribution  $p_Y$ . (b) shows the PDFs of two classes, a unimodal Gaussian class and a bimodal class whose modes are on either sides of the unimodal class. The checkered and the dotted areas correspond to the value of  $|p_{Y|C}(y|c) - p_Y(y)|$  for each of the 2 classes respectively, and the squares of only these areas contribute to the value of QMI, not the grey-shaded area. The grey-shaded area is the overlap between the two class-conditional distributions. The variance of the overall distribution in (a) and (b) are equal. We see that (a) has smaller overlap than (b).

The observation that QMI prefers unimodal class discrimination applies analogously to Shannon’s MI and EMI. This, together with lemma 2, means that the advantage of mutual informa-

tion over LDA, that it is robust against multimodal class distributions (figure 1.3), is irrelevant for high dimensional datasets. An intuitive way of thinking about this is that we can always find low-dimensional projections with unimodal classes, and MI prefers unimodal classes anyway. Furthermore, recall figure 2.7, where we have shown that if we use LDA to initialise the features for QMI maximisation by gradient ascent, QMI maximisation was able to increase the values of QMI along the features. This validates the efficacy of QMI maximisation when LDA features are used as initial features for gradient ascent. Geometrically however, the already tightly packed classes in the LDA feature subspace (figure 2.3a) are made even more compact through QMI maximisation, thus exacerbating the effects of over-fitting.

Based on our observations so far in this chapter and our understanding of over-fitting in high-dimensional datasets in section 1.3, we recommend that in the context of feature extraction for classification of high-dimensional data, future research with the aim of improving classification accuracy should focus more on alleviating the effects of over-fitting, rather than simply maximising mutual information or other kinds of discrimination.

With figure 3.1 in mind, the following questions still remain. Why does PCA still have the best classification performance for high-dimensional datasets, even though it is an unsupervised dimensionality reduction technique? Given that the advantages offered by MI in high-dimensional datasets are limited, can we still find techniques that consistently give better classification performance than PCA? The rest of this chapter addresses these questions, but first we need a convenient measure of discrimination.

## 3.2. A measure of class-discrimination

We use a measure of class-discrimination closely related to LDA. We will call the following the  $\Delta$ -measure of a supervised dataset  $\{\underline{y}_n, c_n\}_{n=1}^N$  in a feature space.

$$\Delta(\{\underline{y}_n, c_n\}_{n=1}^N) := \sqrt{\frac{\sum_{c=1}^K N_c \|\underline{\bar{y}}_c - \underline{\bar{y}}\|^2}{\sum_{c=1}^K \sum_{n \in I_c} \|\underline{y}_n - \underline{\bar{y}}_c\|^2}}. \quad (3.3)$$

This is the square root of the ratio between the average between-class variance and the average within-class variance.

Note that in a two-class problem where the aim is to extract one feature, what is often known as the *Fisher criterion* [3, p.188] between two classes is in fact  $\Delta^2$  in our notation in equation (3.3). For a multi-class problem, the Fisher criterion can be defined in a number of ways [13, p.446]. One of them is  $\text{tr}[(\underline{w}^T \underline{s}_w \underline{w})^{-1} (\underline{w}^T \underline{s}_b \underline{w})]$  and another is  $\text{tr}(\underline{w}^T \underline{s}_b \underline{w}) / \text{tr}(\underline{w}^T \underline{s}_w \underline{w})$ , where the between-class and within-class covariance matrices  $\underline{s}_b$  and  $\underline{s}_w$  are defined in equation (1.11). The former is the criterion used by LDA, and the latter is equal to  $\Delta^2$ . We call equation 3.3 the  $\Delta$ -measure and not the Fisher measure, in order to avoid any confusions.

The value of the  $\Delta$ -measure is high when the average within-class variance is small compared to the average between-class variance. It's an intuitively evident measure of class-

discrimination, but only under the condition that the classes are unimodal, since multimodal classes can have relatively large within-class variances but still be well-distinguished from each other. Indeed, figure 3.3b shows a class with a bimodal distribution, which consequently has relatively large within-class variance, but is nevertheless well-distinguished from the other unimodal class. However, lemma 2 implies that the condition of unimodal classes can always be satisfied for high-dimensional datasets. This justifies the use of the  $\Delta$ -measure as a measure of class-discrimination for the high-dimensional datasets that we study in this thesis.

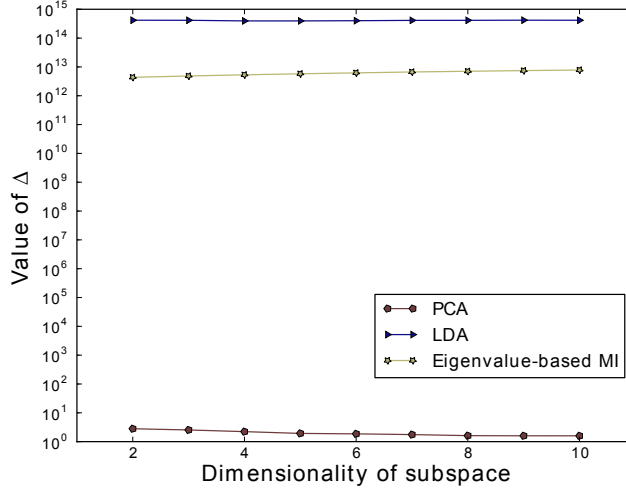


Figure 3.4.: Values of  $\Delta(\{\underline{y}_n, c_n\}_{n=1}^N)$  in the subspaces computed by PCA, LDA, and EMI respectively, using the AT&T Dataset of Faces. The data is pre-whitened. Note the log scale on the vertical axis. Note also that the horizontal axis indicates the dimensionality of the projected data points  $\underline{y}_n$ , and not simply the index of the feature along which  $\Delta(\{\underline{y}_n, c_n\}_{n=1}^N)$  was computed.

Recall figure 3.1, which shows the classification error rates of PCA, LDA, and EMI on the AT&T Dataset of Faces. Figure 3.4 shows the  $\Delta$ -measure in the subspaces computed respectively. Unsurprisingly, LDA achieves the highest values of  $\Delta$  by design. PCA scores the lowest values of  $\Delta$ , which is again unsurprising by looking at its 2D projection in figure 2.3d. The key point however is the following. The order of LDA, EMI and PCA with respect to the  $\Delta$ -measure, which is a measure of class-discrimination, is the reverse of their order with respect to classification accuracy, as shown in figure 3.1. From this it is tempting to make the counter-intuitive suggestion that the lower the discriminative power of a feature extraction algorithm, the higher the classification accuracy. But at the extreme low end of class-discrimination, where the classes are mixed together and indistinguishable from each other, this suggestion is clearly not true. Indeed, figure 2.10 shows that EMI achieves better classification accuracy than PCA and figure 2.11 shows that this is because EMI has better class-discrimination. The implications of our observations so far can be summarised as follows.

For a dataset with a high input dimensionality and small sample size, extremely high levels of class-discrimination will cause over-fitting and consequently the classification accuracy will be low. On the other hand, very low levels of class-discrimination will cause overlap between classes and again the classification accuracy will be low. The optimal level of class-discrimination, with

respect to *generalisation performance* (classification accuracy of new data points), lies somewhere between overlap and over-fitting. PCA often exhibits better classification performance than LDA and EMI for high-dimensional face datasets because the PCA subspace does not have extremely high values of the  $\Delta$ -measure and so does not suffer from over-fitting, but nonetheless often happens to preserve some of the discriminative information in the data too.

Based on this, we now want to focus on finding ways in which class-discrimination in the feature subspace can be controlled, to an extent.

### 3.3. Soft discrimination

We propose a family of feature extractors, parameterised by a single scalar parameter  $\beta$ . This technique strikes a balance between discrimination and overlap between classes, and consequently circumvents the curse of dimensionality and the problem of over-fitting. We simply consider the eigendecomposition of the following objective matrix.

$$\underline{m}_\beta := \underline{s}_b - \beta \underline{s}_w, \quad \beta \in \mathbb{R}. \quad (3.4)$$

The parameter  $\beta$  controls the level of class-discrimination. Large  $\beta$  means small within-class variance and large between-class variance. Recall that LDA is the ratio between the between-class variance and the within-class variance, and notice that (3.4) is the weighted difference between them. We have seen that LDA (and EMI) can suffer from over-fitting in high-dimensional spaces, and that this is because the level of class-discrimination is too high in the sense that the classes are too tightly packed (figure 2.3a). In a way, the class-discrimination in an LDA feature subspace is too “harsh”. Eigenvectors of (3.4) span a subspace in which the class-discrimination is *softer*, as shown in figure 3.5. Hence, we call these eigenvectors *soft discriminants*, and the technique *soft discriminant maps* (SDM).

Figure 3.5 shows the 2D projections computed by SDMs on the AT&T Dataset of Faces, for various values of  $\beta$ . For easy comparison, PCA and LDA are also included. We see how the value of  $\beta$  affects the level of class-discrimination. At this resolution, for  $\beta = 10,000$  and  $\beta = 10,000,000$ , the projections computed by the corresponding SDMs look like perfectly fitting projections, as does the LDA projection. However, figure 3.6 reveals their levels of discrimination. It shows the values of the  $\Delta$ -measure (equation (3.3)) in their respective feature subspaces. Figure 3.7 shows their classification error rates. We can see that SDMs give better classification performance than PCA, and far better than LDA. The key point here is that the reason for SDM’s superior performance over PCA is different from that over LDA. SDM has better classification performance than PCA because it gives better class-discrimination, as can be seen in figure 3.5. On the other hand, it is better than LDA because the class-discrimination in an SDM feature subspace is not too harsh, and so SDMs (for appropriate values of  $\beta$ ) do not suffer as much from over-fitting. Figure 3.8 shows the level of over-fitting in a 2D projection computed by SDM. Compared with figure 1.2b, we can see that the testing data points are relatively close to the corresponding training data points in the same class.

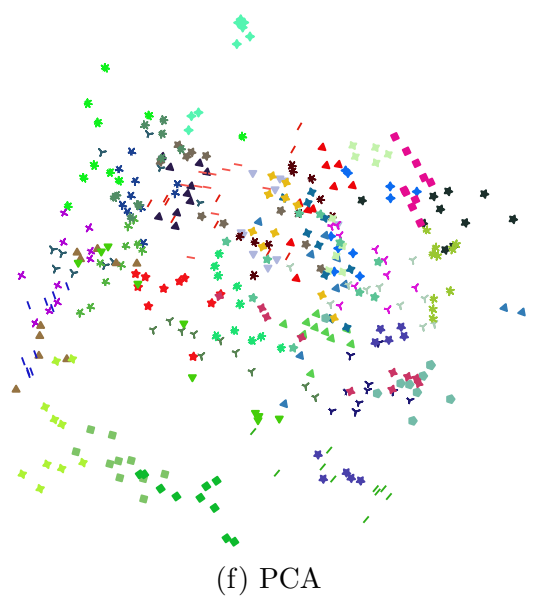
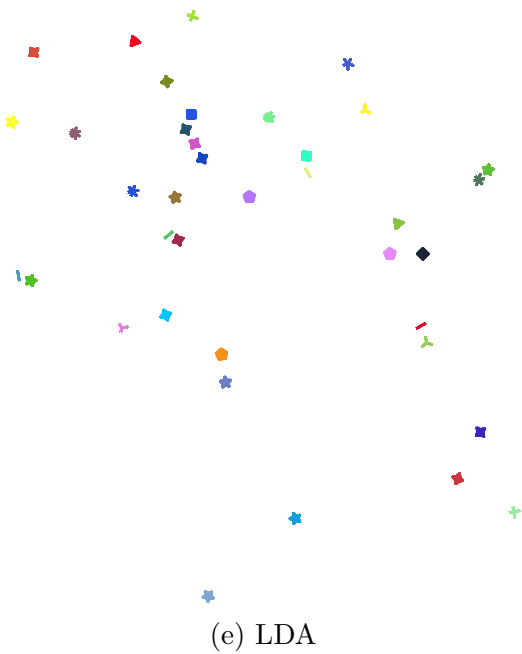
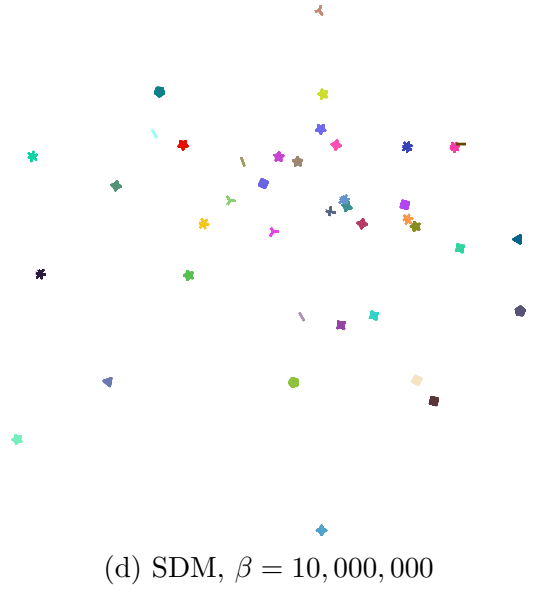
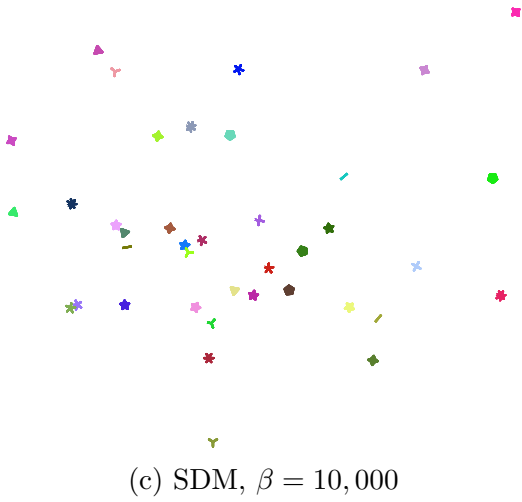
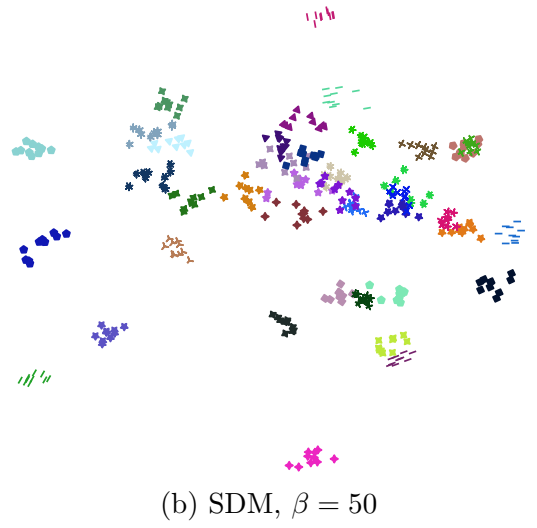
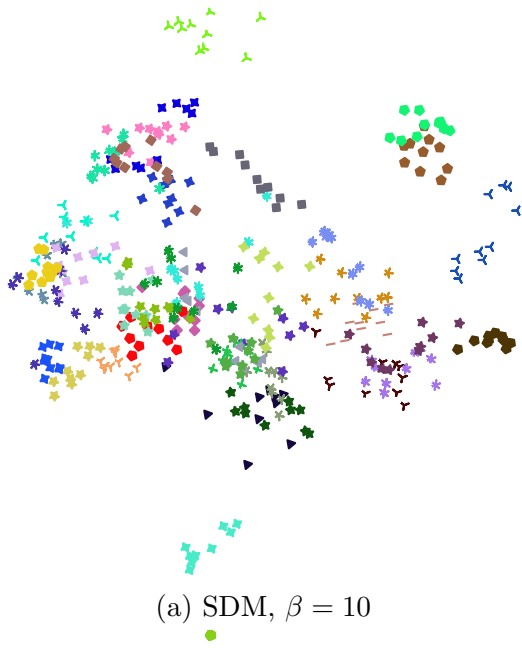


Figure 3.5.: 2D projections of the AT&T Dataset of Faces computed by PCA, LDA and SDMs for various values of  $\beta$ . The data is pre-processed using PCA, but not whitened.

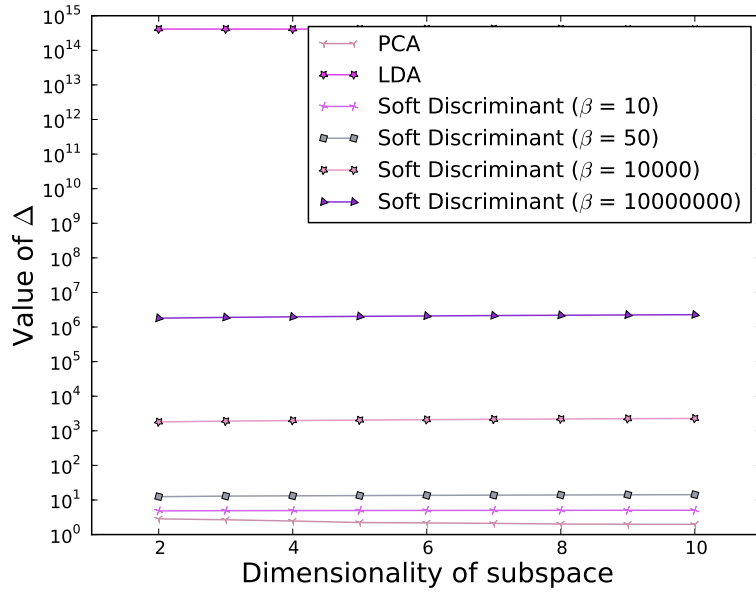


Figure 3.6.: Values of the  $\Delta$ -measure for the feature subspaces computed by SDMs for various values of  $\beta$ , on the AT&T Dataset of Faces. The data is pre-processed by PCA but not whitened.

