University of London
Imperial College of Science, Technology and Medicine
Department of Computing

Estimating Gene Interactions Using Information Theoretic Functionals

Georgia Chan

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Computing of the University of London and
the Diploma of Imperial College, December 2008

Abstract

With an abundance of data resulting from high-throughput technologies, like DNA microarrays, a race has been on the last few years, to determine the structures and functions of genes and their products, the proteins. Inference of gene interactions, lies in the core of these efforts.

In all this activity, three important research issues have emerged. First, in much of the current literature on gene regulatory networks, dependencies among variables in our case genes - are assumed to be linear in nature, when in fact, in real-life scenarios this is seldom the case. This disagreement leads to systematic deviation and biased evaluation. Secondly, although the problem of undersampling, features in every piece of work as one of the major causes for poor results, in practice it is overlooked and rarely addressed explicitly. Finally, inference of network structures, although based on rigid mathematical foundations and computational optimizations, often displays poor fitness values and biologically unrealistic link structures, due - to a large extend - to the discovery of pairwise only interactions.

In our search for robust, nonlinear measures of dependency, we advocate that mutual information and related information theoretic functionals (conditional mutual information, total correlation) are possibly the most suitable candidates to capture both linear and nonlinear interactions between variables, and resolve higher order dependencies.

To address these issues, we researched and implemented under a common framework, a selection nonparametric estimators of mutual information for continuous variables. The focus of their assessment was, their robustness to the limited sample sizes and their expansibility to higher dimensions - important for the detection of more complex interaction structures. Two different assessment scenaria were performed, one with simulated data and one with bootstrapping the estimators in state-of-the-art network inference algorithms and monitor their predictive power and sensitivity. The tests revealed that, in small sample size regimes, there is a significant difference in the performance of different estimators, and naive methods such as uniform binning, gave consistently poor results compared with more sophisticated methods.

Finally, a custom, modular mechanism is proposed, for the inference of gene interactions, targeting the identification of some of the most common substructures in genetic networks, that we believe will help improve accuracy and predictability scores.

Acknowledgements

I would like to express a deep feeling of gratitude and appreciation, to my supervisor Prof. Duncan F. Gillies. His insights into various aspects of my thesis, his patient direction, editing and mentoring advice, have made writing this dissertation one of the most valuable experiences. I am very grateful for having had this opportunity to work with him.

I would particularly like to thank Prof. Michael Stumpf, supervisor to my transfer and now my employer, for his insightful feedback and advice and trusting me with a Postdoc.

I am also grateful to Polyxeni-Margarita Kleniati and George Tzallas-Regas for their technical advice on writing the dissertation, the staff of the Department of Computing for the administrative support, my friends and colleagues in the College, especially David J. Thornley, for the stimulating discussions and advice.

On a personal note, I am greatly indebted to my parents, for their unconditional love and support in every possible way and last but not least, I would like to thank from the bottom of my heart, my husband and my sister for their enduring love, encouragement, understanding and support all these years. Nothing would be the same without them.

Dedication

To my dad and Panteleimon.

‘Just as the constant increase of entropy is the basic law of the universe, so it is the basic law of life to be ever more highly structured and to struggle against entropy. ’

Vaclav Havel

Contents

Abstract **i**

Acknowledgements **iii**

1 Introduction **1**

1.1 Observation & Measurement 1

1.1.1 Gene Expression In a Nutshell 1

1.1.2 DNA Microarrays in a Nutshell 3

1.2 Analysis & Modelling 4

1.2.1 Dependency Models 6

1.3 Known Issues & Motivation 9

1.4 Scope & Objectives 11

1.5 Thesis Outline 14

2 Information Theory **16**

2.1 Basics of Information Theory 17

2.1.1 Entropy 18

2.1.2 Mutual Information 21

2.1.3 Multivariate Mutual Information 24

2.1.4 Information Theory-Related Concepts 28

2.2 Previous Work 32

2.2.1 Reveal 32

2.2.2 Relevance Networks 32

2.2.3 ARACNe 33

2.2.4 Information Theory and K2 35

2.2.5 CLR 36

2.2.6 MRNET 38

3 Estimation of Information Theoretic Functionals 39

3.1 Introduction 39

3.2 Estimators 42

3.2.1 Estimates of MI based on partitioning of the observation space 42

3.2.2 k-Nearest Neighbours 47

3.2.3 Kernel Density Estimators 51

4 Parametric Simulations 57

4.1 Mutual Information 63

4.1.1 Gaussian Distribution 63

4.1.2 Logistic Distribution 71

4.2 Higher Order Mutual Information 77

4.2.1 Gaussian Distribution 77

4.2.2 Logistic Distribution 85

4.3 Comparison of all the methods . . . 94

4.3.1 Discussion . . . 100

5 Synthetic Networks 101

5.1 Benchmarking Strategy . . . 102

5.2 Synthetic Data Generation . . . 103

5.2.1 Known Generators . . . 105

5.3 Mutual Information Matrix Estimation . . . 107

5.4 Network Inference . . . 108

5.5 Network Validation . . . 109

5.5.1 Quality Measures . . . 110

5.6 Empirical Validation - Methodology . . . 112

5.7 Empirical Validation - Results . . . 113

5.7.1 Conclusions . . . 120

6 Improving the Inference of Structure of Gene Interactions 125

6.1 Structure Learning Methodology . . . 126

6.2 Estimation of Conditional Mutual Information . . . 129

6.3 Towards a Complete Picture . . . 131

7 Conclusions 136

7.1 Discussion & Summary of Results . . . 136

7.2 Future Work . . . 139

7.3 Final Remarks . . . 140

Bibliography **140**

List of Tables

4.1 KI-GS-Optimal Number of Neighbours 80

4.2 KII-GS-Optimal Number of Neighbours 81

4.3 SPL-LOG-Optimal Number of Neighbours 91

5.1 Confusion Matrix 109

5.2 Standard quality measures for network validation 110

5.3 Stages and Options for Empirical Validation 112

5.4 Summary of Results 120

List of Figures

1.1 Gene Expression in Eukaryotes - Image from *http://creationwiki.org* 2

2.1 Relationship between $H(X), H(Y), H(X,Y), H(X|Y)H(Y|X)$ and $I(X;Y)$ - Image from the book "Information Theory, Inference, and Learning Algorithms" *[Mac03]* 22

2.2 Relationship between $H(X), H(X,Y), H(X|Y), I(X;Y)$ and $I(X;Y|Z)$ - Image from the book "Information Theory, Inference, and Learning Algorithms" *[Mac03]* 24

2.3 Schema for the background correction estimation of CLR. The z-score of each regulatory interaction depends on the distribution of MI scores for all possible regulators of the target gene (z_i) and on the distribution of MI scores for all possible targets of the regulator gene (z_j) [FHT+07] 37

3.1 Two steps in the sequence of partitions. The shaded element of G2 is named (0,3) in the tree notation and the shaded element of G3 is named (0,3,1) [FS86] 45

3.2 An example of a partitioning that can arise from a recursion step by [Dar98]. . . 46

3.3 Panel (a): Determination of $\epsilon(i), n_x(i)$ and $n_y(i)$ in the first algorithm, for $k = 1$ and some fixed i. In this example, $n_x(i) = 5$, $n_y(i) = 3$. Panels (b), (c): Determination of $\epsilon_x(i)$,$\epsilon_y(i)$, $n_x(i)$ and $n_y(i)$ in the second algorithm for $k = 2$. Panel (b) shows a case in which $\epsilon_x(i)$ and $\epsilon_y(i)$ are determined by the same point, while panel (c) shows a case where they are determined by different points [KSG04]. 49

3.4 Schematic representation of the indicator function that assigns points to bins, for standard binning (left) $k = 1$ and B-spline approach (right) $k = 3$. The weight of each data point to each of the bins is given by the intersection with the respective graphs of the B-spline function with the data point. By definition, all weights contributing to one data point (variable a) sum up to unity [DSSK04]. . 54

4.1 Experimental setup. 62

4.2 [GS/MI/AP] Errors across sample size for different ρ. 64

4.3 [GS/MI/KI-nn] Effect of the number of neighbours for different ρ. 65

4.4 [GS/MI/KII-nn] Effect of the number of neighbours for different ρ. 66

4.5 [GS/MI/Kraskov] Error values across sample size for different ρ. 68

4.6 [GS/MI/KDE] Effect of Spline Order 69

4.7 [GS/MI/KDE-BIN] Effect of Bin Size 70

4.8 [GS/MI/KDE] Error Values across sample size for different ρ. 70

4.9 [LOG/MI/AP] Errors across sample size for different θ. 72

4.10 [LOG/MI/KNN] Effect of the number of neighbours for different θ. 73

4.11 [LOG/MI/KI] KI Error values across sample size for different α and θ. 74

4.12 [LOG/MI/KII] KII Error values across sample size for different α and θ. 75

4.13 [LOG/MI/KDE-bin size] Error Values sample size $N = 100$ for different number of bins and θ. 76

4.14 [LOG/MI/KDE] Error Values across sample size θ. 76

4.15 [GS/HOMI/AP] Errors across sample size for different σ. 79

4.16 [GS/HOMI/KI] Effect of number of neighbours for different σ on KI. 80

4.17 [GS/HOMI/KII] Effect of number of neighbours for different σ on KII. 81

4.18 [GS/HOMI/KNN] Error values across sample size for different σ. 82

4.19 [GS/HOMI/KDE-BIN] Errors across sample size for different θ. 84

4.20 [LOG/HOMI/AP] Errors across sample size for different θ. 86

4.21 [LOG/HOMI/KI] Effect of number of neighbours for different θ. 87

4.22 [LOG/HOMI/KII] Effect of number of neighbours for different θ. 88

4.23 [LOG/HOMI/KNN] Error values across sample size for different θ. 89

4.24 [LOG/HOMI/KDE-bin size] Error values for sample sizes $N = 50$ and $N = 100$ across bin size for different θ. 90

4.25 [LOG/HOMI/KDE-Spl Order] Error values for sample sizes $N = 50$ and $N = 100$ across Spline Order for different θ. 92

4.26 [LOG/HOMI/KDE] Error values across sample size for different θ. 93

4.27 [GS-ALL-MI] Standard and systematic errors of all the methods for different ρ, across N. 96

4.28 [LOG-ALL-MI] Error values of all the methods for different θ. 97

4.29 [GS-ALL-HOMI] Error values of all the methods for different σ. 98

4.30 [LOG-ALL-HOMI] Error values of all the methods for different θ. 99

5.1 [FSC-DV-SS] Effect of sample size on F_β-score for DV and different network inference algorithms. 114

5.2 [FSC-KII-SS] Effect of sample size on F_β-score for KII and different network inference algorithms. 115

5.3 [FSC-KDE-BS-MRNET] Effect of bin size on F_β-score for KDE on MRNET. . . 115

5.4 [PR-ALL-30] P-R & F_β-score curves for all three inference algorithms and sample size = 30. 118

5.5 [PR-ALL-20] P-R & F_β-score curves for all three inference algorithms and sample size = 20. 122

5.6 [PR-ALL-10] P-R F_β-score curves for all three inference algorithms and sample size = 10. 123

5.7 [FSC-ALL-SS] Effect of sample size on F_β-score for all the methods and different network inference algorithms. 124

6.1 Examples of regulation structures, where conditional MI can be helpful. Image from [SOMMA02]. 125

6.2 Examples of regulation structures. 126

Chapter 1

Introduction

Gene interactions are the processes life is made of - careful definition, insightfull experimental setting, accurate measurement and finally, correct analysis and modelling are the necessary steps to their understanding. This cycle of theory, experiments and computational modelling, lies at the core of the discipline of functional genomics.

Genome-wide expression analysis aims to elucidate this continuous interplay of genes and gain further insight on their function and organisation structure, with a powerful combination of computational analysis and in vitro experimentation.

Computational analysis of gene-interactions is the broad area of our research.

1.1 Observation & Measurement

1.1.1 Gene Expression In a Nutshell

Gene expression is a two-stage procedure that maps the DNA code of a gene to a protein or some form of RNA. The two stages are *transcription* and *translation*.

During transcription, DNA is copied or transcribed into a virtually identical single-stranded messenger RNA (mRNA). In eukaryotes, the newly transcribed RNA undergoes further mod-

ifications (*post-transcriptional modifications*) and becomes mRNA. This process occurs in the nucleus.

The mRNA is then carried in the cytoplasm, where it is translated into a chain of aminoacids (polypeptide). The chain of aminoacids, will be exposed to a series of further processes (*post-translational modification*) like cleavage, folding, phosphorylation, before it becomes an active protein. For a graphical illustration of gene expression and details on the individual steps, see figure 1.1.

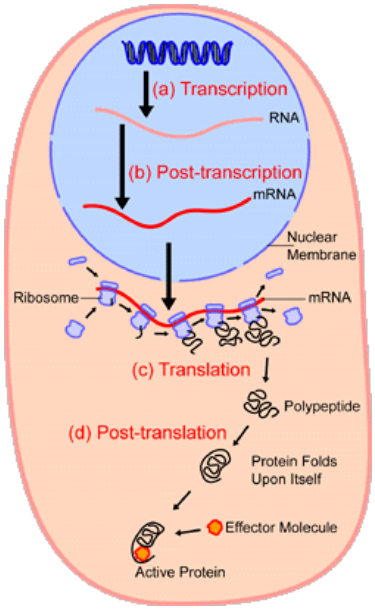


Figure 1.1: Gene Expression in Eukaryotes - Image from *http://creationwiki.org*

The whole gene expression mechanisms is subjected to a number of regulation controls effected by other proteins. For example, the binding of certain proteins, the *transcription factors* onto the *DNA binding sites* 'upstream' of a DNA coding segment (*promoter regions*) has been found to be a key determinant in the process of transcription, either by stimulating or repressing the process of transcription, thus controlling the amount of mRNA produced. Cells also regulate gene expression by post-transcriptional modification; by allowing only a subset of the mRNAs to go on to translation; or by restricting translation of specific mRNAs to only when the product is needed. At other levels, cells regulate gene expression through DNA folding, phosphorylation and intricate "feedback mechanisms" in which some of the gene's own proteins from the

expressed genes, will in turn interfere with the expression of other genes[1], setting off a cascade of changes to many other genes, forming what seems like a complex dependency structure.

The amount of protein that a cell expresses is controlled by the tissue, the developmental stage of the organism and the metabolic or physiologic state of the cell [Ng04] and is the outcome of *gene regulation*. This whole process of expression, forms the backbone of the cell's control over shape and function and is the basis for cellular differentiation, morphogenesis, versatility and adaptability of any organism.

1.1.2 DNA Microarrays in a Nutshell

DNA microarrays offer the technical means to observe and measure the expression profile of thousands of genes, or gene fragments, under different physiological conditions simultaneously.

During the course of a typical microarray experiment, mRNA extracted from a pool of genes, is reverse-transcribed to create either radioactive-or fluorescent labelled cDNA. The labelled cDNA, is then hybridized with a large DNA library of fragments from thousands of genes, that is attached to a glass or membrane support [Sch96] (microarray). The intensity of the fluorescent or radioactive material on each spot, is proportionate to the levels of 'matches' (hybridizations) that took place on each spot, hence the amount of mRNA produced - before transcription - by a specific code fragment[2]. Although the expression of many genes is known to be regulated after transcription (so an increase in mRNA concentration need not always increase expression), microrrays provide an indirect but still invaluable way of assessing gene expression, since transcription factors are themselves proteins.

Therefore, by isolating DNA from different cells, microarrays enable us to measure the relative amounts of mRNA from a multitude of genes in a given cell population in a single assay. Moreover, by controlling the properties of the initial pool of genes from where mRNA is extracted, this whole experimentation process can enumerate genes whose expression alters under different

[1] The process known as *epistasis*

[2] A technique to quantify the cellular concentration of mRNA after transcription is real-time polymerase chain reaction(RT-PCR) [Sch96].

environmental stimuli or stresses at a system-wide level.

This is great news for the biologist who wants to gain insight into how genes are involved in the biological processes and how their interactions are conducted and regulated.

In fact, since its first appearance in 1995, microarray technology revolutionised the course of scientific penetration, to areas that were previously impossible to tackle and became the major experimental instrument of the area of expression profiling in functional genomics.

1.2 Analysis & Modelling

The use of microarrays as an expression profiling technique, inspired a whole new range of exploratory investigation methodologies in functional genomics (static experiments, time-course experiments, knock-out experiments), each with its own analysis requirements. Moreover, as a high-throughput technology, it introduced a new data paradigm with thousands of variables (genes) and a multitude of parameters representing biological factors, experimental conditions etc.

To address the issues of gene interaction and regulation - lacking substantial requisite background biological knowledge - in a fully comprehensive way, we need to resolve to mathematical abstractions and statistical modelling. Probability and statistics are necessary to the analysis of gene expression, as they can model our incomplete picture of the reality [Jak03] and abstract away the 'noise' inherent in such a multi-step procedure. Furthermore, this approach facilitates the application of well-known modelling and computational analysis strategies, that have been successfully employed in other areas.

The domain of modelling gene expression requires a set of key assumptions. First, a gene can be represented as a continuous random variable, with its corresponding expression levels following a certain (unknown) probability distribution. Suitable abstractions and their mathematical representations need also to be decided upon for the concepts of *interaction* and *dependence*.

In systems' biology, there is still no consensus on what dependence or interaction is, in a mul-

tivariate setting[3]. Instead, the field accepts the mandates of data mining techniques handling the computational analysis of gene expression data to define these concepts.

Statistical dependence in a mathematical context, describes the strength of the relationship between a dependent variable and one or more explanatory (predictor) variables. Suitable actors for these roles in a biological context are genes, proteins, fragments of proteins like binding factors, functional gene fragments like operons, etc. For the purposes of our study, we will focus on interactions between genes.

Statistical interaction is also a type of dependence, but without specifying the roles of the implicated variables (e.g. parent-child, regulator-regulatee, etc). The quantitative criterion that determines if two variables interact is called the *similarity* or *dependency metric*, which is essentially a function that maps pairs or groups of variables to real numbers, measuring the strength of the association.

To complete the modelling path, after deciding upon the concept and the metric for dependence and interaction, we can represent regulation structures as probabilistic networks, where genes act as nodes and dependencies/interactions between genes would form the edges that connect them. Thus, an inferred genetic network contains information about the regulation structure to which a gene belongs and which genes it interacts with.

A multitude of dependence models (concepts and metrics) exist to date in system's biology literature, some are generic and others applicable in restricted contexts only [NBdRvS04]. A brief taxonomy is attempted in the discussion that follows.

[3]Currently there are separate concepts of competitive and cooperative, direct and indirect, joint regulations by common factors and conditional independence etc. [WAG+06].

1.2.1 Dependency Models

Dependency as (Linear) Correlation

With data from expression profiling techniques, like microarrays, it is natural to talk about how genes are (dis)similar to one another with respect to their expression levels, sampled under different conditions, individuals, or even time. According to the context of the experiment, similarity can be interpreted either as interaction or as dependence.

The most widespread model of similarity, especially during the early stages of expression profiling, is linear correlation. As the name suggests, linear correlation measures the linear relationship between variables.

For a pair of continuous variables X and Y, correlation is defined using Pearson's correlation coefficient:

$$r_{X,Y} = \frac{covariance(X,Y)}{standard_deviation(X) * standard_deviation(Y)} \tag{1.1}$$

The value of correlation ranges from -1 (perfect negative correlation) to 1 (perfect positive correlation) and is symmetric - i.e. $r_{X,Y} = r_{Y,X}$. Zero correlation means that there is no linear relationship between the variables.

Pair-wise correlation does not model interactions where the resulting expression-level of a gene is the result of combined action of multiple other genes, intrinsically. For the inference of regulatory structures, with the correlation values for all pairs of variables, one possibility is, to construct a distance matrix that holds the similarity of each pair of genes to all other genes. Single linkage hierarchical clustering could then be employed, to find a single linked tree that connects associated genes. This approach was taken by Arkin and Ross in [AR94]. Another approach can be found in [SS05a], where improved (regularized) small-sample point estimates of partial correlation are introduced, to help differentiate between direct and indirect interactions among the variables and construct graphical Gaussian models (GGM) of the inferred dependencies.

Besides the lack of modelling for multiple inputs (combined effect), the restriction to linear

correlations is another severe limitation, since the existence of non-linear relationships in real biological networks is almost a fact (e.g. in positive feedback loops). Both these restrictions, render correlation-based dependency models unfavourable candidates for gene expression analysis.

Dependency in Bayesian Networks Context

In Bayesian Networks [FLNP00], dependency is viewed as the conditional probability of X given its immediate causes (parents). This is defined as:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | parents(X_i)) \tag{1.2}$$

and describes a causal, directed, hence asymmetric, relationship. This setting defines a unique rule for expanding the joint probability distribution in terms of simpler conditional probabilities; the candidate complete network structures are produced by some resampling technique (usually MCMC) and evaluated by a specified scoring function (e.g. MDL) [FLNP00].

In practice, Bayesian networks are difficult to compute, even for modest size genetic networks and this is partly due to the fact that, inferring the joint probability distribution in order to find a simple property (dependency) is not a good idea [Vap99] and a direct estimation without learning the joint probability distribution is preferred [NBdRvS04].

Other limitations of BNs is that they are poor at representing large number of variables, can not deal with regulatory loops (a limitation that dynamic BNs can overcome) and often require the discretisation of expression data prior to the inference process. Finally, they cannot distinguish between independent and cooperative regulation of a gene by others - a problem with equivalence classes $A \leftarrow C \rightarrow B$ and $A \rightarrow C \rightarrow B$ [WAG+06]. Despite these limitations, the formalism of Bayesian networks can be adapted easily to any concept of multivariate dependence, providing at the same time a rigid framework for noise handling, over-fitting and significance testing and perhaps as more and more data will become available, it will become more readily applicable.

Dependency in Information Theory Context

Information-theoretic approaches typically rely on the estimation of mutual information from expression data in order to measure the statistical dependence between variables.

Mutual information is defined as a functional of the probability distributions of X and Y and their union (X, Y):

$$I(X;Y) = \sum_{x,y} P(x,y) log_2 \frac{P(x,y)}{P(x)P(y)} \tag{1.3}$$

It is symmetric and can only be positive definite in two dimensions. It is zero if and only if $P(x) = P(y)$.

In strict terms mutual information it is not a metric, as it violates the property of triangular inequality[4]. If necessary, however, it is easy to create a metric out of mutual information and the joint entropy of variables X, Y as: $d(X,Y) = H(X,Y) - I(X;Y)$.

Mutual information, essentially quantifies the distance between the joint distribution of X and Y and what the joint distribution would be if X and Y were independent. In terms of information, it represents the difference between the information provided by X and Y taken separately and their joint information, or else how much knowing one of the variables reduces our uncertainty about the other. We note here, that mutual information makes no assumptions about any of the involved distributions.

Mutual information, as a dependency measure, handles successfully the issue of non-linearity, since it estimates both linear as well as non-linear relations between variables. As a genetic network model, it will construct relationships between genes, based solely on pair-wise comparisons - in a fashion similar to linear correlation.

Information theory however, provides other measures based on mutual information, that account for more complex dependencies. Higher order mutual information and conditional mutual information are the best examples and will be analysed more formally in section 2. What is of importance here, is that with higher order mutual information we can expand the analysis

[4] $I(X,Z) \leq I(X,Y) + I(Y,Z)$

beyond pairs of variables, to triplets or quadruples, or any higher-order dependency structure. Moreover, with conditional mutual information, we are able to distinguish between direct and indirect interactions.

Genetic network inference using information theoretic functionals, shows a lot of potential and has rsecently held the attention of the bioinformatics community for the inference of large networks [MKLB07]. An overview of some of the most successful studies on the subject is given in section 2.2.

On the account of the aforementioned conceptual connotations, we believe that information theory functionals - entropy, higher order mutual information, conditional mutual information - are ideal dependency models for the analysis of gene expression data and will form the backbone of our research.

1.3 Known Issues & Motivation

Although microarray technology seems to approach an age of maturity, in terms of conformity with standards and popularity within the molecular biology research community, it still presents huge challenges from the computational analysis front. Many of its inherent properties, if not addressed properly can compromise the quality of the results. These issues are presented below.

Noisy Measurements - High Variability The preparation and execution of a microarray experiment, is a lengthy and complex process involving many steps - growth of cultures, extraction of mRNA, reverse transcription, labelling, hybridisation, scanning, and image analysis. Therefore, as with any experimental technique of this kind, variation and error is introduced in every step and cannot be completely avoided. Replication and averaging, (i.e. *normalization*) are the advised remedies to the problem of estimation and reduction of variability. However, microarrays are still a rather expensive technology and though replicates are highly desirable, they are rarely avaiable. In addition, the lack of biological knowledge about the overall behaviour of a gene (or population statistics), hinders the

application of averaging as well.

Curse of Dimensionality As mentioned above, lack of replication is an invariant problem of microarray experiments. The vast majority of datasets available, measure a very small number of arrays -ranging from 5 to a few tens. Considering the fact, that each microarray contains thousands of genes, traditional statistical analysis is not adequate and more often than not, produces inconclusive results.

In order to reduce the dimensionality of the problem, the most common approach so far, has been to consider that gene expression conforms to a particular (most often linear) distribution. This assumption however, we know by now, does not hold - gene expression is highly skewed, with only a few genes highly expressed among a multitude of genes with low expression levels [ARR+08]. Consequently, it would be a futile effort to proceed with the study of gene interactions using parametric models, unless absolutely certain about their applicability (in limited settings).

The same dichotomy applies to the estimators of mutual information as well. The critical tradeoff there, is that estimates of mutual information that make few or no assumptions concerning the nature of gene expression, are bound to suffer biases caused by insufficient samplings, while methods that reduce the dimensionality of the problem by considering a parametric family, suffer biases because true gene expression does not conform with that distribution [Vic02].

Rather than knowingly ignoring the true nature of the data, our choice in the present work, was to take the challenge and address the problem of small sample size explicitly.

Lack of Background Knowledge A significant issue with expression profiling from measured expression data, is the absence of background knowledge about true models and patterns of interactions.

This deficit affects computational analysis in many levels:

- there is not sufficient biological knowledge to restrict the analysis to a subset of core variables for a given biological process.

- given the multitude of different biological conditions, it is difficult, costly and time-consuming to obtain a complete expression profile and consequently a model of expression, even for a single gene.

- there are no standard models of regulatory mechanisms for genes, except generic cause-effect relationships between transcription factors and corresponding binding sites, so it is very difficult to prove or disprove a modeling approach.

In spite of those limitations, with persistent efforts, a combination of successful methodologies - each in its own limited setting - and systematic integration with different information sources, the research community managed to extract a limited amount of reliable information about gene interactions. This information is often consolidated in prototype genetic networks used to generate synthetic expression data.

As a consequence, when it comes to quantitative assessment of the performance of dependency models and inference strategies, we are left with two options: going down the route of reusing data from previous studies, or the use of simulated and synthetic data via generators inspired from existing ground truths. Using the first route, we seek to affirm the previous findings, whereas for the second route we can straightforwardly calculate appropriate quality measures. In either case, the quality of data and knowledge ultimately ascertains the quality of the results.

In the present work and for the purposes of benchmarking the different mutual information estimators, we followed the route of simulations and synthetic data generation.

1.4 Scope & Objectives

The inference of genetic interactions from microarray expression data, is one of the most challenging, yet rewarding, tasks of modern functional genomics. In view of the limitations presented above, it becomes obvious that to be successful, exhaustive study, intuitive sagacity and technical skill is required.

Especially in the past, researchers in the interest of simplicity and ease of use, or lack of collective knowledge often disregarded the issues presented above - a practice that now tends to be abandoned. Increased awareness and cross-checking with other research areas sharing some of these issues, is slowly bringing the field of microarray data analysis to a stage of maturity. In line with this trend, our main objective behind our work, was to search for methods that are sympathetic to the idiosyncrasies of the data (high variability, lack of backround knowledge) and mainly address the issue of small sample sizes.

Mutual information and relevant information theoretic functionals - entropy, higher order mutual information, conditional mutual information - have the potential to fulfill these requirements, as they can model both linear and non-linear interactions and capture more complex dependency structures. Moreover, they are very useful for knowledge representation and are naturally amenable to inference using graph theoretical arguments.

However, what is great news for the inference formalism, is not so great for the actual estimation. Non-parametric mutual information estimation, is far from trivial and is an active research area on its own. To date, most popular, non-parametric, mutual information estimators, are based on frequency counting, which requires discrete variables. By contrast, gene expression data is continuous, thus estimates of mutual information between genes suffer biases due to discretisation.

In response to these demands, we concentrated our efforts to the investigation of non-paramentric estimators for mutual information and higher order mutual information for continuous variables, and assessment of their applicability in undersampled regimes. A secondary goal of our research was to prove, that there is a gain in performance when we move further from conventional non-parametric estimators of mutual information - i.e. histogram based methods - both in variance and in bias. What is also of importance, is that the gain in performance applies also in higher dimensions and in small sample settings, an affirmation that enables a whole new cycle of experimentation.

One of the applied aspect of our work was to implement or port (wherever an algorithm was available) the different estimators in *R* [R D08] and build a common framework for their em-

pirical evaluation.

Empirical evaluation was split in two parts:(1) statistical assessment (2) integration with existing genetic network inference methods and assessment of the networks produced.

In particular to the first part, the purpose was to scrutinise the working aspects of the estimators and assess their bias and variance via parametric simulations, in realistic sample settings - sample sizes range: $5 \dots 200$. To our knowledge, this is a novel kind of benchmark, since most studies focus their assessment on the asymptotic properties of the estimators, or within a conveniently large data range.

To this end, we expanded the assessment in two additional directions: (a) two different parametric families and (b) separate analysis for the extension of the estimators to higher dimensions. The motivation behind the first direction, was the sparsity of studies on the applicability of non-parametric estimators on distributions other than the normal. This sparsity appears at first as a paradox, since one of the particular merits of mutual information, is its detachment from parametric settings. Particular difficulties entailed in the task, convinced us of the reasons behind it. Nevertheless, we performed the same type of statistical assessment of systematic and standard error of the estimators on data following the logistic distribution as well.

Particular emphasis was also given to the performance of the estimators in higher dimensions. Our motivation this time, comes from the acknowledgement, that gene expression is the product of a complex system of interactions. Hence, a suitable inference method should at least be able to scale to higher dimensions. Besides that, multivariate entropy and multivariate mutual information are essential for the computation of other information theoretic quantities, like conditional mutual information, which we use for the last part of our study.

Moreover, in order to link our findings from the fist part of our empirical evaluation, back again to the task of inference of genetic interactions, we carried out a second type of comparative assessment. For this study, we generated synthetic data from established generators and combined the estimators with network inference methodologies from existing studies on reverse engineering of gene regulatory networks. This experimental setting enabled us to perform a

complete testing of the proposed estimators in a fast and reproducible manner in a realistic environment.

Both parts of empirical evaluation, in conjunction, give a complete and conclusive picture of the application of information theoretic functionals to the task of inference of genetic interactions.

The results inform us, that in undersampled settings, the more sophisticated methods in our benchmark have better statistical qualities - bias and variance - than the naive binning method, that is used in the vast majority of studies. In addition, there is a notable gain in the performance of some well-known network reconstruction algorithms, when in place of their own estimators (which are also based on uniform binning) we use the alternatives, to infer known network structures of synthetic - but realistic - data.

Finally, to address the issue of poor fitness values due to pairwise only interactions, we explore how with the use of information theoretic functionals, we can discern between direct and indirect effects among variables, hence infer some of the more complex depencency architectures, frequently appearing in gene regulatory networks. This knowledge, enables us to improve the predictive power of an inference algorithm, by reducing the number of false positive and false negative edges. To apply these ideas in practice we suggest a new, modular algorithm for the inference of gene interactions, using the estimators reviewed, to compute standard pair-wise mutual information, as well as conditional mutual information among triplets of variables.

1.5 Thesis Outline

The rest of the report proceeds as follows: the background section in chapter **2**, is divided in two parts. The first part, contains all the relevant background on information theory that we are going to use throughout the present work. The second part, presents a brief overview of state-of-the-art methods on reverse engineering of gene regulatory networks, using information theoretic functionals. Chapter **3**, is a review on the existing methods for estimating information theoretic functionals - mutual information in particular - from the perspective of the computational paradigms that they are based on and introduces the list of estimators that appear to fit our

criteria for robustness and nonparametricity and will be assessed in the following sections. In particular, chapter **4** provides a thorough statistical validation of the different estimators, using simulated data, in under-sampled regimes. Chapter **5**, discusses the issue of synthetic data generation and examines how the selected estimators affect the performance of some of the network reconstruction methods presented in the background section. Finally, chapter **6** looks into uncovering complex interaction structures via inequalities based on information-theoretic functionals and introduces a prospective methodology for inference of genetic networks, together with some discussion on the problem of significance testing. Chapter **7** concludes the thesis and points further steps to complete the inference of gene interactions using information theoretic functionals.

Chapter 2

Information Theory

Information theory is a division of applied mathematics, that addresses the issues of representation, estimation, storage and transmission of information. The defining work on the subject, was written by Claude E. Shannon in 1948, in the paper "A mathematical theory of Communication" [Sha48]. The paper opened with the assertion that:

> the fundamental problem of communication, is that of reproducing at one point - either exactly or approximately - a message selected at another point.

Shannon introduced the *source coding theorem* as the solution to the two questions entailed with the problem of communication: (1) what is the ultimate data compression, or in other words, how easily can message data be compressed and (2) what is the ultimate transmission rate over a noisy channel [CT91]. The answer to the first question is *entropy*, and to the latter is *mutual information*.

Entropy, is a term inspired by thermodynamics[1], where it denotes a measure of randomness of molecules in a system. In the context of information theory, it denotes a measure of the amount of information in a random variable. Accordingly, mutual information quantifies the information shared between two variables, hence used to find the transmission rate across a noisy channel. Much of the mathematics behind information theory, is based on developments

[1]Second law of thermodynamics states that entropy always increases.

in the field of thermodynamics, by L. Boltzmann and J.W. Gibbs and further connections between the fields were explored by Landauer in [Lan61].

Historically, the conception of information as a measurable quantity, can be attributed to Ralph Hartley and his 1928 paper *Transmission of Information* [Har28]. Shannon's contribution was essentially the introduction of a combined qualitative and quantitative model of information as a statistical process and the mathematical definition of many of its properties.

Since its formation, the scope of the theory has become considerably broader and diverse. Areas of applications range from communication networks and electrical engineering to neuroscience, statistical physics, computer science, language processing, statistical inference and economics. Moreover, it has served as the conceptual foundation behind many important theories like Kolmogorov's algorithmic complexity (Minimum Description Length), Occam's razor, the theory of optimal investment and error rates for optimal hypothesis testing. For a systematic characterisation and a comprehensive list of applications, see [CT91].

2.1 Basics of Information Theory

As stated above, the fundamental quantities of information theory are entropy and mutual information. However, from these quantities, other important functionals can be derived, that can be used in a variety of problems and the inference of gene interactions, in particular. All of these functionals, can be expressed as functions of the probability distributions of the variables involved. Concise explanations and defining formulae of some of the most appropriate functionals falling within the scope of our research, are presented below. Probability distribution functions - PDFs - will be denoted as $p(x)$ instead of $p_X(x)$, for convenience. All of the definitions can be found in [CT91] and [Mac03].

2.1.1 Entropy

Entropy, in an information theory context, quantifies the uncertainty involved when encountering a random variable.

The entropy of a random variable X with probability mass function $p(x)$ is defined by:

$$H(X) = -\sum_{i=1}^{n} p(x) \log p(x). \tag{2.1}$$

In Shannon's paper, the basis of the logarithm is 2 and the metric is bits. The most common alternative base is Euler's number, the natural logarithm e, where the metric is nats. Any other base can be used as well, when the actual metric is of no particular importance and the entropy can be adjusted by multiplying with the appropriate factor.

The entropy can also be expressed as the expected value of $p(X)$, as in: $H(X) = \mathbb{E}_p \log \frac{1}{p(X)} = -\mathbb{E}_p \log p(X)$. Some of the most important properties of entropy are:

- it is always non-negative,
- it is zero when $p(x) = 0$ or $p(x) = 1$, i.e. when there is no uncertainty about the variable any more,
- conversely, it reaches its maximum value when $p(x) = \frac{1}{2}$.

Differential Entropy

Shannon in his original paper [Sha48], defined another quantity, the differential entropy, to capture the entropy of a continuous random variable.

The differential entropy of a continuous random variable X with probability density function p and support set $\mathbb{S}$, is defined as:

$$H(X) = -\int_{\mathbb{S}} p(x) \log p(x)\, dx. \tag{2.2}$$

Unlike the discrete entropy, differential entropy can take negative values[2] and it is not defined for variables whose PDF integral does not exist[3].

Joint Entropy

Joint entropy is the extension of the definition of entropy to a pair of random variables X and Y, with joint PDF $p(X, Y)$. The same definition would apply, if X was a single vector-valued random variable.

$$H(X, Y) = -\sum_{x,y} p(x, y) \log p(x, y). \tag{2.3}$$

Similarly to the single variable, joint entropy can also be expressed as $H(X, Y) = -\mathbb{E} \log p(X, Y)$. Joint entropy is additive for independent random variables:

$$H(X, Y) = H(X) + H(Y) \text{ iff } P(x, y) = P(x)P(y). \tag{2.4}$$

As in the univariate case, we can extend the definition of differential entropy to a pair, or even set of n continuous random variables.

$$H(X_1, X_2, \ldots, X_n) = -\int f(x_1, x_2, \ldots, x_1) \log f(x_1, x_2, \ldots, x_1)\, dx_1\, dx_2 \ldots dx_n. \tag{2.5}$$

Conditional Entropy

Conditional entropy measures the average uncertainty that remains about X, when Y is known.

Variable Y can either take a particular value, for example $y = b_k$, or we can take the average over all values of Y.

For the first case, the conditional probability distribution takes the form $P(x|y = b_k)$, whereas

[2]Consider a random variable, uniform from 0 to α, so that its density is $1/\alpha$ from 0 to α and 0 elsewhere. Its differential entropy will be $H(X) = -\int_0^\alpha 1/\alpha \log 1/\alpha\, dx = \log \alpha$, hence for $\alpha < 1$ it can be negative. [CT91]

[3]Differential entropy is often denoted as h, to distinguish it from the discrete case

for the latter it is just $P(x|y)$. The corresponding formulae are:

$$H(X|y=b_k) = -\sum_x P(x|y=b_k)\log\frac{1}{P(x|y=b_k)}, \tag{2.6}$$

$$H(X|Y) = -\sum_y p(y)\left[\sum_x p(x|y)\log p(x|y)\right], \tag{2.7}$$

$$= -\sum_{x,y} p(x,y)\log p(x|y). \tag{2.8}$$

In terms of entropies, conditional entropy can be expressed as:

$$H(X|Y) = H(X,Y) - H(Y), \tag{2.9}$$

$$H(Y|X) = H(Y,X) - H(X). \tag{2.10}$$

As above, the definitions above, can be extended to the continuous case, by substituting the sums with integrals.

We note here, that it is possible for $H(X|y=b_k)$ to exceed $H(X)$, but the average on the second variation, $H(X|Y)$, is less than $H(X)$. This in effect means, that data are helpful in order to decrease uncertainty, on average.

Chain Rule for Entropy and Differential Entropy

The joint entropy, conditional entropy and marginal entropy are joined by the following chain rule:

$$H(X,Y) = H(X) + H(X|Y) = H(Y) + H(X|Y), \tag{2.11}$$

which reads, that the joint entropy of a pair of random variables is the entropy of one, plus the conditional entropy with the other variable.

By repeated application of the equation 2.11, we can generalise the entropy chain rule to n

variables.

$$H(X_1, X_2, \ldots, X_n) = \sum_{i=1}^{n} H(X_i|X_{i-1}, \ldots, X_1). \tag{2.12}$$

e.g. for 3 variables:

$$H(X, Y, Z) = H(X) + (H(Y|X) + H(Z|X, Y)$$

2.1.2 Mutual Information

Mutual Information captures the reduction in uncertainty of a random variable, due to another random variable and essentially measures the dependency between the two variables. The general rigorous definition of mutual information for random variables was developed by Kolmogorov and Pinsker.