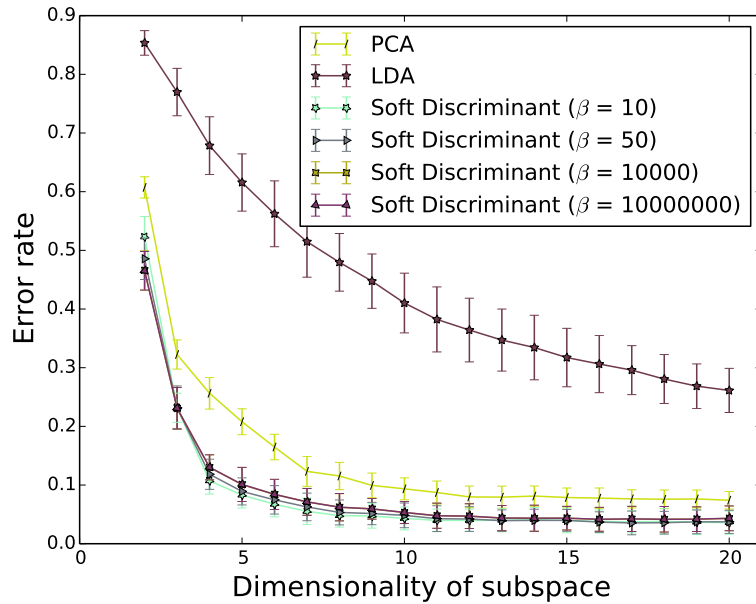


Figure 3.7.: Classification error rates averaged over 12 repeats of 2-fold cross-validation using the nearest neighbour classifier, for the subspaces computed by SDMs for various values of  $\beta$ , on the AT&T Dataset of Faces. The data is pre-processed by PCA but not whitened. For comparison, PCA and LDA are also included.

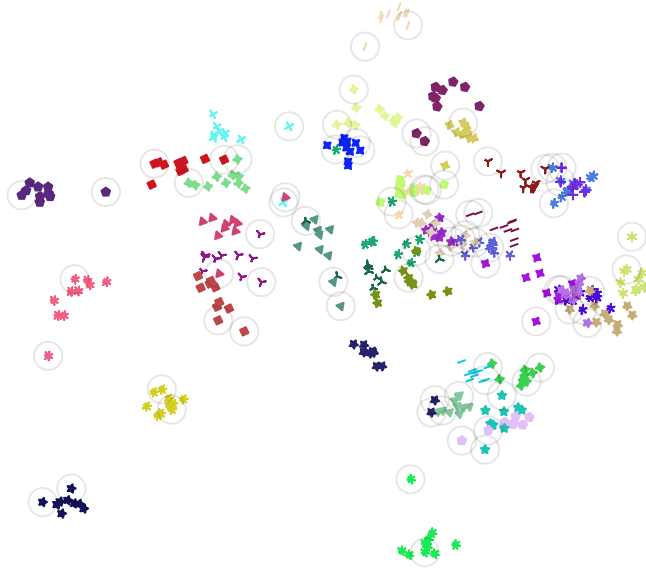


Figure 3.8.: 2D projection of the AT&T Dataset of Faces computed by SDM ( $\beta = 50$ ), where the projection is computed on 80% of the data. The rest 20% are marked by grey circles.

It is also interesting to notice that for this particular dataset, the classification performance of SDMs is rather insensitive to the value of  $\beta$ , in that the classification accuracy of all of the SDMs are quite close to each other, and all clearly better than PCA. This suggests that for this dataset, the range of  $\beta$  for which over-fitting is avoided is quite large, certainly at least as large as  $[10, 10000000]$ . For other datasets though, an SDM can be more sensitive to the value of  $\beta$ . Experiments on other datasets can be found in appendix A. Recall Laplacian maps defined in section 1.7, whose objective matrix is defined in equation (1.7). They are a family of eigenvalue-based feature extraction algorithms that encompasses PCA and Laplacian Eigenmaps, and we have shown in section 2.2.1 that our proposed EMI technique is also a special case. Laplacian maps are based on maximising or minimising the weighted squared pairwise distances  $\omega_{nm} \|\underline{y}_n - \underline{y}_m\|^2$  in the feature subspace. It is worth noting that SDM can also be formulated as a special case of Laplacian maps, since both  $\underline{s}_b$  and  $\underline{s}_w$  can be written as sums of weighted outer products  $\omega_{nm}(\underline{x}_n - \underline{x}_m)(\underline{x}_n - \underline{x}_m)^T$  for a suitably chosen set of weights  $\omega_{nm}$ .

### 3.3.1. Other similar algorithms

In [55], the authors presented a unifying framework for many dimensionality reduction techniques including Isomap, Locally Linear Embedding, Laplacian Eigenmaps, Hessian LLE, and Local Tangent Space Alignment. The framework is called *Patch Alignment*, and has inspired a new technique which the authors call *Discriminative Locality Alignment* (DLA). DLA is based on the eigendecomposition of an objective matrix that is very similar to our SDM objective matrix in equation (3.4). The difference is that instead of using the whole within-class covariance matrix  $\underline{s}_w$ , a matrix is built by considering each data point and only its *nearest* neighbours within its class. Note that in contrast, using the whole within-class covariance matrix  $\underline{s}_w$  is

equivalent to accounting for each data point and *all* of the other data points in the same class. Similarly, instead of using the between-class covariance matrix  $\underline{s}_b$ , a matrix is built by considering each data point and only its nearest neighbours outside of its class. In this way, the authors claim that local geometrical information is accounted for. The authors have demonstrated the superior classification performance of DLA over PCA, LDA and other competing algorithms on several face datasets, and have explained the superiority of DLA by the fact that it accounts for both discriminative information and local structure of the data. By its similarity to SDM however, we conjecture that the superior classification performance of DLA can also be explained in terms of over-fitting, specifically that it does not suffer as much from over-fitting as does LDA, and accounts for discriminative information while PCA does not.

Another algorithm similar to SDM is proposed in [20]. In fact, the algorithm proposed there is precisely a special case of SDM where  $\beta = 1$ . In other words the algorithm is an eigendecomposition of  $\underline{s}_b - \underline{s}_w$ . The authors called this the *Maximum Margin Criterion*. It is worth pointing out two flaws with the theory involved.

First, the name *Maximum Margin Criterion* is misleading. Eigenvectors of the matrix  $\underline{s}_b - \underline{s}_w$  do not maximise the margin between classes, as can be seen in figure 3.5. Indeed, LDA feature subspaces and perfectly fitting projections (see lemma 2 in section 3.1) both exhibit larger margins between classes. In fact, it is because SDMs do not maximise the margin between classes that we can control the level of class-discrimination (with the parameter  $\beta$ ) so as to avoid over-fitting. Therefore, we claim that the superior performance of the “maximum margin criterion” algorithm proposed in [20], when it does occur, is explained by the fact that the algorithm *does not* maximise the margin between classes and therefore avoids over-fitting, while accounting for discriminative information at the same time.

Second, some of the authors’ claims pertaining to equivalences under certain conditions between their maximum margin criterion and LDA are false. As a particular example, the authors claim on page 161 that when the sample size is small, “[Maximum Margin Criterion] is actually equivalent to LDA + PCA”. The justification for this was claimed to be that “in order to maximize  $\text{tr}(\underline{w}^T(\underline{s}_b - \underline{s}_w)\underline{w})$ , the columns of  $\underline{w}$  have to be in the null space of  $\underline{s}_w$ ”. This latter statement is not true. The reader may check this with the following example dataset. Consider a 2D dataset with 2 classes, each with 2 data points, which are  $\{(-1, 21), (-1, 20)\}$  and  $\{(1, -21), (1, -20)\}$  respectively. The null space of  $\underline{s}_w$  in this case is the line spanned by the vector  $(1, 0)^T$ , since in this direction the within-class variance vanishes. However, the difference between the between-class variance and the within-class variance in this direction,  $\underline{w}^T(\underline{s}_b - \underline{s}_w)\underline{w}$ , is equal to 1, and is clearly less than that in the direction of the vector  $(0, 1)^T$ , where the within-class variance is not 0 but equal to  $\frac{1}{4}$ . In general, and intuitively speaking, in order to maximise  $\underline{w}^T(\underline{s}_b - \underline{s}_w)\underline{w}$ , it is not necessary that the vector  $\underline{w}$  be in a direction where  $\underline{w}^T \underline{s}_w \underline{w} = 0$ , because in this direction,  $\underline{w}^T \underline{s}_b \underline{w}$  may be small. In maximising  $\underline{w}^T(\underline{s}_b - \underline{s}_w)\underline{w}$ , a large  $\underline{w}^T \underline{s}_b \underline{w}$  is preferred, sometimes at the expense of a positive  $\underline{w}^T \underline{s}_w \underline{w}$ .

### 3.3.2. Pre-processing by PCA

Prior to an execution of SDM, we recommend that the data be projected onto its maximal PCA space (the space spanned by all of the principal components), which has dimensionality  $\min\{D, N - 1\}$  (for  $N$  linearly independent data points). For high-dimensional datasets where  $D \geq N$ , this is a way of reducing computational cost, since the construction of the matrix  $\underline{s}_w$  is in general  $\mathcal{O}(D^2(N - K))$ , and that of  $\underline{s}_b$  is in general  $\mathcal{O}(D^2K)$ .

It is natural to wonder whether any information has been lost through projecting the  $D$ -dimensional dataset onto the smaller  $(N - 1)$ -dimensional PCA space. However it can be easily shown that no information is lost with respect to  $\underline{s}_w$  and  $\underline{s}_b$  through the application of PCA [53]. Let  $\Phi$  be the  $(N - 1)$ -dimensional PCA space of a high-dimensional dataset with  $D \geq N$ , and let  $\Phi^\perp$  be its orthogonal complement. Any  $\underline{w} \in \mathbb{R}^D$  can be written as  $\underline{w} = \underline{u} + \underline{u}_\perp$  where  $\underline{u} \in \Phi$  and  $\underline{u}_\perp \in \Phi^\perp$ . By construction we have  $\underline{u}_\perp^T \underline{s}_t \underline{u}_\perp = 0$ . Since the matrices  $\underline{s}_w$ ,  $\underline{s}_b$  and  $\underline{s}_t$  are all positive semi-definite, and since  $\underline{s}_w + \underline{s}_b = \underline{s}_t$ , we see that  $\underline{u}_\perp^T \underline{s}_w \underline{u}_\perp = 0$  and  $\underline{u}_\perp^T \underline{s}_b \underline{u}_\perp = 0$ . Therefore we see that  $\underline{w}^T \underline{s}_w \underline{w} = \underline{u}^T \underline{s}_w \underline{u}$  and  $\underline{w}^T \underline{s}_b \underline{w} = \underline{u}^T \underline{s}_b \underline{u}$ . This implies that the SDM objective  $\underline{w}^T (\underline{s}_b - \beta \underline{s}_w) \underline{w}$  is the same whether we work in the input space or the PCA space. The same derivation can be found in [53] where the authors used the argument to justify the use of LDA in the PCA space.

Note that we do not necessarily recommend whitening of the data before applying SDM, as we do require whitening for EMI. For SDM we only recommend pre-processing by PCA without whitening.

### 3.3.3. Controlling the level of class-discrimination

The essence of SDM is to be able to control the level of class-discrimination so as to avoid over-fitting while retaining enough discriminative information in the data. Figures 3.5 and 3.6 are a schematic illustration of how the parameter  $\beta$  affects the class-discrimination. However it would be interesting to see a more precise relationship between  $\beta$  and class-discrimination.

SDM can be formulated as the following optimisation problem.

$$\begin{aligned} \text{Maximise } \underline{w}^T \underline{s}_b \underline{w} \quad \text{subject to } \underline{w}^T \underline{s}_w \underline{w} &= \alpha \\ \text{and } \underline{w}^T \underline{w} &= 1, \end{aligned} \tag{3.5}$$

where  $\alpha$  is a constant. In other words, equation (3.5) says “find a unit vector  $\underline{w}$  along which the between-class covariance is maximised as long as the between-class covariance is equal to a given constant number”. The Lagrangian is

$$L(\underline{w}; \beta, \lambda) = \underline{w}^T \underline{s}_b \underline{w} - \beta(\underline{w}^T \underline{s}_w \underline{w} - \alpha) - \lambda(\underline{w}^T \underline{w} - 1). \tag{3.6}$$

By setting the derivative with respect to  $\underline{w}$  to zero we obtain

$$(\underline{s}_b - \beta \underline{s}_w) \underline{w} = \lambda \underline{w}. \tag{3.7}$$

As expected, equation (3.7) says that  $\underline{w}$  is an eigenvector of  $\underline{s}_b - \beta \underline{s}_w$  with eigenvalue  $\lambda$ . Now we would like to investigate the relationship between  $\beta$  and the class-discrimination along a feature, so we want to express  $\beta$  in terms of a feature  $\underline{w}$  and matrices  $\underline{s}_w$  and  $\underline{s}_b$ . Let  $\underline{1}$  be the vector whose every element is 1,  $\underline{1} := (1, \dots, 1)^T$ . We can produce two simultaneous equations from equation (3.7) by pre-multiplying it by  $\underline{w}^T$  and  $\underline{1}^T$  respectively, and substitute the two equations together to eliminate  $\lambda$ . We then obtain

$$\beta = \frac{\underline{1}^T \underline{w} (\underline{w}^T \underline{s}_b \underline{w}) - \underline{1}^T \underline{s}_b \underline{w}}{\underline{1}^T \underline{w} (\underline{w}^T \underline{s}_w \underline{w}) - \underline{1}^T \underline{s}_w \underline{w}} = \frac{\underline{1}^T [(\underline{w} \underline{w}^T - \underline{1}) \underline{s}_b] \underline{w}}{\underline{1}^T [(\underline{w} \underline{w}^T - \underline{1}) \underline{s}_w] \underline{w}}, \quad (3.8)$$

where  $\underline{1}$  is the identity matrix. Recall the  $\Delta$ -measure defined in equation (3.3), which is a reasonable measure of class-discrimination in the case of unimodal classes. For a one-dimensional feature subspace spanned by  $\underline{w}$ , the  $\Delta$ -measure can be written as

$$\Delta(\underline{w}) = \sqrt{\frac{\underline{w}^T \underline{s}_b \underline{w}}{\underline{w}^T \underline{s}_w \underline{w}}}. \quad (3.9)$$

A comparison between equations (3.8) and (3.9) suggests that  $\beta$  is related directly to  $\Delta$ . Note however the relationship between  $\beta$  and  $\Delta$  is stochastic and not exact, in the sense that it depends on the data which is random. For the AT&T Dataset of Faces, figure 3.9 shows the relationship between  $\beta$  and  $\Delta$  in 5-dimensional feature subspaces. Feature subspaces with other dimensionalities show the same result.

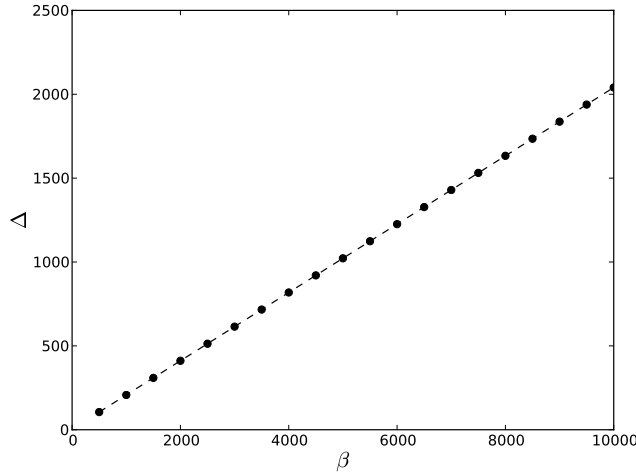


Figure 3.9.: A plot of  $\beta$  against the  $\Delta$ -measure in 5-dimensional feature subspaces computed by SDMs for various values of  $\beta$ , on the AT&T Dataset of Faces. Feature subspaces with other dimensionalities show the same result, that  $\beta$  and  $\Delta$  seem to be linearly related.

We see that  $\beta$  and  $\Delta$  seem to be linearly related to each other, and moreover the deviation from the optimal line is very small. This linear relationship is observed for other high-dimensional image datasets too, see appendix A. This implies that there is a simple way of choosing the value of the parameter  $\beta$ , according to what we want the class-discrimination to be in the resulting feature subspace (as measured by the  $\Delta$ -measure). One way of doing

this is to plot a graph like figure 3.9, use simple linear regression to estimate the parameters of linearity between  $\beta$  and  $\Delta$  (gradient and y-intercept), and evaluate the desired  $\beta$  using a known desired value for  $\Delta$ . This will involve several executions of SDMs for different values of  $\beta$ , which may be undesirable despite the low computational cost of SDM. However, given that the relationship between  $\beta$  and  $\Delta$  is linear for many datasets, it may suffice to simply compute two SDMs, one for relatively small  $\beta$  and the other for large  $\beta$ , and use the line between those two points as an estimate for the linear relationship between  $\beta$  and  $\Delta$ . Note that for some datasets, the level of over-fitting (and therefore the classification performance) is somewhat insensitive to the value of  $\beta$ , as shown in figure 3.7 for the AT&T Dataset of Faces. So for these datasets, the estimation of the optimal  $\beta$  need not be very accurate. The question remains as to how to determine the optimal value of  $\Delta$  in the resulting feature subspace. In general this will depend on  $N$ ,  $D$ ,  $K$  and the particular dataset. In practice, nearly optimal values of  $\beta$  or  $\Delta$  can be obtained by heuristics.

### 3.3.4. Highly representative datasets

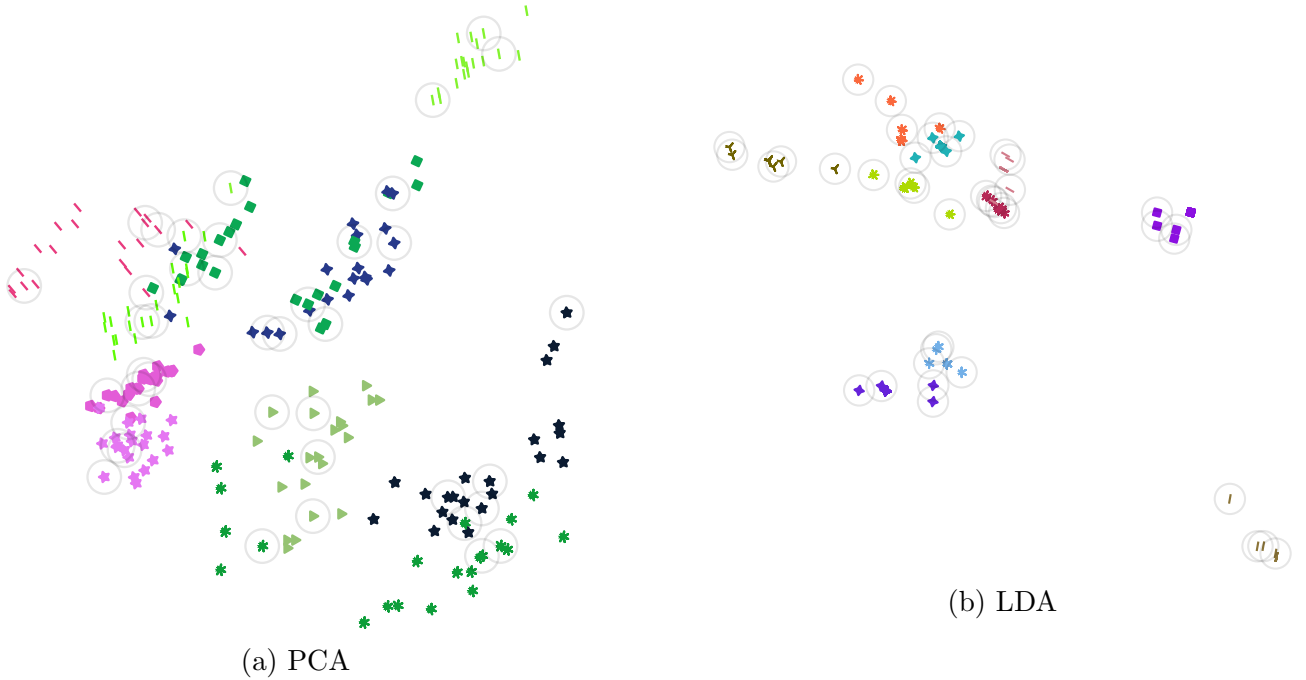


Figure 3.10.: 2D projections of the JAFFE face dataset computed by PCA and LDA respectively, where the projections are computed using 80% of the dataset. The remaining 20% are marked with grey circles. This illustrates the level of over-fitting in the 2D feature subspace. We can see that for this particular dataset, unlike the AT&T Dataset of Faces, LDA does not suffer severely from over-fitting.

We ought to emphasize that there exists high-dimensional datasets on which LDA does not suffer from over-fitting as severely as others, as shown in figure 3.10. Indeed, for the JAFFE face dataset [26], figure 3.11 shows that LDA gives better classification accuracy than PCA. The reason for the robustness against over-fitting is that the training dataset is *representative* of the true data distribution. Geometrically, this means that all non-zero regions of the (unknown)

true data distribution contain suitable numbers of training data points, and in general this is always a good property of a dataset. From figure 3.11 we can also see that SDMs out-perform both PCA and LDA. This could be because that although LDA does not suffer as much from over-fitting for the JAFFE dataset as it does for the AT&T dataset of faces, it is still more affected by over-fitting than SDMs (for appropriate values of  $\beta$ ). Based on this and experiments on other datasets (appendix A), we recommend that for high-dimensional datasets, even when LDA does not suffer very much from over-fitting, it may still be a good idea to use SDMs. In practice, owing to the low computational complexity of PCA, LDA, and SDM, we can always plot graphs using cross-validation on the training dataset to decide which technique is best for a particular dataset.

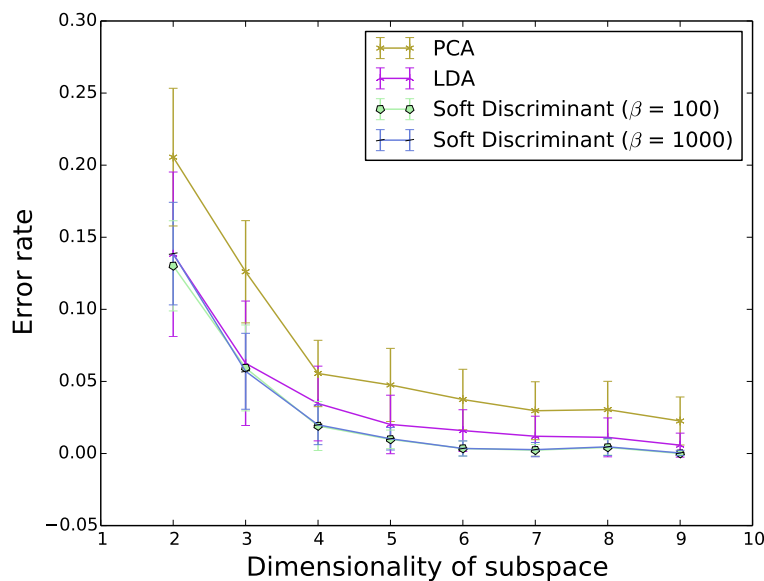


Figure 3.11.: Classification error rates averaged over 12 repeats of 2-fold cross-validation, using the nearest neighbour classifier on the JAFFE face dataset, for LDA, PCA and SMDs ( $\beta = 100, 1000$ ). Note that despite the high-dimensionality of the dataset, LDA out-performs PCA. However, SDMs still out-perform both PCA and LDA.

### 3.4. Summary

In this chapter we have studied the behaviour of feature extraction for high-dimensional datasets. We have seen how the advantages of information-theoretic feature extraction is limited in the high-dimensional setting, because we are always guaranteed to find low-dimensional projections with unimodal classes.

Moreover, for high-dimensional datasets we have seen that in aiming to maximise classification performance, we should not simply maximise class-discrimination (separation or margin), as this may lead to severe over-fitting. Indeed, it is easy to maximise class-discrimination, since lemma 2 implies that we can always find perfectly fitting projections. The prevailing factor affecting the classification performance for high-dimensional datasets is over-fitting, and the challenge of feature extraction for this type of data is in avoiding over-fitting while retaining

enough discriminative information in the data. We have proposed a parametric family of algorithms, collectively called soft discrimination maps (SDMs), that accomplish this task. This family, like EMI, is also a subset of a more general family of eigenvalue-based feature extraction algorithms which we call Laplacian maps, introduced in section 1.7. We have seen how the parameter of SDM controls the level of discrimination in the resulting feature subspace, and in particular that the two form a linear relationship.

## 4. Conclusions

We categorise our conclusions and contributions into two categories: *observations* and *techniques*.

Theoretical and practical observations:

- QMI maximisation by gradient ascent is effective for many comparatively low-dimensional datasets. However, its behaviour can have significant variety for high-dimensional datasets, depending on its free parameters, which include the learning rate and the stopping criterion. The algorithm seems better-behaved when the data is pre-whitened and especially when the features are initialised as LDA features before the gradient ascent iteration process begins.
- A popular view of mutual information is in terms of probability distributions and their similarities with each other. In the context of feature extraction in classification, an alternative view of mutual information in terms of pairwise interactions or pairwise distances provides a theoretically convenient perspective, which can elucidate a connection between information-theoretic and distance-based (or variance-based) feature extraction, through a generalised family of techniques which we call Laplacian maps.
- Information-theoretic feature extraction has limited advantages for high-dimensional datasets, compared with distance-based feature extraction. This is because high input dimensionality means that we can always find smaller subspaces in which each class in the dataset is unimodal, and mutual information prefers unimodal class-separation to multimodal class-separation.
- Lemma 2 (section 3.1) implies that we can always find perfectly fitting subspaces, in which the data points in each class coincide exactly, but such subspaces are prone to over-fitting. More generally, lemma 2 says that we can always find subspaces in which each class is unicentral, where “centre” could be mean, mode, or median.
- Over-fitting tends to occur when the classes in the training dataset are too compact in the feature subspace. This will occur when a feature extraction algorithm puts too much emphasis on maximising the class-discrimination. On the other hand, too little emphasis on maximising class-discrimination may lead to a high degree of overlap between classes. LDA tends to suffer from over-fitting for some high-dimensional face datasets (but not all), while PCA discards discriminative information and therefore tends to suffer from overlap between classes.

- The challenge of feature extraction for high-dimensional datasets is not to maximise class-discrimination (margin, distance, or compactness), but rather to avoid over-fitting while retaining enough discriminative information in the data.

Techniques:

- From the pairwise interaction view of mutual information, we can derive a new estimate of mutual information which we call eigenvalue-based mutual information (EMI), that can be maximised via eigendecomposition of an objective matrix, in the same way that PCA and LDA can be solved by eigendecomposition. EMI maximisation has no free parameters and is significantly more computationally efficient than QMI and Shannon’s MI, which currently can only be maximised using iterative methods. For high-dimensional datasets, EMI maximisation has the same theoretical computational complexity as PCA and LDA.
- Soft discrimination maps (SDMs) are a family of techniques that can circumvent the effects of over-fitting in high-dimensional datasets. Even for highly representative datasets where LDA out-performs PCA, SDMs can still out-perform LDA for appropriately chosen values of the parameter  $\beta$ . We find that  $\beta$  has a linear relationship with the level of discrimination in the resulting feature subspace.
- Both EMI and SDMs are specific cases of the more general family of Laplacian maps.

## 4.1. Future research

### 4.1.1. Mutual information

Our experiments on relatively low-dimensional data, in figure 2.10 and in appendix A, show that information-theoretic feature extraction can be effective and advantageous for classification tasks. However, our observations in section 3.1 imply that we are better off taking a distance-based approach for high-dimensional datasets, with particular emphasis on over-fitting. Dimensionality reduction is always necessary for high-dimensional datasets as long as the curse of dimensionality exists in classification. For relatively low-dimensional data however, with the continually increasing computing power at our disposal, the need for dimensionality reduction is limited. Together, these observations suggest that mutual information may have a limited role in feature extraction for classification.

### 4.1.2. Relevance vector selection and extraction

In this thesis we have mainly focused on reducing the number of dimensions, or equivalently, variables. This is useful in datasets where the input dimensionality is high. However, there are datasets with relatively low input dimensionality but a very high number of training data points. For example, the Letter dataset, available from the UCI database [1], has 20,000 training data points and only 16 input dimensions. For such datasets, dimensionality reduction

is practically redundant, and the prevailing factor limiting the efficiency of their computation is the large number of training data points. When the number of training data points is very large compared to the input dimensionality, we may reasonably expect that the true data distribution is very well represented by the training data, and in fact too well, in the sense that we need not so many data points to give a good representation of the true data distribution. In these situations, an effective way of increasing the efficiency of a classifier is to select a few representative data points from the training dataset, and only use these data points to represent the data distribution in order to classify new data points. The number of these representative points can be significantly smaller than the size of the whole training dataset. As an example, support vector machines (SVMs) [6, 3, sec.7.1] use only the data points on the *margins* (class boundaries), called *support vectors*, to compute decision surfaces. That is to say, if all the other training data points were removed, the resulting SVM trained on the remaining data points would be exactly the same. Another example is the *Relevance Vector Machine* [49, 3, sec.7.2], which performs classification using a sparse subset of representative data points called *relevance vectors*, which are identified among the whole training dataset through a Bayesian approach to maximum likelihood. Figure 4.1 illustrates how a data distribution can be well-represented by just a small subset of the whole training dataset, as long as the subset is selected sensibly. In this particular case, the two modes of the data distribution can be represented using just two data points.

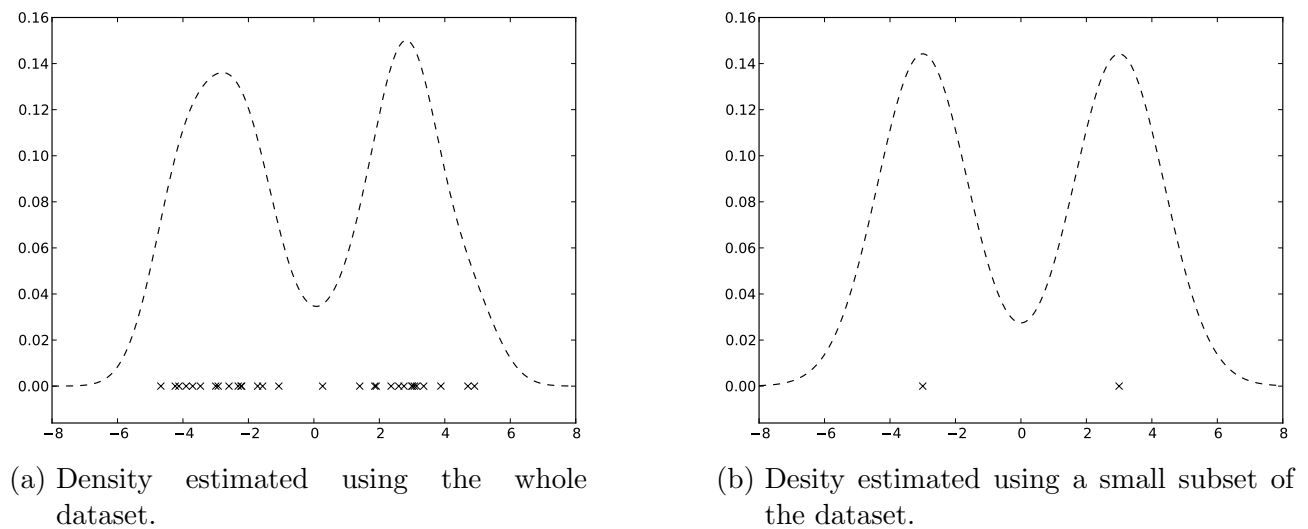


Figure 4.1.: Illustration of how a data distribution can be well-represented by only a small subset of the whole dataset. The data points are marked by  $\times$ , and their estimated distributions, computed by Gaussian kernel density estimation, are shown in dashed lines. The distribution computed using the small subset in (b) is similar to the one computed using the whole dataset in (a).

Instead of selecting a subset of relevance vectors to represent a data distribution as do relevance vector machines, we could construct new data points that are representative of the data distribution. We may call such a technique *relevance vector extraction*. The dichotomy between relevance vector *selection* and *extraction* is analogous to that between feature *selection* and *extraction*. Relevance vector extraction is closely related to *clustering*, for which there

are well-established techniques such as *k-means clustering* and *Gaussian mixture models* [3, chap.9]. For example, in Gaussian mixture models, the data distribution is assumed to be a weighted average of Gaussian clusters, and each cluster can be represented by a single mean data point instead of possibly many training data points, much like how the two Gaussian clusters in figure 4.1a can be represented by two mean data points in figure 4.1b. *k-means clustering* and *Gaussian mixture models* require the *expectation-maximisation algorithm* (EM) [3, chap.9]. Recent advances in *spectral clustering* [52] have seen improvements in efficacy and computational efficiency in clustering, by employing techniques in *spectral graph theory* [4].

It will be interesting in future research to investigate relationships between feature extraction and relevance vector extraction (or clustering). Recall that each row of a design matrix  $\underline{x}$  is a data point  $\underline{x}_n^T$ . Equivalently, each column of a design matrix is a dimension in the data. In this way, we can view feature extraction as the process of finding an alternative dataset with a design matrix  $\underline{y}$  that has fewer columns. Analogously, we can view relevance vector extraction as the dual problem of feature extraction, where we try to find an alternative dataset with a design matrix that has fewer rows. Both feature extraction and relevance vector extraction aim to reduce the size of the data, by reducing the number of columns and rows, respectively, of the design matrix. Note that by “reducing” the number of columns and rows, we do not simply mean selecting a subset of columns or rows from the original design matrix, for that would be feature and relevance vector *selection*. In feature and relevance vector *extraction*, we construct whole new datasets that represent the original input datasets. A more specific (but still informal) connection between feature extraction and relevance vector extraction is the following. We already know that PCA outputs eigenvectors that point in the directions of greatest variance in the data. The advent of *kernel PCA* [3, sec.12.3] has led to a new perspective of PCA, one in terms of similarity between data points. In this view, traditional PCA measures similarity by *alignment* between data points, which is measured by their dot product. This gives us the perspective of PCA that it outputs eigenvectors that point towards clusters of similar data points. Kernel PCA, on the other hand, measures similarity by a chosen kernel. For example, the Gaussian kernel (equation (1.23)) measures similarity by the closeness (inverse Euclidean distance) between two data points. (But it is far from the unique kernel that does this, any *local* kernel will do.) In this way we can see connections between kernel PCA and clustering, which is an example of a connection between feature extraction and relevance vector extraction.

Often equivalent techniques are known by different names in different literatures. For example, the *edge weight matrix* in spectral clustering and spectral graph theory is equivalent to the *Gram matrix* in kernel PCA. Connections between feature extraction and relevance vector extraction (or clustering) will enable us to extrapolate techniques from one field to the other, and elucidate useful insights.