As an expression of entropy functionals it can be defined as:

$$I(X;Y) \quad = \quad H(X) - H(X|Y), \tag{2.13}$$

using equation 2.11:

$$I(X;Y) \quad = \quad H(X) + H(Y) - H(X,Y),$$

and as a function of PDFs of the variables:

$$I(X;Y) \quad = \quad \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}. \tag{2.14}$$

The same definition as in 2.13, can be extended from probability mass functions to probability density functions, in order to accommodate continuous variables, after substituting sums with integrals. The following properties are common to both cases.

Mutual information has some important properties:

- it is symmetric, i.e. $I(X;Y) = I(Y;X)$.
- it is non-negative, i.e. $I(X,Y) \geq 0$.
- $I(X;Y) = 0$ iff $p(x,y) = p(x)p(y)$.
- using the chain rule of entropy, it follows that:

$$I(X;Y) = H(X) + H(Y) - H(X,Y). \tag{2.15}$$

- therefore, $I(X;X) = H(X) - H(X|X) = H(X)$.
- $0 \leq I(X;Y) \leq \min\{H(X), H(Y)\}$.

The relationship of mutual information with entropy, joint entropy, conditional entropy, can be depicted in the diagram 2.1.

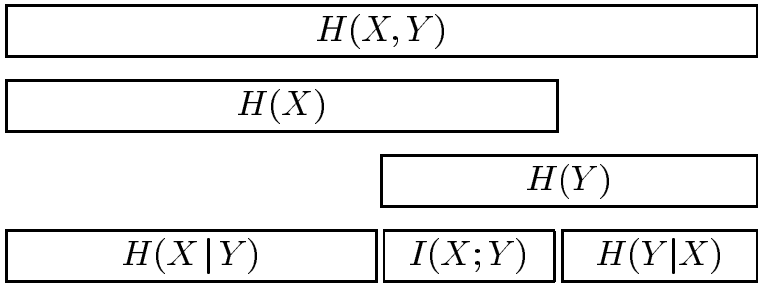


Figure 2.1: Relationship between $H(X), H(Y), H(X,Y), H(X|Y)H(Y|X)$ and $I(X;Y)$ - Image from the book "Information Theory, Inference, and Learning Algorithms" *[Mac03]*

.

It is important to note here, that mutual information is not a metric, as it violates the requisite property of 'triangle inequality'. For applications that require a metric, e.g. clustering, we can use either of the following quantities (among others), that make use of mutual information.

$$\rho_1(X,Y) = \frac{I(X;Y)}{H(X)} \tag{2.16}$$

$$\rho_2(X,Y) = H(X,Y) - I(X;Y) \tag{2.17}$$

Both of these quantities satisfy the following properties that define a metric:

1. $0 \leq \rho_1(X,Y)$ and $0 \leq \rho_2(X,Y)$ (non-negativity)

2. $\rho_1(X,Y) = \rho_2(X,Y) = 0$ iff $p(X) = p(Y)$ (indiscernability)

3. $\rho_1(X,Y) = \rho_1(Y,X)$ and $\rho_2(X,Y) = \rho_2(Y,X)$ (symmetry)

4. $\rho_1(X,Z) \leq \rho_1(X,Y) + \rho_1(Y,Z)$ and $\rho_2(X,Z) \leq \rho_2(X,Y) + \rho_2(Y,Z)$ (triangle inequality or subadditivity)

Conditional Mutual Information

Conditional mutual information, is the first higher order functional in our study. In the simplest case of three random variables, it expresses the reduction in uncertainty of X due to knowledge of Y, given Z.

In a manner analogous to conditional entropy, it manifests itself in two variations: the first variation measures the conditional mutual information between random variables X and Y given a certain instance of Z ($z = c_k$) and the second variation, is the average over all values of Z of the first conditional mutual information.

$$I(X;Y|z=c_k) = H(X|z=c_k) - H(X|Y,z=c_k), \tag{2.18}$$

$$I(X;Y|Z) = H(X|Z) - H(X|Y,Z) \tag{2.19}$$

$$\mathbb{E}_{p(x,y|z)} \log \frac{p(X,Y|Z)}{p(X|Z)p(Y|Z)} \tag{2.20}$$

The relationships between entropy, joint entropy, mutual information, conditional entropy and conditional mutual information can be depicted as follows:

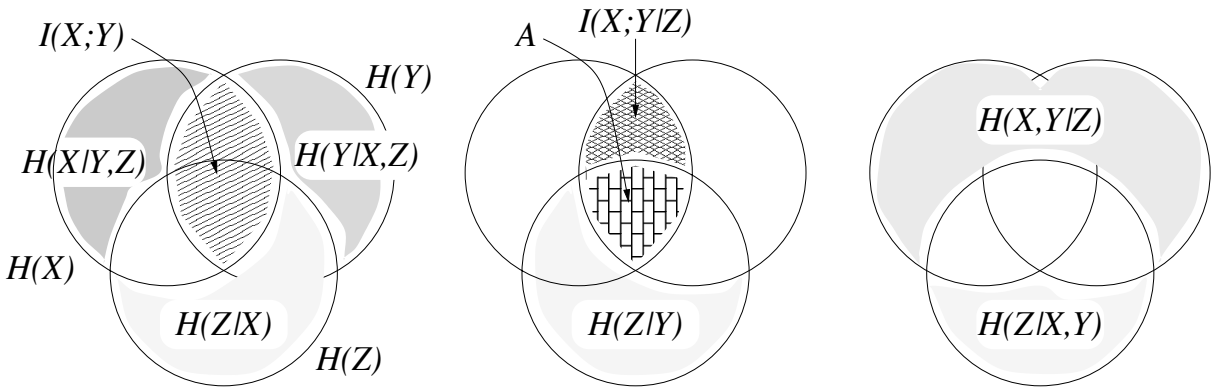


Figure 2.2: Relationship between $H(X), H(X,Y), H(X|Y), I(X;Y)$ and $I(X;Y|Z)$ - Image from the book "Information Theory, Inference, and Learning Algorithms" *[Mac03]*

.

Chain Rule for Mutual Information

By continuous application of formula 2.13, we can obtain the chain rule for mutual information, similarly to equation 2.12:

$$I(X_1, X_2, \ldots, X_n; Y) = \sum_{i=1}^{n} I(X_i; Y|X_{i-1}, \ldots, X_1). \tag{2.21}$$

e.g. for 3 variables:

$$I(X_1, X_2, X_3|Y) = I(X_1; Y) + I(X_2; Y|X_1) + I(X_3; Y|X_2, X_1).$$

2.1.3 Multivariate Mutual Information

Mutual information, as it appeared in [Sha48], was only defined for pairs of random variables without provision for generalisation to higher dimensions. It was Watanabe in 1960 [Wat60], who first introduced *total correlation*, to express the amount of dependency among a set of variables. The same principle, with similar definitions, appeared also with different names: multi-information, total constraint, higher order mutual infromation, interaction information - from different authors - Studený & Vejnarová 1998 [SYV98], Garner 1963 [HG63], Matsuda 1999 [Mat99] and McGill 1954 [McG54], respectively.

The main definition for multivariate mutual information, that is in line with the one for the standard bivariate mutual information is:

$$I(X_1; X_2; \ldots; X_n) = I(X_1; X_2; \ldots; X_{n-1}) - I(X_1; X_2; \ldots; X_{n-1}|X_n). \tag{2.22}$$

E.g. for 3 variables:

$$I(X; Y; Z) \;\; = \;\; I(X; Y) - I(X; Y|Z),$$

where conditional multivariate mutual information, will take the form:

$$I(X_1; X_2; \ldots; X_{n-1}|X_n) = \mathbb{E}_{X_n}\, I(X_1; X_2; \ldots; X_{n-1}|X_n). \tag{2.23}$$

Total Correlation

Let us consider the joint ensemble of random variables $X_1, X_2, \ldots, X_n$ with joint probability distribution $p(x_i, \ldots, x_n)$.

Total correlation is defined as:

$$I(X_1; X_2; \ldots; X_n) \;\; = \;\; \sum_{i=1}^{n} H(X_i) - H(X_1, X_2, \ldots, X_n). \tag{2.24}$$

$$\;\; = \;\; \sum_{x_1 \in X_1} \sum_{x_2 \in X_2} \ldots \sum_{x_n \in X_n} p(x_1, x_2, \ldots, x_n) \log \frac{p(x_1, x_2, \ldots, x_n)}{p(x_1)p(x_2)\ldots p(x_n)} \tag{2.25}$$

The term $H(X_1, X_2, \ldots, X_n)$ is the actual amount of information that is contained in the ensemble of variables. The difference between these terms, represents the absolute redundancy present in the given ensemble, hence providing a general quantitative measure of the cohesiveness of the set of variables [Rot52]. According to the second equation, total correlation can also be interpreted, as the Kullback-Leibler divergence between the actual distribution $\mathrm{p}(X_1, X_2, \ldots, X_n)$ and its maximum entropy product approximation $p(X_1)p(X_2)\cdots p(X_n)$.

Interaction Information

Interaction information is defined by equations 2.22 and 2.19 presented above. In the case of three variables, equation 2.22 can be expanded as:

$$\begin{aligned} I(X;Y;Z) &= I(X;Y) - I(X;Y|Z) \text{ using eq. 2.19} \\ &= I(X;Y) - H(X|Z) - H(Y|Z) + H(X,Y|Z) \text{ using eq. 2.15} \\ &= H(X) + H(Y) - H(X,Y) - H(X|Z) - H(Y|Z) + H(X,Y|Z) \text{ using eq. 2.9} \\ &= H(X) + H(Y) + H(Z) - H(X,Y) - H(X,Z) - H(Y,Z) + H(X,Y,Z). \end{aligned}$$

$I(X;Y;Z)$ measures the intrinsic three-body correlation, which is not governed by two-body statistics.

With four variables X, Y, Z, W and the same set of equations, we get:

$$\begin{aligned} I(X;Y;Z;W) &= I(X;Y;Z) - I(X;Y;Z|W) \\ &= H(X,Y,Z,W) - H(X,Y,Z) - H(X,Y,W) - H(Y,Z,W) - H(X,Z,W) \\ &+ H(X,Y) + H(Y,Z) + H(X,Z) + H(W,X) + H(W,Y) + H(W,Z) \\ &- H(X) - H(Y) - H(Z) - H(W) \end{aligned}$$

Again, $I(X;Y;Z;W)$ measures the intrinsic four-body correlation, that is not governed by standard mutual information.

Proceeding in a similar fashion, the general case of total correlation of n variables in terms of marginal entropies would be:

$$I_n(X_1, \ldots, X_n) = -\sum_{k=1}^{n} (-1)^{k+1} \sum_{i_1 < \ldots < i_k} H(X_{i_1}, \ldots, X_{i_k}), \tag{2.26}$$

where the term $\sum H(x_{i_1}, \ldots, x_{i_k})$ runs over all possible combinations $\{i_1, \ldots, i_k\} \in \{1, \ldots, n\}$. This generalisation to n variables can be attributed to Jakulin & Bratko [JB04].

Higher Order Mutual Infomation

Higher order mutual information, is very similar - except the normalization factor - to total correlation presented above. For the purposes of completeness, we will rewrite the relevant formulae from the perspective of probability functions. In figure 2.2, it is represented with the shaded area marked as A, for the special case of three variables.

Let us consider the joint ensemble of random variables $X_1, X_2, \ldots, X_n$ with joint probability distribution $p(X_i, \ldots, X_n)$. Then recalling equation 2.5, the joint entropy of the ensemble is:

$$H(X_1, X_2, \ldots, X_n) = -\sum_{X_i} p(X_1, \ldots, X_n) \log p(X_1, \ldots, X_n). \tag{2.27}$$

Recalling from equation 2.14, mutual information is the joint entropy of the ensemble, averaged on the variables' range, so in an analogous manner, higher order mutual information can be defined as:

$$I(X) = (-1)^n \sum_{X_i} p(X_1, \ldots, X_n) \times \ln \frac{p(X_1, \ldots, X_n)}{\hat{p}(X_1, \ldots, X_n)}. \tag{2.28}$$

where:

$$\begin{aligned} \hat{p}(X_1, \ldots, X_n) \quad = \quad & \prod_{i_1 < \ldots < i_{n-1}} p(X_{i_1}, \ldots, X_{i_{n-1}}) / \\ & \times \prod_{i_1 < \ldots < i_{n-2}} p(X_{i_1}, \ldots, X_{i_{n-2}}) / \\ & \cdots / \prod_i p(X_i). \end{aligned}$$

In the case of three variables the above equation becomes:

$$I(X_1, X_2, X_3) = -\sum_{X_1, X_2, X_3} p(X_1, X_2, X_3) \ln \frac{p(X_1, X_2, X_3)}{\hat{p}(X_1, X_2, X_3)}. \tag{2.29}$$

with

$$\hat{p}(X_1, X_2, X_3) = \frac{p(X_1, X_2)p(X_2, X_3)p(X_1, X_3)}{p(X_1)p(X_2)p(X_3)}. \tag{2.30}$$

It is important to note here, that total correlation or higher order mutual information, do not share all the properties of mutual information, specifically, the non-negativity. We summarise here, some of the properties and important inequalities implied by the definitions of total correlation/higher order mutual information.

- it is symmetric
- it can be either negative or positive
- $I(X_1, X_2, \ldots, X_n) = 0$ iff $p(X_1, X_2, \ldots, X_n) = p(X_1)p(X_2)\ldots p(X_n)$.
- $I(X_1, X_2, \ldots, X_n) \leq I(X_1, X_2, \ldots, X_{n-1}) \Leftrightarrow I(X_1, X_2, \ldots, X_{n-1}|X_n) \geq 0$.

2.1.4 Information Theory-Related Concepts

Kullback-Leibler Divergence

Kullback-Leibler divergence, or relative entropy $D(p||q)$, is a measure of the "distance" between two PDFs p and q. It is defined as the expected logarithm of the likelihood ratio:

$$D(p\|q) = \sum_x p(x) \log \frac{p(x)}{q(x)}, \tag{2.31}$$

$$= \mathbb{E}_{p,q}\left(p(x) \log \frac{p(x)}{q(x)}\right). \tag{2.32}$$

The relative entropy is conceptually, a measure of the inefficiency of assuming that the distribution of a random variable is q, when the actual distribution is p. Mutual information is a special case of relative entropy, as it expresses the relative entropy between the joint distribution of

two random variables and the product of their individual distributions, as in:

$$\begin{aligned} I(X;Y) &= D\left(p(x,y)\|p(x)p(y)\right), && (2.33)\\ &= \mathbb{E}_{p(x,y)} \log \frac{p(x,y)}{p(x)p(y)}, && (2.34)\\ &= -\sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}. && (2.35) \end{aligned}$$

Relative entropy shares some of the properties of mutual information, i.e. is always non-negative, is zero if and only if $p = q$. However, it is not symmetric and also violates the triangular inequality, hence excluded as a strict measure of distance.

Data Processing Inequality

Three random variables X, Y, Z, are said to form a Markov chain in that order, i.e. $X \to Y \to Z$, if their joint probability function can be written as:

$$p(x,y,z) = p(x)p(y|x)p(z|y). \tag{2.36}$$

The data processing inequality states that:

$$\text{if } X \to Y \to Z \text{ then } I(X;Y) \geq I(X;Z). \tag{2.37}$$

which informs us, that the average information that Z conveys about X, is less than or equal to the average information that Y conveys about X. An important consequence of this, is that the dependence of X and Y actually decreases (or remains unchanged) in light of a "downstream" variable, or in other words, no more information can be obtained from a pair of variables, by processing them, than what was there to begin with.

Equation 2.36 implies that variables X and Z are conditionally independent given Y and also that $Z \to Y \to X$.

In the problem of assessing gene interactions, data processing inequality can offer a means

of reasoning about the type of the dependence and deciding about the directionality of the interaction (see chapter 7).

Partial Mutual Information

Partial mutual information is a concept introduced by Frenzel and Pompe [FP07], as an information theory equivalent for partial correlation. Similar definitions and area of application (discovery of couplings in multivariate time series), appeared in [PcvS03] and [Sch00].

Partial correlation in linear regression, measures the degree of association between two random variables, with the effect of a set of controlling random variables removed [SS05b]. In the context of inferring gene association networks, it was applied successfully, in [SS05a] and [SS05b].

The purpose of partial mutual information - and its linear model equivalent - is to help overcome the limitations of pair-wise mutual information analysis in differentiating direct and indirect interactions, thereby reducing the number of false positives and inferring a more truthful interaction structure.

Partial mutual information, essentially builds on the concepts of higher order mutual information (or interaction information) and conditional mutual information. Especially in the case of three random variables, partial and conditional mutual information coincide. Starting from the formula $I(X;Y;Z) = I(X;Y) - I(X;Y|Z)$ we have:

$$\begin{aligned} I(X;Y|Z) &= I(X;Y) - I(X;Y;Z) \\ &= H(X) + H(Y) - H(X,Y) \\ &- H(X) - H(Y) - H(Z) + H(X,Y) + H(X,Z) + H(Y,Z) - H(X,Y,Z) \\ &= H(X,Z) + H(Y,Z) - H(Z) - H(X,Y,Z). \end{aligned}$$

In terms of probability functions (discrete case) it can be expressed as:

$$I(X;Y|Z) = \sum_{X,Y,Z} p(X,Y,Z) \log \frac{p(Z)p(X,Y,Z)}{p(X,Z)p(Y,Z)} \tag{2.38}$$

In the spirit of partial correlation, we can define the *order of partial mutual information*, which reveals the number of control/conditioning variables. Hence, first-order partial mutual information, is one with a single conditioning variable, second-order partial mutual information, is one with two variables etc. Zero-order partial mutual information is equivalent to the standard bivariate mutual information.

Some of the most straightforward properties of partial/conditional mutual information are:

- $I(X;Y|Z) \geq 0$ (non-negativity)
- $I(X;Y|Z) = I(Y;X|Z)$ (symmetry)
- $I(X;Y|Z) = 0$ iff $p_{xyz}/p_{..z} = (p_{x.z}p_{..z})(p_{.yz}p_{..z})$. i.e. if and only if X and Y are independent under Z.

In chapter 6, we show examples of how by using a combination of conditional mutual information and classic mutual information we can improve the performance of a network inference strategy, by effective distinction of driving and responding elements and the detection of asymmetry in the interaction of subsystems [Sch00].

2.2 Previous Work

This section summarises the work that has been done to date on the reverse engineering of gene regulatory networks using information theoretic functionals, in chronological order.

2.2.1 Reveal

The oldest attempt to apply mutual information to network deconvolution, dates back to the work of Liang et.al. (1999) [LFS98] were it was applied on temporal Boolean networks [Som96] of low input order of $N = 50$ network elements.

The purpose of the algorithm was, given a set of state transition measurements on a given set of elements, to compute for every combination of k genes (maximum 3) the corresponding k-order mutual information. The rules are constructed using look-up tables for input/output

Algorithm 1 Network Construction Algorithm for REVEAL

```
1: identification of k=1 links
2: determination of k=1 rule
3: while i <= N or there are unexplained state transitions do
4:     identification of k=i links
5:     determination of k=i rule
6: end while
```

pairs. Redundant pair listings are eliminated as we progress and optimisations are possible to avoid unnecessary computations. Of course, as k increases the calculations become slower and even for a network of moderate size ($N = 50$) the number of possible configurations becomes too large to compute. Reveal depends on the critical assumption that all genes have to be observed; on the other hand it performs better on small sample settings. Overall, despite the restrictive setting, Reveal is a very interesting effort.

2.2.2 Relevance Networks

Perhaps the best known example of the application of mutual information in genetic network reconstruction, is the work of Butte and Kohane (2000) [BK00] who used (standard) second-

order mutual information to find functional genomic clusters, based on the assumption that higher mutual information scores (TMI) relate to functional similarity. Mutual information

Algorithm 2 Network Construction for Relevance Networks

1: compute pair-wise MI for all pairs of genes
2: compute threshold (TMI)
3: select those interactions > TMI

was estimated using a two-dimensional histogram and in the resulting association network edges with higher mutual information are drawn with thicker lines. The threshold value for the mutual information (TMI) is selected with respect to the significance of the mutual information distribution, that was calculated after repeated (30 times) permutations of the samples.

2.2.3 ARACNe

One of the most recent and promising applications of mutual information in network reconstruction is the Algorithm for the Reverse Engineering of Accurate Cellular Networks developed by Margolin and Nemenman (2004) [MNB+04]. ARACNe has been used to reconstruct the network structure of human B-cells and the substructure of the cMYC-oncogene in particular, using 336 samples in [BMS+05].

Concept of Dependence ARACNe builds upon the notion of conditional (in)dependence in the form of Bayesian Networks, where the goal is to partition the overall statistical dependence, that is the deviation of the joint probability distribution from the product of its marginals, into contributions of interactions among pairs of variables. The joint probability distribution function of the variable vector $\mathbf{X}$ is:

$$p(\mathbf{X}) = -e^{H(\{g_i\})} \tag{2.39}$$

where:

$$H(\{g_i\}) = \sum_i \phi_i(g_i) + \sum_{i,j} \phi_{i,j}(g_i, g_j) \tag{2.40}$$

is the Hamiltonian that defines the system's statistics. Based on the above representation, ARACNe's definition of statistical dependence is the following: "a set of variables interacts if and only if the single potential that depends exclusively on these variables is nonzero".

The focus of ARACNe, is to identify irreducible statistical dependencies among pairs of variables, which can be interpreted as direct regulatory relationships, rather than dependencies arising from cascades of interactions, using the data processing inequality (DPI).

Algorithm 3 Network Construction Algorithm for ARACNe

1: estimate pairwise interactions using MI
2: compute threshold I_0, for a specific p-value p_0
3: threshold MI scores using I_0
4: rank remaining pairs
5: remove indirect interactions using DPI (eq. 2.37)

Estimation of pairwise mutual information For the estimation of mutual information, the original method, uses a Gaussian Kernel Estimator (GKE), appearing in Beirlant et. al. (2001). The idea is to first rank-order x and y, so that the range of the transformed variables is between 0 and 1 and their marginal probability distribution becomes uniform. This step removes the need to consider position-dependent kernel widths h.

Computing the threshold To determine if mutual information is statistical significant we first need to compute the critical p_0-value to be used as a threshold. The p_0-value is computed via MCMC simulationsThe null hypothesis is defined as a pair of existing genes, whose values are randomly shuffled at each iteration.

Following this step, a computed mutual information score is considered statistically significant, if $I_{ij} \geq I_0$, where I_0 is the significance threshold corresponding to the pre-computed p_0. I_0 is itself derived empirically, by calculating the fraction of estimates below I_0, for different sample sizes and for 10^5 pairs, for p-values up to 10^4.

Using DPI Using the data processing inequality, there is a danger of discarding direct interactions as well, if they are in either of the two following forms: (1) cyclic or acyclic loops with exactly three genes and (2) triplets of variables where the pair-wise mutual informations are zero, when the total interaction is not (e.g. symmetric covariance matrix). In an

improved version of the algorithm, some of these interactions get "rescued" by adding an arbitrarily chosen percent tolerance threshold, but overall it performs poorly in the case of networks with plenty of loops. The major advantage of data processing inequality, is that it requires mutual information to be estimated in the two-dimensional space, rather that three, which is clearly a bonus for less accurate estimators.

Remarks The authors claim that, under a controlled set of approximations - only pair-wise interactions, true network having a tree structure and number of samples greater than 100, ARACNe is able to reconstruct the network without errors.

Furthermore, they claim, that ARACNE's performance does not depend directly on the accuracy of the mutual information estimate, but rather on the accuracy of the estimation of mutual information ranks. Yet, the authors of the paper of the CLR method [FHT+07] that we will describe later, implemented the method using an alternative kernel density-based estimator (Daub et.al.[DSSK04]) and show that this improved the method's precision.

2.2.4 Information Theory and K2

Another recent work on applying mutual information on genetic network reconstruction is the paper from Chen and Anantha (2006) [CAW06] that combines information theory and a greedy search algorithm in BN learning. K2 works on **discretised** expression measurements (2-step quantization). The proposed method first constructs an undirected network based on mutual information between two nodes and then splits the structure into substructures: The above

Algorithm 4 Undirected Network Construction Algorithm for IT&K2

```
1: calculate pairwise MI for all pairs of nodes
2: sort in descending order
3: for each MI pair do
4:    if there is no existing path between the nodes then
5:       add an edge between the two nodes
6:    end if
7: end for
```

procedure results in an undirected network without any loops, for which we know the node

ordering. The 2-level quantisation is calculated as the distance of a particular sample of the gene expression from the mean for that particular gene. After this step, mutual information is computed via its discrete formula (frequency counts).

In the second phase, using the K2 algorithm, the directional orientations for the edges that connect two nodes are then obtained, by optimising a scoring function for each substructure. K2 reduces the computational complexity of learning a network structure by requiring a prior ordering of nodes as an input. The ordering is such that if node X_i is prior to node X_j then X_j cannot be parent node to X_i. The overall performance of the algorithm is greatly affected by this initial ordering.

The method targets small networks and the applications on expression profiling include an example with 8 nodes and an example with 20 nodes using 77 samples (from Spellman's yeast dataset).

2.2.5 CLR

Another successful application of information networks is the work Faith, Hayete et.al. [FHT+07]. For their study they assembled a compendium of 445 new and previously published expression profiles from *E. Coli* under a diverse set of conditions (heat shock, growth phases, numerous genetic perturbations etc.) and applied their Context Likelihood of Relatedness (CLR) method to verify the existence of 338 established interactions (taken from RegulonDB) and the discovery of 741 new ones, some of which they tested with location analysis and ChiP on Chip and real-timePCR experiments.

The CLR algorithm is an extension of the relevance networks approach [BK00] described above in the sense that it uses mutual information to score similarities between a pair of genes, but it applies an adaptive background correction step instead of thresholds, in order to select the most significant interactions.

The adaptive background correction step estimates the likelihood of the mutual information score for a particular pair of genes X, Y by comparing it to the "background" distribution of

mutual information scores involving X and all other variables and then Y and all other variables. Background distribution essentially represents the sum of direct and indirect, accidental and "intentional" relationships of a variable and it is the set that any candidate significant interaction should be compared against (via z-scores in the case of CLR). The background distribution is thus constructed using two sets of MI values: MI_X - the set of information scores involving X and every other potential gene (in row and column X of the adjacency matrix) and similarly MI_Y (see figure [2.3] below). We treat MI_X and MI_Y as independent variables and

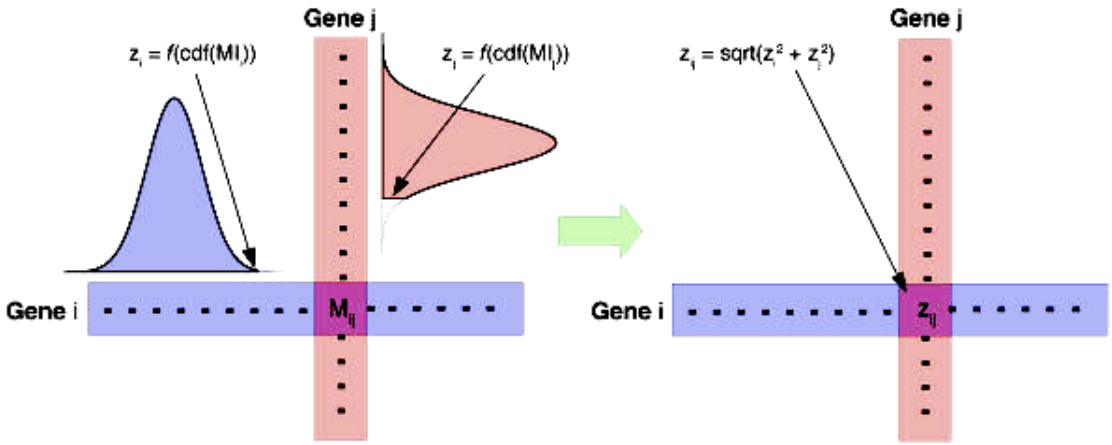


Figure 2.3: Schema for the background correction estimation of CLR. The z-score of each regulatory interaction depends on the distribution of MI scores for all possible regulators of the target gene (z_i) and on the distribution of MI scores for all possible targets of the regulator gene (z_j) [FHT+07]

.

assume that the joint mutual information score $MI_{X,Y}$ can be approximated by a a joint normal distribution. Thus, the final form of the likelihood estimate becomes $f(Z_X, Z_Y) = \sqrt{Z_X^2 + Z_Y^2}$. The algorithm then can be summarised as follows: Mutual information between a pair of vari-

Algorithm 5 CLR

1: calculate pairwise MI for all pairs of nodes
2: calculate the marginal Z-scores Z_X and Z_Y
3: calculate the joint likelihood $f(Z_X, Z_Y)$ of each MI
4: select those interactions whose joint likelihood is significantly larger than the marginals

ables is calculated using the B-spine method of [DSSK04] that we will explore in detail in the following section.

To assess the performance of the resulting network, the final step is to calculate the *precision*

and *recall* of the algorithm, i.e. the ratio of established links to the ones correctly identified also by CLR and the ratio of the total identified links to the established ones. It is reported that the algorithm scores an impressive 60 percent on precision using the specific compendium and the interactions from the RegulonDB database. This study is particularly useful, not only for the impressive performance, but also because the authors provide in their supplementary material a platform for other genetic network inference methods to be tested against.

2.2.6 MRNET

The last method in our survey of information networks, is the work of Meyer, Kontos et. al [MKLB07]. MRNET is insprired by a successful feature selection technique, the *maximum relevance/minimum redundancy* (MRMR) algorithm [DP05], [TFMF01]. The MRMR method first picks a target variable Y. Then selects the set of variables S, that has high mutual information with Y (maximum relevance) and simultaneously are mutually maximally independent, which ensures the minimum redundancy between relevant variables.

Denoting $u_j = I(X_i; Y)$ as the relevance term and $r_j = \frac{1}{|S|} \sum_{X_k \in S} I(X_j; X_k)$ as the redundancy term, the algorithm can be recapitulated as follows:

Algorithm 6 MRNET

1: select target variable Y
2: calculate u_j for every other variable X_j
3: select the variable X_j such that $X_j = \text{argmax}\, X_j \in V \setminus S(u_j - r_j)$
4: the weight of each edge will be the maximum score between the one computed when X_i is the target and the one computed when X_j is the target.

The idea behind the algorithm is that direct interactions - embodied in the most informative variables to target Y - should be highly ranked, whereas indirect ones - those whose information is captured by the direct ones - should be ranked quite low.

Chapter 3

Estimation of Information Theoretic Functionals

3.1 Introduction

There is a copious amount of of literature on the estimation of Shannon's entropy both discrete and **differential** mainly from the fields of neurobiology, econometrics, climatology, etc. For a concise citing entropy estimators see Beirlant [BDGM97] and Hlavackova et al. [HSPVB07]. In principle, standard entropy estimators, designed to extend to multi-dimensional spaces are suitable for mutual information estimation, via equation 2.11 (chain rule). However, there is no general scheme that yields systematic approximations for mutual information or, higher dimensions and other information theoretic functionals that can be incorporated in the study of variable dependencies.

Moreover, a great proportion of the literature dealing with MI estimation, was originally instigated by other areas of research, such as blind source separation (necessary for application of principal or independent component analysis - PCA and ICA) and nonlinear dynamics applications (spectral analysis), where there is an abundance of sample data.

Recalling eq.[2.14], mutual information between two continuous variables can be expressed in

two equivalent forms, as:

$$I(X;Y) = \int\int p(x,y)\log\frac{p(x,y)}{p(x)p(y))}\mathrm{d}x\mathrm{d}y \quad (3.1)$$

$$I(X;Y) = H(X)+H(Y)-H(X,Y) \quad (3.2)$$

In the context of inferring gene interactions from expression microarrays, the typical experimental setting involves a set of N bivariate measurements $z_i = (x_i, y_i), i = 1 \dots N$, which are assumed to be independent identically distributed (IID) realisations of a random variable $Z = (X, Y)$ with density $p(x, y)$.

Treating the expression values of genes X and Y as realisations of random variables with marginal densities $p(x)$ and $p(y)$ respectively, the duty of an estimator of MI is to approximate $I(X, Y)$ using the N measurements, as accurately as possible. According to equation [3.1], computation of MI includes (1) the estimation of variables' - **unknown** - marginal PDF, (2) their joint PDF and after that, (3) a log function of densities' ratios.

The typical case is that the ratios like $p(x_i, y_j)/p(x_i)p(y_j)$ are close to 1, whereas the individual terms can be of arbitrary magnitude.

This subtle balance is very sensitive to statistical fluctuations, as is the difference of entropies in the second equation. Furthermore, the same problems - even more pronounced - arise for the computation of the other information theoretic functionals like higher order entropy, higher order mutual information (total redundancy), conditional and partial mutual information.

These mathematical challenges make progress in general seem a "hopeless" task, given the limited amount of data one can expect to obtain from any biological experimental setting [PQB05]. The most common remedy to these problems, is to make some assumptions on the densities of the expression profile of the genes that constitute our variables. In particular, once we accept that the expression agrees with a distribution from a specific parametric family, it is sufficient to estimate these parameters and then calculate mutual information, analytically or numerically, from the functional form of the density. A handful of such expressions exist for some multivariate distributions (e.g. Gaussian, Γ-exponential etc.).

Once the distribution parameters have been estimated, these functionals can be calculated by numerical integration.

In that situation, we still need to keep in mind, that for random processes with an arbitrary parametric distribution, expressions for mutual information, conditional mutual information and total redundancy are not usually available analytically [KS61] and indeed very difficult to produce, although they are biologically more relevant. Finally, once the parametric assumptions are violated, the generated algorithms will produce erroneous results.

With these considerations in mind, we embark on a search for reliable estimators of information theoretic functionals we set the following list of requirements. Following the discussion above, suitable estimators must fulfill the following requirements:

- be designed for continuous distributions
- make as few assumptions as possible for the sample densities (non-parametric)
- be applicable to small sample settings
- be practically easy to implement

The results of our search are analysed in the following section. In the remainder of this chapter, we will be assessing the properties and compare the performance of the selected candidates.

3.2 Estimators

The non-parametric MI estimators that designed for continuous random variables, that exist in the literature today, could roughly be classified in three different categories: (1) histogram-based, (2) kernel density (KDE)-based and (3) k-Nearest Neighbours (KNN)-based estimators. This classification highlights the defining characteristic of each algorithm and serves as a context for their presentation. For our comparative studies, presented in the next section, we've implemented one method per paradigm (one that shows some promise of robustness in small sample settings) and tested it against the same set of synthetic data. This will enable us to make judgements free from the idiosyncrasies of each method.

3.2.1 Estimates of MI based on partitioning of the observation space

This popular class of estimators, derive from the idea, that differential entropy can be approximated by partitioning the data space (quantization), producing a histogram of the observations and then computing the discrete entropy of that histogram. The partitions can be either fixed a priori independently of the data distribution (see **Binning**), or generated recursively (sequentially) taking the actual distribution of the data into account (see **Adaptive Partitioning**) [HSPVB07].

Naive-Empirical Estimation (Binning)

The most straightforward and widespread approach for the nonparametric estimation of MI, is by using multivariate histograms, whereby the probabilities $p(x_i)$, $p(y_j)$ and $p(x_i, y_j)$ are approximated measuring the relative frequencies of data points occurring in a bin of fixed width [MRL95].

More formally, given an origin x_0 and a bin width h_x, the bins of the histogram are defined through the hypercubes formed by intervals $[x_0 + ah_x, x_0 + (a+1)h_x]$, with $a \in A$ being the partition of x. Thus, the data is partitioned into m_x discrete bins a_i and the histogram is

defined by [Sil86]:

$$\hat{p}(x) = (\sharp x_i \; in \; the \; same \; bin/(Nh_x^d) = k_i/N \tag{3.3}$$

where N is the sample size, d is the dimension of the variables and h_x is the bin width.

In an analogous manner, we estimate $p(y)$, using elements $p(y_j)$ from the bins b_j belonging to the partition B and the joint probability $p_{x,y}(a_i \times b_j) = k_{i,j}/N$ from the XY plane, which is subdivided into boxes (product partition $A \times B$).

Following this procedure, the MI can then be estimated using the formula below:

$$I(X,Y) = \log N + (1/N) \sum_{i=1}^{m_x} \sum_{j=1}^{m_y} k_{i,j} \log \frac{k_{i,j}}{k_i k_j} \tag{3.4}$$

where $k_{i,j}$ is the number of measurements for which x lies in a_i and y lies in b_j.

It is important to note here, that while the limit of the sum (entropy of single variable) for the discrete case diverges as the partition becomes finer, the integral in eq. [3.1] that applies for continuous variables does not diverge, but its value depends on the parameters (coordinates) chosen [1]. Hence, the choices of the origin and the bin width can affect the resulting histogram.

For the commoner and simplest implementations, histograms use a regular mesh of boxes - **marginal equidistantization**. All boxes have the same size and refinements are done uniformly, by decreasing the mesh spacing. To increase the goodness of fit of the histogram to the underlying distribution, the resulting mesh has to be fine enough to resolve all the details of the densities and yet large enough to guarantee convergence to the true density avoiding statistical fluctuations due to the paucity of data samples in each bin. Therefore, successful binning strategies are those that achieve a good compromise between sampling errors and sampling resolution. Graphical illustrations of the effects of bin width and starting point to the resulting histograms can be found in Silverman (1986) [Sil86] and Shinomoto [SS]. Knuth et al. [KGCH05] introduced optimal binning techniques, for a special case of density function approximation (piecewise-constant histogram-style models). For a discussion on the subject of optimal bin sizes see Scott(1979) and for a new method for optimizing time histograms see

[1] The reason is that the argument of the log function in eq. [3.1] has the same units as $1/\mathrm{d}x_i$ [FS86]

Shinomoto [SS07] and Moddemeijer [Mod86].

In the multivariate setting an obvious refinement of the methodology presented above, is to use variable box sizes over the different axes, so that equal numbers of pairs exist in every cell. This idea of adaptive partitioning only locally in places where substructure is statistically significant, is called **marginal equiquantization**.

Marginal Equiquantization

Fraser & Swinney, in [FS86] first pursued the idea of marginal equiquantization to compute mutual information, by recursively splitting each coordinate axis in turn, such that each bin constructed has a nearly uniform distribution of points.

Algorithm 7 Adaptive Partitioning

```
1: Let G_0 be the initial one-cell partition of the XY plane
2: for each cell in the current level of partitioning do
3:    if the distribution of points in the cell is not uniform then
4:       subdivide the cell into 2^m equiprobable segments
5:    end if
6: end for
```

The algorithm results in a hierarchy of partitions $G_0, G_1, ..., G_m$ over the XY plane, organized in a tree structure. Each partition is a rectangular grid of 4^m elements generated by subdividing each axis into 2^m equiprobable segments.

Using tree notation, $R_m(K_m)$ denotes an element of G_m and K_m is an m-tuple $(k_1, k_2, ..., k_m)$ representing the index of the element, that takes one of the 4^m possible values. The number m for the equiprobable intervals is arbitrary, however for most applications $m = 2$, because large m would unecessarily complicate the computations.