### 4.1.3. Relationship between feature extraction, classification and regression.

As well as connections between feature extraction and relevance vector extraction, it will be enlightening to see connections between supervised feature extraction and classification. For example, LDA was originally formulated as a linear classifier for two-class problems, where the projection of a data point along the one extracted feature  $\underline{w}$  determines class membership (or probabilities thereof) [3, sec.4.1.4]. Two further examples of two-class classification techniques that determine class membership by the projection of data points along a vector  $\underline{w}$  are support vector machines (SVMs) and relevance vector machines (RVMs), as mentioned in section 4.1.2. A standard way of generalising two-class classifiers like SVMs and RVMs to multiple classes is the *one-versus-the-rest* technique. For example, in the one-versus-the-rest approach, an RVM is trained  $K$  times, once for each of the  $K$  classes, where each class and the rest of the classes are regarded as two classes, hence the name one-versus-the-rest. In this way, we end up with  $K$  RVMs, each with a corresponding weight vector  $\underline{w}$ . Put together, these  $K$  vectors can be regarded as  $K$  features that span a  $K$ -dimensional feature subspace.

Both RVMs and SVMs can be formulated for regression. (In fact, RVMs are more naturally formulated for regression than for classification.) Given the connections between feature extraction and classification, we can extrapolate the insights therein to feature extraction for regression.

There may be many benefits that can be derived from connections between feature extraction, classification and regression. For example, it is known that SVMs can suffer from the curse of dimensionality and over-fitting. Given the connections between feature extraction and classification, we can extrapolate our insights on over-fitting of feature extractors for high-dimensional datasets, in chapter 3, to investigate the mechanisms that cause over-fitting for SVMs, or to explain why SVMs are so successful in many practical uses.

### 4.1.4. Maximum-spread-zero-overlap and the maximum entropy principle

In this section we describe a specific feature extraction technique that is closely related to soft discriminant maps (SDMs) in principle, and to SVMs in formulation. The principle we employ is one of conditional maximum entropy, and we begin with an intuitive explanation. The algorithm described herein is a natural next step in future research from SDMs in chapter 3.

Recall that there is one free parameter in SDMs, which we denote by  $\beta$ . It controls the level of class-discrimination in the data in the resulting feature subspace. We mentioned that in practice, a good value of  $\beta$  can be chosen by heuristic or by experimentation, since the computational complexity of SDMs is low. However it would be theoretically elegant to be rid of any free parameters. Moreover, the lack of free parameters means that the formulation of the algorithm will be independent of the particular dataset. So is there an autonomous rule for choosing good values of  $\beta$  that works for all (high dimensional) datasets? We approach this

question by thinking in terms of the level of discrimination in the desired feature subspace. Recall that we observed in figure 3.9 that there seems to be a rather precise (and linear) relationship between the value of  $\beta$  and the level of class-discrimination in the feature subspace. So we could ask the following question instead. What level of class-discrimination ought we to aim for in the desired feature subspace?

We approach this question by thinking in terms of the level of uncertainty we have about the true data distribution, or equivalently our level of confidence in the representativeness of the training data. In an over-fitting scenario, each class in the training dataset is too tightly packed in the feature subspace. Figure 1.2 shows an example of this. In terms of uncertainty, we may say that the feature extractor is too confident about the training data, in that it fits the training data almost perfectly but fits new testing data points badly. Conversely, increasing the spread of each class (or decreasing the compactness) corresponds to decreasing our confidence in the training data (or increasing our uncertainty). In general, a high level of uncertainty makes good sense in classification, since it accommodates a good amount of variation in the data. Mathematically, the level of uncertainty is measured by the entropy of a distribution, and for unimodal distributions, high entropy corresponds to large spread (or variance).

However, we ought not to be too uncertain about the training data, lest too much overlap between classes will occur. Figure 2.3d shows an example of this. So a sensible aim is to maximise our uncertainty about each class as long as the classes in the training dataset do not overlap in the feature subspace. Mathematically, this translates to a *conditional maximum entropy principle*, where we maximise the entropy of each class-conditional distribution conditional on the non-overlapping between classes. Practically, this translates to a *maximum-spread-zero-overlap* (MSZO) algorithm.

More precisely, suppose we have a two-class problem, where the two classes are labelled by 1 and  $-1$  respectively. Suppose for the moment that the dataset is *separable*, that is to say that the two classes can be separated by a  $(D - 1)$ -dimensional hyperplane. Suppose each training data point  $\underline{x}_n$  is labelled by  $c_n$  where  $c_n = 1$  or  $-1$ . This is the same setup as in SVMs. For each  $n$ , let us denote  $z_n := \underline{w}^T \underline{x}_n + b$ , for some feature  $\underline{w}$  and *bias parameter*  $b$ . Since we assume that the dataset is separable, there exists  $\underline{w}$  and  $b$  such that  $z_n \geq 0$  whenever  $c_n = 1$  and  $z_n \leq 0$  whenever  $c_n = -1$ . So for such  $\underline{w}$  and  $b$ , we have  $c_n z_n \geq 0$  for all  $n$ . This is the no-overlap condition. So we have the following optimisation problem.

$$\begin{aligned} \text{Maximise } \underline{w}^T \underline{s}_w \underline{w} \quad \text{subject to } \underline{w}^T \underline{w} = 1 \\ \text{and } c_n z_n \geq 0. \end{aligned} \tag{4.1}$$

This is the precise statement of the MSZO problem. We can compare it with the underlying optimisation problem in SVMs.

$$\text{Minimise } \underline{w}^T \underline{w} \quad \text{subject to } c_n z_n \geq 1. \tag{4.2}$$

To solve MSZO, we can employ numerical algorithms in non-linear programming similar to

those used in SVMs. Note that in its current form, the MSZO problem is not quite a quadratic programming problem since the constraint  $\underline{w}^T \underline{w} = 1$  is not linear in  $\underline{w}$ . We refer the reader to [35] for an extensive discussion of constrained optimisation.

The problem stated in (4.1) is under the assumption that the classes are separable. For non-separable datasets, we can project the data points onto a higher-dimensional space through a mapping  $\underline{\phi} : \underline{x} \mapsto \underline{\phi}(\underline{x})$ , in which the classes are separable. One example of such a mapping is  $\underline{\phi}(\underline{x}) := (k(\underline{x}, \underline{x}_1), \dots, k(\underline{x}, \underline{x}_N))^T$ , where  $k(\cdot, \cdot)$  is a kernel. Note however that for high-dimensional datasets, lemma 2 implies that the classes are always separable in the input space. However if  $D > N$ , it may still be a good idea to use a kernel mapping described above, or pre-process the data using PCA, in order to improve computational efficiency and to avoid the curse of dimensionality.

We can extend this technique to multiple classes via one-versus-the-rest or similar approaches. Note that by our discussion in section 4.1.3, the MSZO algorithm can be regarded as either a feature extraction algorithm or a classification algorithm, as can SVMs and RVMs. A fair evaluation of MSZO should include comparisons with SVMs and RVMs.

#### 4.1.5. Unification of feature extraction techniques

In the current feature extraction and selection literature, the variety of techniques is greater than a PhD thesis can comprehensively encapsulate. A variety of perspectives and approaches to problems in a research area can be a good thing. However, too much of certain kinds of variety can distract from the main goals of a subject of research, and a theoretical unification of techniques can be beneficial.

There have been attempts to unify feature extraction and selection techniques [14, 55]. In the introduction of [14], the authors defined the *relevance* of features in terms of their mutual information with the target variables (class labels in the case of classification). We can view this as a *top-down* approach to the unification of techniques, in which an intuitively evident *standard* is defined (definition of relevance of a feature) and various new techniques can seek to conform to it. We can also conceive of a *bottom-up* approach to the unification of algorithms, where we find existing mathematical equivalences and relationships between different algorithms, thus unifying equivalent techniques and possibly generalising similar techniques as belonging to a common family. We have already mentioned that [55] presents a unification of several dimensionality reduction algorithms by showing that they have a common general procedure, one based on “part optimisation” and “whole alignment”. We can view this as an example of bottom-up approach to the unification of these techniques. In this thesis, we have taken provision to unify our proposed algorithms with each other, and with other existing techniques. In particular, we have defined the Laplacian map (section 1.7), which encapsulates EMI maximisation, SDMs, and Laplacian Eigenmaps [2]. (In fact, a slight relaxation of the definition of Laplacian maps would give us an equivalent generalisation to that presented in [55].) Furthermore, we have shown a relationship between EMI and QMI [50] in terms of pairwise interactions.

# A. Supplementary experimental results

This appendix includes some extra experimental results in support of the observations throughout this thesis. The reader may wish to consult this appendix when it is recommended in the main text. As a reminder, all the error bars on the graphs show one sample standard deviation either side of the average classification error rates, averaged over the numbers of repeats and folds.

## A.1. EMI maximisation

### A.1.1. Relatively low-dimensional datasets

The behaviour of low-dimensional datasets is more variable and less predictable than that of high-dimensional face datasets. This is intuitively evident, given the observations we have made in chapter 3. In short, the shape of a data distribution plays a comparatively bigger role in low input dimensionality, whereas in high-dimensional datasets, the exact shape of a data distribution is less relevant than the spread of each class, which affects the level of over-fitting. We’ve seen how EMI maximisation performed on the Pima dataset in figures 2.10 and 2.11. Here we present some more experimental results on datasets available from the UCI Machine Learning Repository [1].

Figure A.1 shows the error rates on the Breast Cancer dataset. This dataset is quite unusual in that the error rates go up as the dimensionality of the feature subspace increases. This is likely because the *intrinsic dimensionality* of the dataset is low, which implies that classification in relatively high-dimensional subspaces is unstable. Note that EMI maximisation achieves the lowest error rates in the lower-dimensional regions. QMI maximisation was not included due to its high computational complexity. A similar evaluation of QMI maximisation using gradient ascent would take more than 24 hours (with a current PC), whereas the evaluation of both PCA and EMI maximisation together, as shown in figure A.1 only took about 20 minutes. However, a 2D projection of QMI maximisation by gradient ascent is included in figure A.2.

The Image Segmentation dataset is provided explicitly with a training subset with 210 training data points, and a much larger testing subset with a total of 2100 testing data points. We carried out two experiments on this dataset. The first is a 5-fold cross-validation on the training dataset only, shown in figure A.3. The second is a direct test where the feature extractors are trained on the training dataset and tested on the testing dataset, shown in figure A.4. In figure A.3 we observe that the information-theoretic methods clearly perform better in terms of classification, and in particular that EMI maximisation performs better than QMI maximisa-

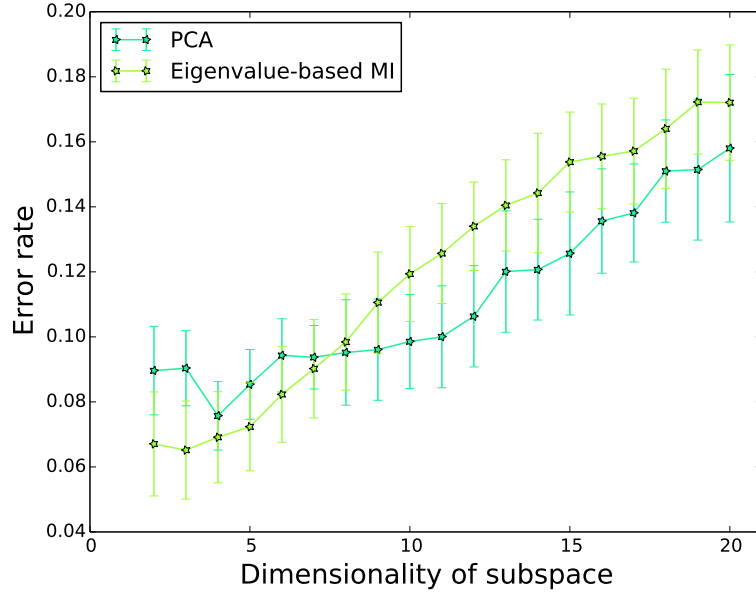


Figure A.1.: Classification error rates averaged over 12 repeats of 2-fold cross-validation of the nearest neighbour classifier, on the pre-whitened Breast Cancer dataset. Note that since this is a 2-class dataset, LDA would only be able to extract one feature, and so was not included. QMI maximisation was also not included due to its high computational complexity.

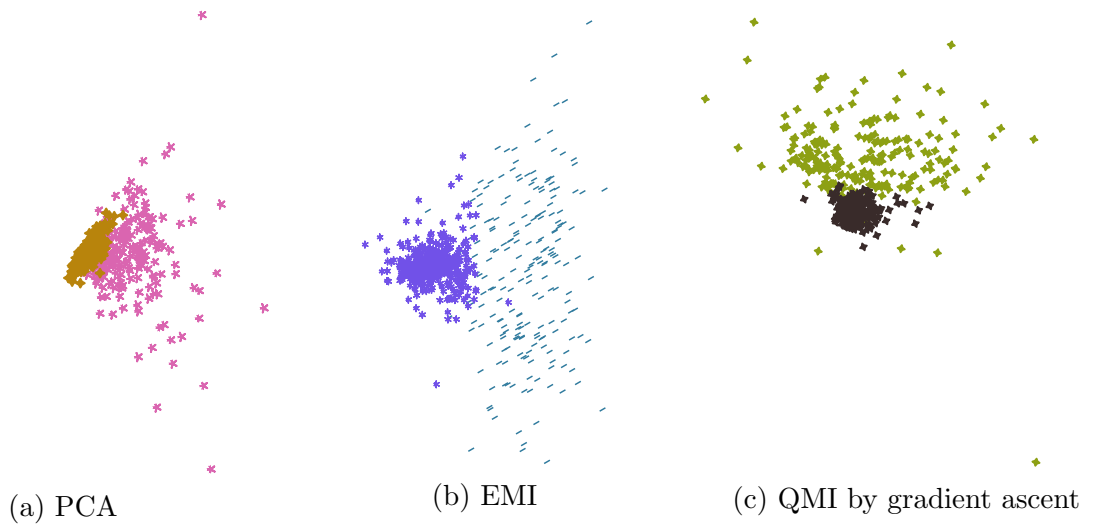


Figure A.2.: 2D projections of the pre-whitened Breast Cancer data.

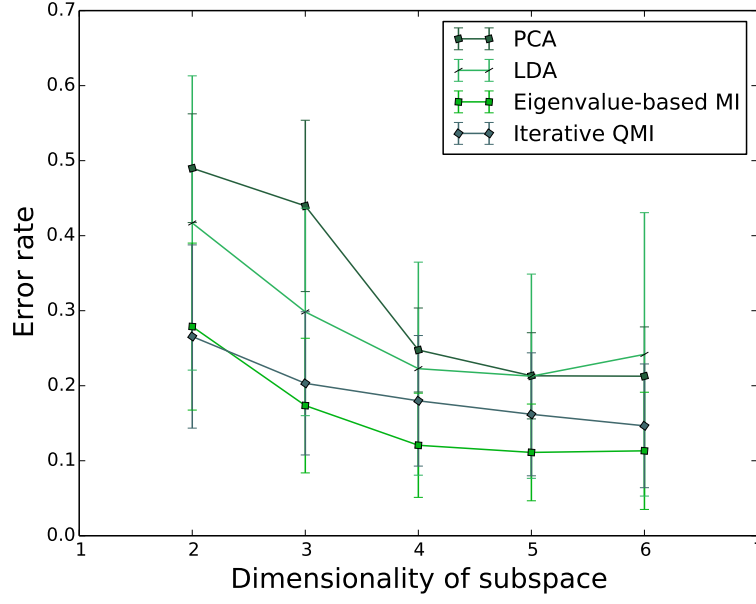


Figure A.3.: Classification error rates averaged over 9 repeats of 5-fold cross-validation of the nearest neighbour classifier, on the pre-whitened Image Segmentation dataset.

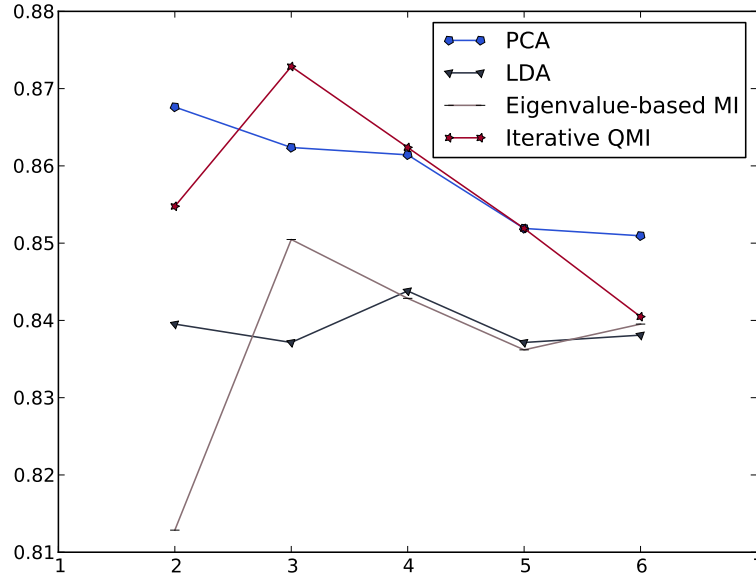


Figure A.4.: A direct test on the pre-whitened Image Segmentation dataset, where the respective feature extractors are trained on the training dataset of 210 data points, and tested on the testing dataset of 2100 data points.

tion by gradient ascent. In figure A.4, we see that although the information-theoretic methods do have slightly better classification performance, all of the techniques perform quite badly. This is not surprising since the testing set is 10 times as large as the training set, which may be too small to represent the true data distribution well. Figure A.5 shows the 2D projections of the training subset for the respective feature extractors, from which we can see that one reason for the under-performance of PCA may be that it is affected by an outlier.