For the basic recursion step, the algorithm checks each element of level m for substructures, which translates to "if joint probability distribution $P(xy)$ in this partition is non- uniform", before proceeding to level $m + 1$. In the case where $P(xy)$ is exactly uniform on $R_m(K_m)$ the authors show [FS86] that dividing the partition element into four subpartitions will have no effect on the contribution to mutual information obtained from that element. To assess

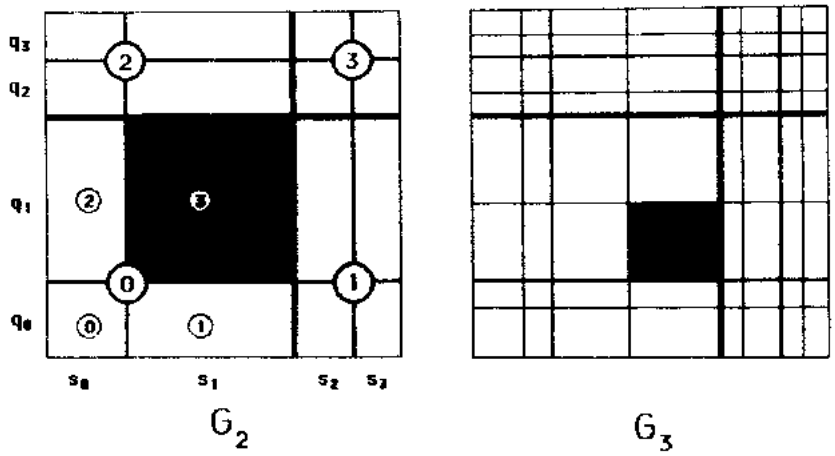


Figure 3.1: Two steps in the sequence of partitions. The shaded element of G2 is named (0,3) in the tree notation and the shaded element of G3 is named (0,3,1) [FS86]

.

uniformity, a χ^2-based test is used. The null hypothesis is that $P(xy)$ is flat over the existing cell, which would imply that the distribution of points in the potential subpartitions are flat multinomials. If then the critical values for each potential subelement are above a certain threshold as defined by the χ^2 test (the level of significance arbitrarily chosen to be 20%), we conclude that there is no substructure in $R_m(K_m)$ and the recursion stops.

The above procedure results in a mixed partitioning of the XY plane, where splitting proceeds only if needed, in areas where $p(x_i, y_j)$ has finer structure.

More recently, Cellucci et. al. [CAR05], suggested a number of improvements to the above algorithm: (1) the number of bins is determined using the minimum description length criterion, (2) the estimated MI is explicitly tested further for its significance by a χ^2 test of the null hypothesis of statistical independence and finally (3) the partition of XY plane should satisfy the Cochran criterion [2] on the expectancies E_{XY} [CAR05]. A comparison of the two methods is also included in the paper.

[2] $E_{XY}(i,j) \geq 1$ for all elements of the partition and $E_{XY}(i,j) \geq 5$ for at least 80% of these elements.

Adaptive Partitioning by Darbellay & Vajda

A successful variation in adaptive partitioning for the estimation of mutual information was suggested by Darbellay and Vajda in [Dar98]. This algorithm is in principle the same to the Fraser-Swinney algorithm, but technically its implementation differs in a couple of points:

- it works with d-dimensional hyperrectangles.
- the partitioning strategy is to check if the mutual information of the variables in the current partition is zero within some statistical fluctuation - i.e. to find out if the variables are locally independent (rather than to test the uniformity of the distribution of points).
- the χ^2 test is used to assert the local independence of X and Y, as a criterion to exit the splitting recursion.
- the partitioning step is to split each of the d edges of a cell into $\alpha \geq 2$ equiprobable intervals. So in effect each cell is split into α^d subcells.

Therefore in practice, the χ^2 test measures the probability of partitioning a cell when it shouldn't be so - if the test fails, that bin is partitioned along a coordinate axis.

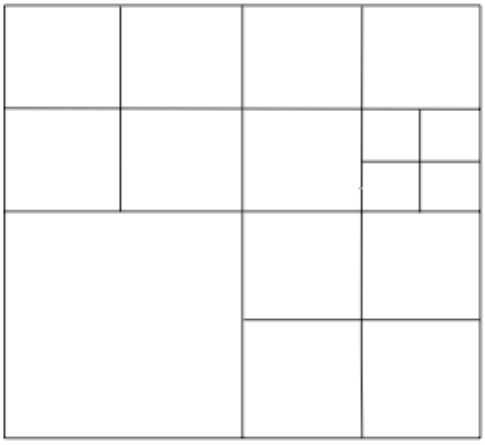


Figure 3.2: An example of a partitioning that can arise from a recursion step by [Dar98].

The main argument behind the partitioning strategy is that, to find out if two variables are independent, it is sufficient to check if the mutual information on an arbitrarily chosen partition

is zero within some statistical fluctuation; in other words it is sufficient to check local independence. This is the theorem of Dorbushin (1959) [Dob59] To put it formally, if Λ is a refinement of the partition Γ such that $C_k \in \Gamma = \bigcup_l D_{k,l} \in \Lambda$ for some set of indices l depending on k then

$$D_\Gamma = D_\Lambda \Leftrightarrow \frac{P_X(D_{k,l})}{P_{X_a \times X_b}(D_{k,l})} = \frac{P_X(C_k)}{P_{X_a \times X_b}(C_k)} \forall k, l. \tag{3.5}$$

Finally, from the final set of partitions , say $G = \{C_k, k = 1, \ldots, m\}$ the mutual information will be evaluated according to:

$$\hat{I}(X_a, X_b) = \frac{1}{N} \sum_{k=1}^{m} N_{\mathbf{X}}(C_k) \ln \frac{N_{\mathbf{X}}(C_k)}{N_{\mathbf{X_a}}(C_k) N_{\mathbf{X_b}}(C_k)} + \ln N \tag{3.6}$$

which is calculating relative frequencies on appropriate partitions and achieving conditional independence on the rectangles of which the partitions are made.

As pointed by Moddemeijer [Mod99] and Kraskov [KSG04], all estimators based on binning, i.e. the approximation of $I(X, Y)$ by $I_{binned}(X, Y)$, suffer from bias caused by the quantisation. This is aggravated in the case of finite sample sizes from approximating logarithms of probabilities by logarithms of frequency ratios if no error correction mechanisms as in [Gra88] are applied.

3.2.2 k-Nearest Neighbours

Entropy estimation using k-nearest neighbours (KNN) methods, appeared nearly 50 years ago and was established as a popular alternative to standard binning approach by Dobrushin [Dob59] and by Ebrahimi et. al. in [Soo92]. However the estimators were limited to the one dimensional case and could not be directly generalised to accommodate higher-dimensional spaces.

For general multivariate densities, KNN methods, are based on the idea that all sample instances correspond to points in the n-dimensional space $\Re^n$. Real-valued target functions, such as PDFs, are then approximated by calculating the mean value of the distances of the k nearest

neighbours of an instance, plus the Euler constant[3]. Distances are defined in terms of the standard Euclidean metric. For the estimation of mutual information the aforementioned principle can be interpreted as follows [KSG04]:

For each point $z_i = (x_i, y_i) \in \Re^2$ rank its neighbours by distance $d_{i,j} = \| z_i - z_j \|$: $d_{i,j_1} \leq d_{i,j_2} \leq d_{i,j_3} \leq \ldots$.
Similar rankings can be done in the subspaces X and Y. Then each of $H(X)$, $H(Y)$ and $H(X,Y)$ can then be estimated separately as the average distance to the k-nearest neighbour averaged over all x_i, y_i and z_i respectively and finally MI will be computed according to our familiar equation $I(X,Y) = H(X) + H(Y) - H(X,Y)$.

Considering point x_i and the points found around it that fall within the range of a ball of radius $\epsilon > 0$, Kozachenco and Leonenko [Koz87] estimated $p_i(\epsilon)$ to be $p_i(\epsilon) \approx c_d \epsilon^d \mu(x_i)$ and the expectation value of $\log p_i(\epsilon)$ to be equal to $\psi(k) - \psi(N)$ where $\psi(x)$ is the digamma function, $\mu(x_i)$ is the distance to the k-th nearest neighbour and c_d is the volume of the d-dimensional unit ball. Recalling that a KNN entropy estimator can be understood as an average of $\log \mu(x)$, setting $c_d = 1$ (maximum norm) for Euclidean distances, we obtain the following estimator (KL):

$$\hat{H}(X) = -\psi(k) + \psi(N) + log c_d + \frac{d}{N} \sum_{i=1}^{N} log(\epsilon_i) \tag{3.7}$$

where ϵ_i is twice the distance from x_i to its k-th neighbour, d is the dimension of x . Equation [3.7] holds for any value of k, so one can first look for k-nearest neighbors in the joint space and then use the hyper-spheres defined by these points to calculate the number of neighbors in the marginal spaces.

The Kozachenco-Leonenko estimator is proven to be an asymptotically biased estimator and minimally biased estimator for small sample problems [Koz87]. It is however important to add, that for any fixed k, the distance to the k-th neighbor in the joint space will be larger than the distances to the neighbours in the marginal spaces. Since the bias from the non-uniformity of the density depends on these distances, the biases in $\hat{H}(X)$, $\hat{H}(Y)$ and $\hat{H}(X,Y)$ would not cancel. Hence, although this approach is tried, tested and improved for univariate Shannon'

[3] $C_E = -\int_0^\infty e^{-t} \log t \, dt$

entropy of continuous distributions from finite sets of empirical samples [Vic02],[Gra03], its use is not recommended for the estimation of MI .

k-NN based method by Kraskov et. al.

As a remedy to the issue of non-cancelling biases, Kraskov et al. [KSG04] recommend not to use fixed k for marginal entropy estimators and two estimators are proposed that can also be applied to higher order redundancies.

Their proposed estimators develop around the following idea: assume that the k-th neighbour of x_i is on one of the vertical sides of a square with size $\epsilon(i)$ (see figure 3.3, panel (a)). If we denote with $n_x(i)$ the number of points within the square then we can say that $\epsilon(i)/2$ is the distance to the $(n_X(i)+1)$-th neighbour of x_i so equation 3.7 can become:

$$\hat{H}(X) = -\frac{1}{N}\sum \psi[n_X(i)+1] + \psi(N) + \log c_{d_X} + \frac{d_X}{N}\sum_{i=1}^{N} log(\epsilon_i) \tag{3.8}$$

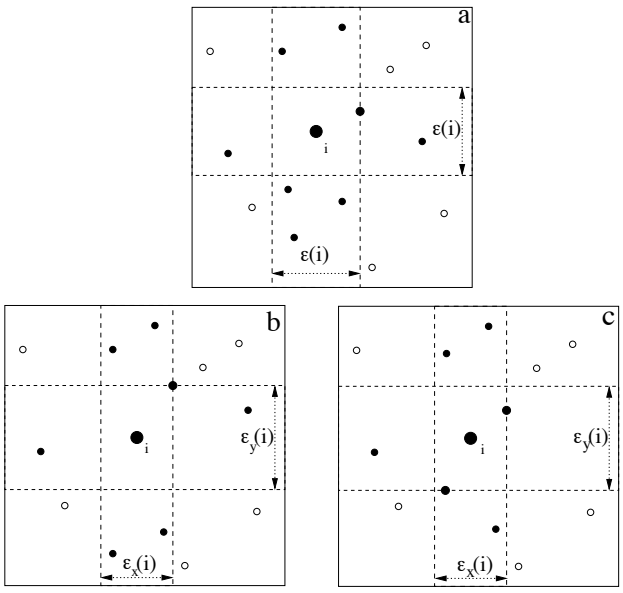


Figure 3.3: Panel (a): Determination of $\epsilon(i), n_x(i)$ and $n_y(i)$ in the first algorithm, for $k = 1$ and some fixed i. In this example, $n_x(i) = 5$, $n_y(i) = 3$. Panels (b), (c): Determination of $\epsilon_x(i)$,$\epsilon_y(i)$, $n_x(i)$ and $n_y(i)$ in the second algorithm for $k = 2$. Panel (b) shows a case in which $\epsilon_x(i)$ and $\epsilon_y(i)$ are determined by the same point, while panel (c) shows a case where they are determined by different points [KSG04].

Moreover, if we introduce for the space $Z = (X, Y)$ the maximum norm, $\| z_i - z_j \| = max\| x_i - x_j \|, \| y_i - y_j \|$ and denote by $\epsilon(i)/2$ the distance of z_i to its kth neighbour and by $\epsilon_x(i)/2$ and $\epsilon_y(i)/2$ the distances between the same points projected onto the X and Y subspaces, we can then count the number of points in the small rectangle around point i containing k neighbours $(n_x(i), n_y(i))$ with $\| x_i - x_j \| < \epsilon_x(i)/2$ and $\| y_i - y_j \| < \epsilon_y(i)/2$ as the new number of neighbours in the marginal space. It follows that $\epsilon(i) = max\{\epsilon_x(i), \epsilon_y(i)\}$. This enables us to adjust equation [3.7] to estimate the mutual information as:

$$\hat{I}^{(1)}(X, Y) = \psi(k) - 1/k - \langle \psi(n_x + 1) + \psi(n_y + 1) \rangle + \psi(N) \tag{3.9}$$

Alternatively, if we count the number of points $n_x(i)$ and $n_y(i)$ as those whose distance is strictly less than $\epsilon_x(i)$ and $\epsilon_y(i)$ respectively, then we get a similar estimator:

$$\hat{I}^{(2)}(X, Y) = \psi(k) - 1/k - \langle \psi(n_x) + \psi(n_y) \rangle + \psi(N) \tag{3.10}$$

For both estimators, we denote by $\langle \dots \rangle$ averages both over all $i \in [1, \dots, N]$ and over all realisations of the random samples,

$$\langle \dots \rangle = N^{-1} \sum_{i=1}^{N} E[\dots(i)] \tag{3.11}$$

Applying the same principle to d dimensions the first (hyper-cubes) estimator becomes:

$$\hat{I}^{(1)}(X_1, \dots, X_d) = \psi(k) - (d-1)\psi(N) - \langle \psi(n_{x_1}) + \dots + \psi(n_{x_d}) \rangle \tag{3.12}$$

and the second (hyper-rectangles):

$$\hat{I}^{(2)}(X_1, \dots, X_d) = \psi(k) - \frac{d-1}{k} - (d-1)\psi(N) - \langle \psi(n_{x_1}) + \dots + \psi(n_{x_d}) \rangle. \tag{3.13}$$

For the implementation of the methods, we would like to point out that for both classes of estimators, we used the suggestion of Kraskov, in [KSG04], to calculate MI after first ranking the x_i by magnitude and coranking the y_i with them. Nearest neighbors of (x_i, y_i) can then be

obtained by searching x neighbours on both sides of x_i and verifying that their distance in the y direction is not too large. However, even with this optimisation, both algorithms are quite slow. An alternative way of searching for nearest neighbours is the *box counting* method as found in Grassberger in [Gra83], but for average d it did not prove to have any clear advantages. Moreover, since many points in a large set may have identical coordinates, the numbers $n_X(i)$ and $n_Y(i)$ are no longer unique. That results in errors for the calculations and as a remedy we followed the suggestion in [KSG04] to add a very low-amplitude noise to the data.

Extension to higher dimensions is made also possible, under the same minimal-bias framework, with the use of the recursive formula:
$I(X_1, X_2, \ldots, X_m) = I((X_1, X_2, \ldots, X_{m-1}), X_m) + I(X_1, X_2, \ldots, X_{m-1})$ where

$$\begin{aligned} \hat{I}(X_1, X_2, \ldots, X_m) = \psi(k) - (m-1)/k + (m-1)\psi(N) \\ -\langle \psi(n_{x_1}(i) + \psi(n_{x_2}(i) + \ldots + \psi(n_{x_m}(i) \rangle \end{aligned} \tag{3.14}$$

3.2.3 Kernel Density Estimators

Both the histogram-based and the nearest-neighbour approaches described in the previous sections can be thought of as approximating the target pdf at a single observation point.

One obvious refinement to this concept is to weight the contribution of each of the k neighbours according to their distance to the observation point giving greater weight to closer neighbours. Once we add distance weighting, we can allow all instances to have an influence on the approximation of the target function [Mit97].

The distance functions, used to determine the relative weight over the distance of the sample points to the point of estimate are called *kernel functions* $K()$ and the resulting pdf estimators are called kernel density estimator (KDE).

With kernel methods the density functions are estimated by a local weighted average of the relative frequency of observations on the neighbourhood of the point of estimate, i.e. by the

summation of the kernels centered at each observation point.

$$\hat{p}(\mathbf{y}) = \frac{1}{n}\sum_{i=1}^{B} K_H(d(\mathbf{x},\mathbf{y})) \tag{3.15}$$

where B is the number of bins, $\mathbf{x}$ is the d-dimensional random vector whose density is being estimated $\mathbf{y}$ is the sample vector and d is a distance function and H is a smoothing parameter called *bandwidth* [Sil86].

Kernel functions $K(d(\mathbf{x},\mathbf{y}))$ are themselves also required to be a valid symmetric probability density functions [MRL95]. Possible choices are Gaussian, Epanechnikov, with Gaussian being the favourite, as the properties of this kernel are well understood, especially for bivariate data [THS01]. For the Gaussian kernel function:

$$K(d(\mathbf{x},\mathbf{y})) = exp(-\frac{\|\mathbf{x}-\mathbf{z}\|^2}{2\sigma^2}) \tag{3.16}$$

and the Epanechnikov kernel function:

$$K(d(\mathbf{x},\mathbf{y})) = \frac{3}{4\sqrt{5}}(1-\frac{d(\mathbf{x},\mathbf{y})^2}{5}, |d(\mathbf{x},\mathbf{y}))| < \sqrt{5} \tag{3.17}$$

Bandwidth is the parameter of the kernel function that specifies the range of data values over which the weighted average is computed. In fact, the bin width in binning methods can be also understood as a form of bandwidth, if only a rigid one [MRL95]. In KDE, bandwidth acts as a smoothing parameter around the bin edges, by defining the kernel's spread and orientation (variance), thus achieving more accurate results in approximation. The choice of bandwidth is a compromise between smoothing enough to remove insignificant bumps and not smoothing too much to smear out real peaks. In other words, large values of H are insensitive to finer details of the distributions (detect little beyond linear relationships), whereas small H values produces estimates overly influenced by individual data points [KSG04][4].

[4]In the case of a multivariate PDF, only one smoothing parameter means that the bandwidth is the same in every direction; however there could be different smoothing parameters for different directions, which results in a *bandwidth matrix* [WJ95]

Wand and Jones [WJ95] have shown than the form of the kernel has little impact on the final kernel density estimate, compared to the impact of the choice of smoothing parameter. Several bandwidth selection mechanisms can be found in the literature [Sil86],[WJ95],[HP91],[THS01].

According to Steuer et al. [SKD+02], KDE methods were found to be superior to the classical histogram methods from the following reasons: (1) they have a better mean square error rate of convergence of the estimate to the underlying density; (2) they are insensitive to the choice of origin; (3) the window shapes are not limited to the rectangular window.

Estimation of mutual information using KDE was first proposed by Moon et.al. (1995) [MRL95]. Other applications can be found in Harrold et.al. (2001) [THS01]. All these efforts measure dependence in time series, however the methodology followed is easily applicable to static datasets as well.

For Moon et.al. [MRL95] the densities are approximated by sums of N Gaussian kernels with fixed prescribed width h -"optimal" Gaussian bandwidth corresponding to the Gaussian kernel as suggested by Silverman [Sil86]- centered at the data points.

In the paper of Harrold et.al. [THS01] on hydrologic time series, empirical trials are used to select a set of appropriate bandwidth parameters instead, for a range of moving average models with normal and near-normal parent distributions. This choice follows the observation that "the calculation of MI involves a ratio of probability densities. When the sample estimate of the MI score is calculated using [3.1] two bandwidths are used: one for the calculation of the marginal (univariate) PDF's and one for the calculation of the joint (bivariate) PDF. Then, the Gaussian bandwidth cannot be expected to be optimal or near-optimal, because of the function of PDF's involved". In their approach, they assume that the badwidth parameters will be multiples of the Gaussian reference bandwidth and an algorithm for the selection of the optimal scaling factors is suggested based on the data at hand.

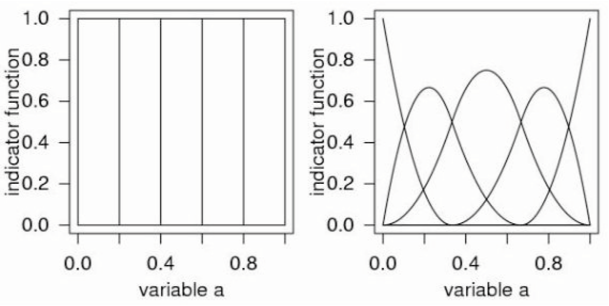


Figure 3.4: Schematic representation of the indicator function that assigns points to bins, for standard binning (left) $k = 1$ and B-spline approach (right) $k = 3$. The weight of each data point to each of the bins is given by the intersection with the respective graphs of the B-spline function with the data point. By definition, all weights contributing to one data point (variable a) sum up to unity [DSSK04].

KDE-based method by Daub et.al.

In search of strategies to simplify the standard KDE algorithm and reduce the computational power requirements, Daub et.al. [DSSK04] propose a hybrid method combining multivariate histograms and KDE-based techniques, whereby the indicator function of kernel methods is replaced by polynomial B-spline functions and the typical hypercube shape of bins is replaced by the shape defined by a sum of Gaussians around the observation point. The bins now overlap and the weight of each data point to each of the bins is given by the value of the respective B-spline function. This approach also has the advantage of alleviating the choice-of-origin problem of stereotypical histogram methods.

More precisely, each measurement is assigned to one or more bins $i = 1 \dots M$ simultaneously, with weights given by the indicator (B-spline) functions $B_{i,k}$. The spline order k determines the shape of the weight functions and thereby the number of bins each data point can be assigned to. For $k = 1$ each point is assigned to one bin, for $k = 2$ to two bins etc., with the respective weights determined by the values of the B-spline function at the point of observation.

The first step in the definition of the B-spline function is the specification of a knot vector t_i

for a number of bins $i = 1, \ldots, B$ and a given spline order $k = 1, \ldots, k$:

$$t_i := \begin{cases} 0 & if \quad i < k \\ i - k + 1 & if \quad k \leq i \leq B - 1 \\ B - 1 - k + 2 & if \quad i > B - 1 \end{cases} \tag{3.18}$$

The spline order determines the degree of the polynomial functions. The domain of the B-spline functions lies in the interval $z \in [0, B - k - 1]$. To cover the range of the variables, the new indicator function needs to be linearly transformed to map their range. The values of the design matrix for the B-spline function correspond to the weight coefficients for the data point in each bin and they are determined according to the following recursive definition [DSSK04]:

$$B_{i,1}(z) := \begin{cases} 1 & if \quad t_i \leq z \leq t_{i+1} \\ 0 & otherwise \end{cases} \tag{3.19}$$

$$B_{i,1}(z) := \begin{cases} B_{i,k-1}(z)\frac{z - t_i}{t_{i+k-1} - t_i} + \\ B_{i+1,k-1}(z)\frac{t_{i+k} - z}{t_{i+k} - t_{i+1}} \end{cases} \tag{3.20}$$

The algorithm can be summarised as follows:

Algorithm 8 Estimating MI using B-spline functions

1: **for** each variable $a : X$ and Y **do**
2: determine the knot vector (Eq.[3.18])
3: transform the data linearly into the domain of the B-spline function
4: evaluate the design matrix for the B-spline (Eq.[3.20])
5: determine the probability for each bin (Eq.[3.21])
6: calculate marginal entropies $H(a)$
7: **end for**
8: estimate joint probabilities for XY for all $B_X \times B_Y$ (Eq.[3.22]) and the joint entropy (Eq.[3.23])
9: calculate $I(X, Y) = H(X) + H(Y) - H(X, Y)$

B_X is the number of bins for variable X and B_Y for variable Y. The joint entropy of the two variables, as well as the marginal entropies can then be calculated by taking the weighted

average of the observations in each bin according to the following equations:

$$p(a_i) = \frac{1}{N}\sum_{u=1}^{N} B_{i,k}(x_u) \tag{3.21}$$

$$p(a_i, b_j) = \frac{1}{N}\sum_{u=1}^{N} B_{i,k}(x_u) \times B_{j,k}(y_u) \tag{3.22}$$

$$H(A, B) = -\sum_{i,j=1}^{M_A, M_B} p(a_i)p(b_j)log(p(a_i, b_j)) \tag{3.23}$$

Chapter 4

Parametric Simulations

The purpose of this chapter is twofold: first, to explore the applied aspects of the methods besides their theoretical particularities and secondly, to provide a common framework to assess their performance. The latter formed a major part of our work involving porting these algorithms in **R** [R D08].

Since an assessment of accuracy requires example cases where the true value of mutual information is known, we opted to provide that assuming that the PDF underlying the samples belongs to known parametric families. For the purposes of comparison we limited our attention to two information theoretic functionals: mutual information and higher order mutual information. Other functionals (conditional MI, partial MI) are either derivable from the aforementioned ones or unsuitable for this type of predictive testing.

In a nutshell, our assessments may be classified under two broad headings:

- bivariate mutual information $I(X;Y)$;
- multivariate (higher order) mutual information[1].

Each one of those headings will be analysed under two different settings:

[1]To further simplify our tests, we restrict the multivariate to the trivariate higher order mutual information $I(X;Y;W)$, introducing a third random variable W.

- Gaussian distribution;
- Alternative distribution.

This second dichotomy is largely a division of labour. Considerable amount of works carried out on simulation and testing has been based on the multivariate (or bivariate) normal distribution. The most profound reason behind this neglect of other distributions, is that no finite sample of data can determine density or mutual information directly. Therefore, some assumption about either the functional form of the density or about its smoothness is desirable and the most common solution is to assume that the density has a parametric form [HSPVB07].

The inclination to favour the normal distribution is partly due to its attractive properties - (1) generating multivariate normal samples is very easy, (2) mutual information of a multivariate normal can be directly calculated from its variance - and partly to the absence of analytical expressions of mutual information for distributions other than the normal.

Analytical functional forms for alternative continuous multivariate distributions are difficult to produce and indeed very sparse. Only a handful of papers exist giving a list of such formulae, from Ahmed et al. [AG89], Nadarajah et al. [NZ03] and [DV00] and almost all of them without any details of the calculations. The breakthrough for this part of our work was enabled by the formulae found in the paper by Darbellay and Vajda [DV00]. The sparsity of papers on the subject confirms the difficulty of the problem of producing such formulations as it requires in depth knowledge of multivariate distribution theory as well as information theory.

Further from that, our final choice for the distributions to serve as an *alternative* testbed was influenced primarily by the simplicity of expressions and secondarily by the handiness of a method for generating random vectors preserving the properties of the family of distributions for which the formulae are available.

For the latter we were confronted with two issues:

- Univariate random number generator methods do not generalise to the multivariate case.

- There is no *global* algorithm for generating random vectors of a known distribution - just some general methodologies. Generation of random numbers is custom to the class and subclass of a distribution family at hand.

The available multivariate distributions from [DV00] are type IV Pareto, logistic, Burr, exponential, Weibull, Weinman exponential, Ordered Weinman exponential and Gamma exponential. In our study, we considered the logistic and exponential. We will take the multivariate **logistic** distribution as our prototype distribution for experimenting with the performance of the estimators.

The family of bivariate and multivariate logistic distributions have always been considered as a potential alternative to the Gaussian distribution for testing purposes. Despite its relative popularity however, there is no 'universal' version applied in the different fields; rather a number of multivariate logistic (i.e. multivariate distributions with logistic marginals) exist emerging from different 'real problems' (see [JK72], [Dev86]).

Gumbel [Gum60], introduced the bivariate logistic distribution, a special case of the generalised multivariate logistic distribution with c.d.f.

$$F(x_i) = \left(1 + \sum_{i=1}^{d} e^{-x_i}\right)^{-1} (x_i > 0, i = 1, 2, \ldots, d). \tag{4.1}$$

Satterthwaite and Hutchinson [SH78] extended the formula to a multivariate logistic as:

$$F(x_i) = \left(1 + \sum_{i=1}^{d} e^{-x_i}\right)^{-\alpha} (x_i > 0, i = 1, 2, \ldots, d). \tag{4.2}$$

where x_i are independent samples from the distribution $p(x_i) = e^{-\lambda \sum_i e^{-x_i}}$ and λ is not a constant, but follows Gamma distribution where α is the shape parameter .

Very close to this version, is the version of Johnson and Kotz [JK72] for which the generalised PDF is featured in the paper of Darbellay and Vajda [DV00] and is used to derive the

information theoretic formulae. The density of this n-dimensional logistic distribution is:

$$p(\mathbf{y}) = \prod_{i=1}^{n} \frac{\alpha + i - 1}{\theta_i} e^{-\frac{y_i - \lambda_i}{\theta_i}} \left[1 + \sum_{j=1}^{n} e^{-\frac{y_j - \lambda_j}{\theta_j}} \right]^{-(\alpha+n)} \tag{4.3}$$

This variation of logistic distribution can be obtained from the Cook-Johnson family with parameter α using the following transformation applied to each random variate:

$$Y_i = -\log(1/X_i - 1).$$

An obvious advantage of using this version of the multivariate logistic distribution, is that we can adapt the simple algorithm of generating Cook-Johnson random variates using the aforementioned monotone transformation, to generate our multivariate logistic variates. The adapted generator algorithm is the following:

Algorithm 9 Multivariate Logistic Generator

1: Generate independent identically distributed exponential random variates $E_1, \ldots, E_{d+1}$.
2: RETURN

$$(\log(\frac{E_1}{E_{d+1}}), \ldots, \log(\frac{E_d}{E_{d+1}}))$$

As a final remark, we would like to point out here, that estimation of any multivariate distribution, can also be done, using any simulation methodology (e.g. MCMC). To perform MCMC, a model for the data is constructed and repeatedly evaluated. The accuracy of the simulated results, is related to the number of simulation iterations performed. Depending on the complexity of the PDF, this process can be quite time consuming. Therefore, simulation techniques are used when analytical solutions are unavailable or impractical to perform. On the other hand, analytical methodologies are highly accurate - by definition - and in most cases do not take very long to compute. Exploring the availability of such formulae for the case of multivariate mutual information, was an added dimension to our research.

Experimental Framework

The experimental setting consists of three steps: (i) the estimation of mutual information (MI) or higher order mutual information (HOMI) of the parametric distribution via the analytic formula (ii) the basic iteration and (iii) the generation of error bars.

The basic iteration can also be broken in three steps: (i) simulated data generation (ii) the estimation of MI or HOMI, using one of the five different estimators in our benchmark and (iii) the calculation of errors.

All the methods were tested against a limited set of empirical sample sizes:
$200, 150, 100, 50, 30, 20, 15, 10, 5$. This set of sample sizes aims to represent typical sample sizes occurring in microarray studies. Indeed perhaps only a handful of experimental settings fall out of that range - microarrays still being a very desirable yet relatively expensive technology.

Two different measures of error are considered: *statistical error* and *systematic error* and are explained below. The basic iteration on each sample size, is repeated 300 times - for sample sizes $100, 150, 200$ - to 500 times - sample sizes $5 - 50$. The use of different iteration steps has only practical significance: it reduces overall run time, cutting from the cases where iterations are less needed, i.e. when sample sizes are not very small.

This is the number of realisations of mutual information estimates that are compared with the corresponding analytic values and the quantiles for the errors are produced. To decrease graphical 'clutter' the 3rd quantile values instead of the whole errorbars were displayed. The experimental set-up is graphically illustrated in figure 4.1 below.

In addition, wherever a method required parameters i.e. number of neighbours for k-NN methods, or spline order and number of bins for the KDE method, a complementary set of experiments were run to assess the effects of the parameters on the algorithms accuracy.

Finally, our focus to compare the methods at hand remaining, taking into account to this severe limitation, complements the vast majority of studies available, concerned exclusively with the asymptotic properties of the estimators or comfortably large sets of empirical samples drawn

from Gaussian distribution only.

Standard Error is the estimated standard deviation of the error of the method divided by $\sqrt{N}$ [KK51]. Specifically, it estimates the standard deviation of the difference between the estimated values and the true values, divided by $\sqrt{N}$.

Systematic Error In the plots that follow we refer to systematic error as the ratio I_{estim}/I_{exact} of the estimated value to the true value of mutual information. It serves as a indicator of the inherent imprecision of the estimator (similar to bias) assessing the systematic divergence of results from the correct value.

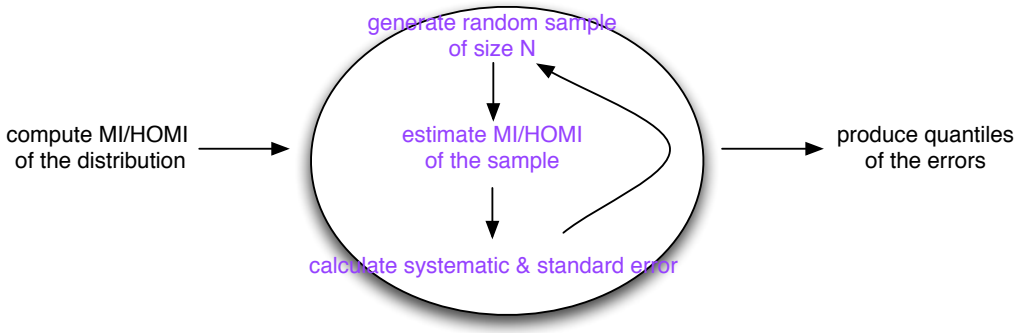


Figure 4.1: Experimental setup.

4.1 Mutual Information

In this section, we study the performance of the algorithms on the estimation of the standard mutual information of two variables. The study begins with the case of two variables drawn from a bivariate Gaussian distribution and is followed by the case of variables drawn by a bivariate logistic.

4.1.1 Gaussian Distribution

Let X and Y be two random variables following a bivariate Gaussian distribution with zero mean and unit variance. From a well known property of the bivariate Gaussian we know that each of the variables X and Y is also Gaussian with zero mean and unit variance and ρ is their correlation coefficient, or the degree of their linear dependence. $\rho = 0$ means that the variables are totally independent and $\rho = 1$ means total dependence. Mutual information $I(X;Y)$ for the bivariate Gaussian distribution has the analytic form of:

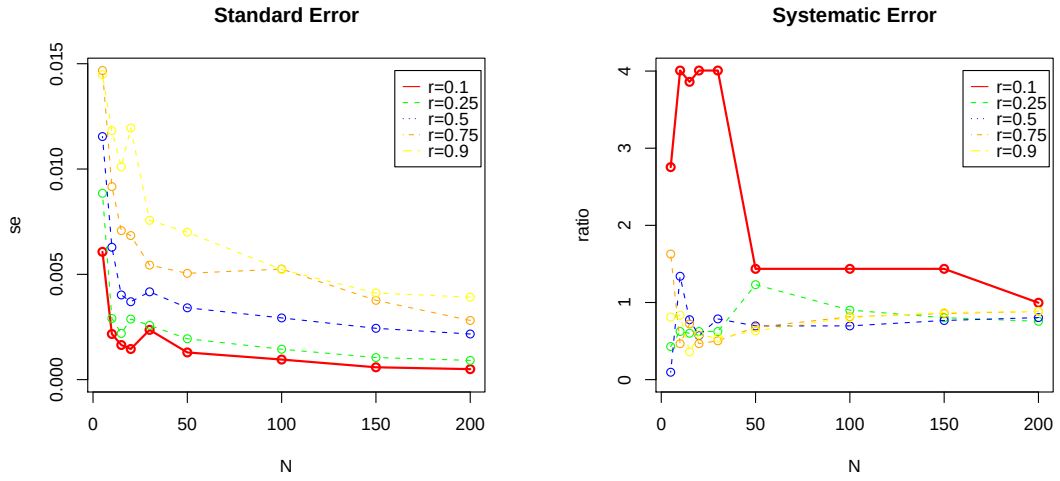
$$I_{Gauss}(X,Y) = -\frac{1}{2}\log(1-\rho^2) \tag{4.4}$$

A detailed derivation of the formula can be found at [CT91].

Generating samples from a bivariate Gaussian with known covariance is a routine task. A standard algorithm can be found at [RV94] and suitable implementations are available in every statistical package. In R [R D08] the function is `multnorm`.

Adaptive Partitioning Estimator

This method of Darbellay and Vajda [DV99] does not require any parameters. Figure 4.2 below shows how standard error [fig4.2(a)] and systematic error [fig4.2(b)] varies across sample sizes ranging from $5 - 200$ samples. Different curves correspond to different correlation coefficients - $\rho \in 0.1, 0.25, 0.5, 0.75, 1$ - to account for different degrees of dependence.



(a) (b)

Figure 4.2: [GS/MI/AP] Errors across sample size for different ρ.

From the graphs we see that the statistical error is reduced as sample size increases for all correlation values. It converges to 0 for nearly independent Gaussians ($\rho = 0.1$) and to values very near 0 for correlated Gaussians. Overall, statistical error is bigger for larger correlation values. Systematic error for all correlation values converges to 1 as the sample size increases. For nearly independent variables MI is overestimated for the very small sample sizes, but the estimator gives nearly exact values for sample sizes > 150.

k-Nearest Neighbours Estimator

To assess the KNN methodology, we first need to test its behaviour with respect to its parameters, i.e. the number of nearest neighbours k, before proceeding to test the effect of increasing sample sizes (data points) N. Figure 4.3 below shows how statistical and systematic errors vary for sample sizes 50 and 100 and different values of nearest neighbours for the first method presented in the paper by Kraskov [KSG04]. The different values of nearest neighbours were $2, 3, N/4, N/3, N/2, 3 * N/4$, which for $N = 50$ gives axis values of $2, 3, 13, 17, 25, 38$ and for $N = 100$ the values $2, 3, 25, 33, 50, 66$ respectively. From figures 4.3(a) and 4.3(c), we observe that, for sample size equal to 100, statistical error is already smaller than that of 50, but both decrease almost log-linearly with increasing k and for all correlations, converging to a value of 0.001 for $3 * N/4$, i.e. as we increase the number of neighbors, the estimator needs the support

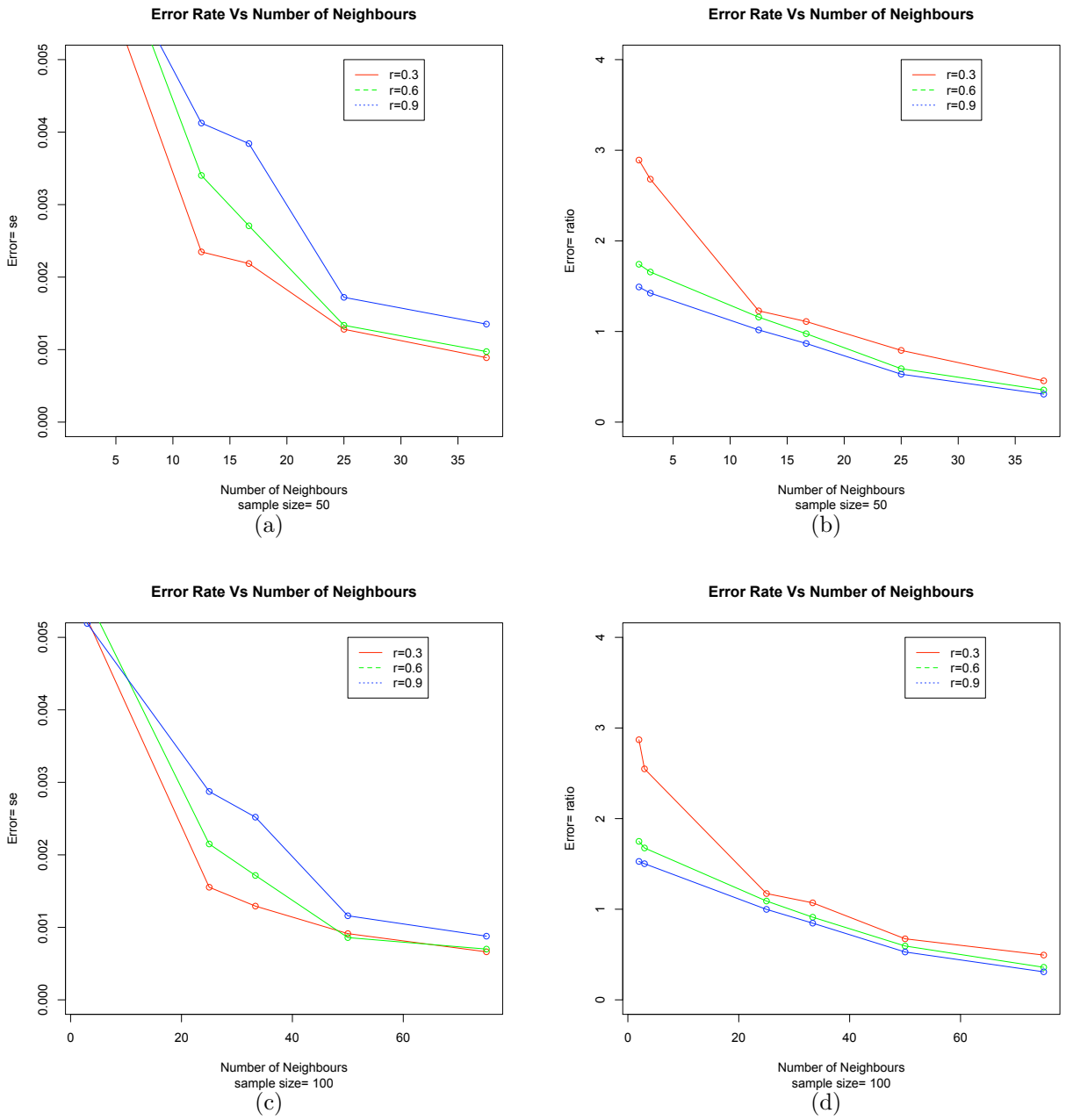


Figure 4.3: [GS/MI/KI-nn] Effect of the number of neighbours for different ρ.

of a higher number of neighbours to estimate accurately the value of MI.

As to the systematic error, (figures 4.3(b) and 4.3(d)) Kraskov I is almost exact for $k = N/4$ or $k = 13$ and $k = 25$ for all correlations. MI is underestimated for $k > N/4$ and overestimated for $k < N/4$.

For Kraskov's second method, sample sizes 100 and 200 values of the x-axis will be $2, 3, 25, 33, 50, 66$ and $2, 3, 50, 67, 100, 150$ respectively. Errors are displayed in figure 4.4.

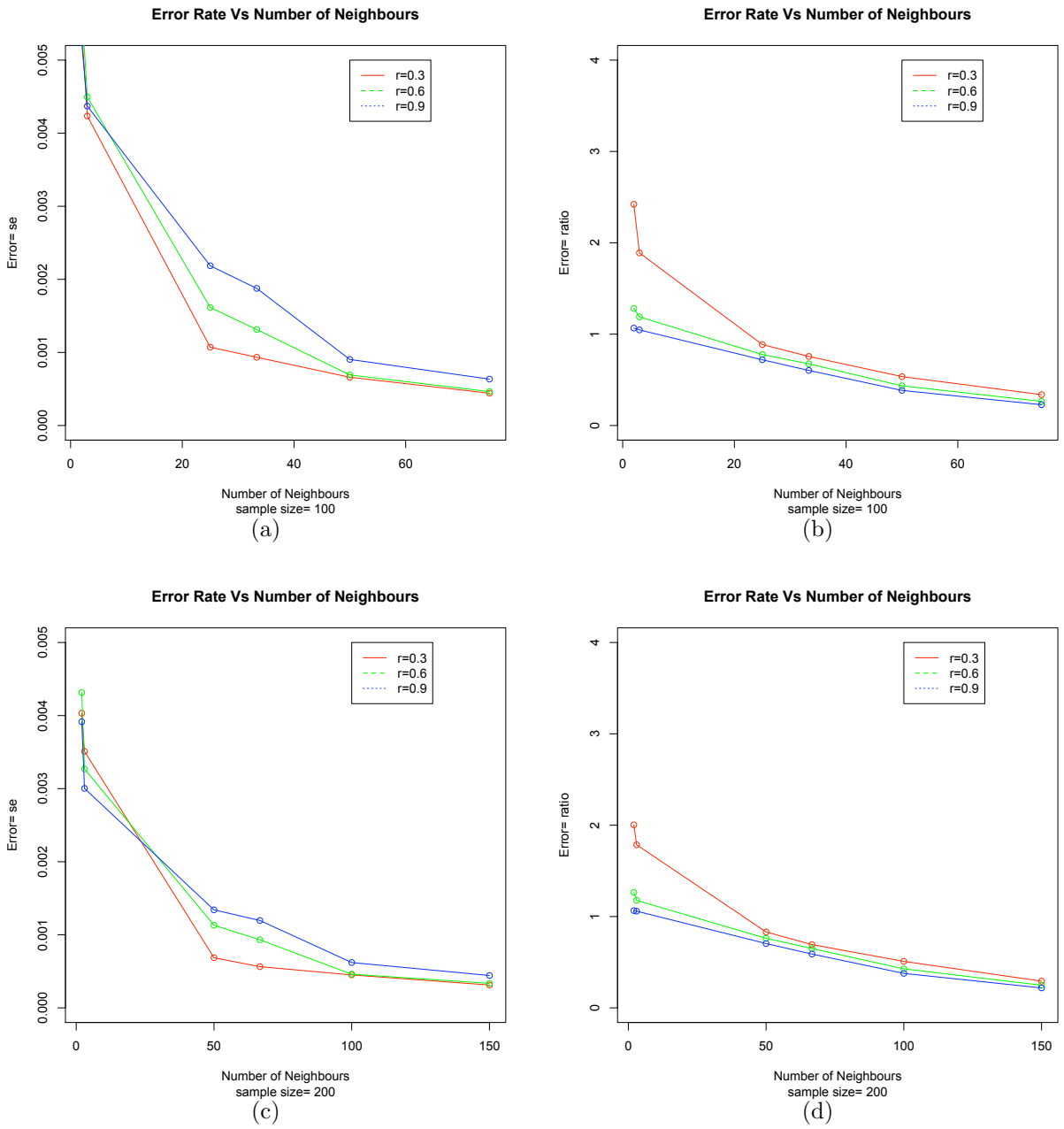


Figure 4.4: [GS/MI/KII-nn] Effect of the number of neighbours for different ρ.

Figures 4.4(a) and 4.4(c) show, that for both sample sizes, statistical errors decrease almost log-linearly with increasing k and the errors are smaller for the largest N. The estimator is almost exact for $k = 2$ and average to low correlation values and $k = N/4$ for high correlation values. Increasing the number of neighbours introduces negative bias to the estimation. Also it seems, that for both estimators and every correlation coefficient, the best value of k is consistently $3*N/4$. So, similarly to the first method of Kraskov, we need the support of as many neighbours

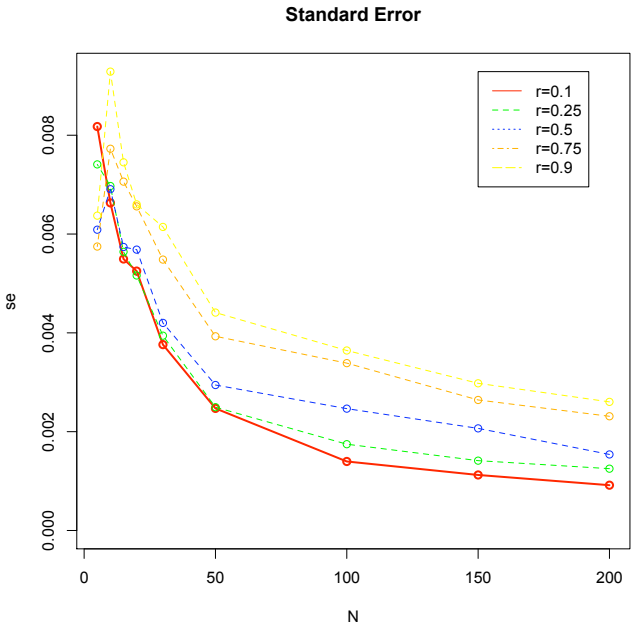
as possible to reduce statistical error.

Examining the ratios, for KII the optimal value is $k = 2$ for average to high correlation values and $k = N/4$ for $\rho = 0.3$. Thus, weakly correlated variables seem to need more support to reduce the positive bias. Any k above these values, causes overestimation and any value below that underestimation.

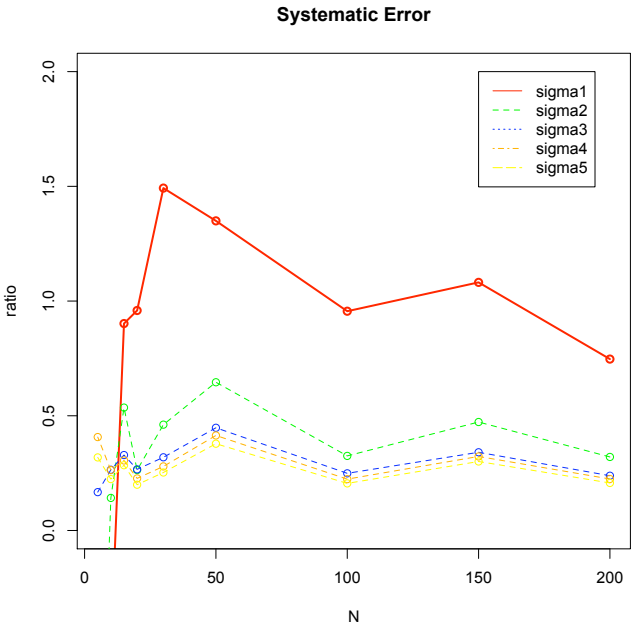
From both figures 4.3 and 4.4, it becomes obvious that we cannot minimize both statistical and systematic errors with a single choice of parameters. This result is consistent with the findings of Paninski in [Pan03]. Therefore the choice for the optimal value of the parameter is a compromise we need to make depending on the application.

Having tested the behaviour of the estimators with respect to their basic parameter, we test the behaviour of the k-NN estimators across sample size for different correlation values. The number of neighbours for each sample size is decided to be $k = N/4$ for KI and $k = 2$ for KII, which gave the 'best' results for systematic error.

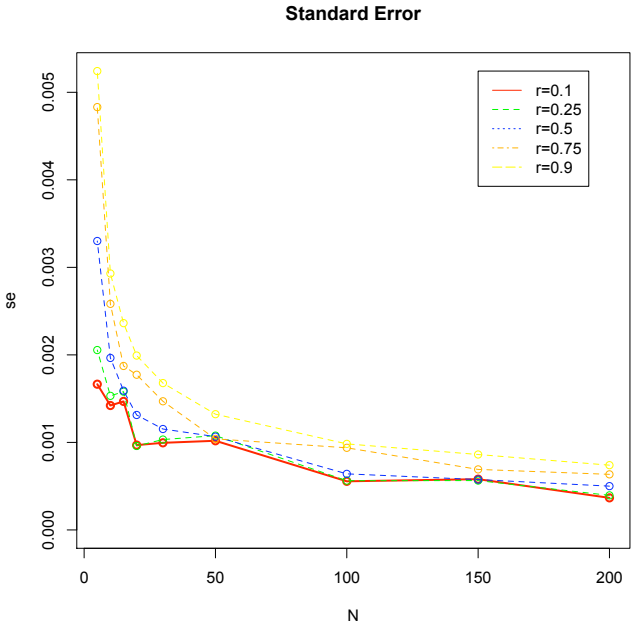
From figures 4.5(b) and 4.5(d) we see that KII overall has smaller (nearly half) the systematic errors; in fact KI seems to consistently underestimate true MI values (by 50%), whereas KII is almost unbiased from sample sizes as small as 30. This is in accordance to what is written in the paper. Other conjectures of the paper, that seem to be verified by our results, are that for the same k the second estimator gives slightly smaller statistical errors, because n_x and n_y tend to be larger and have smaller relative fluctuations. Finally, systematic errors seem to be larger for less correlated Gaussians, for both estimators, but careful selection of the number of neighbors can mitigate these biases.



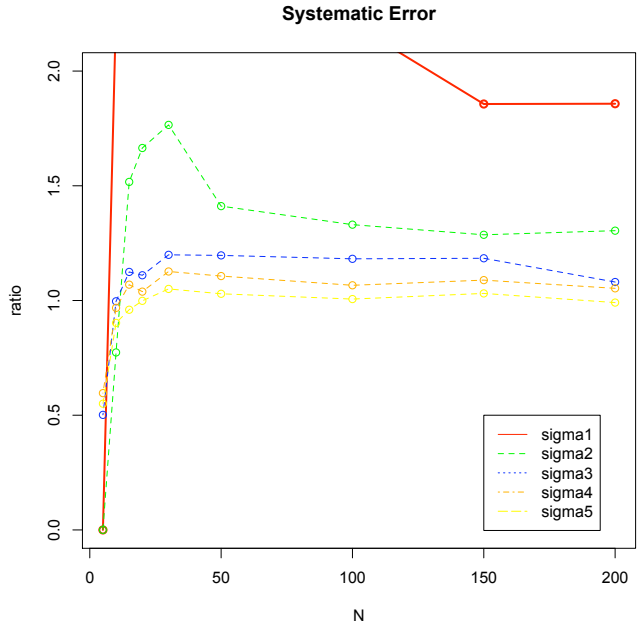
(a) KI



(b) KI



(c) KII



(d) KII

Figure 4.5: [GS/MI/Kraskov] Error values across sample size for different ρ.

Kernel Density Estimator

To assess the B-spline methodology, we first have to test its behaviour against the its parameters, i.e. spline order and the number of bins B. For the effect of spline order for sample sizes, we

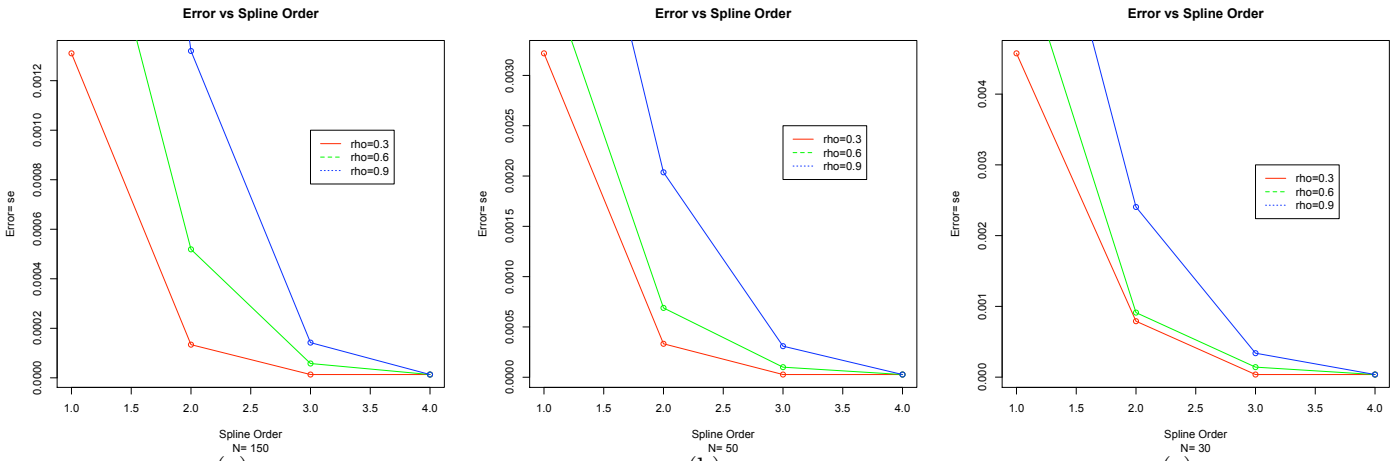


Figure 4.6: [GS/MI/KDE] Effect of Spline Order

observe that the increase of the rate of convergence is achieved from changing the spline order from 1 to 2, where the standard error is halved. After that, the change is not that significant. We observe also, how the standard error, albeit being small, quadruples from sample size $N = 150$ to $N = 20$ (figures 4.6(a) and 4.6(c) respectively. Finally, the standard error increases as the variables become more correlated.

For the effect of bin size, it is shown in the figure 4.7, that for sample size equal to 200, fewer bins improve both for systematic and standard error, or in other words, increasing the number of data points per bin, reduces both systematic and statistical error for any correlation.

Finally, the error values for the KDE method across sample size and for different correlation values are shown in figure 4.8 below. For these tests the spline order is equal to 3 and the bin size $B = 2$. Sudden changes to error values are due to failures of the spline design function to produce the weights for each bin. This phenomenon is very frequent for sample sizes smaller than $30 - 50$.

Figure 4.8 shows us, that both standard and systematic errors decrease with more samples.

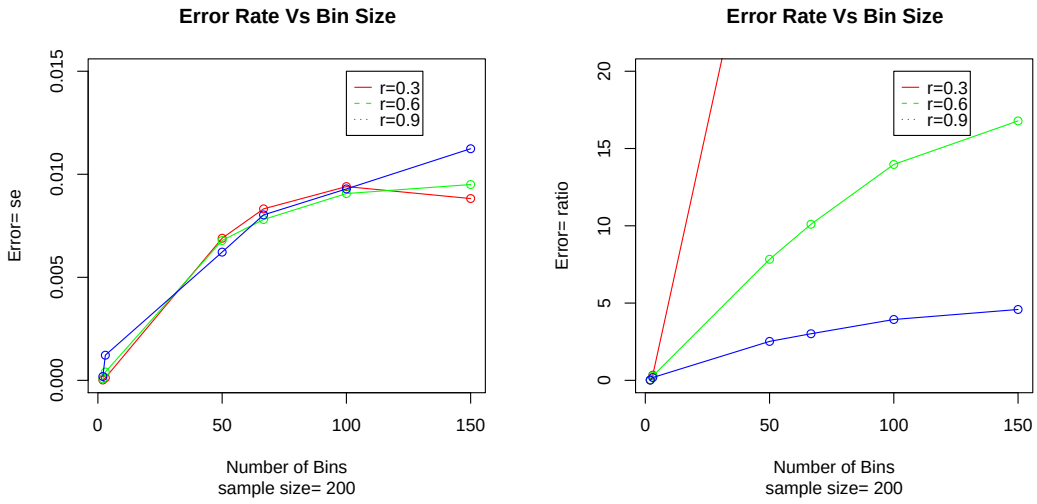


Figure 4.7: [GS/MI/KDE-BIN] Effect of Bin Size

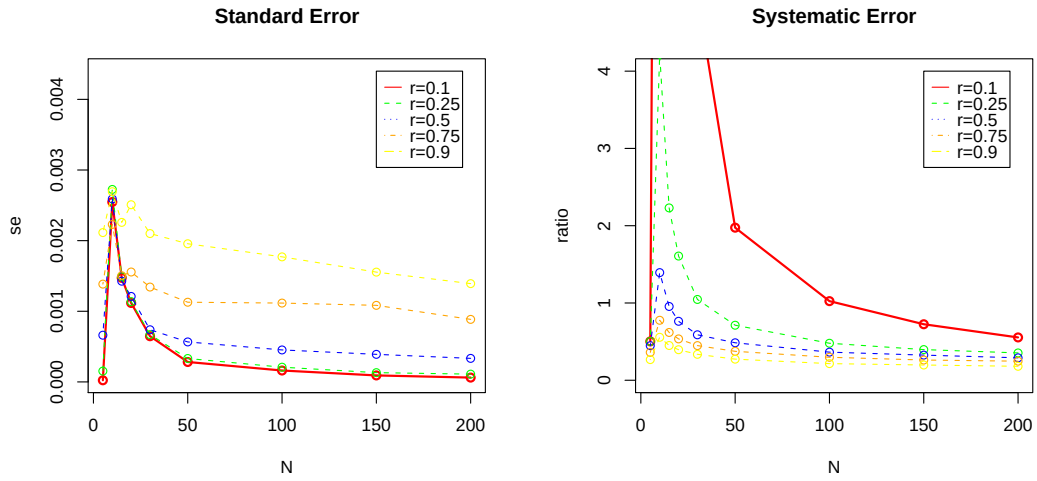


Figure 4.8: [GS/MI/KDE] Error Values across sample size for different ρ.

In the case of systematic error, a more appropriate choice of bin size - custom to sample size rather than a global one - would shift the convergence value to 1 rather that 0.6. So, overall the algorithm has the potential of being exact for sample size > 30, but still is extremely sensitive to the choice of bin size. Finally, we should also point out that the implemented algorithm was applied to rank ordered data, as in the original publication. However, testing with non-transformed Gaussians, the results were very similar.

4.1.2 Logistic Distribution

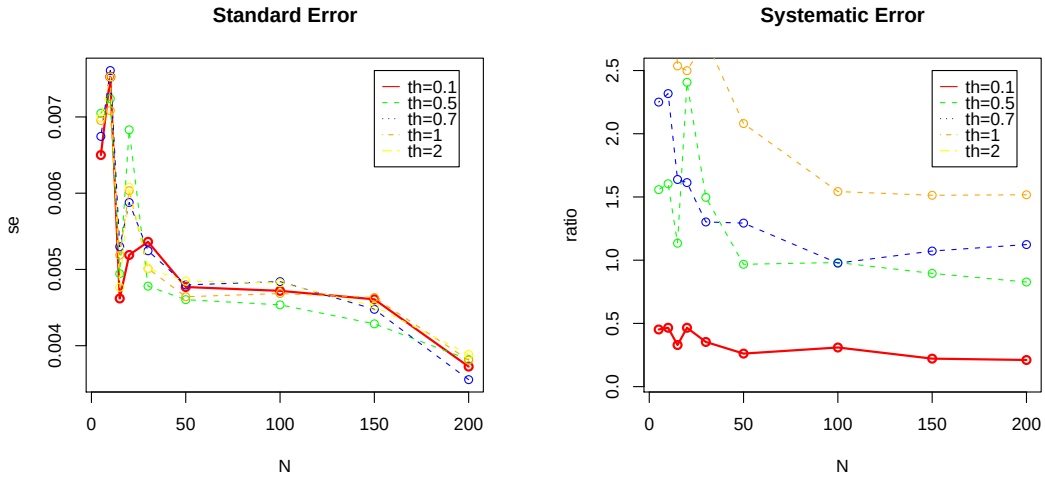
Let X and Y be two random variables following a bivariate logistic distribution (Cook-Johnson family). According to Darbellay and Vajda [DV00] their mutual information $I(X;Y)$ for the bivariate logistic distribution can be estimated via:

$$
\begin{aligned}
I(X,Y) &= \ln \frac{\prod_{j=n_x+1}^{n}(a+j-1)}{\prod_{j=1}^{n_y}(a+j-1)} && (4.5)\\
&+ (a+n_x)\sum_{j=1}^{n_x}\frac{1}{a+j-1} && (4.6)\\
&+ (a+n_y)\sum_{j=n_x+1}^{n}\frac{1}{a+j-n_x-1} && (4.7)\\
&- (a+n)\sum_{j=1}^{n}\frac{1}{a+j-1} && (4.8)
\end{aligned}
$$

where α is the shaping parameter. For the set of experiments on the following graphs $\alpha = 1$. Other set of experiments were carried out for varying values of α $(2, 3, 4)$, but the results were not remarkably different and hence are not shown. Instead, the different curves correspond to different values of the θ parameter of the exponential distribution used to create the bivariate (and in later section the multivariate) logistic distribution (see algorithm 9 in the previous section).

Adaptive Partitioning Estimator

We repeat that the adaptive partitioning estimator takes no parameters, so we can proceed directly to the tests against sample sizes. Figure 4.9(a), and 4.9(b) shows the statistical error and systematic errors of the method, respectively. From the graphs we see that statistical error decreases for larger sample sizes for all θ and systematic error converges to a certain value for sample sizes larger that 100 and any θ. This value is different for every *theta* and the estimator seems to be exact for intermediate values of θ. For $\theta = 1$ MI seems to be underestimated (half of the exact), whereas as θ increases MI gets overestimated (one and a half to over double the exact estimate). In other words, for large values of θ the algorithm seems to 'recognise' false



(a) (b)

Figure 4.9: [LOG/MI/AP] Errors across sample size for different θ.

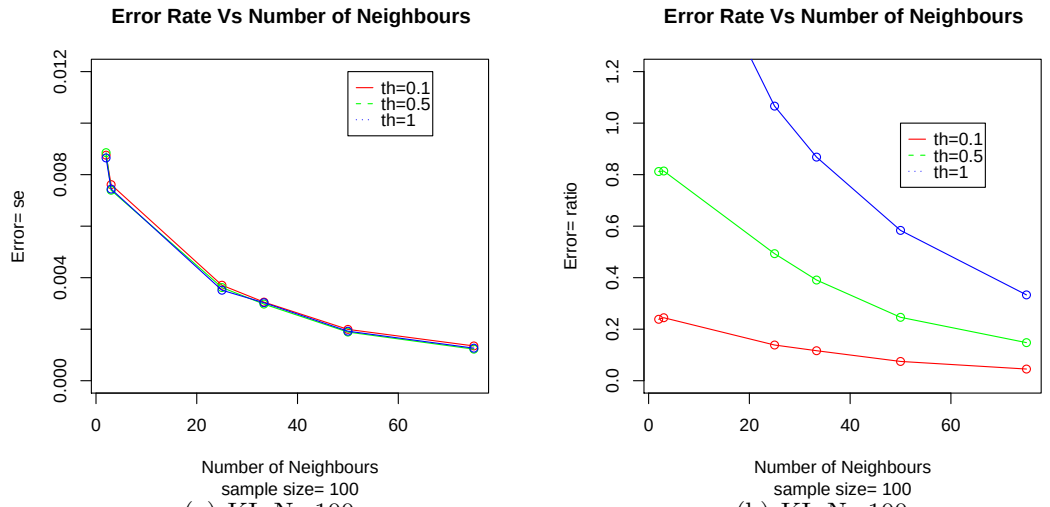
structures in the data and does not split a cell even if we increase the sample size.

k-Nearest Neighbours Estimator

As with the case of the Gaussian distribution, we first test the methods with respect to their own parameters - number of neighbours (k) - and also with respect to the parameter of the logistic distribution (α). The top row of figure 4.10 (figures 4.10(a) and 4.10(b)), shows standard and systematic error plots for KI estimator and sample size equal to 100 across different number of neighbours and the bottom row (figures 4.10(c) and 4.10(d)), shows the same error plots for KII and sample sizes equal to 100, as well.

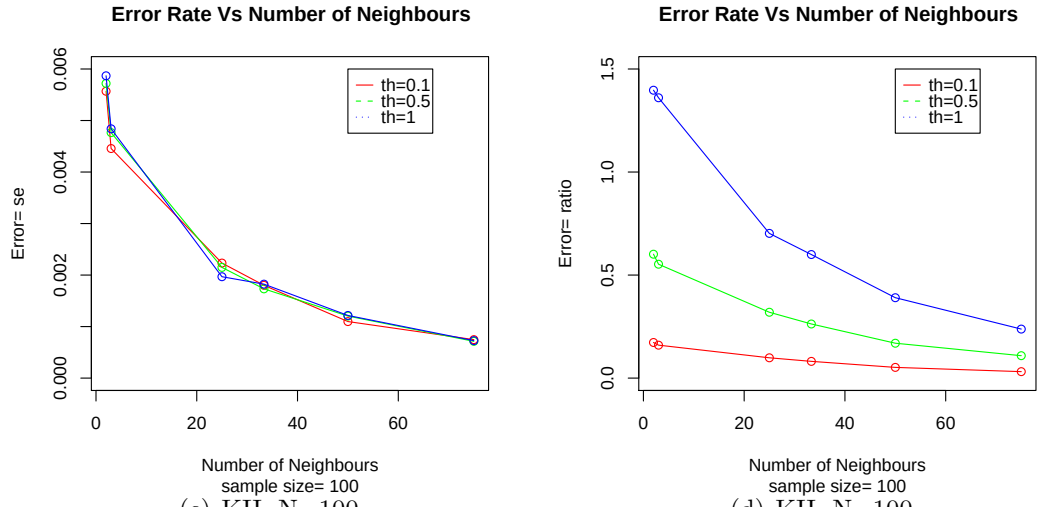
For both estimators statistical errors decrease almost log-linearly with increasing number of neighbours $k \approx 3 * N/4$ and error curves almost coincide for different θ values. Moreover, both estimators have very similar systematic error curves. For the largest θ value both estimators seem to be exact for k between 2 and $N/4$ (or $k/N = 0.02-0.25$) for KII and KI respectively. For $\theta = 0.5$ the critical value is $k = 2$ and for $\theta = 0.1$ MI seems to be consistently underestimated for any k and converging to 0 - for both θ best k is 2[2]. Overall, less neighbours improve systematic errors whereas more neighbours improve statistical errors, so once more there seems to be no

[2]Convergence to 0 practically means that the estimated MI value is very close to 0 itself.



(a) KI, N=100

(b) KI, N=100

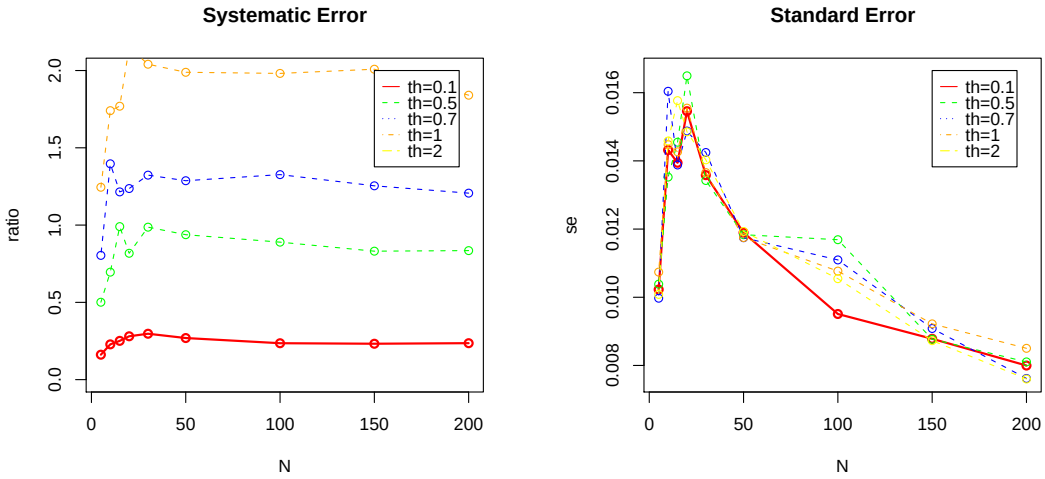


(c) KII, N=100

(d) KII, N=100

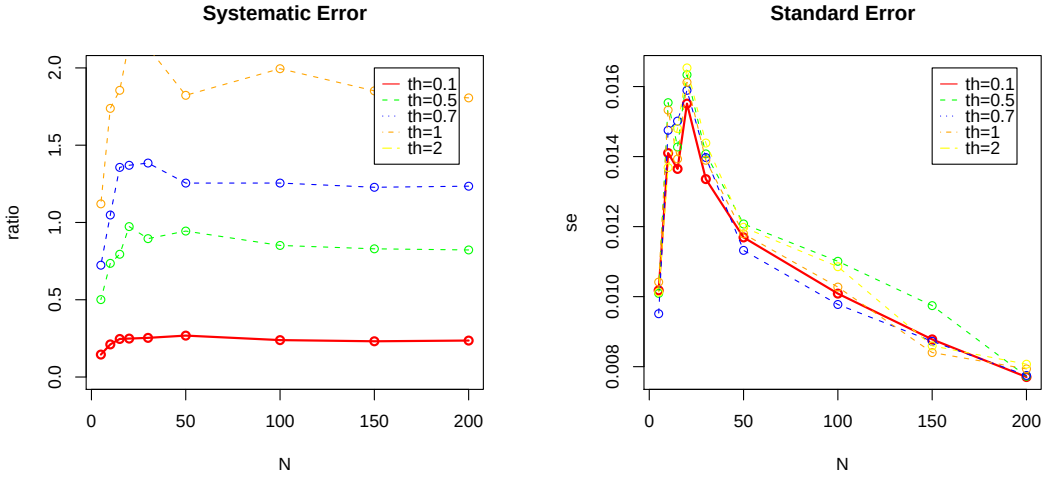
Figure 4.10: [LOG/MI/KNN] Effect of the number of neighbours for different θ.

single best choice for k/N (or number of neighbours for each sample size) that will minimise both statistical and systematic error simultaneously. Finally, figures 4.11 and 4.12 show the error values across sample size (and varying θ values) for KI and KII respectively. For each figure, the top row shows statistical and systematic errors for parameter $\alpha = 1$ and the bottom row for $\alpha = 2$. For the experiments the number of neighbours is $k = 2$. From the plots we can see that for this ratio k/N, both methods show an evident reduction on the standard error as



(a) $\alpha = 1$

(b) $\alpha = 1$



(c) $\alpha = 2$

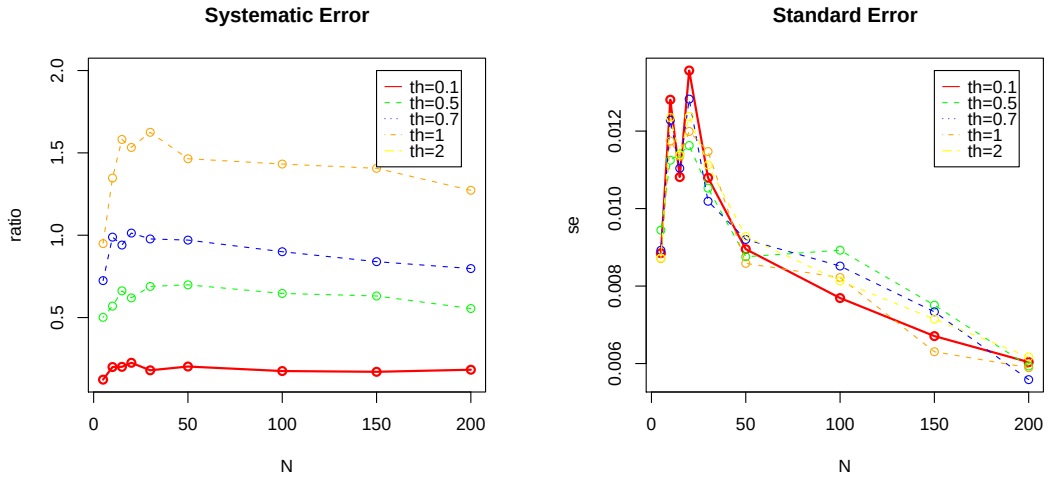
(d) $\alpha = 2$

Figure 4.11: [LOG/MI/KI] KI Error values across sample size for different α and θ.

sample size increases, while the systematic error remains nearly constant. Therefore, for these parameters, θ becomes the most influential factor for the systematic error of both methods. As to the effect of the α parameter, we observe that there is no significant difference between the corresponding error curves for either method.

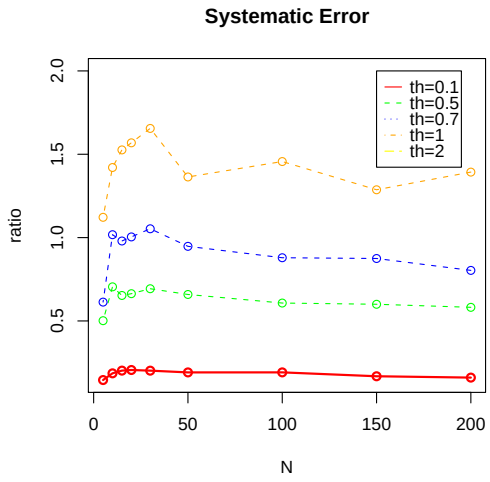
Kernel Density Estimator

In figure 4.13, we assess the effect of increasing bin size (B) to error values for different θ. The sample size was fixed to 100. Standard error seems to be minimum for bin size equal to 2 or

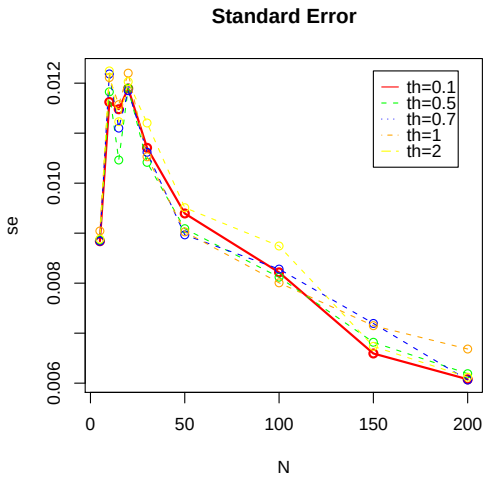


(a) $\alpha = 1$

(b) $\alpha = 1$



(c) $\alpha = 2$



(d) $\alpha = 2$

Figure 4.12: [LOG/MI/KII] KII Error values across sample size for different α and θ.

3 and the bias seems to be minimised at $B = N/3$ for $theta = 0.3$, $B = N/4$ for $\theta = 0.6$ and at $3 < B < N/4$ for $\theta = 0.9$. As with the case of k-NN estimators, "steeper" logistic curves seem to need finer control from the methods, via more samples per bin, or else the result will be significantly overestimated.

On the generalised plots of error across sample size for different values of θ, we kept the number of bins $B = 2$ throughout the experiment. Clearly, for smaller N values we could have increased B to $N/4$ to make the curve appear constant, but either way it is evident that the shape of the bivariate logistic is the critical factor on the bias performance of the method. The same

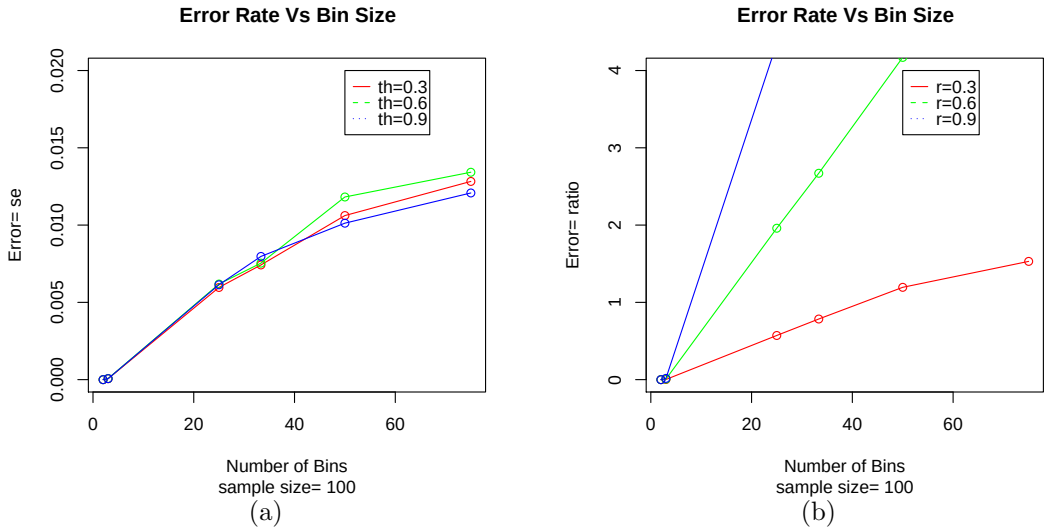


Figure 4.13: [LOG/MI/KDE-bin size] Error Values sample size $N = 100$ for different number of bins and θ.

spread of biases with respect to the θ factor was observed for the k-NN methods; the larger the θ, more the positively biased estimates.

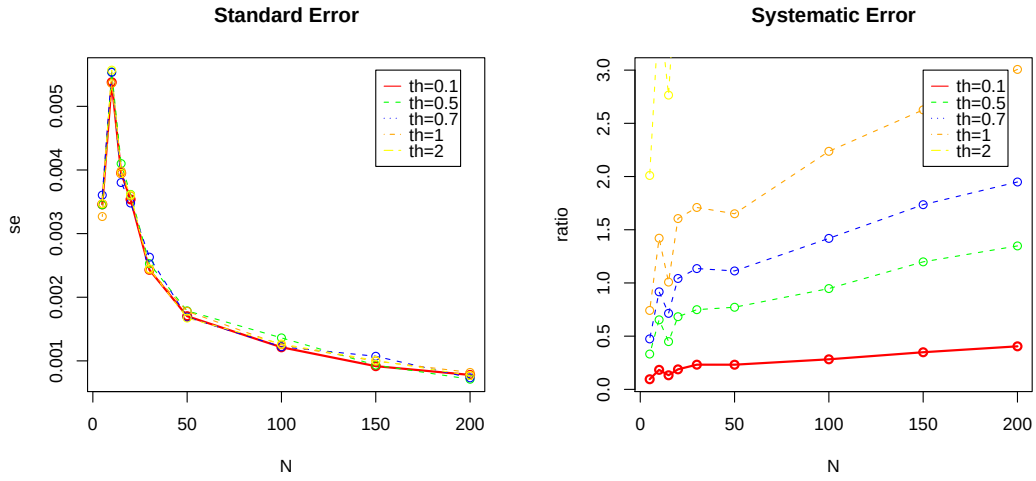


Figure 4.14: [LOG/MI/KDE] Error Values across sample size θ.