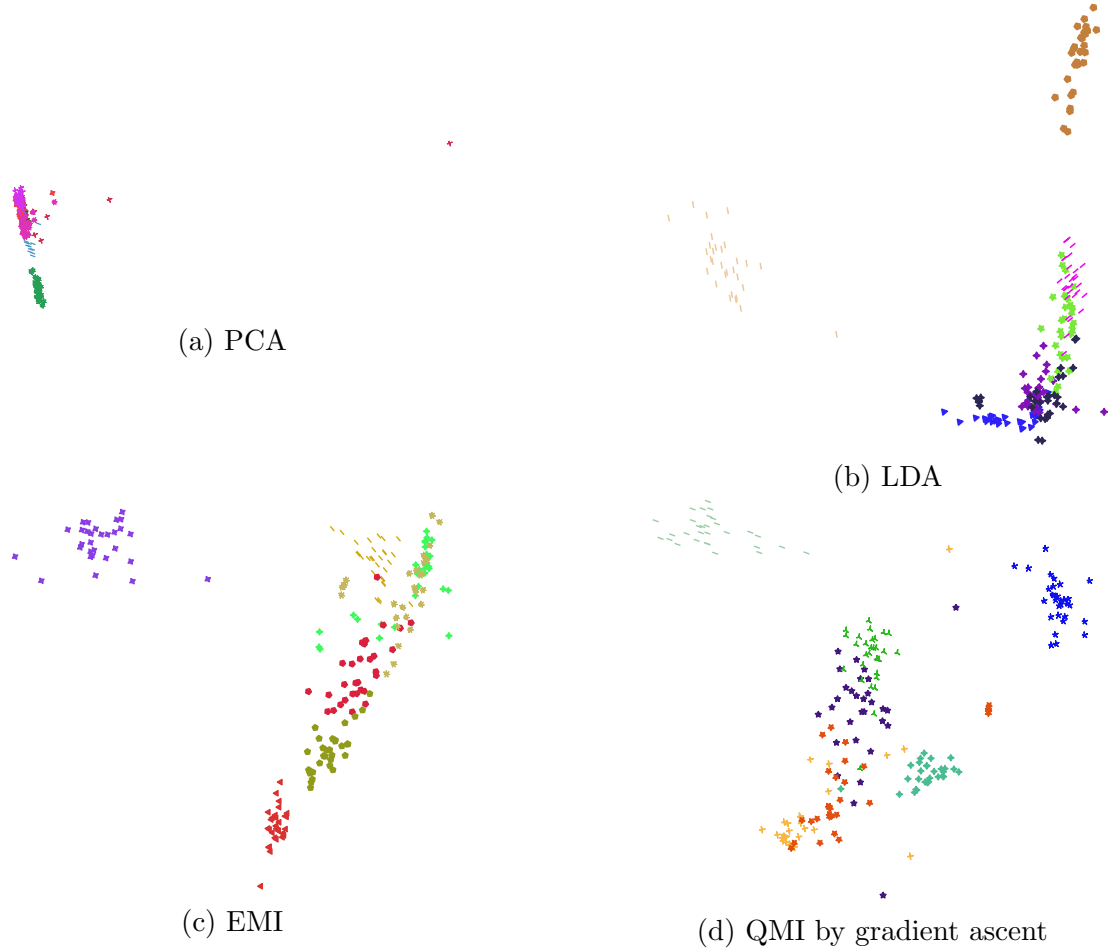


Figure A.5.: 2D projections of the pre-whitened training subset of the Image Segmentation dataset.

On the Parkinsons dataset, PCA, QMI and EMI maximisation have comparable performance, as shown in figure A.6. Figure A.7 shows the respective 2D projections.

EMI maximisation seems to perform relatively well on the Silhouette dataset. This is a 4-class dataset, so LDA could only be evaluated for subspace dimensionalities of 2 and 3. Figure A.8 shows the results. Again due to the high computational complexity of QMI maximisation by gradient ascent, we did not include an evaluation of it here. An evaluation of QMI maximisation by gradient ascent, equivalent to that for PCA and EMI maximisation would take more than 30 hours, whereas for PCA and EMI maximisation, the evaluation took just over 30 minutes. However, a 2D projection of the dataset computed by QMI maximisation is included in figure A.9.

Figures A.10 and A.11 show the error rates and 2D projections of the Yeast dataset. Again,

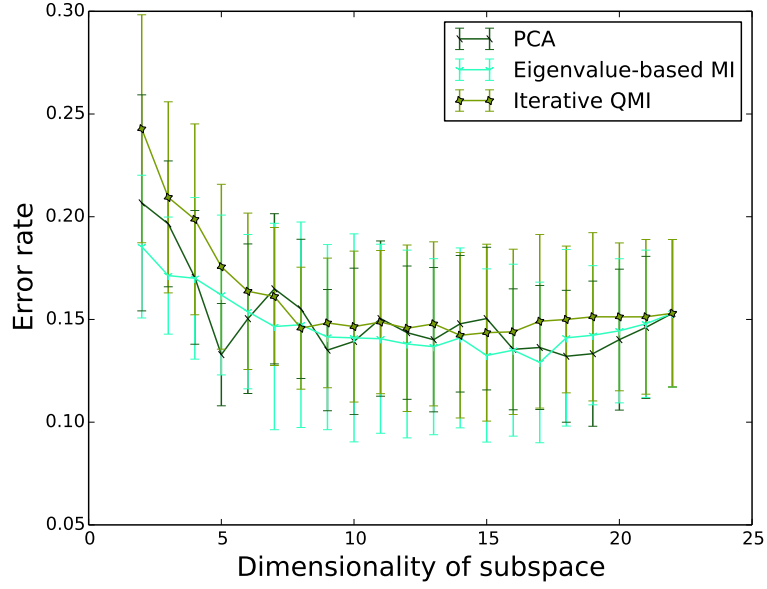


Figure A.6.: Classification error rates averaged over 12 repeats of 2-fold cross-validation of the nearest neighbour classifier, on the pre-whitened Parkinsons dataset. Since this is a 2-class dataset, LDA would only be able to extract one feature as so is not included in this experiment.

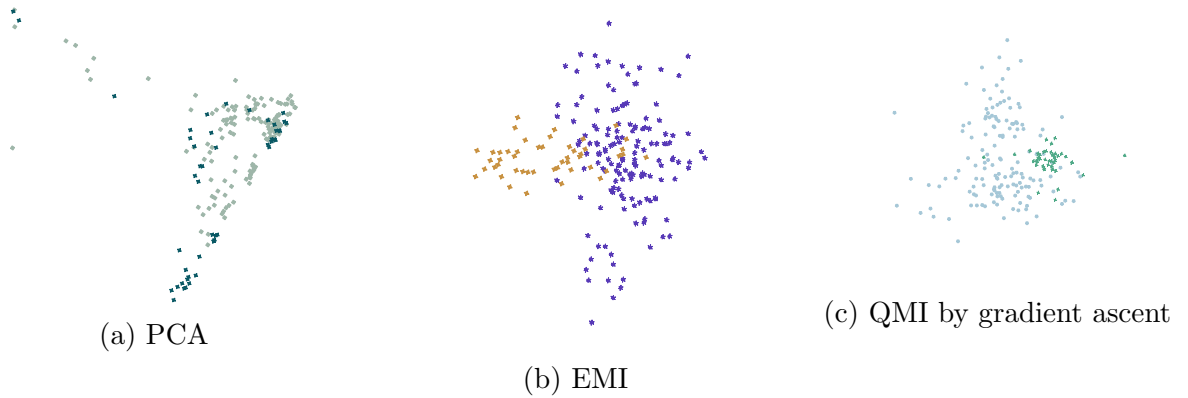


Figure A.7.: 2D projections of the pre-whitened Parkinsons dataset.

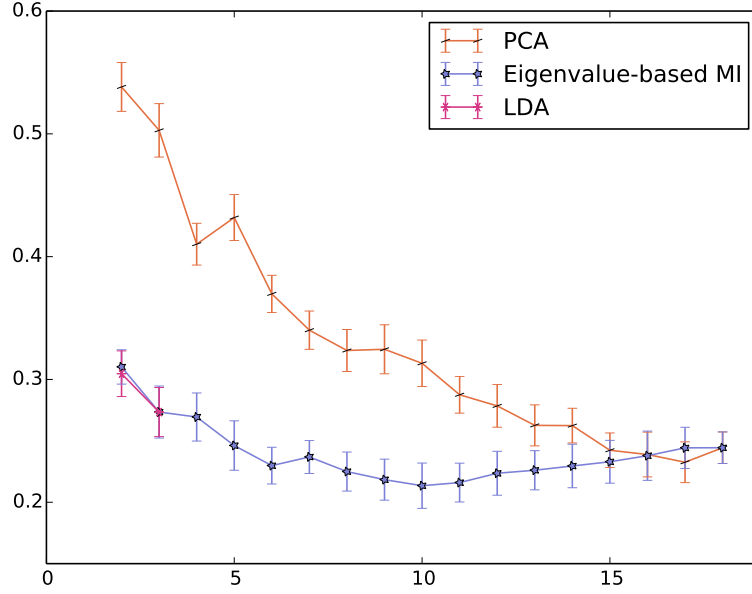


Figure A.8.: Classification error rates averaged over 12 repeats of 2-fold cross-validation of the nearest neighbour classifier, on the pre-whitened Silhouette dataset. Due to the high computation complexity of QMI maximisation by gradient ascent, we did not include it here. This is a 4-class dataset and LDA can only give up to 3 features.

due to the high computational complexity of QMI maximisation by gradient ascent, its evaluation is not included here.

All of the datasets described so far are available on the UCI machine learning database [1]. Table A.1 is a table listing the characteristics of these relatively low-dimensional datasets, including the Pima dataset shown in figures 2.10 2.11 in section 2.2. We briefly note that EMI maximisation solves a little problem associated with LDA, that it cannot output more than  $K - 1$  features.

| Dataset name  | $N$                      | $K$ | $D$ | Class sizes                               |
|---------------|--------------------------|-----|-----|-------------------------------------------|
| Pima          | 768                      | 2   | 8   | 268, 500                                  |
| Breast Cancer | 569                      | 2   | 30  | 212, 357                                  |
| Image Seg.    | 210 (train), 2100 (test) | 7   | 19  | 30                                        |
| Parkinsons    | 195                      | 2   | 22  | 147, 48                                   |
| Silhouette    | 846                      | 4   | 18  | 199, 217, 218, 212                        |
| Yeast         | 1484                     | 10  | 8   | 244, 429, 463, 44, 35, 51, 163, 30, 20, 5 |

Table A.1.: Relatively low-dimensional datasets and their characteristics.  $N$  = total number of data points;  $K$  = number of classes;  $D$  = input dimensionality.

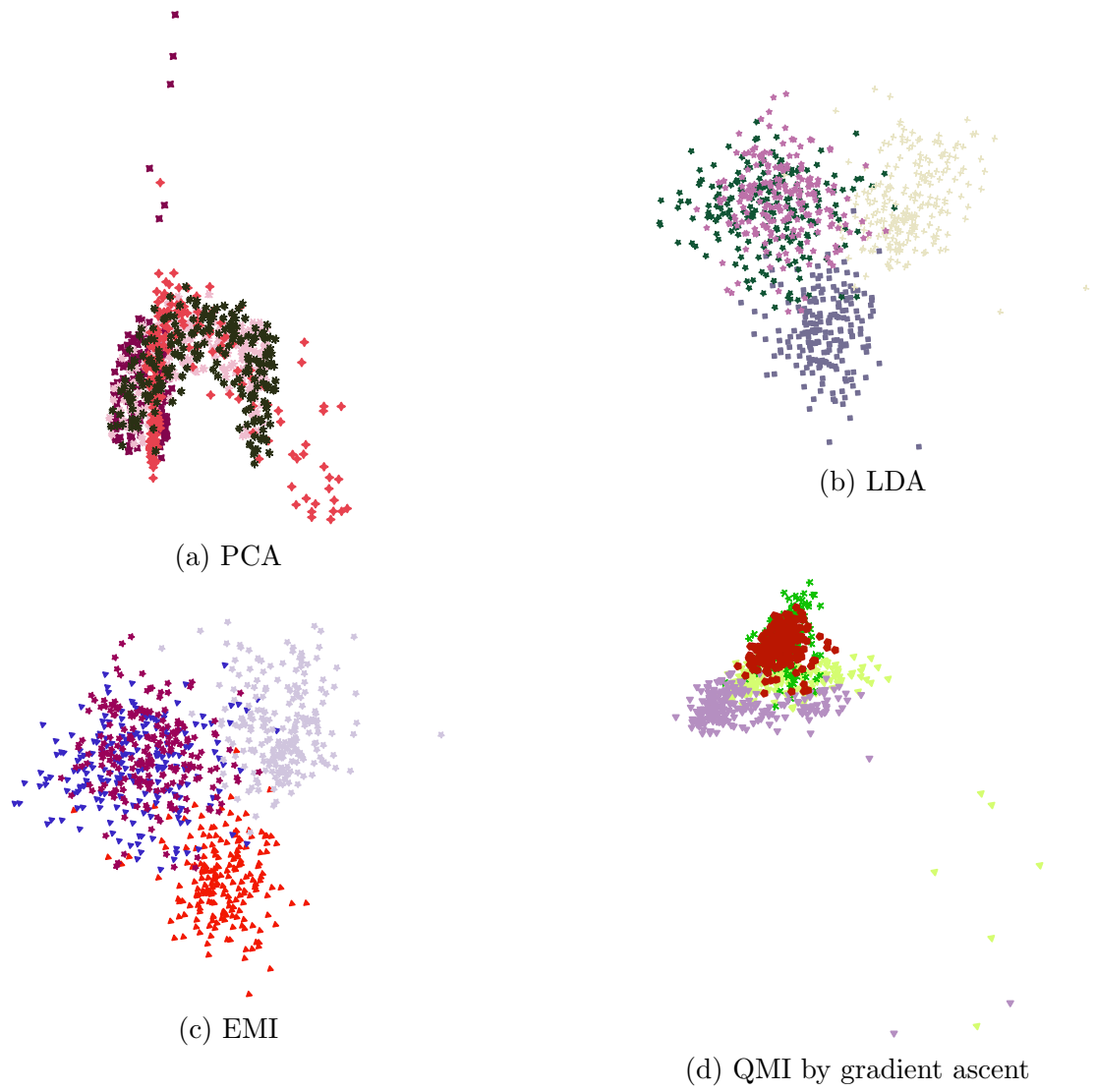


Figure A.9.: 2D projections of the pre-whitened Silhouette dataset.

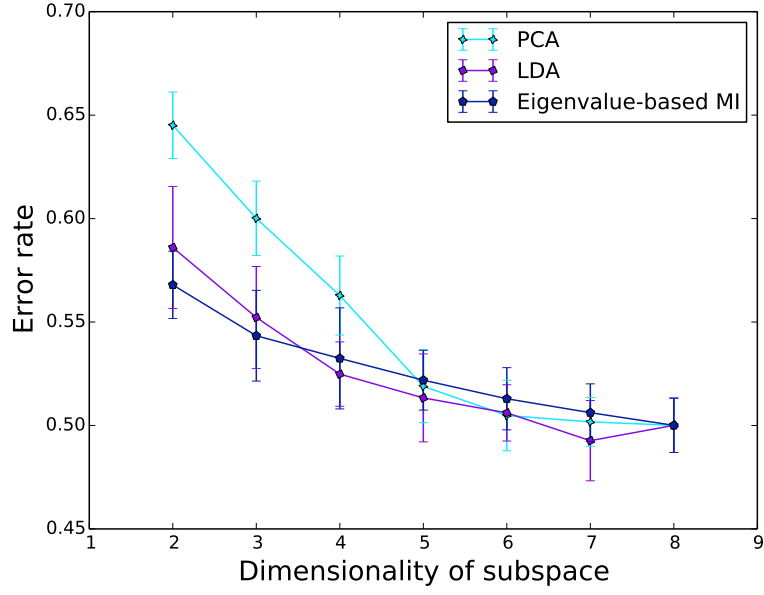


Figure A.10.: Classification error rates averaged over 12 repeats of 2-fold cross-validation of the nearest neighbour classifier, on the pre-whitened Yeast dataset. Due to the high computation complexity of QMI maximisation by gradient ascent, we did not include it here.

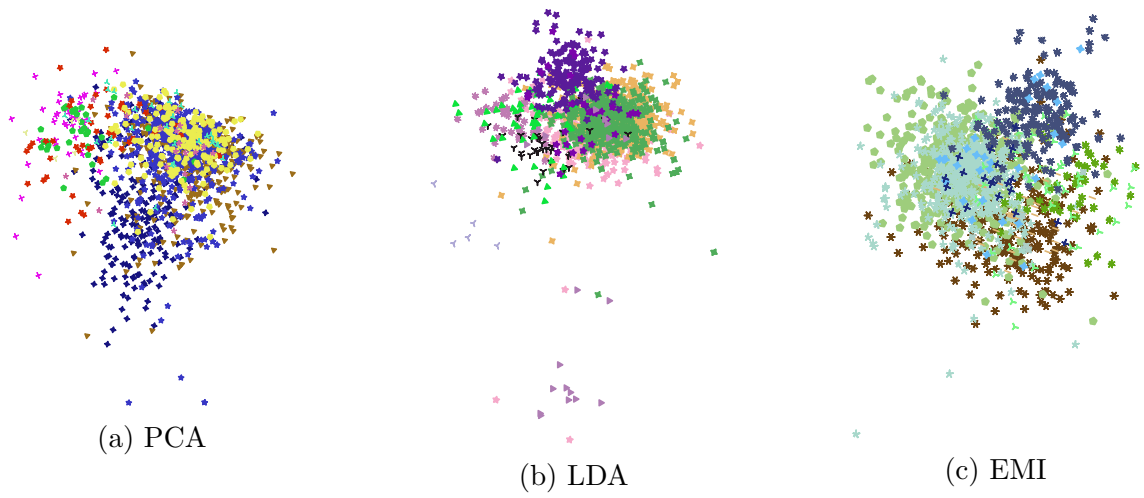


Figure A.11.: 2D projections of the pre-whitened Yeast dataset.