Statistical errors on the other hand are affected solely by the number of samples available, diminishing nearly at logarithmic rate for any θ.

4.2 Higher Order Mutual Information

As explained at the beginning of the section to further simplify our tests we restrict higher order(multivariate) mutual information to the trivariate case; i.e. we are focusing on the estimation of $I(X;Y;Z)$, introducing a third random variable Z.

Recalling from section 2, the ensemble of variables X, Y, Z will have joint probability $p(x, y, z)$, joint entropy:

$$H(X, Y, Z) = \sum p(x, y, z) \ln p(x, y, z),$$

and higher order mutual information:

$$I(X, Y, Z) = -\sum p(x, y, z) \ln \frac{p(x, y, z)p(x)p(y)p(z)}{p(x, y)p(y, z)p(x, z)},$$

or in terms of entropies:

$$I(X, Y, Z) = H(X) + H(Y) + H(Z) - H(X, Y) - H(Y, Z) - H(X, Z) + H(X, Y, Z).$$

Therefore, knowing the distribution of each variable, as well as the distribution of their pairwise joint and triple joint ensembles, can lead us to an exact estimate of their joint mutual information, or a degree of their interdependence after, of course, we solve the sum (integral in the case of continuous variables) of the first equation.

4.2.1 Gaussian Distribution

Assuming that variables X, Y and Z are each a Gaussian with zero mean and unit variance, we know that all their pairwise ensembles, as well as the triplet, will also be of Gaussian distribution with the same parameters. Let σ_{ik} $(i, k \in \{1, 2, 3\})$, be their correlation matrix. Then the higher order mutual information (HOMI) of the group is:

$$I(X, Y, Z) = -\frac{1}{2} \log[det(\sigma)]. \tag{4.9}$$

For the experiments shown on the following plots, we used 5 different correlation matrices σ_x.

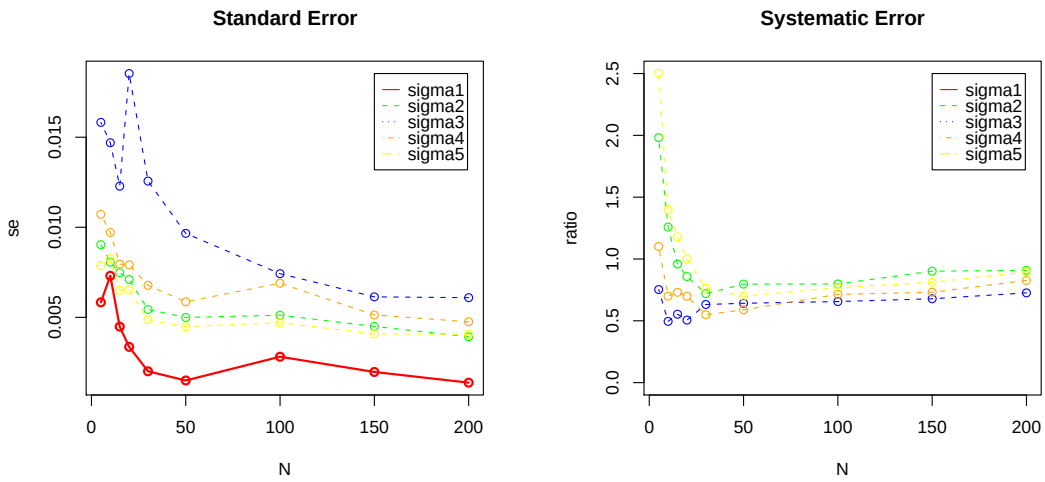
$$\begin{bmatrix} 1 & .01 & .01 \\ .01 & 1 & .01 \\ .01 & .01 & 1 \end{bmatrix}_{\sigma_1} \quad \begin{bmatrix} 1 & .5 & .5 \\ .5 & 1 & .5 \\ .5 & .5 & 1 \end{bmatrix}_{\sigma_2} \quad \begin{bmatrix} 1 & .9 & .9 \\ .9 & 1 & .9 \\ .9 & .9 & 1 \end{bmatrix}_{\sigma_3} \quad \begin{bmatrix} 1 & .3 & .6 \\ .3 & 1 & .3 \\ .6 & .3 & 1 \end{bmatrix}_{\sigma_4} \quad \begin{bmatrix} 1 & .9 & .3 \\ .9 & 1 & .01 \\ .3 & .01 & 1 \end{bmatrix}_{\sigma_5}$$

The first 3 correlation matrices were chosen so as to represent a triplet of variables, bound with common low-to-high degree of linear dependence. The last 2 matrices, represent a slightly more complex dependence structure between the three variables - e.g. relatively low correlation between the 1st and the 2nd variable and low between 2nd and 3rd and relatively high between 1st and 3rd for σ_4. The determinants of the matrices are $0.999702, .5, .28, .568, .1053$ respectively. Especially for the case of the higher order mutual information (HOMI), the determinant of the correlation matrix is the most important factor on the absolute values of statistical and systematic errors, rather than the actual dependence structure. So, measuring HOMI does not reveal a lot about the pairwise dependencies of the ensemble. Instead, it is indicative of the overall interdependence between the variables. Similar correlation matrices ($\sigma_1 - \sigma_3$) were used in [KSG04] and this design facilitates the comparison of numerical results.

Adaptive Partitioning Estimator

Overall, from figure 4.15 above, we see that statistical errors reduce to 0 and systematic errors converge to 1 as we increase the number of samples N, so the method produces nearly exact estimates for sample sizes as small as 200 and onwards.

We need to add however that for σ_1 - very small correlations between the variables, i.e. when all variables are nearly independent - although the statistical error was very low, the method produced strongly (positively) biased measurements - values larger than 2.5 that do not appear at all in the right panel of fig.4.15. From the figure we can also deduce that very small $\sigma_{i,k}$ anywhere in the correlation matrix, introduces positive bias to the results, despite the absolute value of the determinant being intermediate to high (case of σ_5).



(a) (b)

Figure 4.15: [GS/HOMI/AP] Errors across sample size for different σ.

This observation in effect means that the adaptive partitioning method and in particular the condition to exit the splitting recursion, are not sensitive enough to identify the independence of the variables and stop in time. Conversely, this method still produces very good results and is as stable and quick as its 2 variable equivalent.

k-Nearest Neighbours Estimator

Assessing the effect of the number of neighbours for the first k-NN method, we see from figure 4.16 that increasing the number of neighbours decreases the statistical errors for both sample sizes 200 and 30 - figures 4.16(a) and 4.16(c) respectively. The variation on systematic errors indicates the sensitivity of the method to the choice of k, for each sample size and correlation structure. Generally, smaller N requires the support of more neighbours, especially as the dependence between the variables decreases. On smaller sample sizes systematic errors are very high for nearly independent variables (red curve falls outside plot limits). A tabular presentation of the best k values with respect to N and σ is in table 4.1 below:

Repeating the same assessment for Kraskov's second method and sample sizes 60 and 100, we get the curves in fig. 4.17. There we can see that that statistical errors effectively diminish for

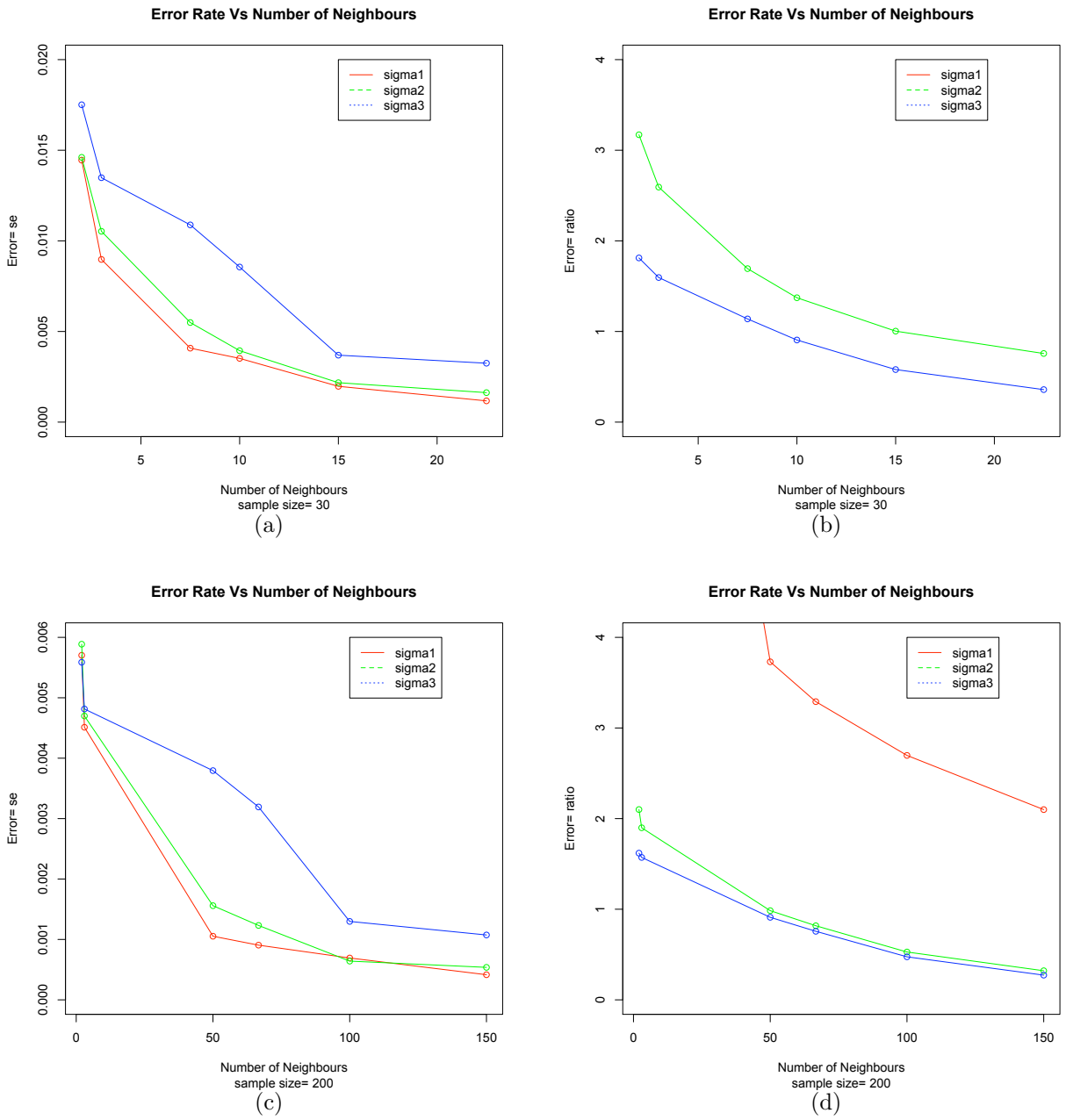


Figure 4.16: [GS/HOMI/KI] Effect of number of neighbours for different σ on KI.

	σ_1	σ_2	σ_3
30	-	$N/2$	$N/3$
200	$> 3 * N/4$	$N/4$	$N/4$

Table 4.1: KI-GS-Optimal Number of Neighbours

$k = N/4$ for both N and all σ. Also, we observe that HOMI is always negative, but can reach an exact absolute value with careful selection of k and provided that σ is not very small. Table 4.2 gives a presentation of the best k values for KII:

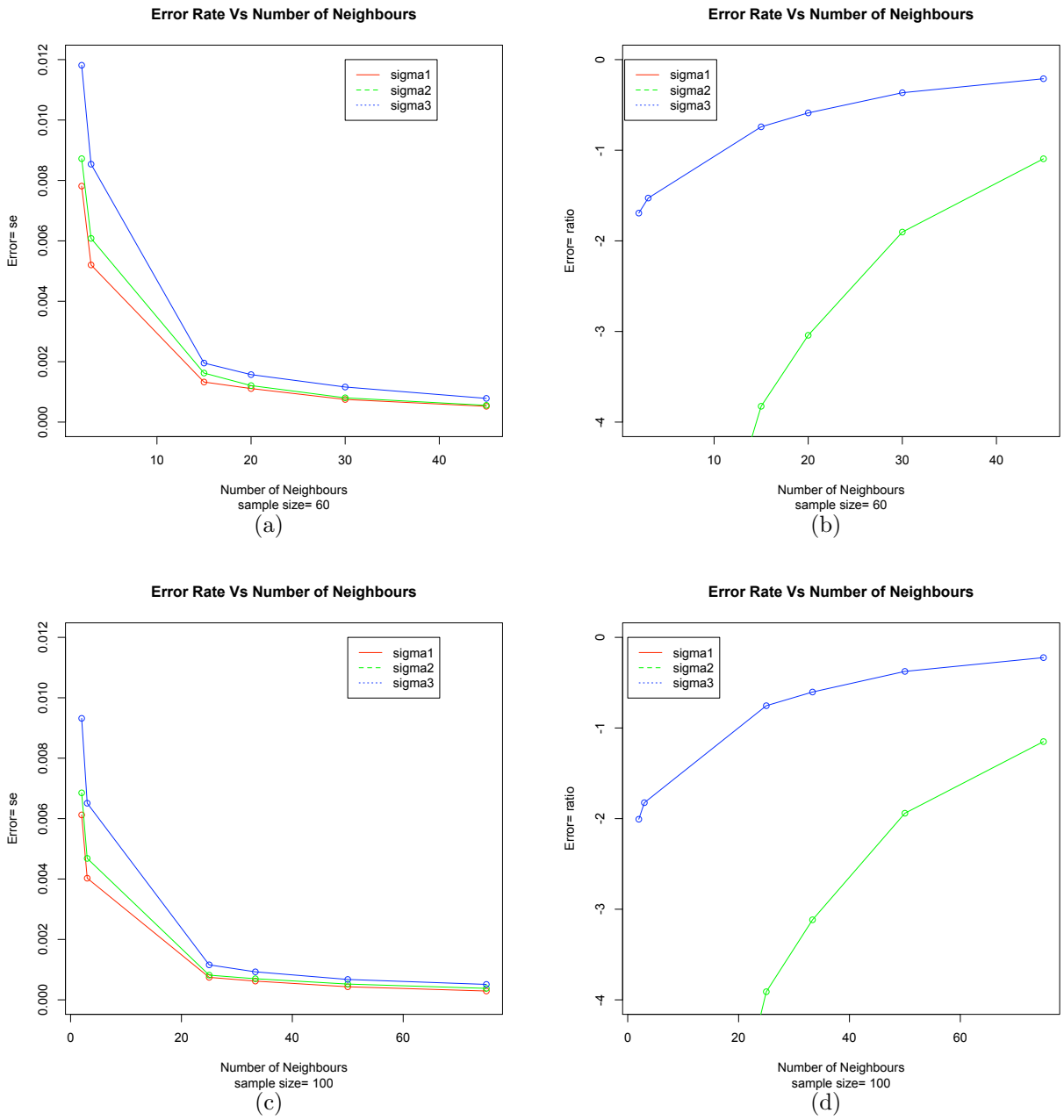


Figure 4.17: [GS/HOMI/KII] Effect of number of neighbours for different σ on KII.

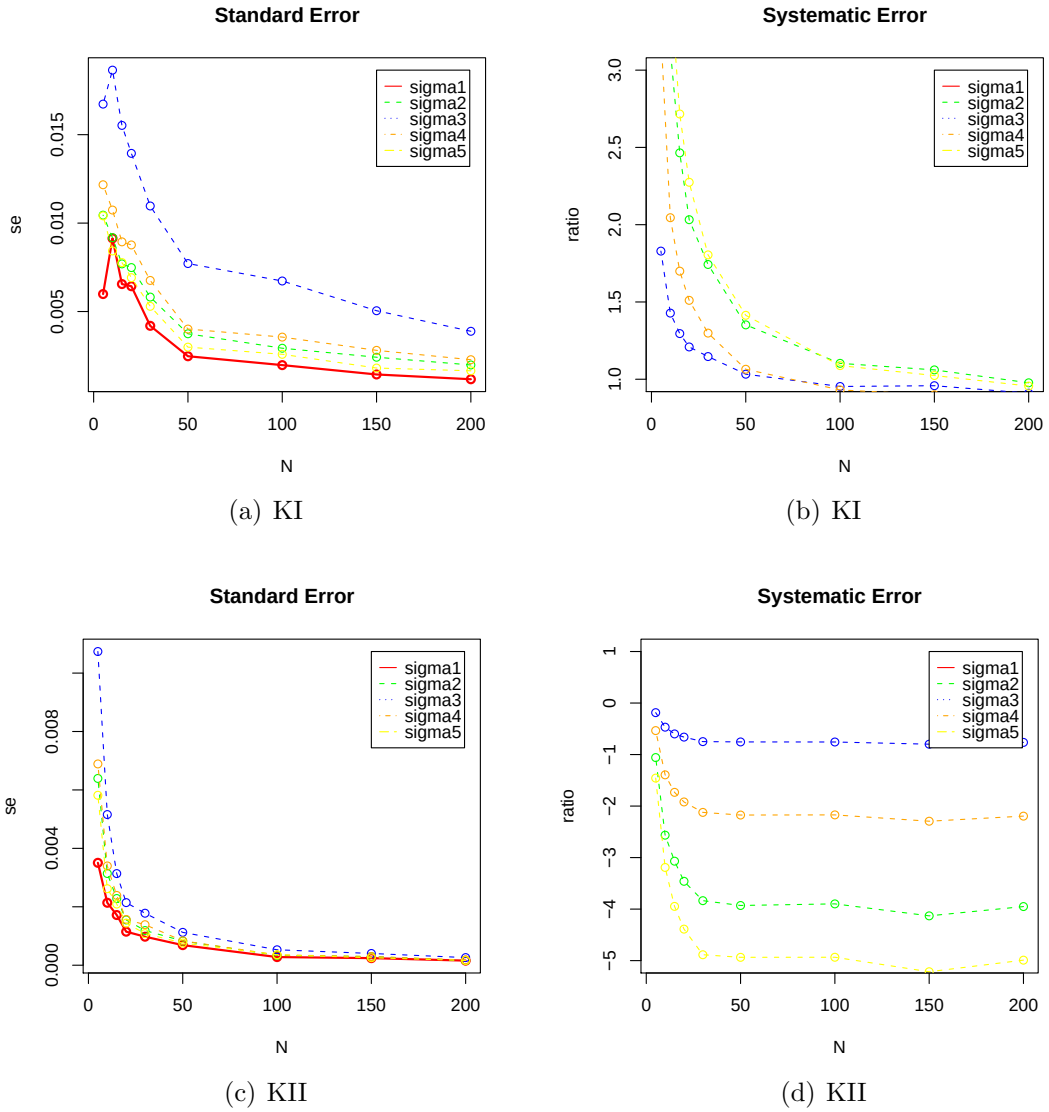
	σ_1	σ_2	σ_3
60	-	$N/4$	$N/4$
100	–	$3 * N/4$	$3 * N/4$

Table 4.2: KII-GS-Optimal Number of Neighbours

Overall, both methods seem to have the capacity of being exact, but depend on a delicate balance between the number of neighbours and sample size. The fact that this balance is also dependent on the parameters of the data distribution, makes the choice of the k/N ratio more

difficult.

This is also evident from plots in figure 4.18 that show the behaviour across sample sizes for different correlation values. Keeping the ratio $k/N = 0.25$ both for KI and KII, we get that KI has higher statistical errors than KII, or in other words KII evaluated for N is roughly equal to KI evaluated for $2N$, reflecting the fact that KII effectively uses smaller length scales, which is one of the conjectures in [KSG04].



(a) KI (b) KI

(c) KII (d) KII

Figure 4.18: [GS/HOMI/KNN] Error values across sample size for different σ.

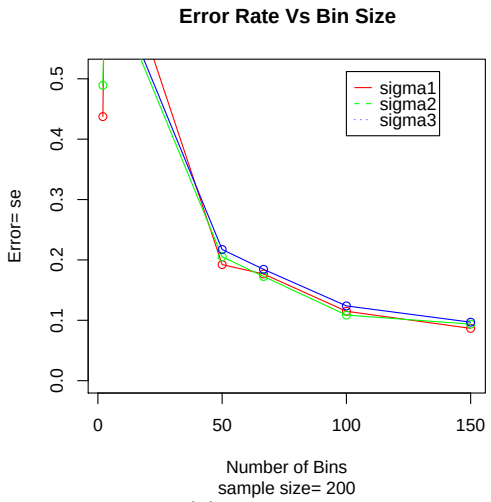
Also, for the same k, the first estimator gives slightly smaller systematic errors in absolute values, that improve by increasing the sample size. Perhaps our choice of σ_1 was too small

compared to the one in [KSG04] (0.01 vs 0.1), but the systematic errors for this correlation coefficient are too high for either method and fall outside the limits of the plot. Conversely, standard errors were very small. Finally, the authors propose to use $k = 2-4$ for either method, except when testing for independence. Their tests though are for sample sizes over 100 points and up to 10^5 or more. For our range of sample sizes this suggestion is obviously no longer valid, but either way both methods seem to be precise and stable with some careful tuning.

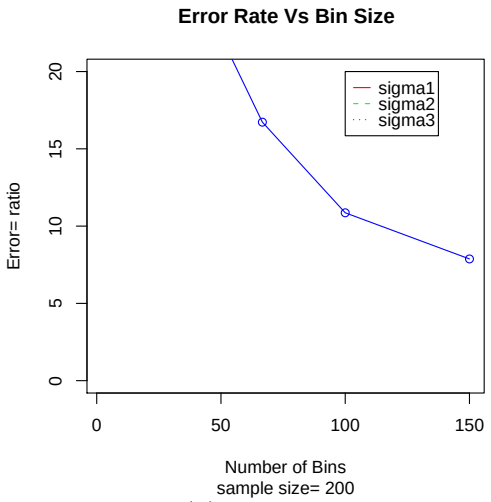
Kernel Density Estimator

To assess the B-spline methodology, we first have to test its behaviour against its parameters, i.e. spline order k and the number of bins B. Figure 4.19 shows how standard and systematic errors vary for different bin sizes when $N = 200$. Each row corresponds to a different spline order $1 - 3$ from top to bottom[3]. From the figure it is obvious that systematic errors are too high for the method to be applied anywhere in our data range, for every spline order tested.

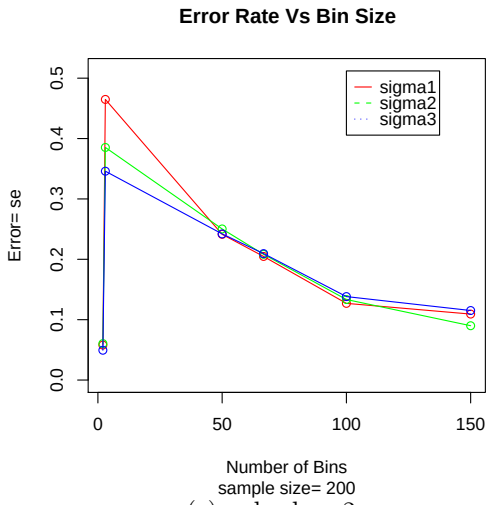
[3]The implemented algorithm is applied to rank ordered data, as in the original publication. However, testing to non-transformed Gaussians, the results are very similar.



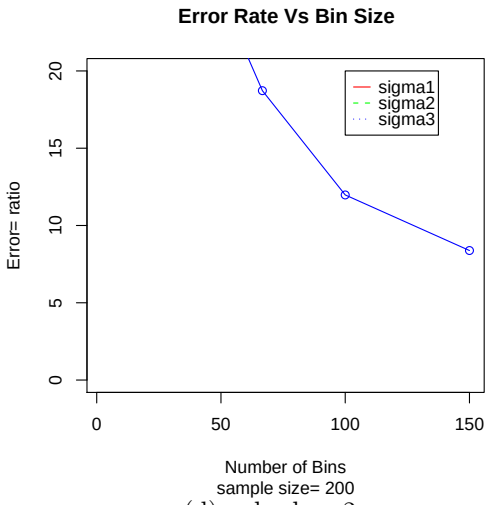
(a) splorder=1



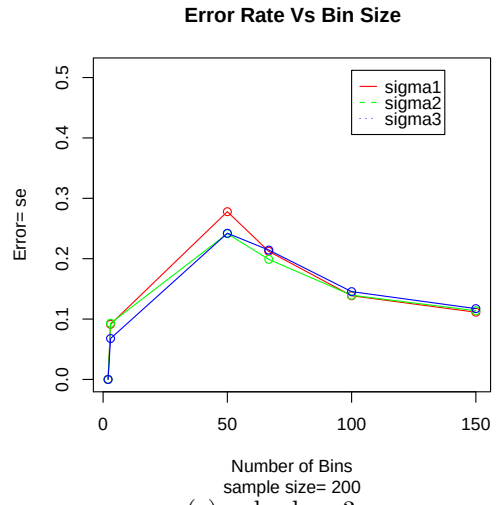
(b) splorder=1



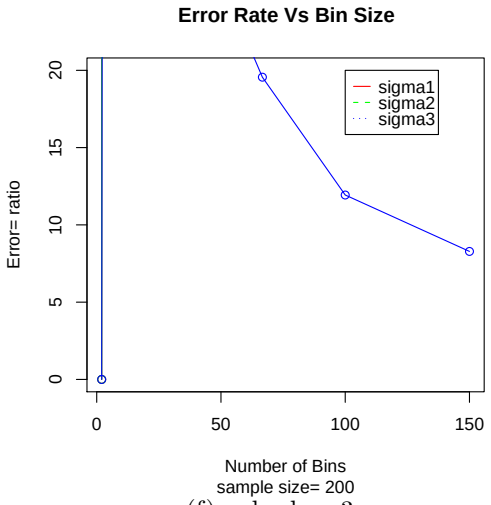
(c) splorder=2



(d) splorder=2



(e) splorder=3



(f) splorder=3

Figure 4.19: [GS/HOMI/KDE-BIN] Errors across sample size for different θ.

4.2.2 Logistic Distribution

As with the case of multivariate Gaussian, we will be estimating HOMI using the formula:

$$I(X,Y,Z) = H(X) + H(Y) + H(Z) - H(X,Y) - H(Y,Z) - H(X,Z) + H(X,Y,Z).$$

The analytic form of the entropy for the logistic distribution provided by Darbellay and Vajda in [DV00]is:

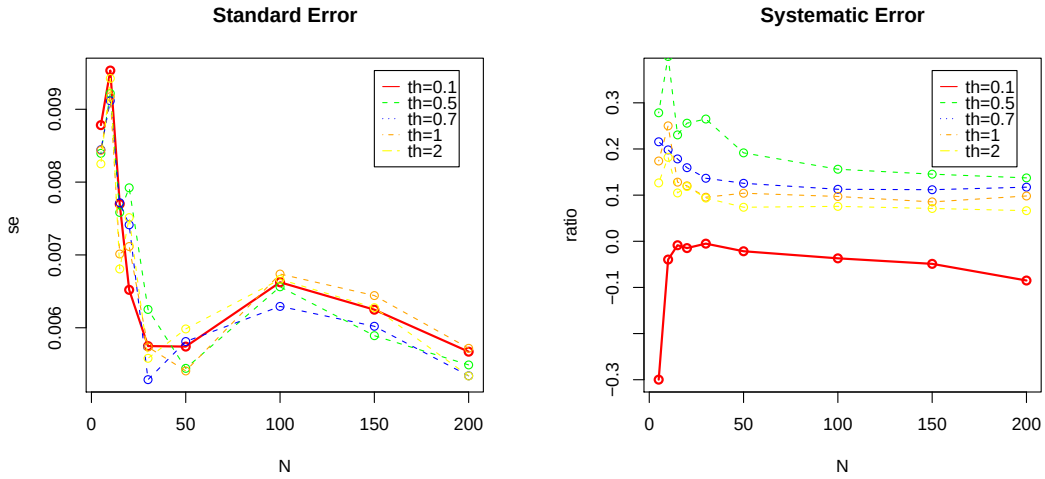
$$-\sum_{i=1}^{n} \ln\left(\frac{\alpha+i-1}{\theta_i}\right) + (\alpha+n)\sum_{i=1}^{n}\frac{1}{\alpha+i-1} + n[\Psi(\alpha) - \Psi(1)] \ . \tag{4.10}$$

where $n = 3$ and α is the shaping parameter. For the set of experiments on the following graphs $\alpha = 1$. Other sets of experiments were carried out for varying values of α $(2, 3, 4)$, but whenever the results where not remarkably different, they are not shown. As with the bivariate case, the different curves in the figures, correspond to different values of the θ parameter of the exponential distribution, used to create the multivariate logistic.

This final part of the comparative study, is also the most challenging, as the methods are assessed both on the grounds of very small samples and on data that deviate significantly from the 'normality' of Gaussians. We have to point out, that all the methods produced very small HOMI estimates often close to 0, hence systematic errors which are in effect the ratio $\widehat{I_{estim}}/I_{exact}$ also take near zero values. Finally, although not shown here, the error bars for each estimate were also larger than the case of Gaussians (bivariate of trivariate) and the bivariate logistic especially for sample sizes < 50.

Adaptive Partitioning Estimator

From figure 4.20(a), we see that $\widehat{I_{estim}}$ converges slowly to $1/10th$ of the true value, as we increase the sample size. Standard errors, although very small, present erratic behaviour, with a high peak at $N = 100$ which disappears as N increases. This gross under-estimation of the true HOMI, is consistent with the findings of the comparative results presented in [KSG04] .



(a) (b)

Figure 4.20: [LOG/HOMI/AP] Errors across sample size for different θ.

k Nearest Neighbours Estimator

As above, we are first testing the methods against different numbers of neighbours k and different θ, to find if there is an optimal ratio k/N that minimises statistical and systematic errors. We tested the first Kraskov method on $N = 50$ and $N = 100$ - figures 4.21(a)-4.21(b) and 4.21(c)-4.21(d), respectively. For both sample sizes, we see that the method actually produces negative HOMI estimates, that in absolute value come closer to 1 for $k = 2$. Since statistical errors seem to be very small, we would choose that value of k as the number of neighbours at which KI performs best.

KII was tested on $N = 100$ and $N = 200$ - figures 4.22(a)-4.21(b) and 4.21(c)-4.21(d) respectively. Again, statistical errors were very small (even half the size of KI) and HOMI estimates improved, when keeping the k/N small. $k = 2$ was again the best value. Even at that value though, the estimator KII only got at about 40% of the actual HOMI value, which is still far better than KI, which got about 6% of the exact HOMI value.

Fixing the number of neighbours at 2, we assess the performance of both methods across sample sizes for different correlation values. From figure 4.23(a)-4.23(b), we observe that KI converges to a (very low) finite value after $N = 50$ for any θ. The second method however, seems to

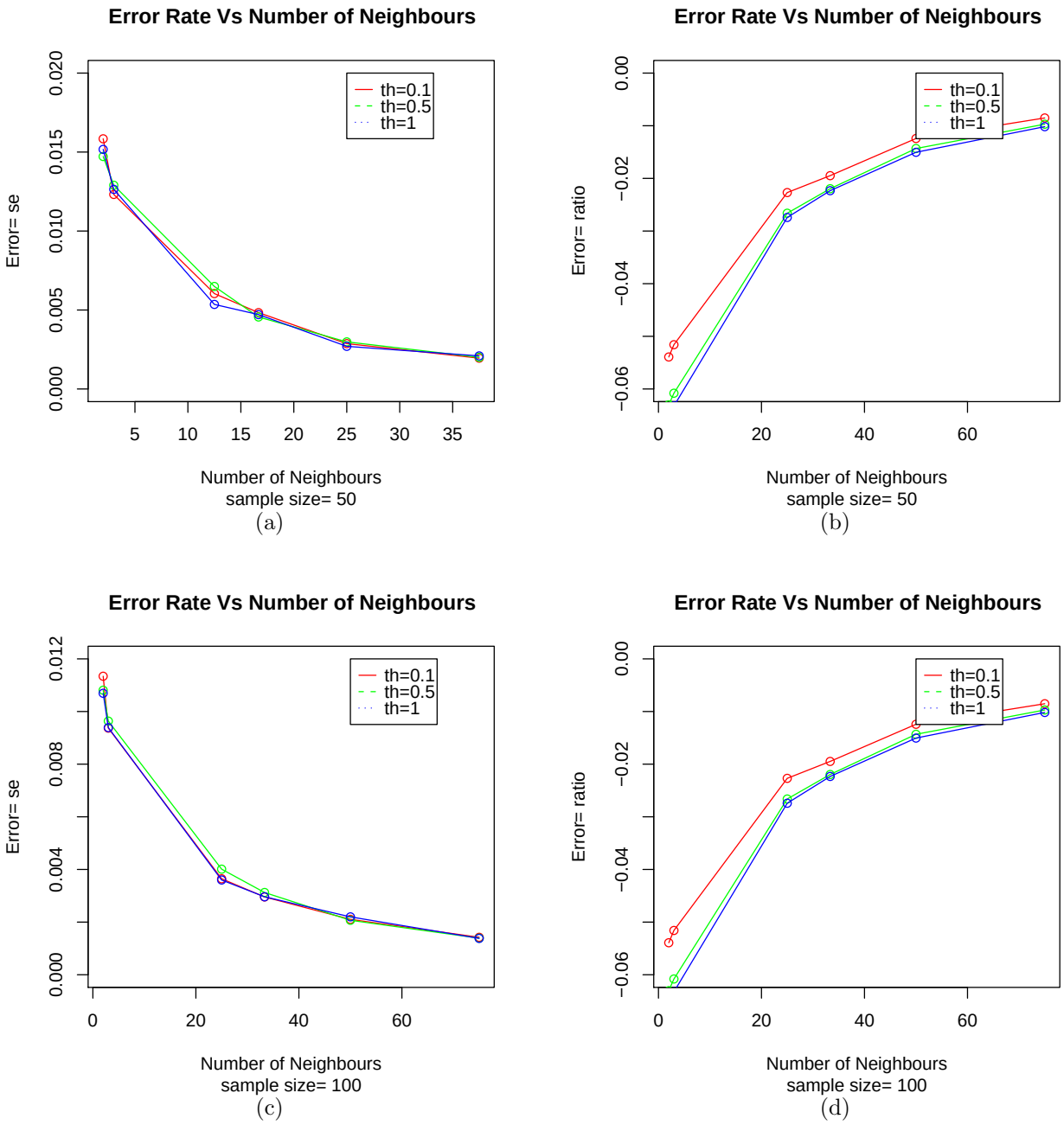


Figure 4.21: [LOG/HOMI/KI] Effect of number of neighbours for different θ.

improve its approximation, as we increase the sample size - again for all θ. The scaling factor appears to be close to $log(N)$. Systematic errors also appear to improve as θ increases. Overall, for the same k the second estimator gives smaller statistical errors and a much less bias most probably because n_x and n_y tend to be larger and have smaller relative fluctuations which turns the method more sensitive to the complex structure of the data.

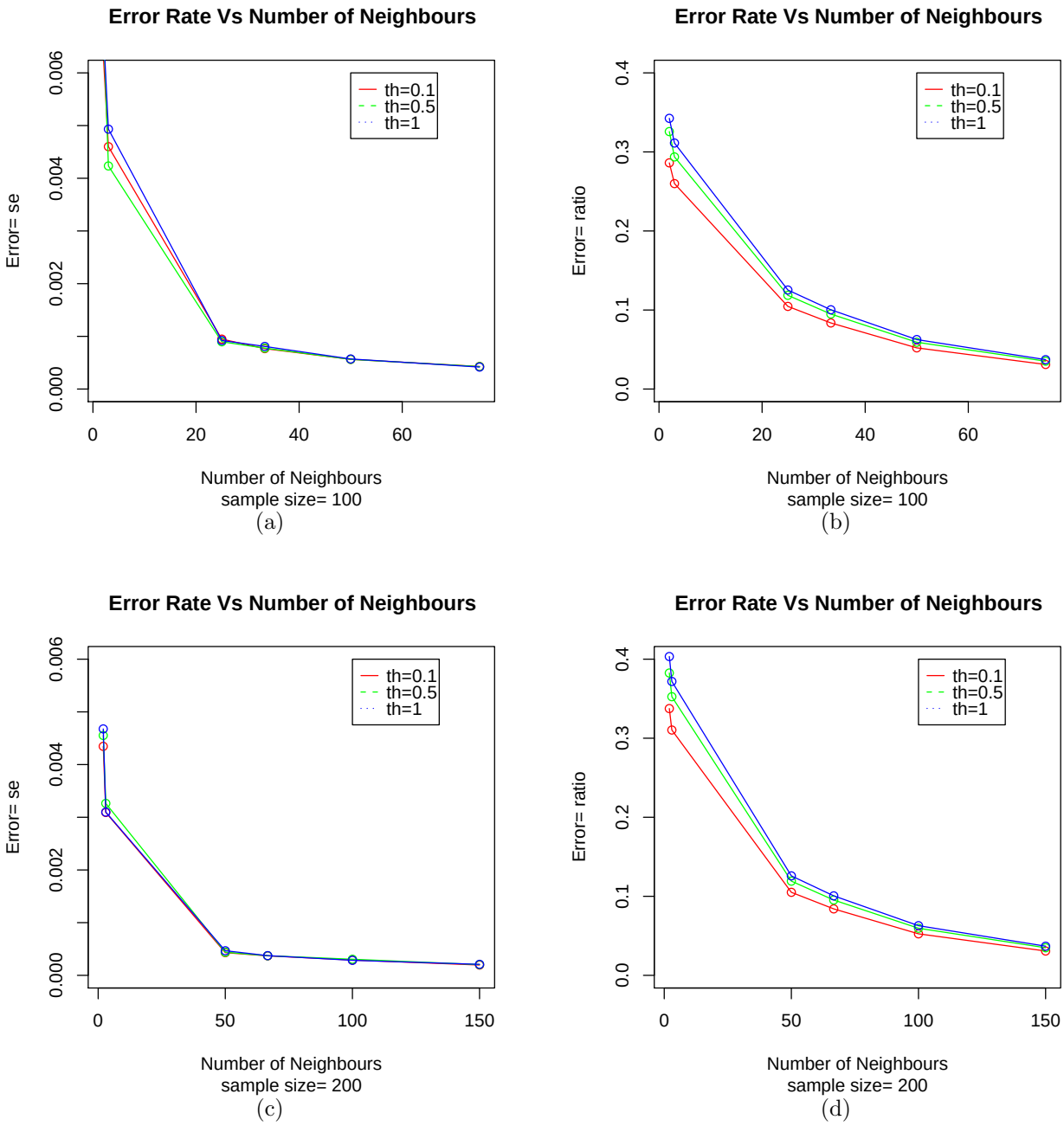
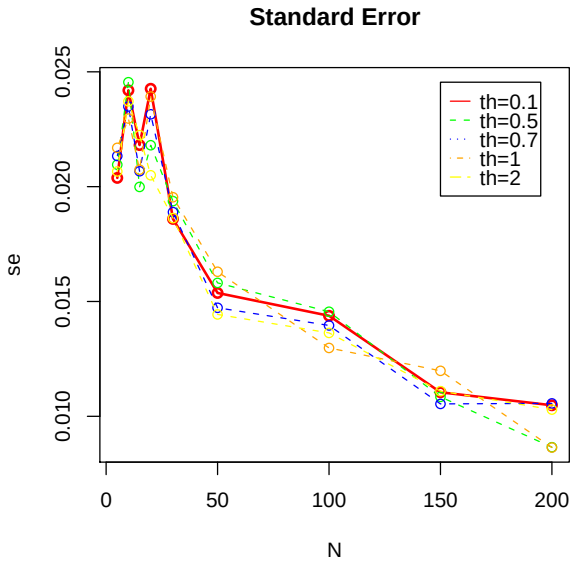
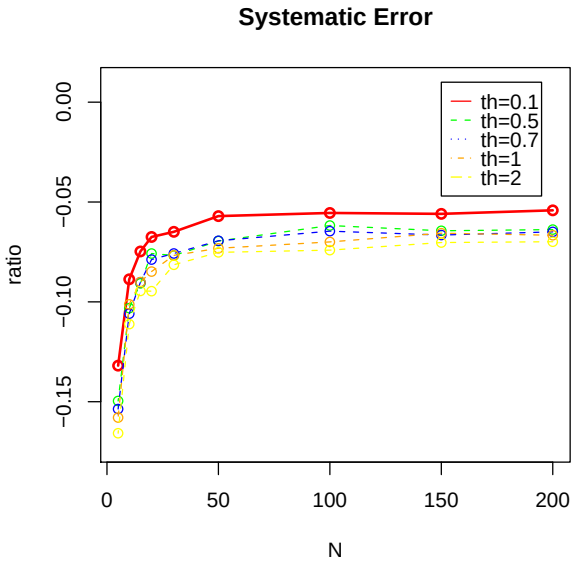


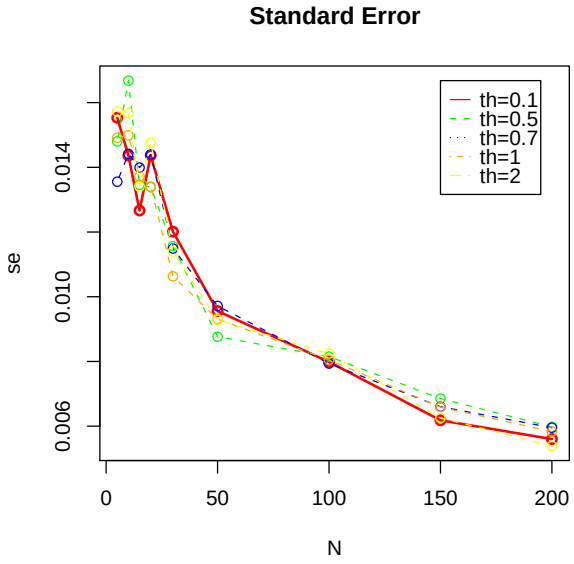
Figure 4.22: [LOG/HOMI/KII] Effect of number of neighbours for different θ.