### A.1.2. High-dimensional datasets

Here we present experimental results in support of the observation in chapter 3 that LDA and EMI often under-perform compared to PCA in terms of classification performance, for high-dimensional datasets. Recall that in short, this is due to over-fitting in LDA and EMI, in particular that the training data classes in the LDA and EMI subspaces are too compact. Our aim here is to illustrate the over-fitting effect in techniques that put too much emphasis on the compactness of classes and class-separation in the feature subspace. We have observed in section 2.1 that the behaviour of QMI maximisation by gradient ascent can be unpredictable in high-dimensional input spaces, and that its computational complexity is significantly higher than PCA, LDA and EMI maximisation. We observed in figures 2.2 and 3.3 that QMI prefers large class separation and unimodal classes. In conclusion, due to the unpredictability of the gradient ascent algorithm in high-dimensional input spaces, QMI maximisation will not converge properly in general; and when it does, it will give feature subspaces that have the same property as those of LDA and EMI maximisation, that the classes are compact and well-separated from each other — precisely the mechanism that causes over-fitting. In regard of these observations, we do not include evaluations of QMI maximisation by gradient ascent in the experiments presented here.

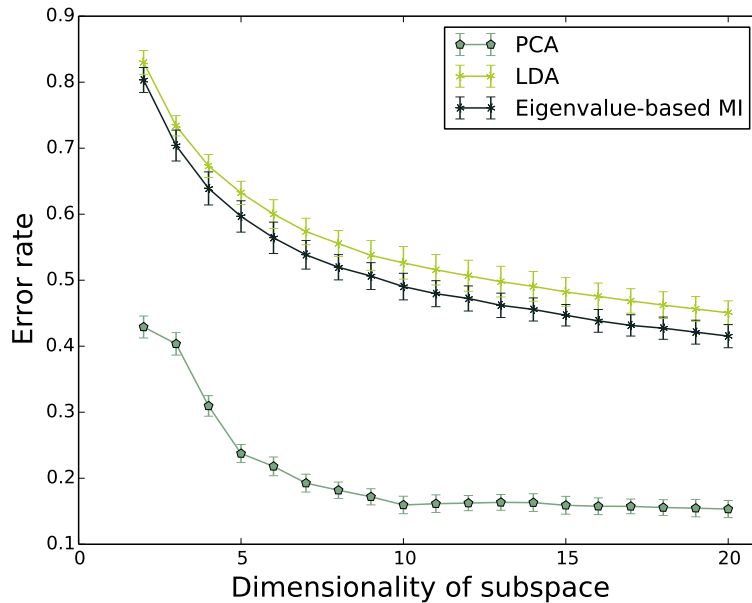


Figure A.12.: Classification error rates averaged over 12 repeats of 2-fold cross-validation of the nearest neighbour classifier, on the pre-whitened Essex95 face dataset.

The Essex face database (maintained by Dr. Libor Spacek, University of Essex [48]) has four datasets, called Essex94, Essex95, Essex96, and Essex Grimace. Essex94 and Essex96 are significantly larger than Essex95 and Essex Grimace. See table A.2 for the characteristics of the high-dimensional datasets used in this thesis. Figure A.12 shows the classification results on the Essex95 dataset, and figure A.13 shows the respective 2D projections. Figures A.14 and A.15 are the corresponding figures for the Essex Grimace dataset. We make two important observations from these figures. First, the classification performance of LDA and EMI is worse

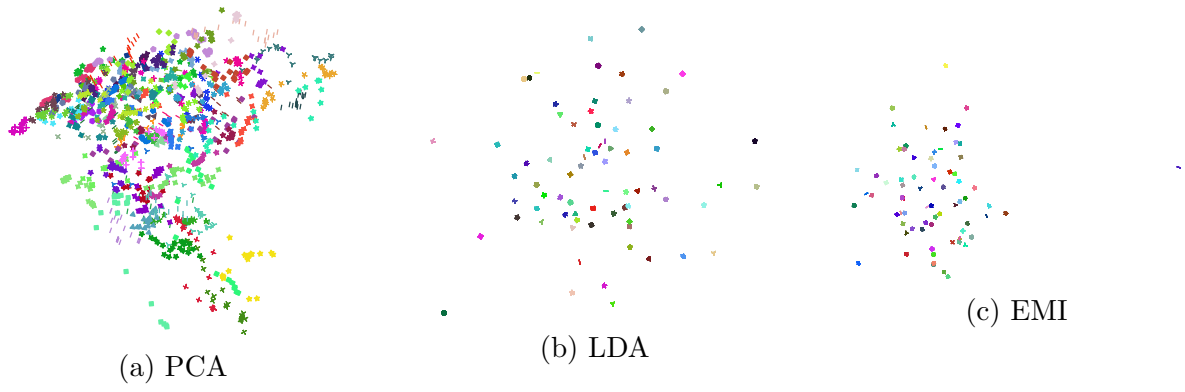


Figure A.13.: 2D projections of the pre-whitened Essex95 dataset.

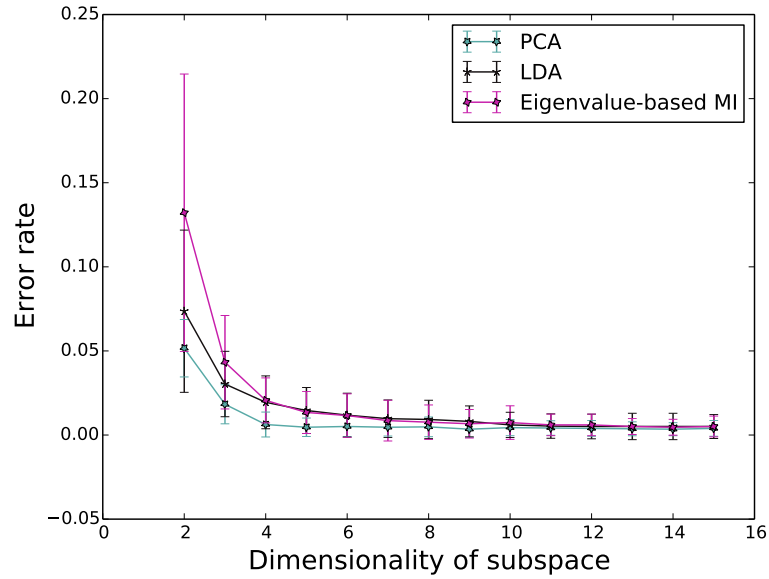


Figure A.14.: Classification error rates averaged over 12 repeats of 2-fold cross-validation of the nearest neighbour classifier, on the pre-whitened Essex Grimace dataset.

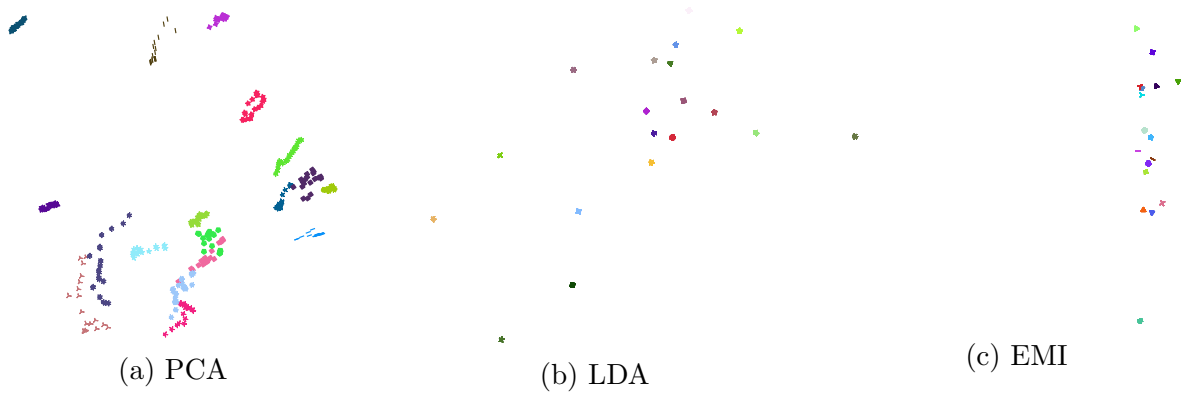


Figure A.15.: 2D projections of the pre-whitened Essex Grimace dataset.

than that of PCA. Second, the 2D projections computed by LDA and EMI maximisation are ‘better’ than those computed by PCA, in the sense that the classes in the LDA and EMI feature subspaces are well-separated from each other and individually compact, so much so that at this resolution, each class appears as a singleton point. We can make exactly the same observations for the Essex94 and Essex96 datasets, and so to save space we do not include the corresponding diagrams here.

An interesting point to note is that on Dr. Libor Spacek’s official website for the Essex Datasets, he mentioned that Essex Grimace is one of the “most difficult” datasets (of the four), due to “extreme variations of [facial] expressions”. With the insight we have obtained through this thesis, we note that the word “difficult” should be understood with care, and in context. By comparing figures A.14 and A.12, we see that Essex Grimace is actually *easier* compared with Essex95, since the classification results on Essex Grimace are better. The reason for this, from the over-fitting perspective of feature extraction that we explored in chapter 3, is precisely *because of* the “extreme variations” in facial expressions. Indeed, with large variations in facial expressions, the training data points become *more representative* of the true data distribution (or, of all the possible facial expressions), and consequently techniques like EMI and LDA suffer less from over-fitting.

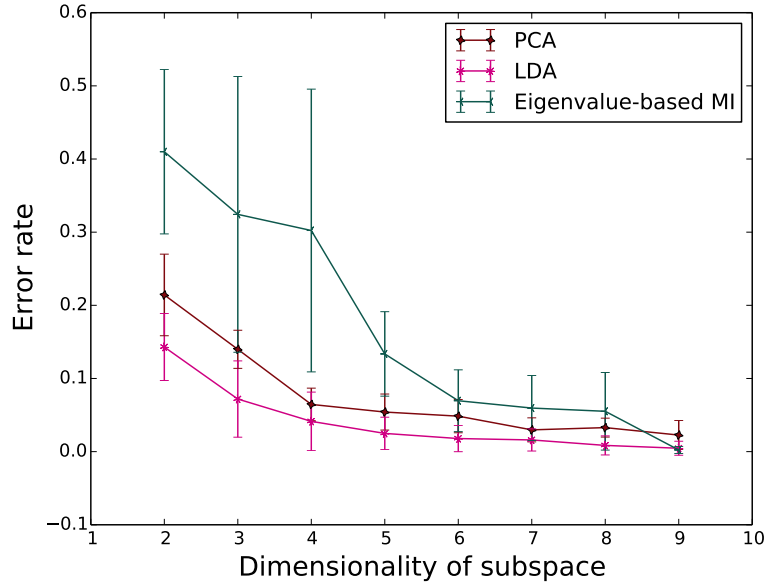


Figure A.16.: Classification error rates averaged over 12 repeats of 2-fold cross-validation of the nearest neighbour classifier, on the pre-whitened JAFFE dataset.

We mentioned the JAFFE face dataset [26] in section 3.3.4, which seems particularly well-suited for LDA, since LDA gives better classification performance than PCA even though the input dimensionality is high. It is worth mentioning that for this dataset, EMI maximisation curiously under-performs compared with both LDA and PCA, as shown in figure A.16. However, as shown in figure 3.11, the SDM technique gives better results than all of LDA, PCA, and EMI maximisation. The cause of the bad classification performance of EMI maximisation on JAFFE can be seen in figure A.17, where we can see that some classes are overlapping with each other. The mechanisms that cause this is not clear at this stage. It is plausible that some outliers or

artifact in the data are causing the problem. However, since we see that SDMs consistently give superior performance for high-dimensional datasets, and that we do not recommend the use of LDA or EMI maximisation for high-dimensional datasets because of the observations made in chapter 3, we have not dwelled on the investigation of this problem. Rather, this experiment serves as a warning that EMI maximisation, as formulated in section 2.2 of this thesis, can occasionally give curious results.

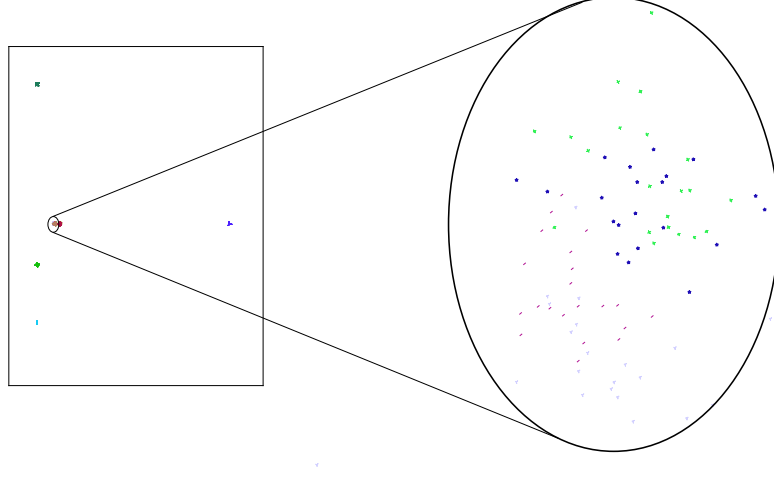


Figure A.17.: 2D projection of the pre-whitened JAFFE face dataset, computed by EMI, where part of the projection is zoomed in. We can see that, although some classes are well-isolated, others are overlapping with each other.

We end this section with evaluations of EMI maximisation on two more datasets. Classification results for the Sheffield Face Dataset (formerly UMIST) [42] are shown in figure A.18. The FVC (Fingerprint Verification Competition) [11] datasets used here are subsets of the original full database. Only these subsets are available to download from [11]. There are 4 subsets, corresponding to different sensor technologies (one of them, FVC4, is synthetically generated). The classification results for the 4 subsets are very similar, and so we will show only one of these, in figure A.19. Note that on all of the high-dimensional datasets mentioned in this section, SDMs can give superior classification performance to all of LDA, PCA and EMI maximisation, as we will see in section A.2.

## A.2. Soft discriminant maps

In figures 3.7 and 3.11 we have shown the classification performance of SDMs on the AT&T Dataset of Faces and the JAFFE face dataset respectively. In this section we present some more experiments on various datasets. The datasets used and their characteristics are summarised in table A.2. All evaluations are averages of 12 repeats of 2-fold cross-validation of the nearest neighbour classifier. Moreover, datasets are always pre-processed by PCA before the application of SDMs. However the datasets are not whitened during pre-processing, in contrast to the evaluations of EMI where the datasets are pre-whitened as per prerequisite.

Classification results for the four Essex datasets are shown in figure A.20. Figure A.21 shows

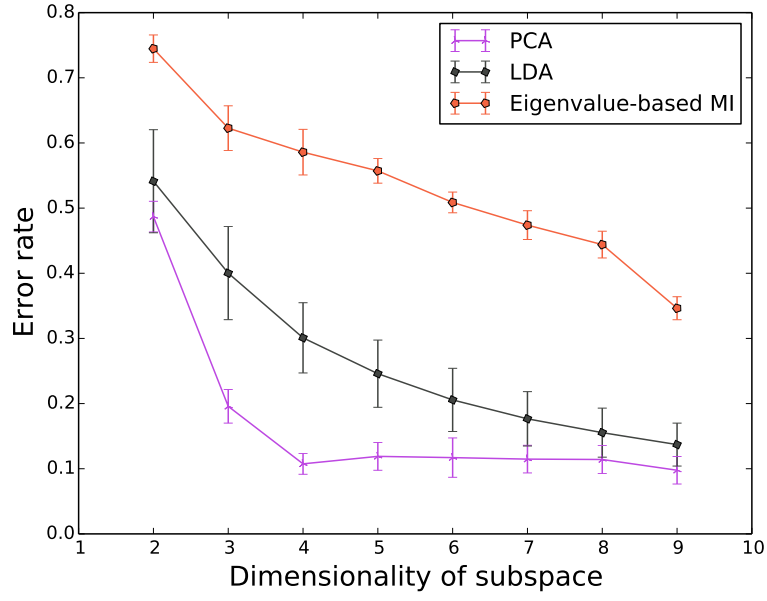


Figure A.18.: Classification error rates averaged over 12 repeats of 2-fold cross-validation of the nearest neighbour classifier, on the pre-whitened Sheffield Face Dataset.