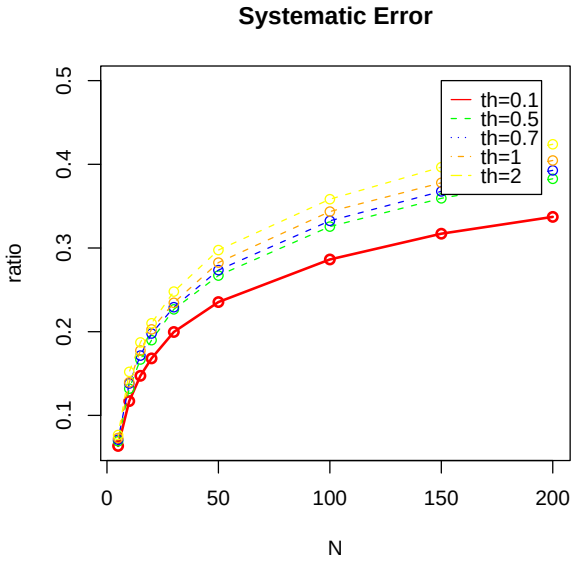
(a) KI



(b) KI



(c) KII



(d) KII

Figure 4.23: [LOG/HOMI/KNN] Error values across sample size for different θ.

Kernel Density Estimator

To assess the B-spline methodology, as before, we first have to test its behaviour against the its parameters, i.e. spline order and the number of bins B. Figure 4.24 shows the error values

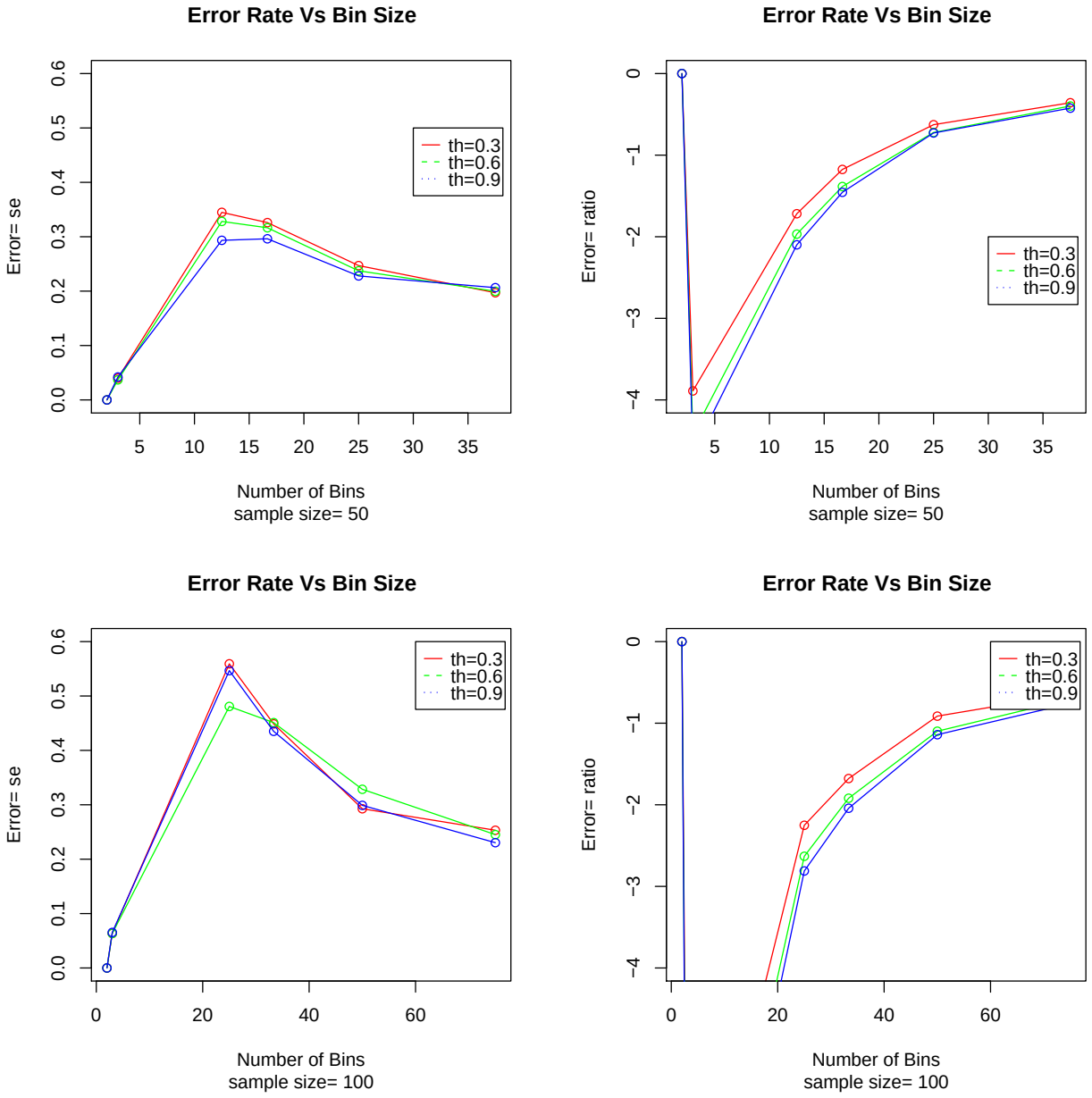


Figure 4.24: [LOG/HOMI/KDE-bin size] Error values for sample sizes $N = 50$ and $N = 100$ across bin size for different θ.

at different bin sizes, for different θ. We chose $N = 50$ and $N = 100$ for our tests. There we can see, that systematic error increases as we decrease the number of bins per sample size. The optimal values of data per bin, are displayed in the table 4.3 below. We note that $B = N/2$,

	$\theta = 0.3$	$\theta = 0.6$	$\theta = 0.9$
50	$N/2 - N/3$	$N/2$	$N/2$
100	$N/2$	$N/2$	$N/2$

Table 4.3: SPL-LOG-Optimal Number of Neighbours

brings the absolute value of systematic error close to 1 for every θ. This ratio however, is not the optimal for standard error. For this, the best value is 2.

Fixing bin size to $N/2$ for the same N we used above, we now plot the effect of spline order on standard and systematic error - figures 4.25(a)-4.25(c) and 4.25(b)-4.25(d) respectively. In this figure, we observe that increasing the spline order increases almost linearly the standard error, but decreases, again almost linearly, the systematic error. The slope of the change is more steep for smaller N. Despite this trend, the optimum spline order - that both minimises standard error and brings to 1 systematic error - is 1, for both sample sizes and almost all (except $\theta = 0.3$) θ. Finally, from the aggregated plot of errors across sample sizes for different θ, we set $B = N/2$ for all N and spline order equal to 1. Systematic errors seem to improve and the method becomes exact as sample size increases. Unfortunately, fixing the ratio B/N to 0.5 causes the standard error to increase, as we increase the sample size (since B has to increase as well). However, the importance of systematic error outweighs the importance of the statistical error (especially when standard error is that small anyway), therefore we are happy to make this compromise.

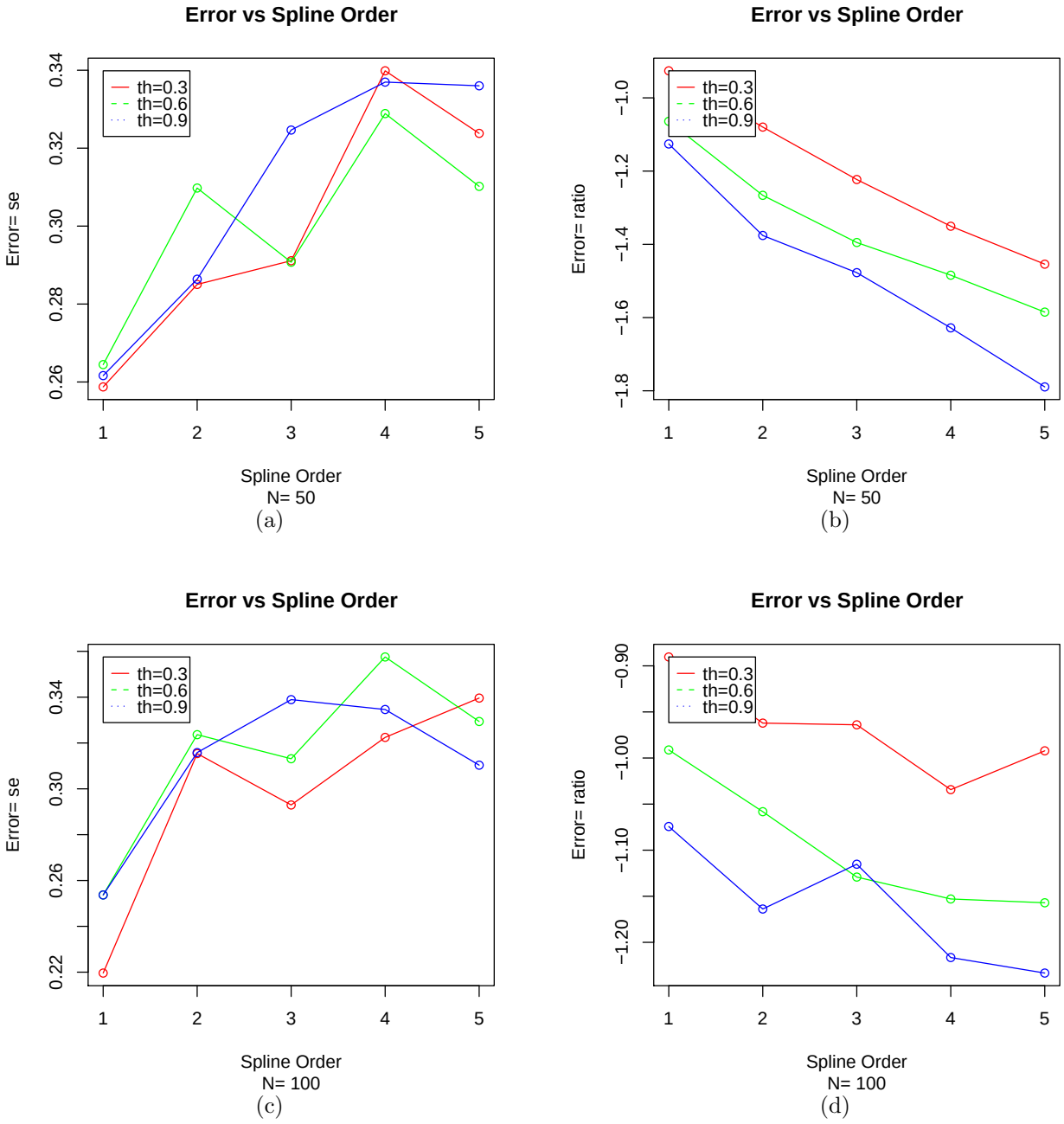


Figure 4.25: [LOG/HOMI/KDE-Spl Order] Error values for sample sizes $N = 50$ and $N = 100$ across Spline Order for different θ.

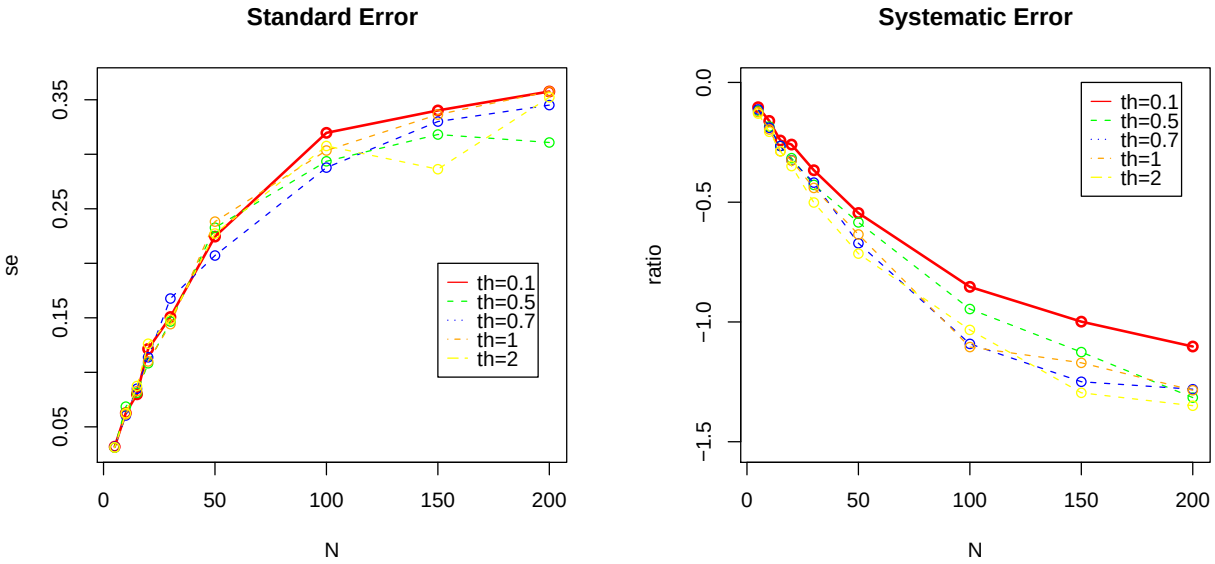


Figure 4.26: [LOG/HOMI/KDE] Error values across sample size for different θ.

4.3 Comparison of all the methods

For the final part of our investigation, on the performance aspects of the methods, we will be aggregating the individual error statistics of each method under one common experimental set-up. Moreover, we will include error measurements from an implementation of the **naive histogram** method, to assess if and how we gain, by using the more sophisticated methods on the calculation of information theoretic functionals, in undersampled data applications.

This comparative study is consolidated into four figures; one for each different part of our work:

- Mutual Information & Gaussian Distribution (fig.4.27),
- Mutual Information & Logistic Distribution (fig.4.28),
- Higher Order Mutual Information & Gaussian Distribution (fig.4.29) and
- Higher Order Mutual Information & Logistic Distribution (fig.4.30).

Each figure is composed of six plots: the left-column plots display statistical errors and the right-column plots display systematic errors. Each row, corresponds to a different parameter setting for the distributions under study.

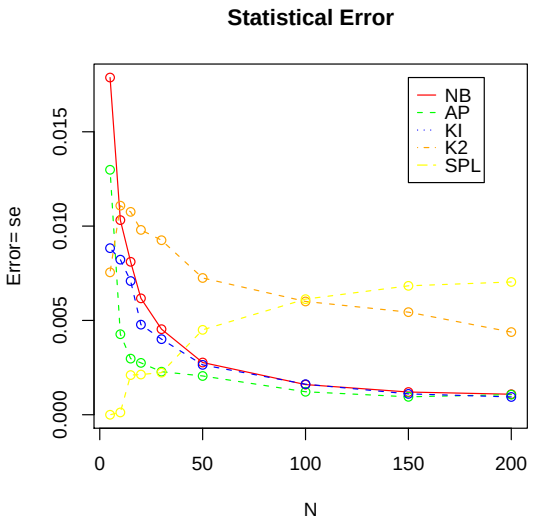
As for the parameters of the individual methods, like number of neighbours **KI** and **KII**, spline order and bin size for the KDE method, we tried to pick values for which the methods displayed their average performance - rather than fine tuning - to ensure some level of fairness.

Figure 4.27, shows that the adaptive partitioning method of Darbellay and Vajda [DV99] is almost exact for every correlation coefficient and from quite early on the sample size scale ($N > 20 - 30$). For smaller N systematic error fluctuates between 3 and 0.5 which is still acceptable. The fact that it operates tuning-free and is the fastest of all the methods participating in this comparison makes it a clear winner. For KDE and k-nn methods keeping the neighbour over sample size ratio at a specific value we can more or less reach their 'best' value or converge to 1 at a very slow rate. The adaptive partitioning and k-nn methods consistently perform better than the naive histogram method with every correlation coefficient.

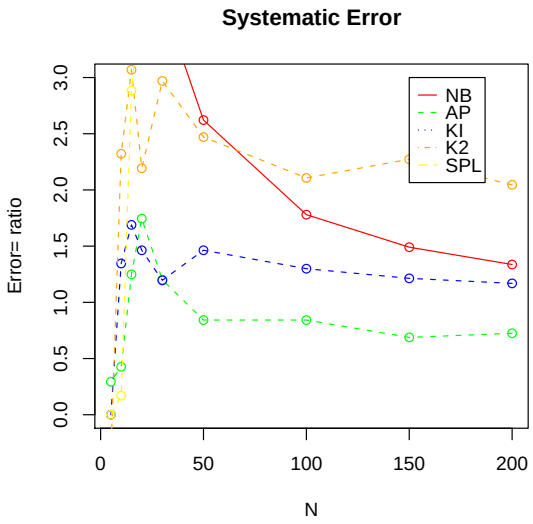
Figure 4.28, shows that the adaptive partitioning method again converges to near exact values as N increases. The naive histogram method is also unbiased for $N > 50$ and has the lowest statistical error as well. k-nn methods consistently over-estimate MI with KI slightly more elevated. The KDE method was found to have both bias and statistical error scaling of e^N.

Figure 4.29, shows that the KI and AP methods are nearly exact for $N > 20$ and KDE consistently consistently over-estimates HOMI for every σ. Oddly enough the naive histogram method over-estimates HOMI for Gaussians with correlation matrices σ_1 and σ_2 but is exact for Gaussians with correlation matrix σ_3.

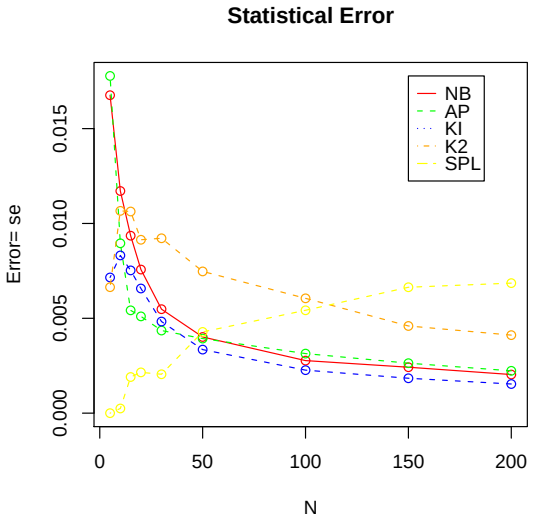
Figure 4.30, is the last set of plots showing that the performance of AP, KI and NB is similar for all θ values under-estimating the true HOMI. Again the estimates of KDE method are significantly under the predicted values and scale to lower values as N increases. The KII estimator seems to converge asymptotically to exact values.



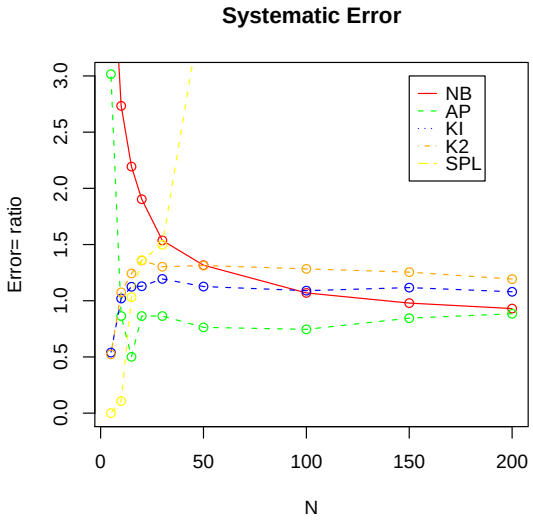
(a) $\rho = 0.3$



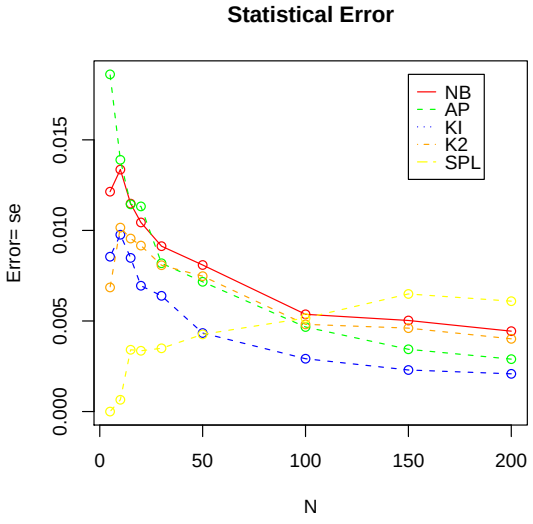
(b) $\rho = 0.3$



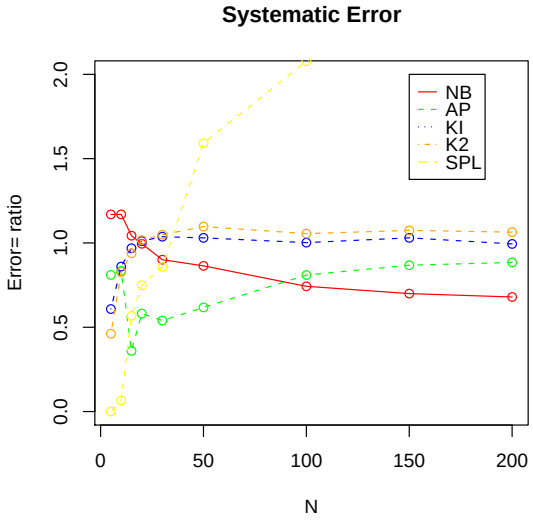
(c) $\rho = 0.6$



(d) $\rho = 0.6$

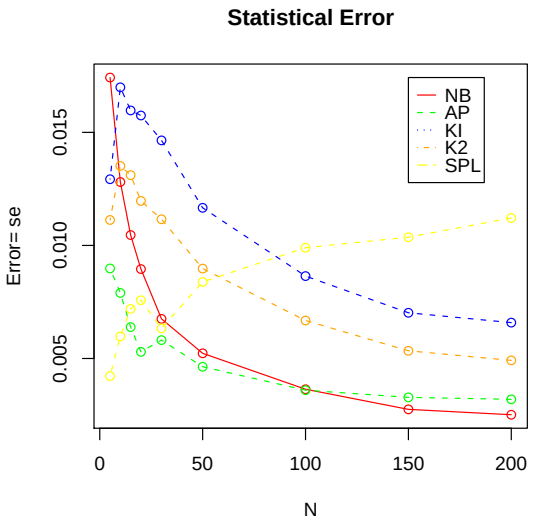


(e) $\rho = 1$

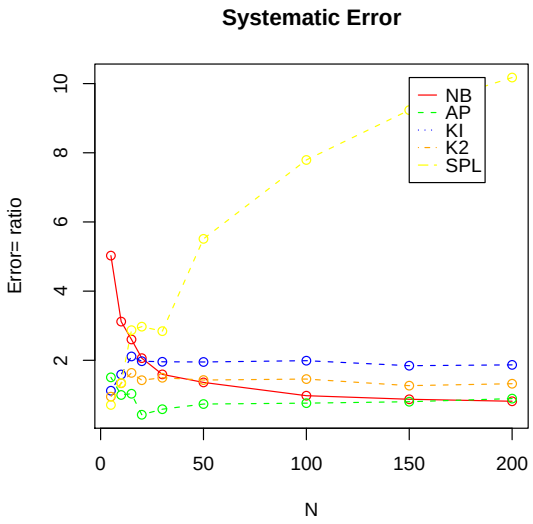


(f) $\rho = 1$

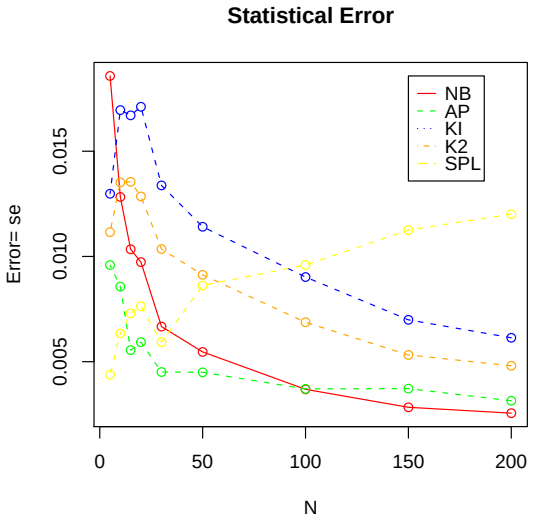
Figure 4.27: [GS-ALL-MI] Standard and systematic errors of all the methods for different ρ, across N.



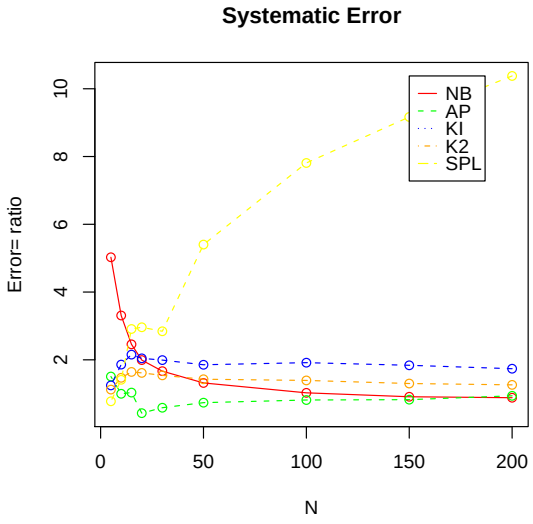
(a) $\theta = 0.3$



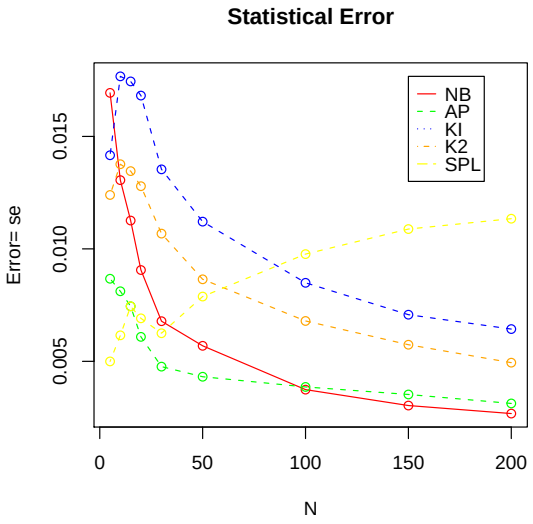
(b) $\theta = 0.3$



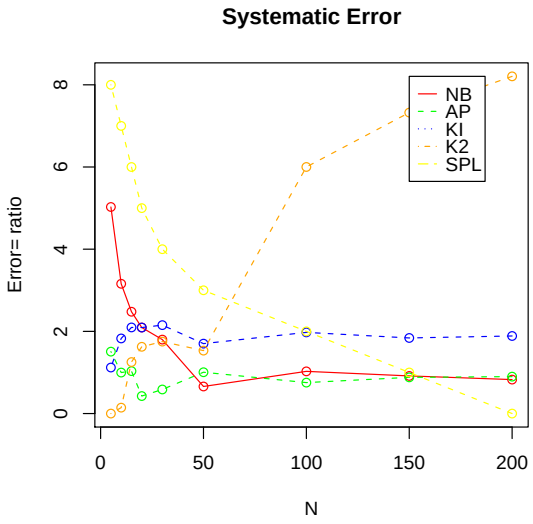
(c) $\theta = 0.6$



(d) $\theta = 0.6$

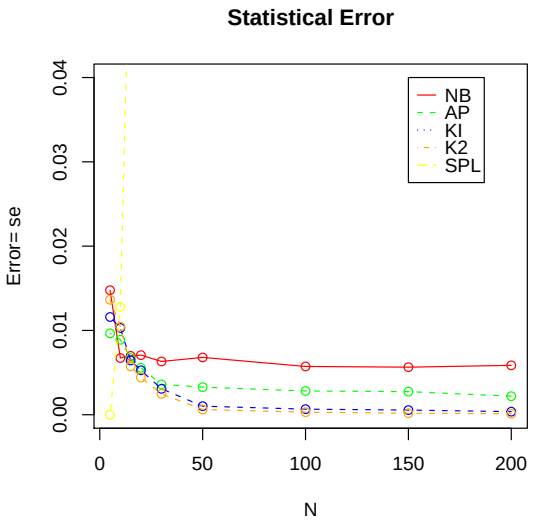


(e) $\theta = 1$

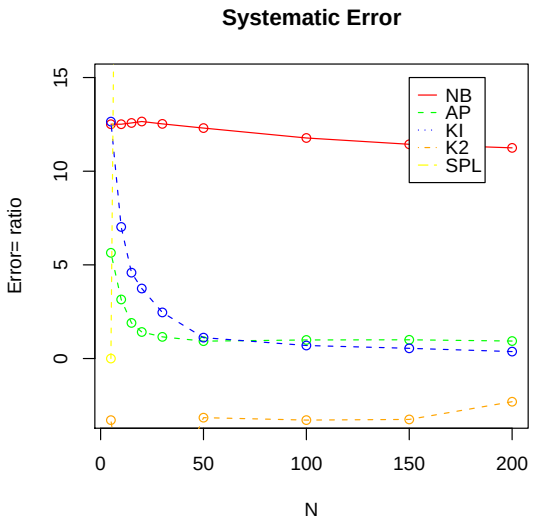


(f) $\theta = 1$

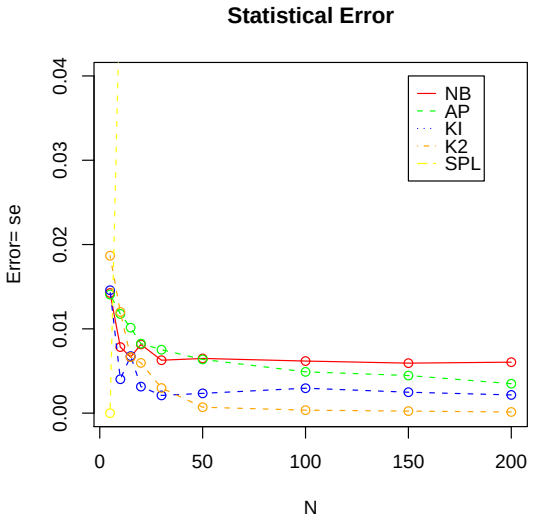
Figure 4.28: [LOG-ALL-MI] Error values of all the methods for different θ.



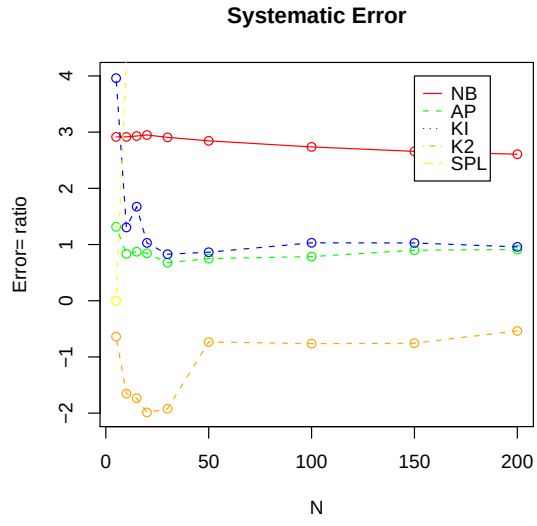
(a) σ_1



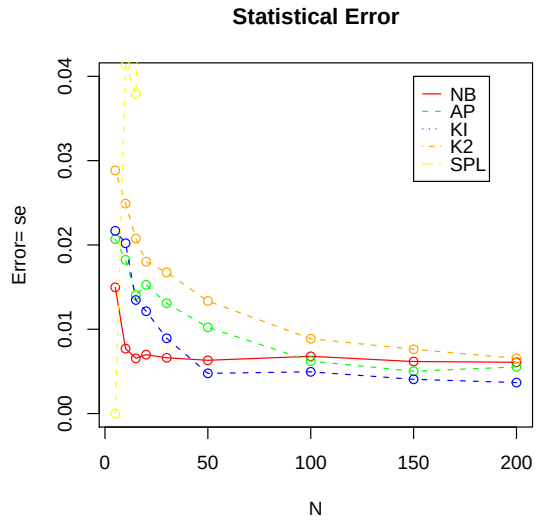
(b) σ_1



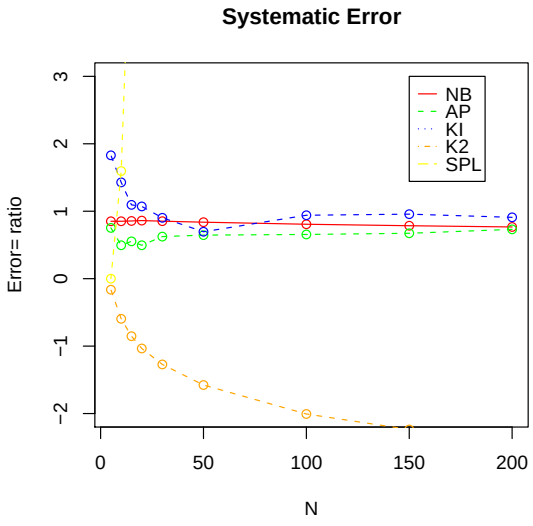
(c) σ_2



(d) σ_2

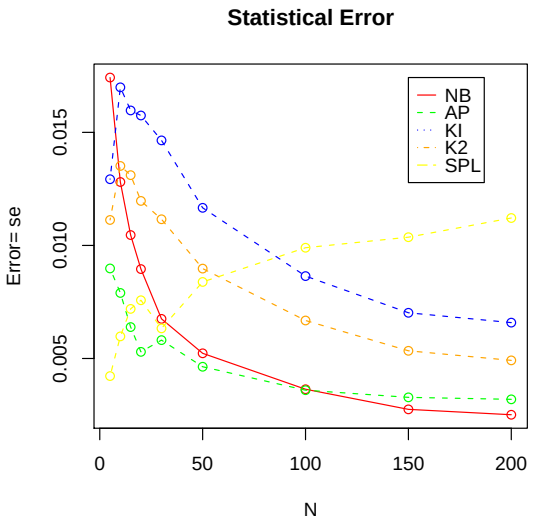


(e) σ_3

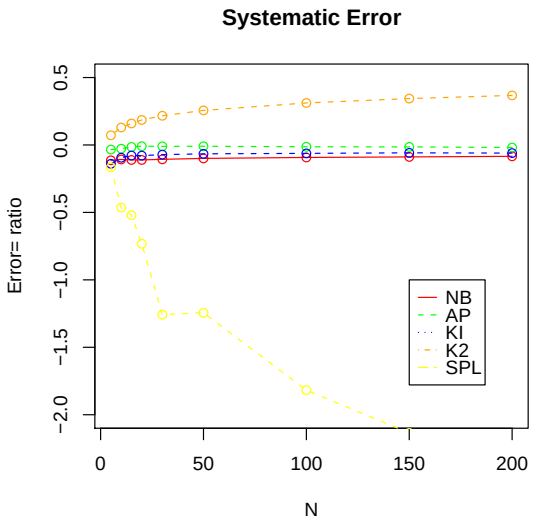


(f) σ_3

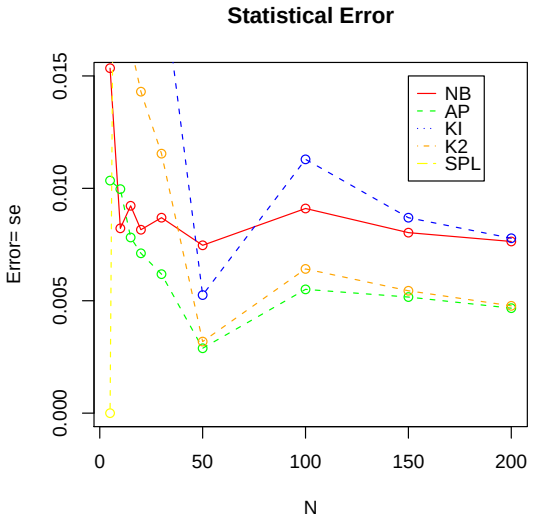
Figure 4.29: [GS-ALL-HOMI] Error values of all the methods for different σ.



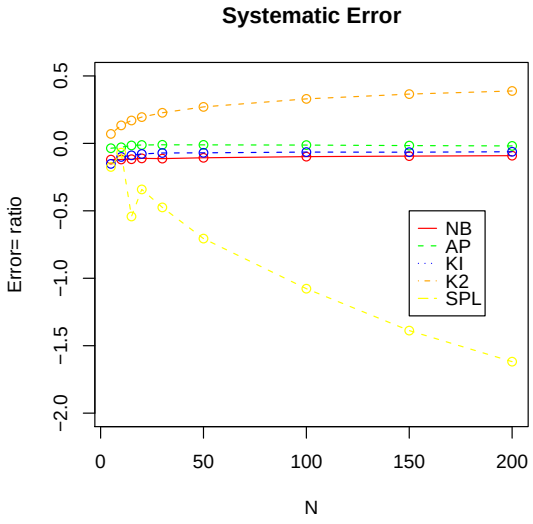
(a) $\theta = 0.3$



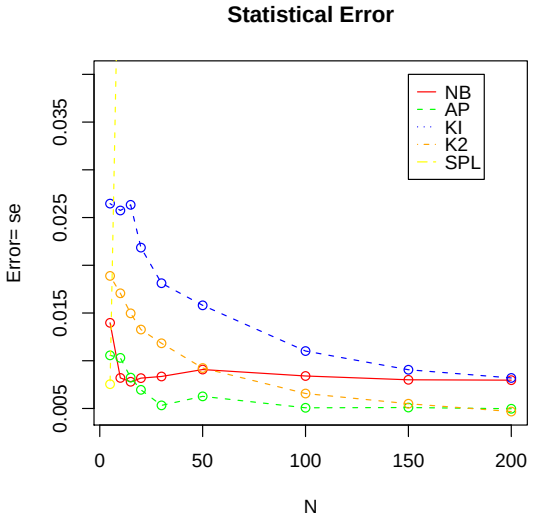
(b) $\theta = 0.3$



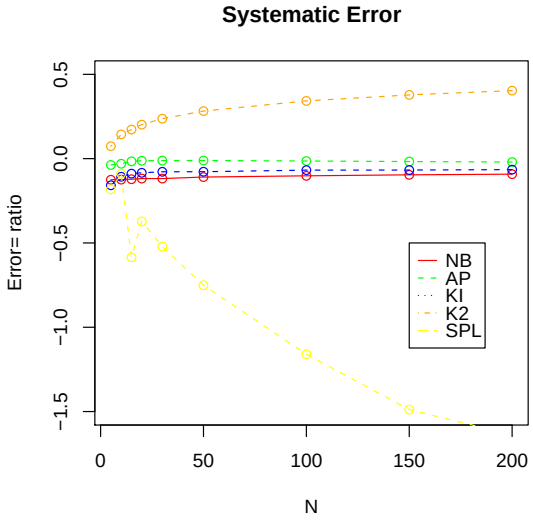
(c) $\theta = 0.6$



(d) $\theta = 0.6$



(e) $\theta = 1$



(f) $\theta = 1$

Figure 4.30: [LOG-ALL-HOMI] Error values of all the methods for different θ.