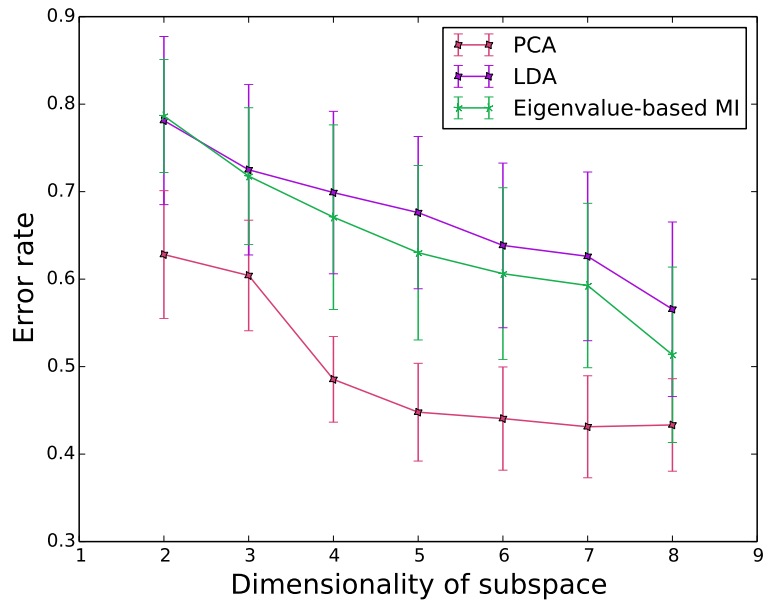
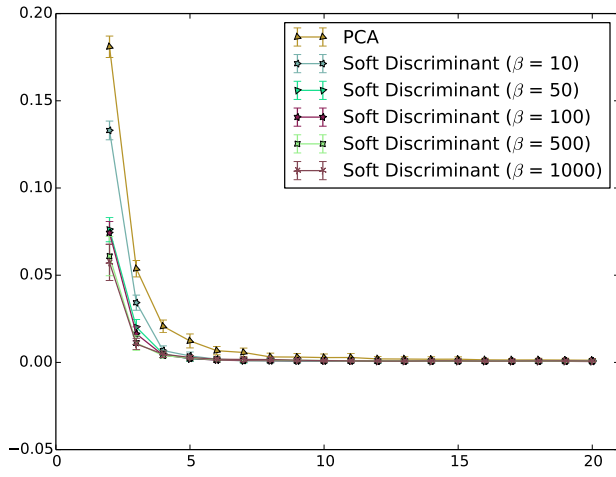
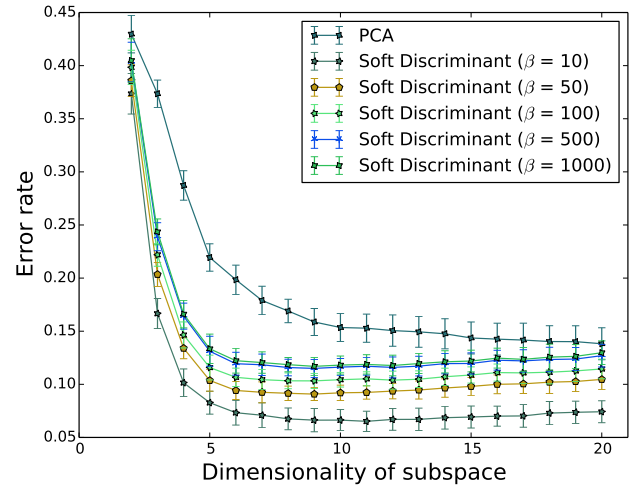


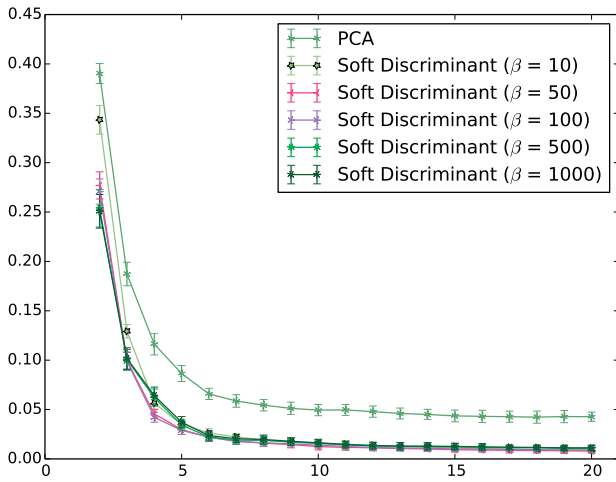
Figure A.19.: Classification error rates averaged over 12 repeats of 2-fold cross-validation of the nearest neighbour classifier, on the pre-whitened FVC1 fingerprint dataset.



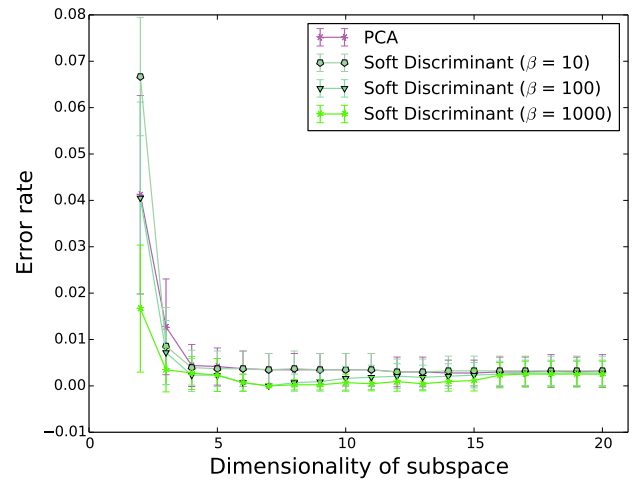
(a) Essex94



(b) Essex95



(c) Essex96



(d) Essex Grimace

Figure A.20.: Classification error rates for the four Essex face datasets.

| Dataset name  | N    | K   | D      |
|---------------|------|-----|--------|
| AT&T          | 400  | 40  | 10304  |
| Essex94       | 3040 | 152 | 36000  |
| Essex95       | 1440 | 72  | 36000  |
| Essex96       | 3018 | 151 | 38416  |
| Essex Grimace | 360  | 18  | 36000  |
| JAFFE         | 213  | 10  | 65536  |
| FVC1          | 80   | 10  | 90000  |
| FVC2          | 80   | 10  | 93184  |
| FVC3          | 80   | 10  | 214144 |
| FVC4          | 80   | 10  | 76800  |
| Sheffield     | 565  | 20  | 10304  |

Table A.2.: High-dimensional datasets and their characteristics.  $N$  = total number of data points;  $K$  = number of classes;  $D$  = input dimensionality.

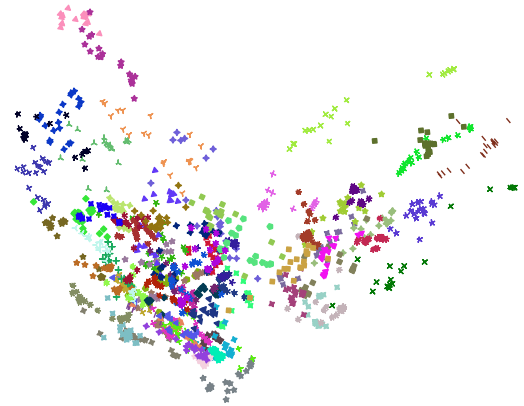
2D projections of the Essex95 dataset computed by SDMs for various values of  $\beta$ . We can see how the classes become more compact in the 2D feature subspace as the value of  $\beta$  increases. Figure A.20b suggests that the best value for  $\beta$  may be around 10, and as the value of  $\beta$  increases so does the error rate. This is the gradually increasing effect of over-fitting as the level of class-discrimination increases. However, when the level of class-discrimination is too low, as is the case for PCA, we obtain high classification error again. The Essex95 dataset seems quite sensitive to the value of  $\beta$ , unlike the other datasets. The Essex96 dataset, for example, shows little difference in classification error rate for different values of  $\beta$ . The 2D projections of the other Essex datasets are completely analogous to figure A.21, and so are not shown here.

Next we study the four FVC datasets (Fingerprint Verification Competition) [11]. Note that this is a subset of the full database. Only this subset is available from the website [11], and not the full database. Figure A.22 shows the classification results. Note that there are classification algorithms and suites (feature extractor plus classifier) far better designed for the specific task of recognising finger prints based on these training datasets. Our focus here is not to design the best classifier specifically for this data, but rather to illustrate the breadth of applications and performance of SDMs as a feature extractor. We can see that all of these datasets are relatively insensitive to the value of  $\beta$ , quite like the AT&T Dataset of Faces (figure 3.7) and unlike the Essex95 dataset (figure A.20b). Also notice that the performance of both PCA and the SDMs are quite bad for the FVC4 dataset, though note that this is a synthetic dataset. Figure A.23 shows the 2D projections of the FVC1 dataset. Again, the 2D projections for the other three FVC datasets are completely analogous and so only FVC1 is shown here.

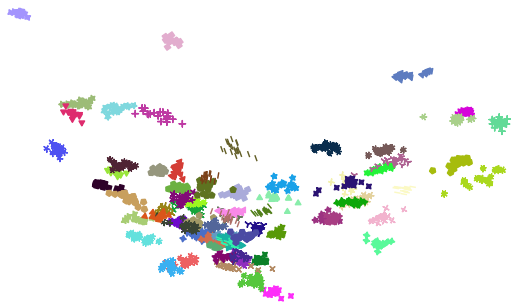
Classification results and 2D projections of the Sheffield face dataset (formerly UMIST) [42] are shown in figures A.24 and A.25. In the context of face recognition, this is an interesting dataset because each class (corresponding to each individual) contains several angles of view of an individual's face, varying gradually from profile to frontal. This rotation in our physical world would translate to a one-dimensional manifold (a curve) in the input space, where the



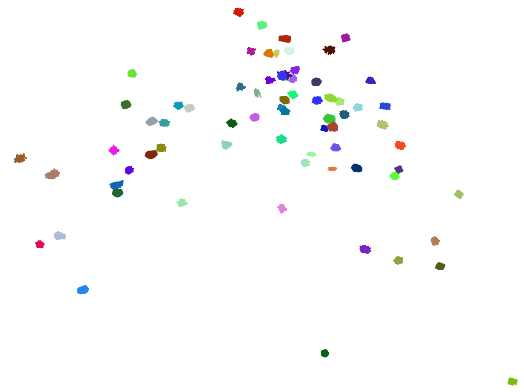
(a) PCA



(b) SDM,  $\beta = 10$

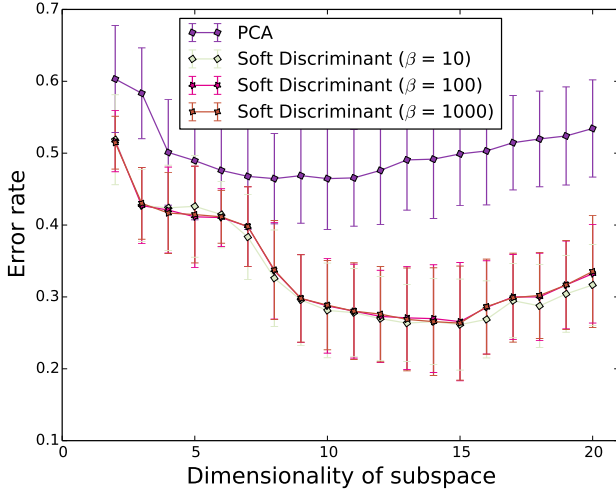


(c) SDM,  $\beta = 100$

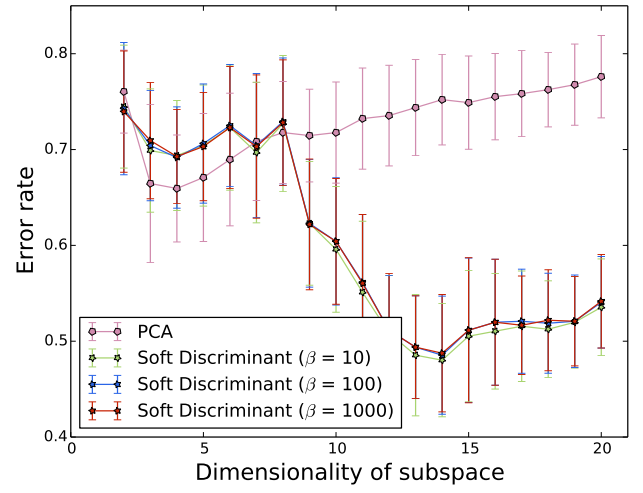


(d) SDM,  $\beta = 1000$

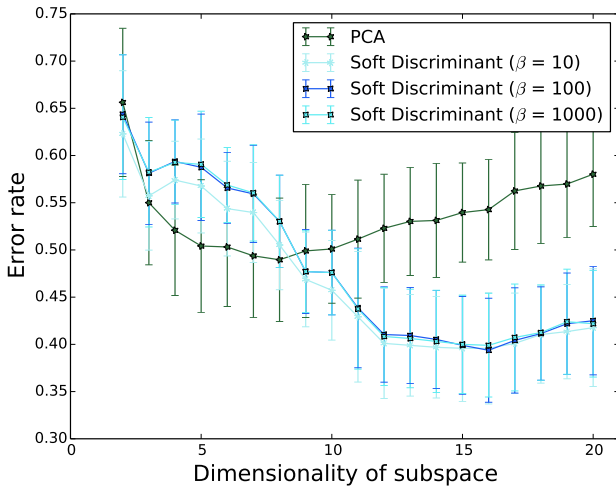
Figure A.21.: 2D projections of the Essex95 face dataset. Notice the inverse relationship between  $\beta$  and the level of class-discrimination in the projection.



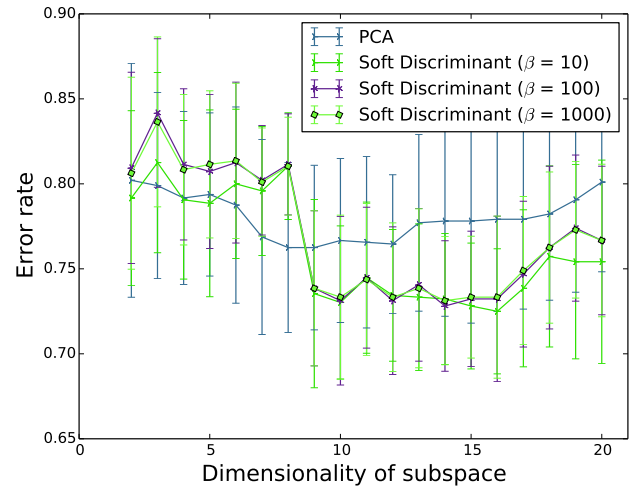
(a) FVC1



(b) FVC2

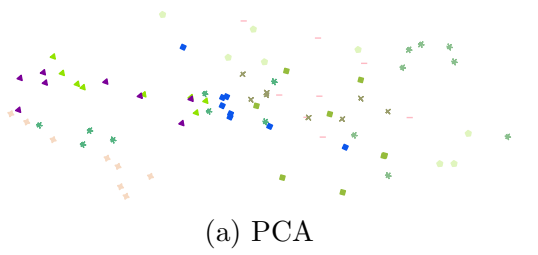


(c) FVC3

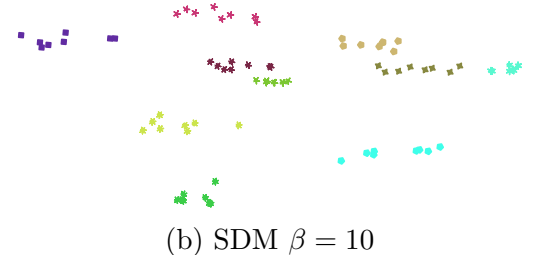


(d) FVC4

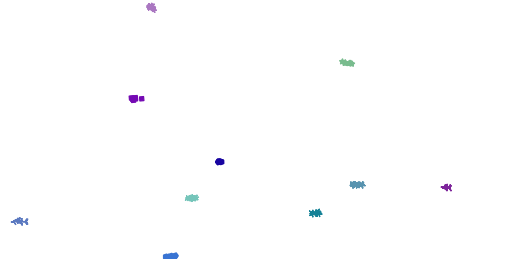
Figure A.22.: Classification error rates for the four FVC fingerprint datasets.



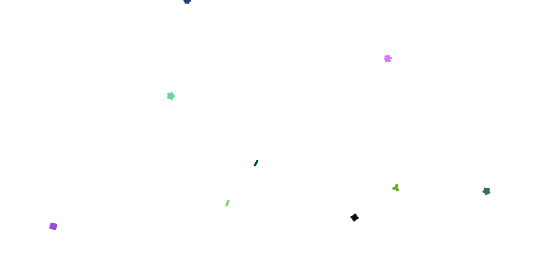
(a) PCA



(b) SDM  $\beta = 10$



(c) SDM  $\beta = 100$



(d) SDM  $\beta = 1000$

Figure A.23.: 2D projections of the FVC1 dataset computed by SDMs for various values of  $\beta$ . Notice the inverse relationship between  $\beta$  and the level of class-discrimination in the projection.

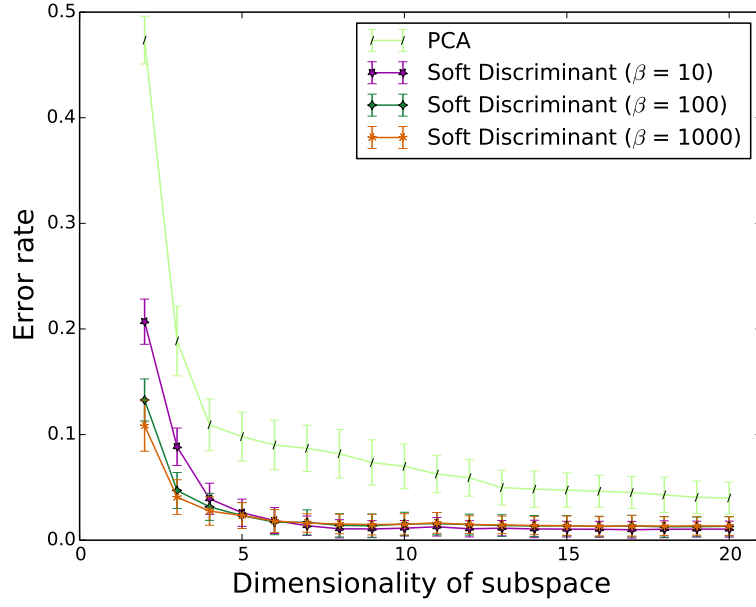


Figure A.24.: Classification error rates for the Sheffield dataset.

one dimension corresponds to the one degree of physical freedom in the rotation of the head. We can verify this by looking at figure A.25a, where we can see the intuitively satisfying pattern that the shape of each class seems to be of a one-dimensional curve.