4.3.1 Discussion

The following list summarizes the key observations on the performance of each method, following by our benchmarking methodology.

Naive Binning (NB): This popular method, is very easy to implement and quite fast in computational time. However, it is shown that in small sample settings, it is not universally consistent. We consider a method inconsistent, when the results show an opposite trend to the one that they are supposed to show, e.g. systematic error diverging from near unit values, when we increase the sample size (see figure 4.27(f)).

KDE-based method by Daub et. al (KDE): This method is theoretically an improvement to the standard binning method, by solving the issue of choosing optimal bin width found with NB. It's implementation is quite uncomplicated and R code for the bivariate case, is provided by the authors. In practice however, it is very slow, very sensitive to the choice of parameters and not universally consistent. Further tests in the following chapter, add more evidence to these findings.

k-nn based method by Kraskov et. al. (KI): Both these methods, are rather difficult to implement and very slow. As they are sensitive to the choice of k, extensive prior testing is required to find the optimal value, but once this is found, the methods are consistent. Usually positively biased and it converges to a nearly constant value after $N = 30$.

k-nn based method by Kraskov et. al. (KII): Like the first variation, KII is sensitive to the choice of k, but once we find an optimal value it performes better than KI in the dedicated study and it is second best overall.

Adaptive Partitioning (AP): The adaptive partitioning method, is fairly straightforward to implement and it is very fast, without any dependent parameters. Furthermore it is consistent and nearly exact, with minimal standard error, in every case. According to this parametric benchmark, It is the best alternative to naive binning.

Chapter 5

Synthetic Networks

For the first part of the performance analysis of our selected algorithms in chapter 4, we focused on the study of their practical properties and their statistical assessment with parametric models of expression. Arguably, this exploratory investigation is not perfect, as parametric scenaria may lack the features and complexity of real expression data. Ideally, we would like to see how these estimators compete with other estimators in the literature, beyond calculations of MI or HOMI, possibly in the context of structure learning algorithms and genetic network inference, like the methodologies presented in our background section.

Abstracting away from the technical details of each work in section 2.2, we can see that most methods generate the genetic networks by sequentially applying: a) a function to estimate mutual information between genes and b) a method to infer the transcriptional genetic network (TRN).

The main idea behind this chapter is to keep our proposed estimators as plug-in components for the first part of this framework and reuse the network inference component for the second part. This way, we can create alternative networks and thereby assess the usefulness of the algorithms in a biological context.

Similar to the parametric testing, statistical knowledge about the network performance of the estimators requires repeated tests on high-quality data sets obtained from many experimen-

tal conditions, preferably derived from different well-characterized networks. Unfortunately though, experimental data sets available from public databases or statistical packages, rarely meet these requirements. Moreover, straightforward application to real data can only create partial confidence in the algorithm since, with real data we lack knowledge about the phenomena under study; therefore conclusions about the method itself are entangled with our uncertainty about the underlying biology [VdBVLN+06].

Since here the emphasis is on the objective assessment of the various estimators rather than on the recovery of novel biological information, we suggest the use of synthetic expression data for which the underlying network is known. The main advantage of synthetic expression data, is that the generating process is supposedly well known and so one is able to judge the relative merits of the methods, unhampered by the lack of ground truths and the complexity of biological processes. Having said that, however, does not imply that we intend to blindly use data of any size or design. Use of simplistic or untruthful data would offset our commitment to realism and defeat the purpose of this assessment.

5.1 Benchmarking Strategy

In a nutshell, our proposed benchmarking strategy consists of the following steps:

1. generation of artificial network and corresponding expression data,
2. computation of the mutual information between all the pairs of genes,
3. inference of the network,
4. validation of the results by comparison with the original network
5. comparison of the performance of different estimators

In the following sections we address each of these steps in detail.

5.2 Synthetic Data Generation

Artificial networks, like the their real-life counterparts, are conceptual models of genetic regulation, where each gene is considered to be directly or indirectly influenced by a number of other genes. It can be represented as a directed or undirected graph.

1) As such, the first issue a designer of a network needs to address is the *network topology* i.e. the wiring diagram of the genes' interactions.

Small-scale networks can often be crafted by hand, from experienced biologists, but for large-scale networks, designers resolve to using random graph models like (a) Erdős-Rényi, (b) Albert-Barabási, (c) Watts-Strogatz, (d) small-world and (e) directed scale free and others. For a detailed description of each network model, in the context of systems biology see [MSK03], [VdBVLN$^+$06].

Topological characteristics that form part of the designs, can be classified as:
Important Factors: density of connections, large number of genes, presence of nonlinear dynamics and combination of different topologies.
Evaluated Important Factors: average direct path length, average undirect path length, average in-degree, average out-degree, average clustering coefficient etc.

2) Once the network topology is decided, regulatory interactions (i.e. roles) are selected for each gene. Successful synthetic networks, consider a spectrum of possible interactions: activation, inhibition and - more rarely - complex regulation dynamics such as cooperativity, synergism or antagonism. Regulatory interactions of each gene are embodied in *transition functions*, inspired by enzyme kinetic equations. A transition function defines how the mRNA concentration of a gene varies in the presence of each of its input regulators. This whole system of equations synthesize a full dynamic model that is a genetic network in a steady-state.

Several dynamic models have been suggested as transition function, including Boolean [Kau71], continuous [ZDS01] and Bayesian [SJH02],[YAP$^+$04]. Most current network simulators use a set of ordinary differential equations (ODE's) [MSK03], or Michalis-Menden and Hill enzymatic kinetic models [VdBVLN$^+$06].

3) Having prepared the network topology and the dynamic model of its operation in normal steady-state condition, we are ready for the generation of *microarray-like gene expression data*, which is essentially sampling from the dynamic system of equations that embodies our network.

We are reminded, that expression datasets are in effect by-products of biological experiments that interfere with the system's normal state of affairs. Two examples of triggers that cause the network to depart from its steady state are mutant strains and any external environmental conditions.

Within the sampling process, deviation can be modelled by choosing a set of genes without regulatory inputs and setting their expression level to a different value for each experiment. Expression levels can subsequently be calculated as specified by their transition functions, starting from the input genes. If the system allows loops and oscillatory behaviours, these are resolved separately. Remaining genes with undefined regulatory inputs can be assigned random constitutive expression values.

After the calculations have propagated through the entire network, they are repeated until transient effects have disappeared before generating the output expression levels for that sample [VdBVLN$^+$06] and the whole process is repeated for as many samples as we need.

For this final stage, we have to be careful, to ensure that sampling is done in accordance with the type of experiment that we wish to emulate and the type of methods that we wish to test. For example, if the method requires time series data or does comparisons of mutant strains, then the generator must produce time series for the former and comparative mutant data for the latter [MSK03].

From the discussion above, it becomes apparent, that construction of an artificial data generator can be very challenging, both theoretically and practically. A compelling amount of choices need to be made, in order to come close to capturing the complexity of real expression data. Some of these projects from the current literature are presented below.

5.2.1 Known Generators

Most generator systems to date (see [SJH02], [YAP+04], [ZDS01], [KDB04]), were tailor-made to showcase particular applications, and are very difficult to transform and adapt to other methodologies. In our short treatise on the subject, we choose three methods that in our opinion are more generally applicable and, to a certain extend mimic the characteristics of real datasets well enough.

sRogers generator

The *sRogers* generator ([RG05]), specialises in the simulations of datasets arising from knock-out experiments and uses a Bayesian regression approach to network construction. It employs the same data generation procedure as Yeung et al. ([YTC02]) and Rice et al. ([RTS05]) whereby a network is generated following an approximate power-law distribution on the number of regulatory connections out of each gene. It is possible to encode both excitatory or inhibitory connections. The different gene expression samples are generated by resampling (R) times from the steady-state of the network, which is in turn obtained by integrating a system of differential equations during which the expression of the knocked-out gene is held to zero. A total of k wild and k knock-out measures can be obtained this way, resulting to $N = 2kR$ samples in total.

Mendes data collection

Mendes et al. ([MSK03]) provide a collection of data sets organised in series of models. Each model represents a wild-type of network topology: (a) Erdős-Rényi, (b) Albert-Barabási, (c) Watts-Strogatz and (d) small-world. Mathematically, each network class is embodied in a kinetic model via a set of differential equations - one for each gene. Each equation represents general rate laws of transcription that approximate mRNA concentrations of the gene arising from elementary chemical steps such as synthesis and breakdown of mRNA in the presence of one inhibitor or one activator.

The Gepasi simulator ([Men93]) - an earlier product of the same authors - generates samples representing different stages of a time series experiment or different mutants/perturbations in a static experiment. Controlled noise is finally added to the expression measurements generated from the previous stage. Via this complicated but modular framework, the authors have compiled four series of datasets, in an effort to cover as many *important factors* of real gene expression data as possible, such as the ones mentioned above, while not committing to a single organism or experimental setting. Since each one of the synthetic networks alone cannot cover two or more *important factors* simultaneously, we are advised to assess the average performance of our algorithms against as many of the synthetic data sets as possible - i.e. all the data sets in a series - to compensate for the lack of strong biological meaning of the networks.

SynTReN **generator**

The authors in [VdBVLN$^+$06] argue that is is not possible to choose a single set of parameters for a typical random graph model (such as Erdős-Rényi or Albert-Barabási etc.), such that more than one *evaluated important factor* of the synthetic network produced, resembles a realistic biological network. For example, we can choose parameters so that the average directed path length is realistic, but then the average in-degree will not be biologically plausible and *vice versa* [1].

As a remedy, they suggest the use of previously characterised transcriptional genetic networks as a source for the synthetic network topologies. A limited number of well-studied organisms like *E. coli* ([SOMMA02], [MKD$^+$04]) and *S. cerevisiae* ([GBBK02]) currently fit these requirements. As more true interactions are added or deleted, so our knowledge about the properties of real transcriptional networks will improve and better values/models for the parameters of network topologies will emerge.

The *SynTReN* generator starts the creation of a network topology by randomly selecting one node and all of its neighbours from the model organism of our choice. In each iteration, a

[1] A special class of random graphs models, the *directed scale free*, is according to the authors the class that picks most of the characteristics of real TRNs with careful selection of parameters. .

randomly selected node together with all its neighbours (i.e. a cluster of nodes) are added to the graph. Therefore, only nodes with at least one connection to the current graph are retained. Alternatively, we can add a single node at a time without its neighbours. Finally, a number of 'background genes' is added to represent the part of the network not elicited by the simulated experiment from pathways that are not directly affected by the experiment but yet change in a correlated way.

Different possible interaction structures are accommodated including cooperative, competitive, non-competitive, synergistic or antagonistic. The transition functions representing the interactions are based on the Michaelis-Menden and Hill kinetic models, under different conditions. Biological noise is added using a lognormal distribution, which is superimposed on the kinetic equations. Generation of gene expression data proceeds as described in the previous section. Finally, random experimental noise, again using the lognormal distribution is added to the normalised data.

SynTRenN was the generator of our choice for our comparative study and more specific details on the final dataset produced and used throughout this testing procedure, see section 5.6.

5.3 Mutual Information Matrix Estimation

To learn the connectivity of a network from scratch using mutual information we first need to make a list of all pairs of variables under consideration and then estimate their mutual information. The results can be stored in a matrix - the mutual information matrix - a $N \times N$ matrix, with $MIM_{ij} = I(X_i; X_j)$ and $i, j \in 1 \dots N$ where N is the number of genes.

The four differential mutual information estimators (i.e. for continuous variables) presented in the previous sections - Adaptive Partitioning, k-NN I and II and KDE - are in the benchmark together with two variations of the empirical mutual information estimator: (1) the naive-empirical estimator that we presented in section 3, and (2) the shrinkage estimator, that we present below. The main difference between differential and empirical estimators is that the

former computes

$$I(X;Y) = \int f(x,y) \log \frac{f(x,y)}{f(x)f(y)} dxdy,$$

whereas the latter computes the MI of the empirical probability distribution,

$$I(X;Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)},$$

after discretization of the variables.

Shrinkage Estimation

The shrinkage estimate [Hou06] estimates the probability density function of the random variable X in bin B_i, $p(x_{B_i})$ as:

$$\hat{p(x_{B_i})} = \lambda \frac{1}{|B|} + (1-\lambda)\frac{y_{B_i}}{N},$$

where y_{B_i} is the number of points in bin B_i.

This estimator is in fact a mixture of two estimators: the one in the first part of the formula ($\frac{1}{|\mathcal{B}|}$), which is the same estimator as the naive, that has zero variance but a large bias and the other in the second part of the formula ($\frac{y_{B_i}}{N}$), that has larger variance but is unbiased.

Parameter λ is chosen so that the *mean square error* $E\{\sum_i (\hat{\theta}_i - \theta_i)^2\}$ is minimised [2].

Both the empirical and the shrinkage estimatros can be found in packages MINET (from Meyer et al. [MLB07]) and ENTROPY (from Margolin et al. [MNB+04]) in R [R D08].

5.4 Network Inference

Mutual information matrices give a full account of the dependency between a set of variables, by scores of MI. Network inference methods, take a dependency score between a pair of variables and decide, in the context of the other score, if it describes a direct interaction between the variables. A handful of network inference algorithms using mutual information as the de-

[2] Applied to the problem of bin probabilities $\lambda = \frac{|B|(N^2 - \sum_i y_{B_i}^2)}{(N-1)(|B| \sum_i y_{B_i}^2 - N^2)}$

pendency measure, were presented in 2.2. Earlier efforts begin the construction of the network by adding edges between all pairs of genes with MI scores that exceeded a certain threshold (RelNet). By this non-rigorous selection method, we expect that many of the inferred edges, would not represent true parent-child/direct interactions. Relations where genes are regulated by a common parent gene, or even a gene a few nodes upstream can also be included. The more advanced methods (CLR, ARACNe, MRNET), address this problem with a suitable pruning process to remove connections that are in effect overwritten by more direct paths through the graph. The remaining network structure is then considered a better representative of causal relationships in the set of variables.

For our benchmark we considered all three more advanced network inference algorithms: CLR, ARACNe and MRNET - sections 2.2.5, 2.2.3 and 2.2.6 respectively. All methods receive as input the mutual information matrix and return the edge matrix of the network. This matrix is in effect a symmetric adjacency matrix with 1s in the slots were MI was decided by the network inference algorithm to be significant and 0s in the slots were judged as redundant.

5.5 Network Validation

Using the model network produced from the synthetic data generator, we can assess the performance of the network inference algorithm - and in extension of our mutual information estimator - by measuring the quality of reconstruction in our inferred network.

The first step for this assessment is to build the *confusion matrices* that summarise the decisions made by the network inference algorithm. Confusion matrices (Kohavi and Provost, 1998 [PK98]) contain information about the actual and predicted edges in the form seen in table 5.1.

EDGE	Predicted Positive	Predicted Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

Table 5.1: Confusion Matrix

The entries in the confusion matrix in the context of network inference can take the following

forms:

TP: number of positive predictions when the edges actually exist.

FP: number of positive predictions when the edges actually do not exist.

FN: number of negative predictions when the edges in fact exist.

TN: number of negative predictions when the edges in fact do not exist.

5.5.1 Quality Measures

PRECISION	$p = \frac{TP}{TP+FP}$
RECALL	$r = \frac{TP}{TP+FN}$
SPECIFICITY	$s = \frac{TN}{TN+FP}$
TRUE POSITIVE RATE/SENSITIVITY	$tpr = \frac{TP}{TP+TN}$
FALSE POSITIVE RATE	$fpr = \frac{FP}{FP+FN}$
F_β-SCORE	$F_\beta = (1+\beta)\frac{p}{\beta p+r}$

Table 5.2: Standard quality measures for network validation

Table 5.2 shows a number of quality measures derived as combinations of elements from the confusion matrix. These measures have become a standard in tasks like reverse engineering of networks, classification and other information retrieval applications.

Precision can be seen as the number of derived existing edges divided by the number of correctly inferred edges - both positive and negative. Precision can be otherwise interpreted as a measure of the algorithm's *exactness*.
Recall measures the number of derived and existing edges divided by the total number of edges that are actual, thereby being a measure of the algorithm's *completeness*.

The maximum score of 1.0 in precision means that *every* edge inferred by the algorithm represents an actual dependence of the variables. We still do not know however if *every* covers *all*. Moreover, the maximum score of 1.0 in recall means that *all* the actual dependencies of genes were retrieved, but cannot inform us, if along with the actual ones, the method retrieved some non-actual as well. It becomes obvious, that the two measures complement each other

and it is usually pointless to consider them in isolation. *P-R curves* join precision and recall in a common plot.

F_β-score (van Rijsbergen, 1979), combines precision and recall, in a 'total' measure of accuracy. In effect, it is a weighted average of the two factors, where β is a real number determining the contribution of each in the total score. In most cases $\beta = 1$, which amounts to equal contribution.

True positive rate (TPT) and *false positive rate* (FPR) are another set of operating quality measures.
TPR captures the amount of derived actual predictions over all the actual edges - also called *sensitivity*.
FPR on the other hand measures the fraction of non-edges that are misclassified as dependencies - also called *fall-out*. These two measures, also are almost always considered together and their ratio is captured in *ROC curves*.

P-R & ROC curves

P-R and ROC curves provide a graphical representation of an algorithm's performance. ROC plots use FPR and TPR as x and y axes respectively. Similarly, P-R plots, set recall on the x-axis and precision on the y-axis. It is possible by examining these curves to expose differences between the algorithms and their learned networks.

For a given expression dataset, their corresponding existing and non-existing edges and the evidence of a number of samples, ROC and PR curves essentially "cover the same points" [DG06] - the visual representation of the information however, is quite different.

In the case of genetic networks - even our synthetic ones - the actual dependencies are quite sparse compared to the number of gene-pairs involved. This is in fact translated as a very high ratio of negative to positive examples.

The effect of this irregularity on the validation of the algorithm, is that a large change in the number of false positives (FP) only induces small changes in false positive rate (FPR), hence

causing ROC analysis to produce more positive results for the performance of the method.

Precision on the other hand, by comparing false positives (FP) to true positives (TP) rather than true negatives (TN), captures this anomaly, hence is better suited to highlight the differences of the algorithms for heavily skewed datasets [Pro98].

5.6 Empirical Validation - Methodology

Having introduced the background on the empirical validation of our mutual information estimators via the inference of synthetic genetic networks, including the motivation and the rationale for the study, we will now analyze the technical details of every stage, in order to better comprehend the results that follow.

Table 5.3 below shows the four steps of performance evaluation and the different methods for each stage that we have available for the benchmark.

STAGE I	STAGE II	STAGE III	STAGE IV
Synthetic Network + Expression Data Generation	MIM construction Methods	Network Inference Methods	Validation Scores
SynTreN (E. coli)	Empirical (EMP) Shrinkage (SHR) Adaptive Partitioning (DV) k-nn (KI) k-nn (KII) KDE (KDE)	CLR ARACNe MRNET	P-R curve F_β-score

Table 5.3: Stages and Options for Empirical Validation

STAGE I: Synthetic Network and Expression Data Generation. Of the three generators presented in section 5.2.1 we opted for a dataset created by SynTreN generator[3].

For model organism, we chose *E. coli* as the model network and kept the default parameters for noise and connectivity. The resulting expression dataset consists of 100 samples of 200 genes and the corresponding genetic network has 293 edges that are supposed to

[3]The latest version of the free Java toolkit can be found at the webpage http://homes.esat.kuleuven.be/ kmarchal/SynTReN/index.html.

be actual dependencies sourced from bioinformatics literature on *E. coli*. We note here that a fully connected network of 200 genes would have a total of 19900 edges, thus the ratio of proven versus potential edges is a little over 1%. The ratio of genes vs samples is 2.

This network is in fact a very good imitator of real life scenarios and it is therefore suitable for testing the behaviour of the methods under these circumstances.

STAGE II: MIM Construction. For this stage, as we described previously in section 5.3, we take all the possible pairs of genes and compute their mutual information, six times - one for each estimator (four from the previous section plus the empirical and shrinkage estimators). This process yields six MIM matrices of size 100×100.

STAGE III: Network inference. As outlined previously in section 5.4, each of the six matrices will be passed on to each one of the network inference routines. This step will result in $6 \times 3 = 18$ matrices for each dataset, that represent the network structure inferred by MRMNET, ARACNe and CLR methods. As stated above these matrices are in effect adjacency matrices with 1s in the slots were the MI was deemed by the network inference algorithm to be significant and 0s in the slots that were judged as redundant.

STAGE IV: Validation. Each of the 18 matrices generated by the previous step, is compared to the actual network matrix, available from the design of the synthetic data. 18 confusion matrices are computed, via which we obtain the precision-recall measures and the corresponding F_β-scores. For reasons discussed above, we will not display any ROC-curves.

5.7 Empirical Validation - Results

The purpose of this chapter is not to do an exhaustive presentation of all the possible method combinations (all 18 confusion matrices), or expand our assessment to network inference algorithms, but rather to highlight the relative merits of using estimators other than the empirical for the estimation of mutual information when the sample size is small.

To this end, we begin by showing the effect of increasing sample size on the overall accuracy of the reconstruction algorithms. Figure 5.1 shows the variation of F_β-scores for the adaptive partitioning method of Darbelay & Vajda (DV) on MRNET (5.1 left panel) and CLR (5.1 right panel) respectively.

The x-axis on the plot represents the number of samples and the y-axis represents the corresponding F_β-scores.

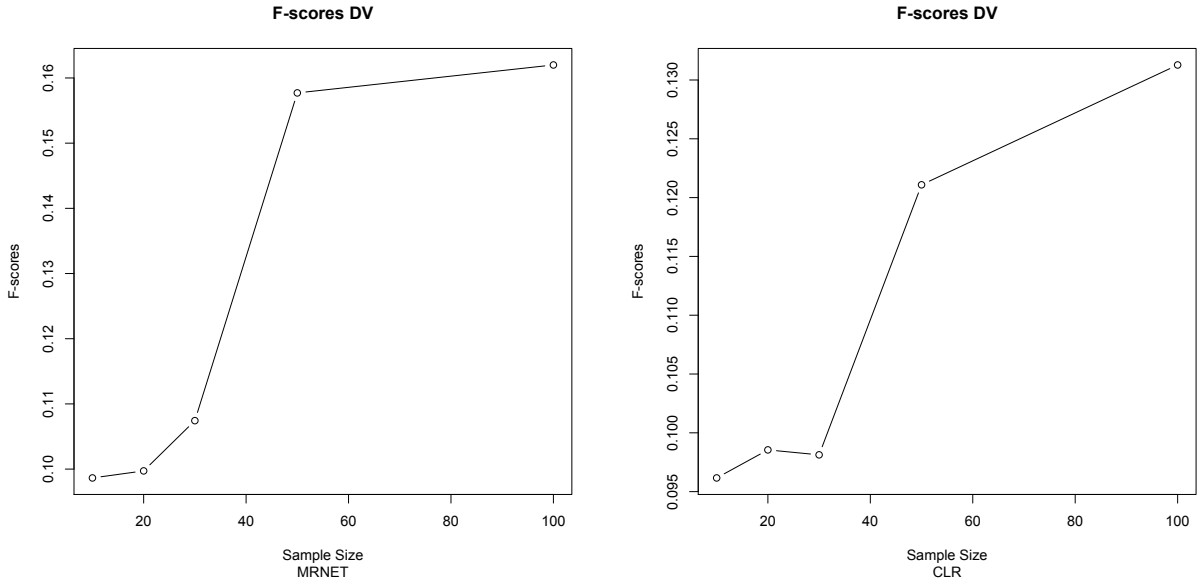


Figure 5.1: [FSC-DV-SS] Effect of sample size on F_β-score for DV and different network inference algorithms.

In figure 5.2 we show again the effect of sample size on the performance of MRNET and CLR (5.1 left and 5.1 right), for the second version of the k-nn algorithm of Kraskov (KII) ([KSG04]).

Both for DV and KII methods, the figures show that the accuracy of the network reconstruction with either of the inference algorithms (MRNET, CLR) is improved linearly with increasing sample size.

Figure 5.3 highlights the sensitivity of the KDE on the selection of bin size. The overall accuracy of the reconstructed network is rather invariant to the sample size when bin size $|B| = 3$ for every N (5.3(b)) and it vanishes almost completely, when we change the bin size to adapt to sample size $|B| = N/3$ (5.3(a)). Minimum accuracy and instability was also observed for the other reconstruction algorithms (CLR, ARACNe).

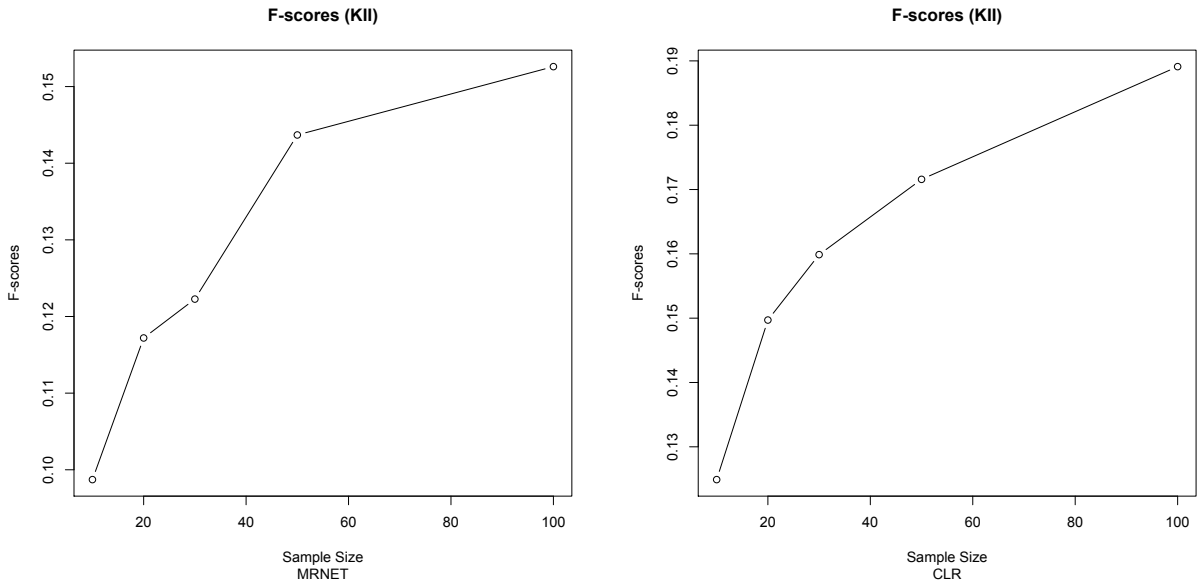
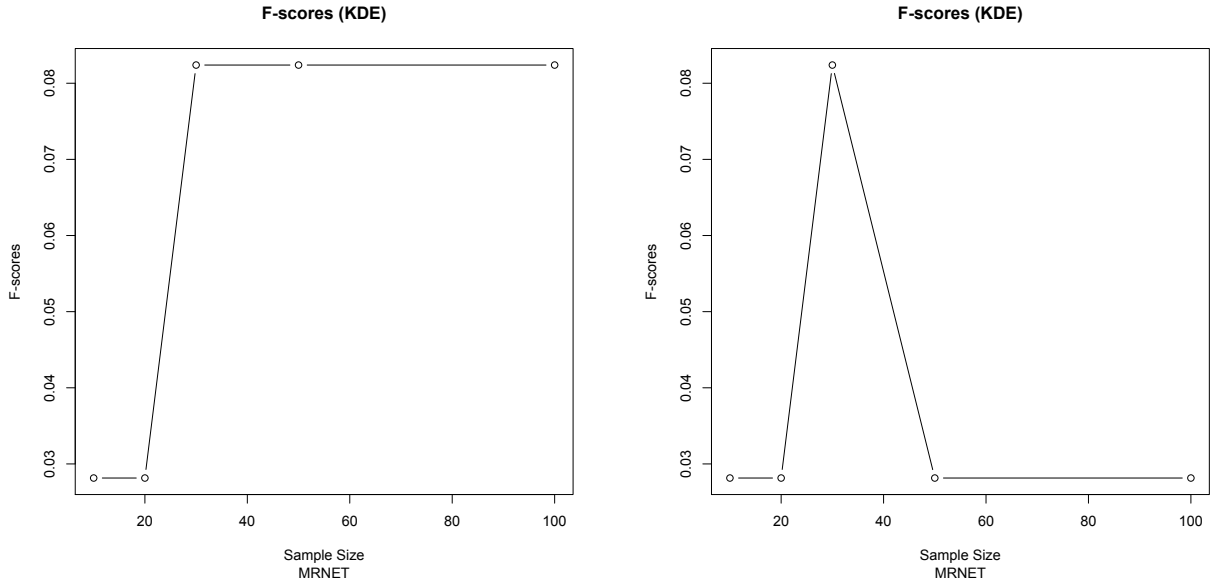


Figure 5.2: [FSC-KII-SS] Effect of sample size on F_β-score for KII and different network inference algorithms.



(a) bin size $|B| = N/3$ (b) bin size $|B| = 3$

Figure 5.3: [FSC-KDE-BS-MRNET] Effect of bin size on F_β-score for KDE on MRNET.

Since the focus of the present work is the study of the behaviour of the methods in realistic sample sizes, we will now look at the performance of each estimator in sample size $N = 30, N = 20$ (Fig. 5.4 and 5.5 respectively) and $N = 10$ (Fig. 5.6) in further detail.

For this study we will plot the P-R curves and the corresponding F_β-score curve for each

method in a single plot. All three network inference methods were considered, so the best performing method will be judged from its scores as well as its consistency - i.e. presence in all inference algorithms. For the F_β-scores, the x-axis on the plots represents a step in the validation process. Each one of the steps, is assigned a threshold value[4]. Then at each step, the edges of the reconstructed network, whose weight is below this threshold value are eliminated. The resulting network, is then compared to the known network and a confusion matrix (TP,TN, FP, FN) is constructed for this threshold (or validation step) and the corresponding F_β-score value, is also computed.

For a method to be present in the graph, the underlying confusion matrices must not contain any NaN values resulting from extremes in MI estimation. We will observe, that this case is far from uncommon for some estimators under restricted sample settings.

In fig. 5.4 the first row, presents the P-R & F_β-score curves of the methods after the ARACNe method was applied for the network reconstruction. We note the absence of any results from the shrinkage method (green line) on the plots. In fact, the inferred network was inconclusive (just 1730 pairs with mutual information above zero), which caused erroneous confusion matrices and thus F_β-scores. The best scoring function in this case appears to be k-NN Kraskov's second method, because it is closer to the diagonal in the P-R diagram and higher up the y-axis in F_β-score diagram. Adaptive partitioning and the simple empirical method are very close for the second position - DV is achieving higher recall (completeness) but lower precision than EMP. The KDE method has an overall poor performance.

Both empirical methods (EMP and SHR) fail to produce any significant interactions when subjected to the CLR inference algorithm. The first k-NN method also had 3 NaN values (out of 50) in its confusion matrix, and hence was eliminated by the study. The best scoring functions for CLR are the k-NN second method again and DV (adaptive partitioning) as a second best. Using the MRNET inference method, all the estimators produce plausible network structures. Validation shows that the second k-NN estimator has the best precision-recall balance and second best overall accuracy. Equivalent performances come from KI, DV and SHR - with KI

[4]The threshold values are estimated proportionally to the range of mutual information values produced by each estimator.

showing higher recall and top scoring overall accuracy in lower threshold steps and SHR higher precision. All methods seem to be in accord, that KII is the best performing algorithm for $N = 30$.

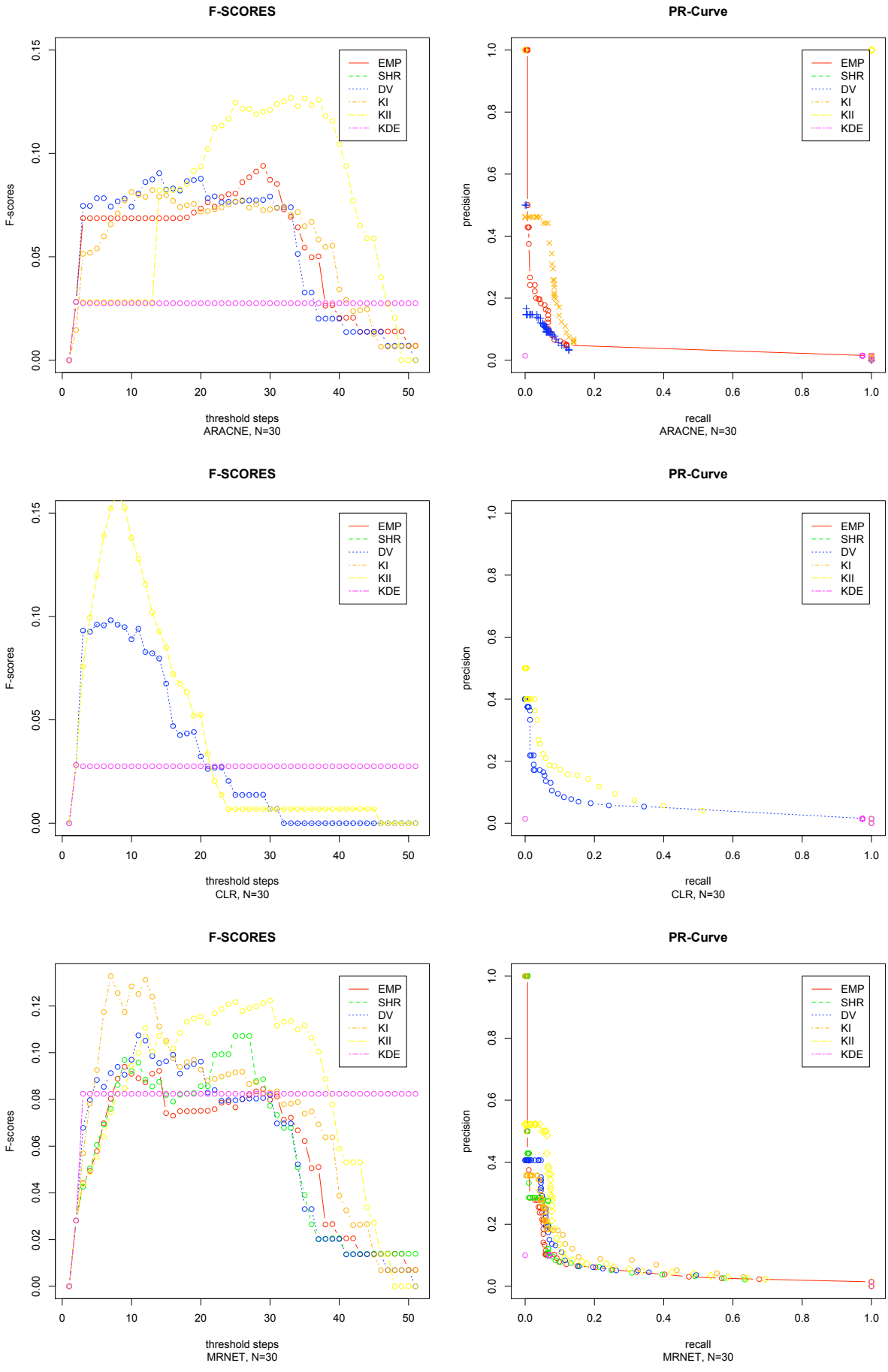


Figure 5.4: [PR-ALL-30] P-R & F_β-score curves for all three inference algorithms and sample size = 30.

In fig. 5.5, we repeat the analysis as in fig. 5.4. As with $N = 30$, both empirical estimators fail to produce any meaningful results with CLR. In fact, this combination (EMP or SHR with CLR) works only for sample sizes greater than 100, so when using the CLR network inference algorithm the empirical methods do not suite small sample settings.

The ranking of the remaining methods is (1) KI (2) KII and (3) DV for overall accuracy and from the P-R diagram we gather that KII hits the maximum value of precision, but recall values remain low, whereas DV and KI methods achieve a better balance.

For ARACNe, presented in the second row of fig. 5.5, again the SHR method produced a lot of NaN values, thus had to be excluded from the study. It is very difficult to make clear decisions about the exchange of precision and recall for any of the remaining algorithms, except maybe that KII and KI seem to form more of a balanced trade-off (diagonal line) rather than a mutually exclusive relationship of DV, EMP and KDE, which is the effect of very small sample sizes. Again, KII scores the highest overall accuracy - KI and DV are second best.

For MRNET we can just about see from P-R curve that KII and KI are the best scoring algorithms with DV and EMP following.

As before, KDE is persistently a poor performer and KII is a persistently good one.

Finally, fig. 5.6 presents the assessment of all the methods with $N = 10$. As with the case of $N = 30$ KI, SHR and EMP are excluded from the graph. KII achieves higher accuracy scores as well as better P-R balance than DV.

For ARACNe the ranking of the methods is (1) DV (2) KII (3) EMP (4) KI and (5) KDE. The adaptive partitioning estimator is again performing quite well when using the MRNET reconstruction method, although it fails to achieve high precision - KII appears to have better balanced performance.

5.7.1 Conclusions

this chapter, we have attempted to use the chosen robust estimators in the task of inferring a known network. This task was performed to highlight the relative merits of using estimators of mutual information and higher order mutual information, alternative to the standard binning estimators. The benchmarking strategy involved plugging the estimators to known network reconstruction algorithms and using synthetically produced data.

By using data, produced by SynTren, we achieved two goals simultaneously: first, to compare all the algorithms outside the 'comfort zone' of a parametric setup and second, to have a realistic but controlled environment, by which a comparison would be fair, quantifiable by standard methods and easily reproduced. Finally, in practical terms, we gain a better idea about the methods' performance under completely new networks, where previous knowledge about the data is missing.

Table 5.4 below, summarises the best scoring methods for every reconstruction algorithm from the previous section (sample sizes $10, 20, 30$).

Fig. 5.7 presents a collective summary of the accuracy of all the methods - both continuous and empirical - for all sample sizes $10, 20, 30, 50, 100$ under MRNET on the right panel and under ARACNe on the left.

	MRNET	CLR	ARACNE
10	DV, KII	KI, DV	DV, KII
20	KI, KII	KII, DV	KI, KII
30	KI, KII	KII, DV	KII, EMP (DV)

Table 5.4: Summary of Results

Because the performance of all the methods is poor, we need to stress, that even with that low precision and recall, in absolute values, the significance of using a robust estimator is even more pronounced.

To that effect, table 5.4 and figure 5.7 show that in undersampled regimes it is indeed worth using estimators alternative to the naive empirical ones, for the calculation of MI. For sample sizes above 50 or 100 - conditional to the network reconstruction algorithm - empirical methods

can also be applied, as there seems to be a linear increase in their performance. Finally, table 5.4 above, highlights the second variation of k-nn method of Kraskov [KSG04] and the adaptive partitioning method of Darbellay and Vajda [DV00] as the most suitable methods to use in any setting.