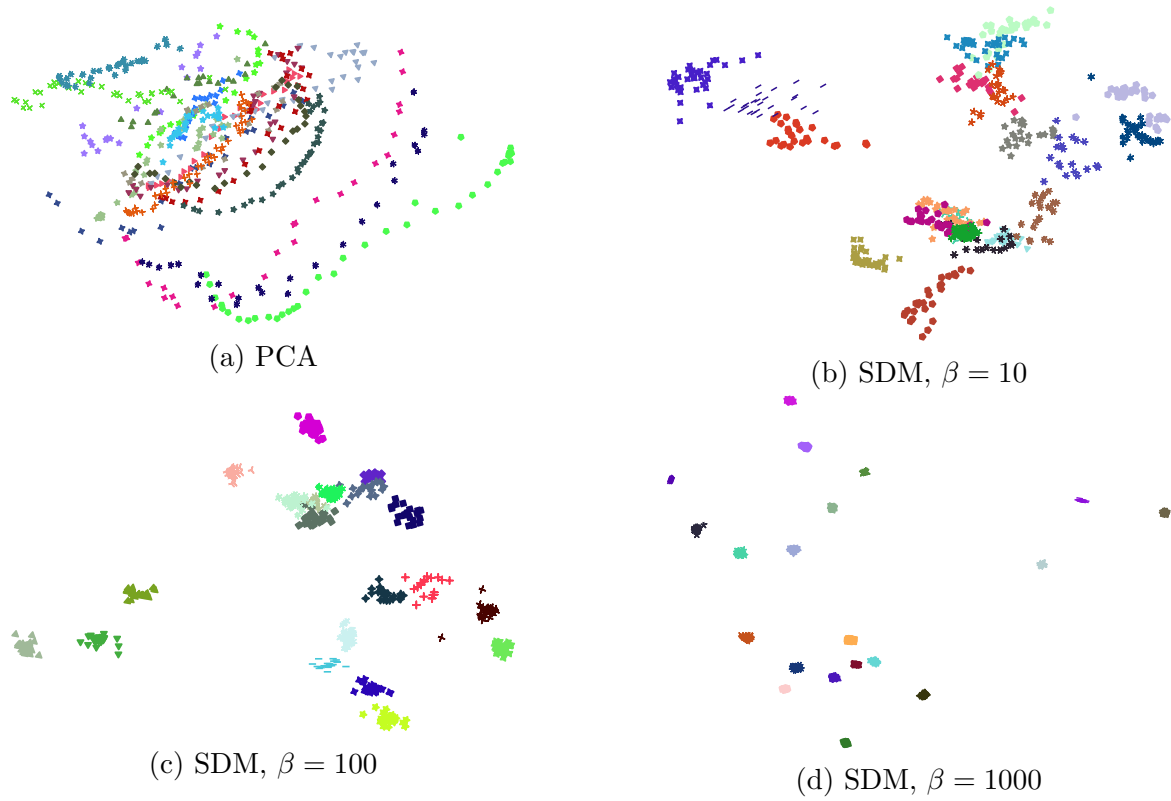


Figure A.25.: 2D projections of the Sheffield dataset, computed by SDMs for various values of  $\beta$ .

### A.2.1. Relationship between $\beta$ and class-discrimination

In section 3.3.3, and in particular in figure 3.9, we observed that the value of  $\beta$  in SDMs seems to be linearly related to the level of class-discrimination in the desired feature subspace, as measured by the  $\Delta$ -measure defined in equation 3.3. This relationship is *stochastically* linear, however the deviation from linearity seems to be negligibly small, as can be seen in figure 3.9. In this section we present some more plots on different datasets, similar to figure 3.9. These plots (figure A.26) look extremely similar to each other, and therefore not overwhelmingly fascinating in themselves, but they serve as experimental evidence for the observation that  $\beta$  is linearly related to the  $\Delta$ -measure.

The corresponding plots for the Essex94 and Essex96 datasets are very similar to those for Essex Grimace and Essex95, and so are not included. Similarly, the corresponding plots for FVC2, FVC3 and FVC4 are almost identical to that for FVC1, and so are not included. We note that in the Essex and Sheffield datasets, the deviation from linearity is more severe at relatively low values of  $\beta$ , where  $\Delta$  seems to be sub-linear with  $\beta$ . We have found in our experiments that this tends to happen for relatively large datasets at small values of  $\beta$ . However, the over-all pattern of linearity is clear. In any case, whether the relationship between  $\beta$  and  $\Delta$  are *precisely linear* is less relevant than the observation that their relationship does indeed seem to be *quite precise*, which means that we can usually find near-optimal and reliable values for  $\beta$  through experimentation.

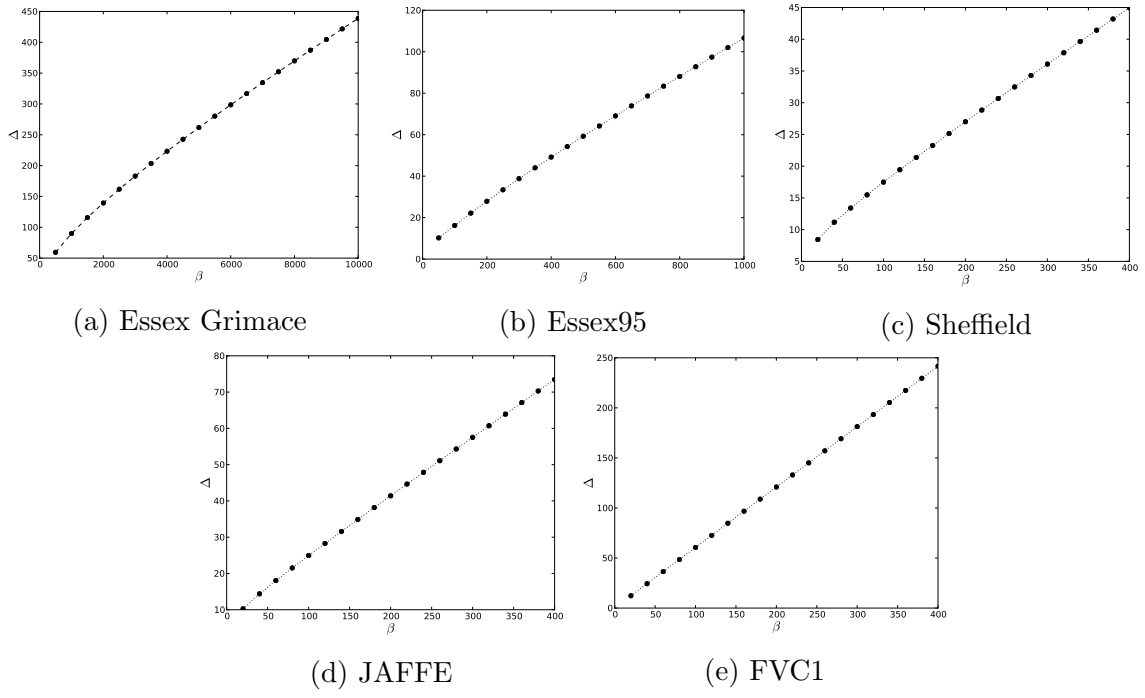


Figure A.26.: Plots of  $\beta$  against the  $\Delta$ -measure in 5-dimensional feature subspaces computed by SDMs for various values of  $\beta$ . Feature subspaces with other dimensionalities show the same pattern.

# Bibliography

- [1] K. Bache and M. Lichman. *UCI Machine Learning Repository*. 2013. URL: <http://archive.ics.uci.edu/ml>.
- [2] M. Belkin and P. Niyogi. “Laplacian Eigenmaps for dimensionality reduction and data representation”. In: *Neural computation* 15.6 (2003), pp. 1373–1396.
- [3] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006. ISBN: 0387310738.
- [4] Béla Bollobás. *Modern graph theory*. Vol. 184. Springer Verlag, 1998.
- [5] L. Breiman et al. *Classification and decision trees*. Wadsworth and Brooks, 1984.
- [6] C.J.C. Burges. “A tutorial on support vector machines for pattern recognition”. In: *Data mining and knowledge discovery* 2.2 (1998), pp. 121–167. ISSN: 1384-5810.
- [7] T.M. Cover and J.A. Thomas. *Elements of information theory*. Wiley-interscience, 2006.
- [8] Inderjit S Dhillon, Subramanyam Mallela, and Rahul Kumar. “A divisive information theoretic feature clustering algorithm for text classification”. In: *The Journal of Machine Learning Research* 3 (2003), pp. 1265–1287.
- [9] Chris Ding and Hanchuan Peng. “Minimum redundancy feature selection from microarray gene expression data”. In: *Journal of bioinformatics and computational biology* 3.02 (2005), pp. 185–205.
- [10] David L Donoho and Carrie Grimes. “Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data”. In: *Proceedings of the National Academy of Sciences* 100.10 (2003), pp. 5591–5596.
- [11] *Fingerprint Verification Competition Database*. <http://bias.csr.unibo.it/fvc2000/download.asp>. Accessed: 12 June 2013.
- [12] George Forman. “An extensive empirical study of feature selection metrics for text classification”. In: *The Journal of machine learning research* 3 (2003), pp. 1289–1305.
- [13] Keinosuke Fukunaga. *Introduction to statistical pattern recognition*. Academic press, 1990.
- [14] I. Guyon et al. *Feature extraction: foundations and applications*. Vol. 207. Springer, 2006.
- [15] Isabelle Guyon and André Elisseeff. “An introduction to variable and feature selection”. In: *J. Mach. Learn. Res.* 3 (2003), pp. 1157–1182. ISSN: 1532-4435.

- [16] A.K. Jain, R.P.W. Duin, and Jianchang Mao. “Statistical pattern recognition: a review”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 22.1 (Jan. 2000), pp. 4–37. ISSN: 0162-8828.
- [17] J.N. Kapur. *Measures of information and their applications*. Wiley New Delhi, India, 1994.
- [18] Ron Kohavi and George H John. “Wrappers for feature subset selection”. In: *Artificial intelligence* 97.1 (1997), pp. 273–324.
- [19] J.M. Leiva-Murillo and A. Artes-Rodriguez. “Maximization of Mutual Information for Supervised Linear Feature Extraction”. In: *Neural Networks, IEEE Transactions on* 18.5 (2007), pp. 1433–1441. ISSN: 1045-9227.
- [20] Haifeng Li, Tao Jiang, and Keshu Zhang. “Efficient and robust feature extraction by maximum margin criterion”. In: *Neural Networks, IEEE Transactions on* 17.1 (2006), pp. 157–165.
- [21] S.Z. Li and A.K. Jain. *Handbook of face recognition*. Springer, 2011.
- [22] Raymond Liu and Duncan F. Gillies. “An eigenvalue-problem formulation for non-parametric mutual information maximisation for linear dimensionality reduction”. In: *International conference on image processing, computer vision, and pattern recognition*. Vol. 2. 2012, pp. 905–910.
- [23] Raymond Liu and Duncan F. Gillies. “An Estimate of Mutual Information that Permits Closed-Form Optimisation”. In: *Entropy* 15.5 (2013), pp. 1690–1704. ISSN: 1099-4300. DOI: 10.3390/e15051690. URL: <http://www.mdpi.com/1099-4300/15/5/1690>.
- [24] Raymond Liu and Duncan F. Gillies. “Feature selection using order statistics”. In: *Proceedings of the Eleventh International Conference on Pattern Recognition and Information Processing*. 2011.
- [25] Marco Loog, Robert P. W. Duin, and Reinhold Haeb-Umbach. “Multiclass linear dimension reduction by weighted pairwise Fisher criteria”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 23.7 (2001), pp. 762–766.
- [26] Michael Lyons et al. “Coding facial expressions with gabor wavelets”. In: *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*. IEEE. 1998, pp. 200–205.
- [27] LJP Van der Maaten, EO Postma, and HJ Van den Herik. “Dimensionality reduction: A comparative review”. In: *Journal of Machine Learning Research* 10 (2009), pp. 1–41.
- [28] Thomas Mitchell. *Machine Learning*. McGraw-Hill, 1997. ISBN: 0071154671.
- [29] V.D.M. Nhat and S.Y. Lee. “Two-dimensional weighted PCA algorithm for face recognition”. In: *Computational Intelligence in Robotics and Automation, 2005. CIRA 2005. Proceedings. 2005 IEEE International Symposium on*. IEEE. 2005, pp. 219–223.

- [30] Umut Ozertem, Deniz Erdogmus, and Robert Jenssen. “Spectral feature projections that maximize Shannon mutual information with class labels”. In: *Pattern Recognition* 39.7 (2006), pp. 1241–1252. ISSN: 0031-3203.
- [31] Emanuel Parzen. “On Estimation of a Probability Density Function and Mode”. In: *The Annals of Mathematical Statistics* 33.3 (1962), pp. 1065–1076. ISSN: 00034851.
- [32] Karl Pearson. “Principal components analysis”. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 6.2 (1901), p. 559.
- [33] Jaakko Peltonen and Samuel Kaski. “Discriminative components of data”. In: *Neural Networks, IEEE Transactions on* 16.1 (2005), pp. 68–83.
- [34] Xipeng Qiu and Lide Wu. “Info-margin maximization for feature extraction”. In: *Pattern Recognition Letters* 30.16 (2009), pp. 1516–1522. ISSN: 0167-8655.
- [35] Singiresu S Rao. *Engineering optimization: theory and practice*. Wiley, 2009.
- [36] A. Renyi. “On measures of entropy and information”. In: vol. 1. University of California Press, 1961, pp. 547–561.
- [37] Irina Rish. “An empirical study of the naive Bayes classifier”. In: *IJCAI 2001 workshop on empirical methods in artificial intelligence*. Vol. 3. 22. 2001, pp. 41–46.
- [38] Sam T Roweis and Lawrence K Saul. “Nonlinear dimensionality reduction by locally linear embedding”. In: *Science* 290.5500 (2000), pp. 2323–2326.
- [39] Yvan Saeys, Iñaki Inza, and Pedro Larrañaga. “A review of feature selection techniques in bioinformatics”. In: *bioinformatics* 23.19 (2007), pp. 2507–2517.
- [40] C. Shannon. “A mathematical theory of communication”. In: *The Bell System Technical Journal* (1948).
- [41] Tatyana Sharpee, Nicole C Rust, and William Bialek. “Analyzing neural responses to natural signals: maximally informative dimensions”. In: *Neural computation* 16.2 (2004), pp. 223–250.
- [42] *Sheffield Face Database*. <http://www.sheffield.ac.uk/eee/research/iel/research/face>. Accessed: 12 June 2013.
- [43] B. W. Silverman. *Density estimation for statistics and data analysis*. Chapman and Hall, 1998. ISBN: 0412246201.
- [44] D. Skočaj, A. Leonardis, and H. Bischof. “Weighted and robust learning of subspace representations”. In: *Pattern recognition* 40.5 (2007), pp. 1556–1569.
- [45] Donna K Slonim et al. “Class prediction and discovery using gene expression data”. In: *Proceedings of the fourth annual international conference on Computational molecular biology*. ACM. 2000, pp. 263–272.
- [46] Taiji Suzuki and Masashi Sugiyama. “Sufficient dimension reduction via squared-loss mutual information estimation”. In: *Neural computation* 25.3 (2013), pp. 725–758.

- [47] Joshua B Tenenbaum, Vin De Silva, and John C Langford. “A global geometric framework for nonlinear dimensionality reduction”. In: *Science* 290.5500 (2000), pp. 2319–2323.
- [48] *The Essex face Database*. <http://cswww.essex.ac.uk/mv/allfaces/>. Accessed: 11 June 2013.
- [49] Michael E Tipping. “Sparse Bayesian learning and the relevance vector machine”. In: *The Journal of Machine Learning Research* 1 (2001), pp. 211–244.
- [50] Kari Torkkola. “Feature extraction by non parametric mutual information maximization”. In: *Journal of Machine Learning Research* 3 (2003), pp. 1415–1438. ISSN: 1532-4435.
- [51] GV Trunk. “A problem of dimensionality: A simple example”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 3 (1979), pp. 306–307.
- [52] Ulrike Von Luxburg. “A tutorial on spectral clustering”. In: *Statistics and computing* 17.4 (2007), pp. 395–416.
- [53] Jian Yang and Jing-yu Yang. “Why can LDA be performed in PCA transformed space?”. In: *Pattern Recognition* 36.2 (Feb. 2003), pp. 563–566. ISSN: 00313203.
- [54] H.H. Yue and M. Tomoyasu. “Weighted principal component analysis and its applications to improve FDC performance”. In: *Decision and Control, 2004. CDC. 43rd IEEE Conference on*. Vol. 4. IEEE. 2004, pp. 4262–4267.
- [55] Tianhao Zhang et al. “Patch alignment for dimensionality reduction”. In: *Knowledge and Data Engineering, IEEE Transactions on* 21.9 (2009), pp. 1299–1313.
- [56] Zhenyue Zhang and Hongyuan Zha. “Principal manifolds and nonlinear dimension reduction via local tangent space alignment”. In: *arXiv preprint cs/0212008* (2002).
- [57] Deli Zhao, Zhouchen Lin, and Xiaoou Tang. “Laplacian PCA and its applications”. In: *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE. 2007, pp. 1–8.
- [58] W. Zhao et al. “Face recognition: A literature survey”. In: *ACM Computing Surveys (CSUR)* 35.4 (2003), pp. 399–458.