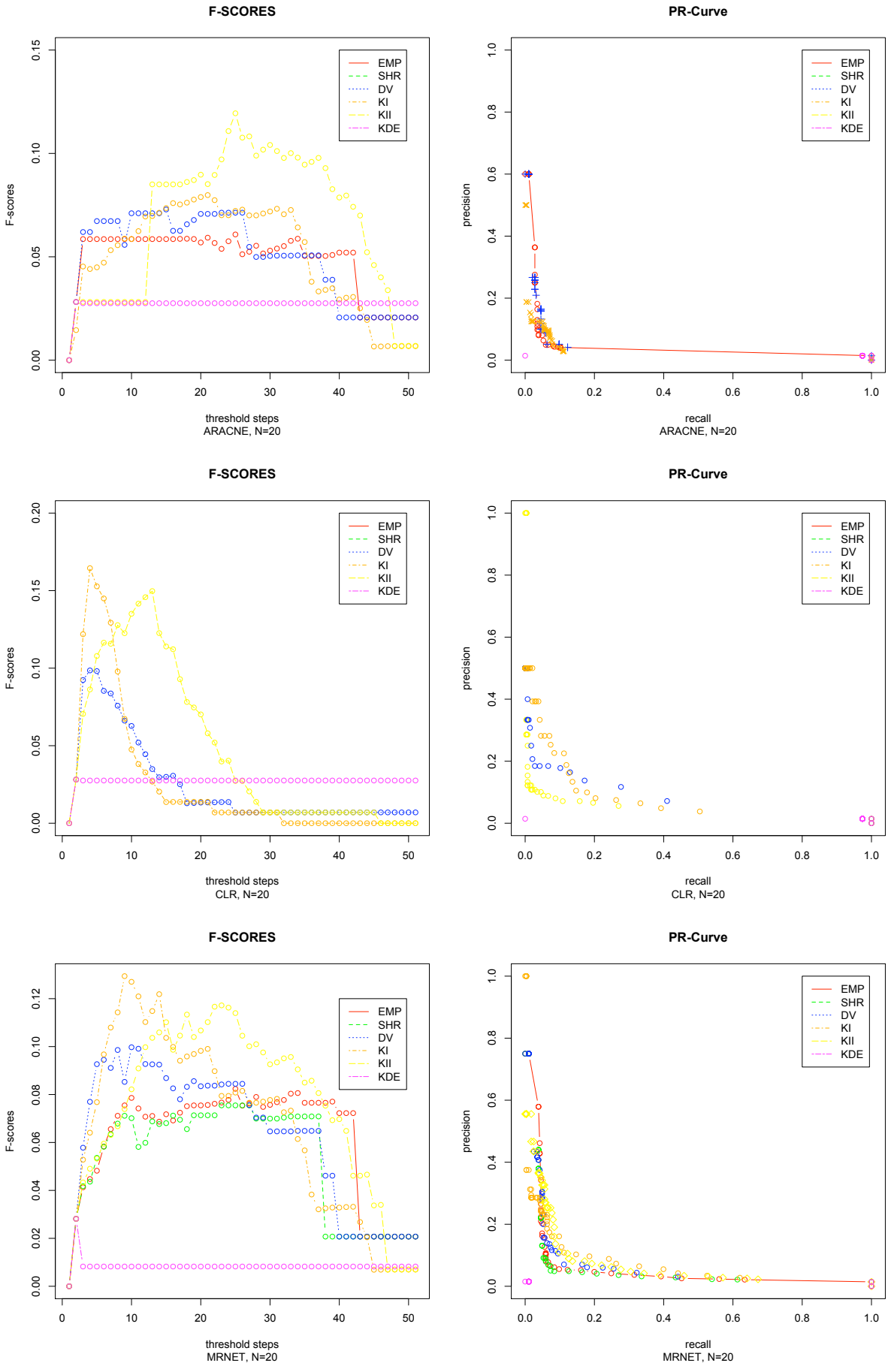


Figure 5.5: [PR-ALL-20] P-R & F_β-score curves for all three inference algorithms and sample size = 20.

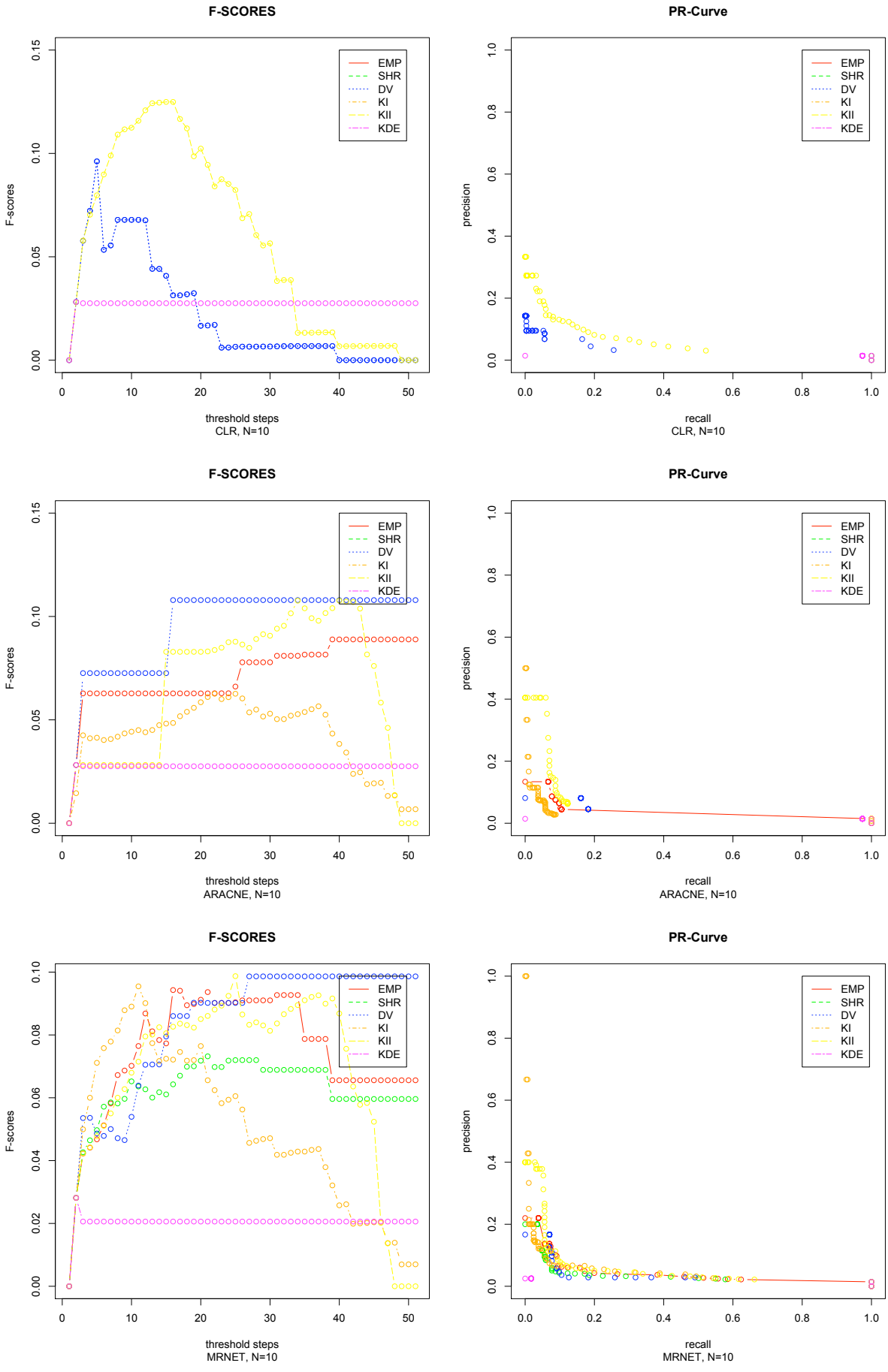


Figure 5.6: [PR-ALL-10] P-R F_β-score curves for all three inference algorithms and sample size = 10.

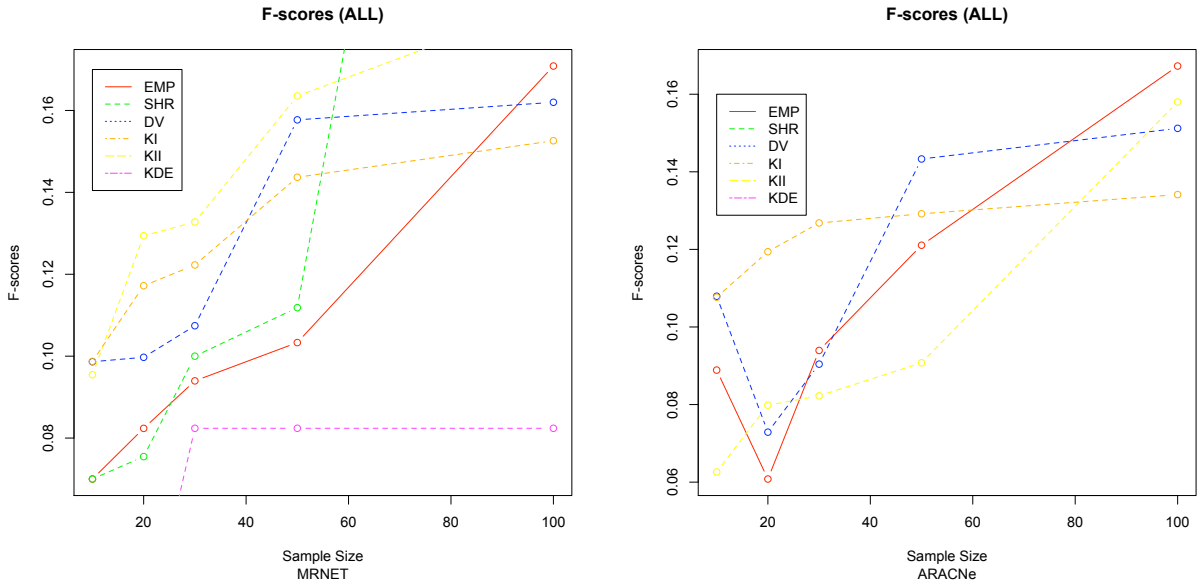


Figure 5.7: [FSC-ALL-SS] Effect of sample size on F_β-score for all the methods and different network inference algorithms.

Chapter 6

Improving the Inference of Structure of Gene Interactions

Extensive research so far has demonstrated, that genetic biological networks often show properties that combine elements from different model network architectures; i.e. they have elements of random graphs, small-world, scale-free and other networks, simultaneously.

Within these architectures, complex sub-structures often arise, that require better suited techniques for their correct detection and evaluation. Figure 6.1 from [SOMMA02], shows a comprehensive list of possible substructures of size 3, that can be encountered in a genetic network.

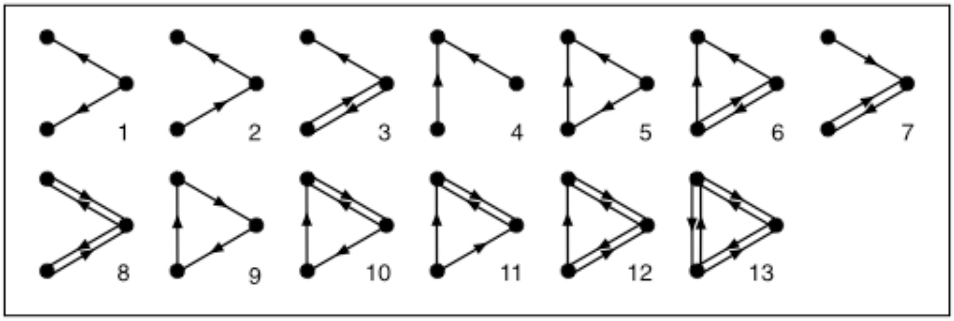
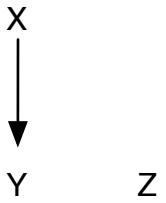


Figure 6.1: Examples of regulation structures, where conditional MI can be helpful. Image from [SOMMA02].

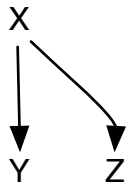
6.1 Structure Learning Methodology

From figure 6.1 above, it becomes apparent, how two processes may interact indirectly and still be strongly dependent, but not linked via a direct edge. Many of these sub-graphs cannot be detected by ordinary bivariate analysis and the actual dependencies are often missed, or superseded by other dependencies and the resulting networks have many false positive or false negative edges.

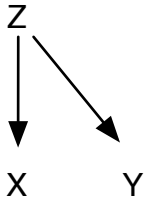
Faced with the critical issues of sparsity of data and lack of ground truths, what appears to be the most reasonable strategy, is to try to take small steps at a time and slowly build up the complete picture. A first step in this direction, would be to focus on the recovery of small network structures, like the ones presented in figure 6.1. Correct inference of such substructures, could confirm computationally a big part of established interactions, correct part of the disputed ones and finally, help predict a lot more new interactions. Figures 6.2(b-e) show a selection



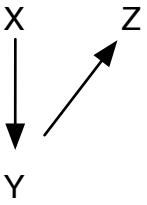
(a) no effect



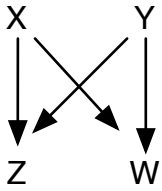
(b) corregulation of Y and Z by X.



(c) explanation of X and Y by Z.



(d) Markov chain between X, Y and Z .



(e) Bi-fan motif of corregulation.

Figure 6.2: Examples of regulation structures.

of model gene interactions, whose structure is undetected by conventional mutual, or higher order mutual information. We will show that it is possible to uncover each of these motifs, by detecting asymmetries in information theoretic functionals - conditional/first-order partial

mutual information, in particular.

Figure 6.2(a), displays the simplest regulation structure, that of a pair of regulator-regulatee, with no effect on other variables. The relationship $I(X;Y) = I(X;Y|Z)$ describes this structure.

Figure 6.2(b), indicates that Y and Z are regulated by the same mechanism, via X. Computation of pair-wise mutual information, would show $I(X;Y) = I(X;Z) = I(Y;Z) > 0$ and if we were to stay on these facts alone, we would have to draw a fully connected network of interactions, i.e. add the edge between $Y - Z$. Computing however, the conditional mutual information estimates for the three variables, we would see that:

$$\begin{aligned} I(X;Y|Z) &\approx 0, \\ I(X;Z|Y) &\approx 0, \\ I(Y;Z|X) &\ll I(Y;Z). \end{aligned}$$

The last inequality states, that having observed X that regulates both Y and Z, the information provided about Y by observing Z, is reduced.

Figure 6.2(c) is very similar to figure 6.2(b), in terms of characteristic inequalities and states that X and Y are conditionally independent given Z.

$$\begin{aligned} I(Z;X|Y) &\approx 0, \\ I(Z;Y|X) &\approx 0, \\ I(X;Y|Z) &\ll I(X;Y). \end{aligned}$$

We can view this interaction structure from a causality inference perspective. Assuming that we assign variables X, Y, Z the roles of regulator-regulatee and conditioning/control respectively, this structure would arise when the conditioning variable is a preceding cause of the regulator-regulatee. We need to point out here, even with these inequalities, we cannot differentiate between this arrangement and $X \rightarrow Z \rightarrow Y$, where the control variable intervenes on the path

of the regulator-regulatee.

Figure 6.2(d) captures the structure of a Markov Chain, were X regulates Y and Y regulates Z. This is the type of relation, that the data processing inequality (see eq. 2.37) could also detect.

$$\begin{aligned} I(X;Y|Z) &\approx I(X;Y), \\ I(X;Z|Y) &\approx I(X;Z), \\ I(Y;Z|X) &\approx I(Y;Z), \\ I(X;Z|Y) &\ll I(X;Y|Z) \ll I(Y;Z|X). \end{aligned}$$

From the last equation, we can see that Z is indirectly regulated by X, hence their conditional mutual information is a minimum. According to equation 2.37, we could also confirm that $I(X;Y) \geq I(X;Z)$.

Finally, in figure 6.2(e), is a step up from case in 6.2(b), where both variables Z and W are regulated by X and Y. This network structure is known to the biology community, as bi-fan motif.

In this case, mutual information between Z and W would be very high - even higher than the interactions $X - Z$ and $Y - W$. We note, that using the data processing inequality would result in the erroneous addition of an edge between Z and W. Characteristic conditions for this structure are:

$$\begin{aligned} I(X;Z|W) &\approx 0, \\ I(X;W|Z) &\approx 0, \\ I(Z;W|X) &< I(Z;W), \\ I(Z;W|Y) &< I(Z;W). \end{aligned}$$

The last two inequalities state, that, having observed X (or W for the second inequality) regulates both Z and W, the information provided about Z by observing W, is reduced.

The network motifs and related information-theoretic inequalities presented above, pretty much capture the whole range of interactions that conditional mutual information, DPI and higher order mutual information can contribute to the improvement in the performance of genetic network inference.

Other important types of interaction structures, like feed-forward loops, structures of the type appearing in figure 6.1$(3, 6, 7)$, or higher order regulatory cascades, need the explicit treatment of time in the equations for all the functionals presented, whenever time-series experiments are available. Prichard and Theiler [Pri95] among others, have extended the concept of conditional redundancies to multiple variables and propose a time-lagged version of mutual information. However, these extensions, were suggested for problems in other areas, with large data sets and repeated measurements, therefore extensive studies need to be done in order to apply them in genetic network interaction problems [PV07].

6.2 Estimation of Conditional Mutual Information

A large part of our work was devoted into exploring the statistical properties of a set of non-parametric mutual information estimators in undersampled regimes. Both of the functionals that we scrutinised, i.e. mutual information and higher order mutual information are essential for the estimation of conditional mutual information, via equation 6.1, the first form of which we repeat here:

$$\begin{aligned} I(X;Y|Z) &= I(X;Y) - I(X;Y;Z) \\ &= H(X) + H(Y) - H(X,Y) \\ &- H(X) - H(Y) - H(Z) + H(X,Y) + H(X,Z) + H(Y,Z) - H(X,Y,Z) \\ &= H(X,Z) + H(Y,Z) - H(Z) - H(X,Y,Z). \end{aligned}$$

From all these variations, and the application of a chain rule for more that three variables, we see that it is possible to decompose the problem of estimating higher order conditional

dependencies, into multiple problems of entropy and joint entropy estimation, or mutual and higher order mutual information.

This type of decomposition suits well three out of four of the estimators examined: standard binning, kernel density estimator and the adaptive partitioning. However, for the k-nn based estimators, the authors [KSG04] advise against the use of such an approach. The main argument there, is that their estimators would suffer from significant bias, incurred by the different length scales used for the computation of conditional mutual information.

Instead, the authors propose the direct estimation of conditional mutual information via the formula using the conditional probability distribution of the variables involved, whenever this is available, i.e.

$$I(X;Y|Z) = \sum_{X,Y,Z} p(X,Y,Z) \log \frac{p(Z)p(X,Y,Z)}{p(X,Z)p(Y,Z)}$$

and again using the same type of approximation, via endowing the sample space with a metric to calculate the distances between points.

There exists, however, a non-parametric k-nn based estimator of conditional mutual information, suggested by Prichard et. al., in [Pri95]. As with the principal estimator by Kraskov et. al., the local conditional PDF in the metric space is estimated, by fixing a volume of given shape (hyper-rectangular) around the point where the PDF is to be computed and counting the number of points that fall within the shape. The final equation for this estimator is:

$$I(X;Y|Z) = \psi(k) - \langle \psi(n_{xz}+1) + \psi(n_{yz}+1) - \psi(n_z+1) \rangle. \tag{6.1}$$

In this paper, it is also shown [Pri95], that the k-nn partial mutual information estimator is also consistent and converges to the true value of CMI, for smaller volumes for the hypersphere. Smaller volumes, however cause larger bias in limited data settings.

The problem of bias and variance of the conditional mutual information estimators is a persistent problem throughout limited data problems, like those we explore here. Having shown

the performance of these estimators with standard MI and higher order MI, we expect that conditional MI will follow the same pattern of behaviour.

As discussed in [PV07], it can be stated that, using synthetic data, low variance of an estimator should always be preferred over low bias. This is because, any bias of the estimator, translates to a systematic error inherent with the method and tends to be almost identical with real data, as with the synthetic.

6.3 Towards a Complete Picture

We have seen how systematic evaluation of a combination of information theoretic functionals, is an effective first step to improve the inference of structure of gene interactions.

Here, we outline the general idea of a network reconstruction algorithm, that employs successively, both mutual information and conditional mutual information, in order to determine the existence of direct and some of the indirect dependence structures and alleviate part of the difficulties in the detection of false positive and false negative interactions. A similar inference scheme, was also suggested in [LW08], and used as a testbed for a purpose-made conditional mutual information estimator based on adaptive partitioning and the KDE-based method of Daub et al. [DSSK04].

Our reconstruction strategy may lead to undirected, directed, or partially directed graphs of linear and non-linear dependencies.

Algorithm 10 Schema for a Network Reconstruction Algorithm using MI and CMI

1: Compute pair-wise mutual information for every gene pair.
2: Apply a significance testing procedure, to select the non-trivial ones.
3: Generate all the possible gene triplets.
4: **for** each triplet whose genes all have significant MI scores **do**
5: generate a set of estimates of the CMIs from each of the three possible combinations
6: **end for**
7: **for** each set of CMIs of a gene triplet **do**
8: assess the existence or not of more complex dependency structures, like the examples in fig 6.2 via separate tests.
9: **end for**

To assess the feasibility of our idea, we implemented a simplified version of the above algorithm and tried it on a real dataset that was available to us. The data set takes the measurements of 2000 genes from *Neisseria Menigitidis* in a culture of glucose across 15 arrays. The data set is part of a larger experiment on cells of N. Menigitidis grown under different media and different knock-outs. It was provided to us by the group of Dr. Nigel Saunders, in Sir William Dunn School, Department of Pathology, University of Oxford and it is subject to publication.

The complete analysis of the data, without the guidance and expert knowledge of a biologist, is beyond the scope of our treatment, therefore we picked randomly 10 variables for illustration, which translates to to 45 couples, 120 triplets and 210 quadruplets and so on.

In our test implementation, we run tests for the existence of Markov-chain structures, as well as the bifan motif, using the adaptive partitioning method, that we analyzed in detail in the previous chapters. The significance of the mutual information and conditional mutual information scores, was assessed via resampling, using the bootstrap method by Efron [Efr87] and Tibshirani [DE96], explained below. The same principle was applied to assess parts of the tests required to assert the existence of the structures (like $I(X;Z|Y) \ll I(X;Y|Z) \ll I(Y;Z|X)$ in the final rule of Markov-chain and $I(X;Z|W) \approx 0$) in the first rule of the bifan structure. This part of the computation, adds severe delays to the overall run-time of the algorithm, but it is essential to establish computationally complex and abstract mathematical relations, like those required by the individual rules.

Moreover, in our model implementation, we run separate sets of experiments for the triplets in the data set and the quadruples. However, since the computation of conditional mutual information requires the computation of mutual information, and the computation of 4-body total correlation in turn, relies on the computation of some conditional mutual information scores, it becomes apparent, that there is a huge potential for speeding the whole procedure, by reusing computed tests and information-theoretic scores, where appropriate. This is essentially a numerical problem and we defer it for our future work.

A final issue with the scheme proposed, is the 'reconciliation' of the results, after uncovering the more complex structures. Specifically, from our test implementation, we end up with a set

of pairwise dependencies, a set of co-regulating structures, a set of Markov-chain structures, a set of bifan motifs, depending on the rules available. Merging these results in a unified network, is not so trivial.

Current network reconstruction algorithms, rely on the computation of 2-dimensional $(n \times n)$ dependence matrices, where each cell of the matrix, holds the association score between a pair of variables. With the proposed algorithm, we effectively go beyond the 2-dimensional matrix and adding a third variable, we are raising the structure to a 3-rd order tensor. Accommodating higher order conglomerations, requires either an effective reconciliation of the results from iterative application of the algorithm, or the a priori construction of hypercubes of the desired dimension. So far, to the best of our knowledge, there are no algorithms for neither.

Finally, although a very small study has been undertaken, the results do form a proof of concept for the method of discovering dependency motifs in genetic networks, using more complex information theoretic functionals.

Significance Testing

The first step of the schema, that requires computation of MI for every gene pair, produces a $G \times G$ matrix of MI values, where G is the number of genes considered. For the purpose of building a network of relevant interactions, we need to be more selective at this stage and apply some kind of non-specific filtering, that will prune the number of pairs to be considered for the next stage, to a manageable or informative size. Significance testing is not compulsory, it has been shown however, that the effect of considering the whole population of genes and gene pairs that are irrelevant to the problems at hand, is to overstate the resultant "reliability" scores of the results [AAR$^+$06].

Half of the network reconstruction methods that we reviewed in our background section consider all variable pairs (CLR, K2, MRNET) whereas the other half apply threshold values, used as cut-off points for non-informative pairs. The choices for the threshold values, range from more naive (Relevance Networks), to more sophisticated ones (ARACNe).

A second option for significance testing, that we suggest here, is the popular method based on resampling, the (non-parametric) bootstrap confidence intervals. The main goal of resampling methods (jackknife, bootstrap, cross-validation etc.), is to assess the value of a statistic and determine how likely it is to have occurred by chance. Should the population statistic of our choice (standard/higher order/conditional mutual information) for a pair or triplet of variables fall within our chosen confidence intervals, it is included to the next stage of the network reconstruction algorithm.

Non-parametric Bootstrap *Confidence intervals*, are literally bounds within which a certain population statistic is to be found and indicate the reliability of an estimator for the statistic. How broad these bounds will be, is determined by our choice of the *confidence level* α. In other words, the confidence level shows how likely the interval is to contain the parameter. Therefore, increasing the desired confidence level, widens the confidence interval and vice versa.

The bootstrap was introduced by Efron [Efr87] and Tibshirani [DE96] and many variations of it exist today, some generic and others with particular areas of application[1]. Bootstrap builds on the principle, that we can approximate the sampling distribution of the statistic of interest, by repeated sampling from the existing data at hand - i.e. the empirical distribution function. This is achieved, by simulating what would happen if the same experiment was conducted multiple times. The simulation runs, by drawing with replacement samples of size m, from the vector of available values of the statistic x_i. Thus, in a particular bootstrap sample, a given subject from the original study may appear once, twice, many times, or not at all [BH03].

Overall, there are $\begin{pmatrix} 2m-1 \\ m \end{pmatrix}$ distinct samples, with respective probabilities:

$$P(x^{(r)}) = \frac{m}{j_1^{(r)} j_2^{(r)} \cdots j_1^{(r)}} (\frac{1}{m}^{m}),$$

where $j_1^{(r)}$ is the number of copies of x_i in the bootstrap sample. The bootstrap distribution

[1]It is out of the scope of our thesis to discuss the different variations of non-parametric bootstrap, but the interested reader is referred to Chernick (1999) "Bootstrap Methods: A Practioners Guide." John Wiley & Sons.

of the statistic, is then derived by calculating the realisation of the statistic for each bootstrap sample and as $m \to \infty$, the bootstrap distribution becomes an asymptotically valid approximation of the population distribution value of the statistic.

It is computationally impracticable to compute all $\begin{pmatrix} 2m-1 \\ m \end{pmatrix}$ values of the statistic, even for moderate values of m. Therefore, once again, we are faced with the choice of parametric versus non-parametric assumptions about the form of the distribution of the statistic, upon which depends the accuracy of the confidence intervals. In the cases of standard/higher order/conditional mutual information on gene expression data, it is almost impossible to make appropriate assumptions about the relevant distributions and we have to proceed with the non-parametric route.

In either case, the next step, is to draw a large number of samples - around 1000 or more is adequate - and generate the empirical sampling distribution of the functional/statistic. To construct $100(1-\alpha)\%$ confidence intervals for the statistic, we simply take the $100(\alpha/2)$ and $100(1-\alpha/2)$ percentiles of this distribution. A common value for α is 0.1, so the confidence interval is 90%. In practice, this means that in 10% of the cases, on average, the true value of the statistic will not fall within these bounds [BH03]. That, however, should not deter us from setting strict confidence intervals, since the narrower they are, the more information we have about the reliability of the estimator.

As a concluding remark, all the steps outlined in algorithm 10, with the additional choice of bootstrap confidence intervals for the significance testing step, is our proposed scheme for the inference of gene interactions using information theoretic functionals. The full implementation of the algorithm and its evaluation using all the MI estimators considered in our study, is the first priority in our plans for future work.

Chapter 7

Conclusions

The advent of high-throughput technologies like DNA microarrays, initiated a shift in the perspective of expression profiling from a focus on individual genes and functionalities to a more global view of genetic networks and systems. This shift set off an unprecedented demand for meaningful and effective analysis of such data, especially in the area of modelling gene interaction and regulation. With a huge amount of ground biological truths waiting to be uncovered, many of the earlier studies were intended to generate hypotheses rather than to confirm them, so rigid validation of the assumptions was not feasible or relevant, nor often, were the results. However, it slowly becomes apparent, that we are reaching the end of this era; carefully targeted experimental set-ups, specific hypotheses and more sophisticated analysis procedures are slowly emerging. The present work aims to follow this trend.

7.1 Discussion & Summary of Results

In our thesis, we have addressed three critical issues in microarray-based gene expression profiling: non-linearity, sparsity of samples and inference of complex structures. Below, we summarise our choices and findings on each of these issues.

Non-Linearity It is now established by a critical amount of studies, that gene interactions are in general, not linear in nature. Violation of this condition, when modelling gene interactions, results to systematic deviation and poor fitness values. In the present thesis, we address this issue by using information-theoretic functionals, which typically rely on the estimation of mutual information from expression data, in order to measure the statistical dependence between variables. Information theory was preferred as a modelling approach over Bayesian networks, because it can provide equal expressive power and at the same time may require less data and computational time, depending on the assumptions. Moreover, in chapter 5, we show that its non-parametric implementation gives better fitness scores, when compared to its parametric form, on (realistic) synthetic data.

Small Sample Size Arguably, the biggest shortcoming of microarray experiments to date, is the abnormally small number of samples, compared to the number of variables considered. In a typical expression data set, there are over two orders of magnitude fewer observations than factors observed and that, of course, poses unprecedented problems on any estimator or inference method.

Moreover, each different hypothesis we need to generate or confirm, has its own sample requirements, based on the type and, more importantly, number of factors to be considered.

For our study, we addressed the problems by searching the literature for non-parametric mutual information estimators, that would show robustness in under-sampled regimes. Other requirements were, that the estimator had to be able to accommodate continuous random variables, echoing the nature of gene expression and also be scalable to multivariate/higher order mutual information.

Our research started with the most common histogram-based methods - ranging from the widely used naive implementations to the sophisticated adaptive partitioning schemes - and progressed through alternatives based on kernel density estimators and k-nearest neighbours. The main principles of each approach were discussed, along with successful examples of each paradigm.

Five different estimators were selected, to be assessed on the criteria stated above: the naive-

histrogram, the adaptive partitioning method of Darbellay and Vajda [DV99], both of k-nn based estimators of Kraskov [KSG04] and finally, the B-splines based method of Daub et al. [DSSK04].

All the estimators are non-parametric and can be adjusted to compute any information-theoretic functional of our interest, namely classic mutual information, higher order mutual information and combining the two, conditional mutual information. For the small sample behaviour, we estimated the systematic error (bias) and statistical error (standard error) of the estimators on simulated data, drawn from the Gaussian and the logistic distribution, in two different benchmarks. The test were exhaustive on the parameters of each method as well. The sample sizes considered in our simulations, were in the range of $5 - 200$, which definitely represents a very large percentage of microarray studies available.

The primary goal of this assessment on simulated parametric data, was to show, that in limited sample settings, the choice of the particular estimator of the mutual information is particularly important. It became evident throughout the assessment, that both bias and standard error are quite large on the lower range for any estimator, however more or less all of the alternatives, performed consistently better than the naive binning estimator for both distributions and hence should be preferred for the task of network inference.

To take this point a bit further, we "plugged" the estimators considered in various state-of-the-art network inference algorithm and assessed their performance on the quality of the reconstructed network. For a fair comparison, we used a synthetic data generator, which provided the network structure - a combination of different existing networks of E.coli - and the matching expression data.

Again, the alternative algorithms proved consistently better that the naive methods for the small sample sizes, and any of the reconstruction methods. We acknowledge that overall, the precision was very low and the number of false positives is unavoidably very high, but that only highlights the importance of our findings.

Inference of Complex Structures Information theory equips us with the conceptual framework to account for many types of interactions - from the most simple and intimate relationships of pairs of variables to the most complex dependence structures comprising a large set of features.

From the definitions of conditional mutual information and the successive application of the chain rule, we saw in chapter 6, that it is possible to decompose the problem of estimating higher order conditional dependencies, into multiple problems of entropy and joint entropy estimation, or mutual and higher order mutual information.

This observation, inspired a custom, modular network reconstruction algorithm, with the potential of correctly identifying interaction structures of three variables and possibly more, beyond the standard pairwise interactions. Most notably, we show examples of how some of these structures - like Markov-chain structures and the bin-fan motif - can be elicited using a combination of inequalities comprising mutual information and conditional mutual information. For the role of significance assessment, we suggest the use of non-parametric bootstrap confidence intervals, which will help in filtering-out the features which have no chance of being predictive or informative, regardless the inference method.

The proposed algorithm is expected to be more demanding, both computationally (due to the bootstrap) and in terms of sample size (due to the use of conditional mutual information). The demands however, would translate immediately in improvements of predictive power and the reduction of false positive/ false negative edges, compared to the other network reconstruction algorithms. In other words, should more data samples become available, whatever we can save from the estimation of the information-theoretic functionals, we can now dispose on any other improvement technique that proves useful to the problem at hand.

7.2 Future Work

The immediate extension of our work in the present thesis, would be:

1. Use the proposed algorithm with the k-nn methods as well,

2. Find a suitable real dataset and use it to assess its performance by contrasting it to the existing inference methods and most notably a linear model counterpart: partial correlation.

3. Implement ways to improve the efficiency of the algorithm, perhaps by finding computed parts that can be reused.

4. Bundle all the code available - benchmarks as well as our proposed algorithm - in an R package.

Further investigation could also be useful, on uncovering more sets of characteristic equation of the small motifs, via a study of their representative inequalities.

Finally, as a different project, it is possible to remodel this work to accommodate time series experiments, again focusing on how earlier work can be improved upon and focusing on estimation methods that are robust to sample size limitations.

7.3 Final Remarks

If unsupervised, graph learning problems are notoriously difficult [Pea00] and for large-scale gene regulatory network inference, the conditions under which they must be studied, with the sparsity of samples, pose even more challenges.

We showed through simulations and through synthetic data, that making reasonable biological assumptions and with careful choice of estimators - robust to small sample limitations - we can increase the predictive power of the current network reconstructing methodologies.

We hope, that as more background knowledge becomes available, we will be able to compose better surrogate data to test and improve the inference methodologies and predict the correct network dynamics. A great wealth of knowledge, still awaits to be harvested.

Bibliography

[AAR+06] Alexa, Adrian, Rahnenfuhrer, Jorg, Lengauer, and Thomas. Improved scoring of functional groups from gene expression data by decorrelating go graph structure. *Bioinformatics*, 22(13):1600–1607, July 2006.

[AG89] Nabil Ali Ahmed and D. V. Gokhale. Entropy expressions and their estimators for multivariate distributions. *IEEE Transactions on Information Theory*, 35(3):688–, 1989.

[AR94] Adam Arkin and John Ross. Computational functions in biochemical reaction networks. *Biophys. J.*, 67:560–578, 1994.

[ARR+08] Mark D. Alter, Daniel B. Rubin, Keri Ramsey, Rebecca Halpern, Dietrich A. Stephan, L. F. Abbott, and Rene Hen. Variation in the large-scale organization of gene expression levels in the hippocampus relates to stable epigenetic variability in behavior. *PLoS ONE*, 3(10):e3344, Oct 2008.

[BDGM97] J. Beirlant, E. J. Dudewicz, L. Gyrfi, and E. C. Meulen. Nonparametric entropy estimation: An overview. *International Journal of the Mathematical Statistics Sciences*, 6:17–39, 1997.

[BH03] Michael R. Berthold and David J. Hand. *Intelligent Data Analysis*. Springer, April 2003.

[BK00] A. J. Butte and I. S. Kohane. Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. *Pac Symp Biocomput*, pages 418–29, 2000. Using Smart Source Parsing.

[BMS+05] Katia Basso, Adam A A. Margolin, Gustavo Stolovitzky, Ulf Klein, Riccardo Dalla-Favera, and Andrea Califano. Reverse engineering of regulatory networks in human b cells. *Nat Genet*, March 2005.

[CAR05] C. J. Cellucci, A. M. Albano, and P. E. Rapp. Statistical validation of mutual information calculations: Comparison of alternative numerical algorithms. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 71(6):066208, 2005.

[CAW06] Xue-wen Chen, Gopalakrishna Anantha, and Xinkun Wang. An effective structure learning method for constructing gene networks. *Bioinformatics*, 22(11):1367–1374, 2006.

[CT91] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & sons, 1991.

[Dar98] Georges E. Darbellay. An estimator of the mutual information based on a criterion for conditional independence. *Computational Statistics and Data Analysis*, 32:1–17, 98.

[DE96] T. J. Diciccio and B. Efron. Bootstrap confidence intervals. *Stat. Sci.*, 11(3):189–228, 1996.

[Dev86] Luc Devroye. *Non-uniform random variate generation*. Springer-Verlag, Berlin, Germany / Heidelberg, Germany /London, UK / etc., 1986.

[DG06] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 233–240, New York, NY, USA, 2006. ACM.

[Dob59] R.L. Dobrushin. General formulation of Shannon's main theorem of information theory. *Usp. Math. Nauk.*, 14:3–104, 1959. Translated in Am. Math. Soc. Trans., 33:323-438.

[DP05] C. Ding and H. Peng. Minimum redundancy feature selection from microarray gene expression data. *J Bioinform Comput Biol*, 3(2):185–205, April 2005.

[DSSK04] Carsten O. Daub, Ralf Steuer, Joachim Selbig, and Sebastian Kloska. Estimating mutual information using b-spline functions – an improved similarity measure for analysing gene expression data. *BMC Bioinformatics*, 5:118, 2004.

[DV99] G. Darbellay and I. Vajda. Estimation of the information by an adaptive partitioning of the observation space. *IEEE Trans. on Information Theory*, 45:1315–1321, 1999.

[DV00] A. Darbellay and I. Vajda. Entropy Expressions for Multivariate Continuous Distributions. *IEEE Transactions on Information Theory*, 46(2):pp. 709–712, 2000.

[Efr87] Bradley Efron. Better bootstrap confidence intervals. *Journal of the American Statistical Association*, 82(397):171–185, 1987.

[FHT+07] Jeremiah J. Faith, Boris Hayete, Joshua T. Thaden, Ilaria Mogno, Jamey Wierzbowski, Guillaume Cottarel, Simon Kasif, James J. Collins, and Timothy S. Gardner. Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles. *PLoS Biology*, 5(1):e8+, January 2007.

[FLNP00] Nir Friedman, Michal Linial, Iftach Nachman, and Dana Pe'er. Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, 7(3):601–620, August 2000.

[FP07] Stefan Frenzel and Bernd Pompe. Partial mutual information for coupling analysis of multivariate time series. *Physical Review Letters*, 99(20):204101, 2007.

[FS86] Andrew M. Fraser and Harry L. Swinney. Independent coordinates for strange attractors from mutual information. *Phys. Rev. A*, 33(2):1134–1140, Feb 1986.

[GBBK02] Nabil Guelzim, Samuele Bottani, Paul Bourgine, and Francois Kepes. Topological and causal structure of the yeast transcriptional regulatory network. *Nature Genetics*, 31(1):60–63, May 2002.

[Gra83] P. Grassberger. Characterization of strange attractors. *Physical Review Letters*, 50(5):346–349, 1983.

[Gra88] P. Grassberger. Finite sample corrections to entropy and dimension estimates. *Physics Letters A*, 128:369–373, April 1988.

[Gra03] P. Grassberger. Entropy estimates from insufficient samplings, 2003.

[Gum60] E. J. Gumbel. Distribution des valeurs extrêmes en plusieurs dimensions. *Publications de l'Institut de Statistique de l'Université de Paris*, 9:171–173, 1960.

[Har28] R. V. L. Hartley. The transmission of information. *Bell System Technical Journal*, 7:535–564, July 1928.

[HG63] Ray Hyman and Wendel Garner. Uncertainty and structure as psychological concepts. *Science*, 140(3568):799–, 1963.

[Hou06] Jean Houser. *Improving entropy estimation and the inference of genetic regulatory networks.* PhD thesis, Dept. of Biosciences, INSA Lyon, 2006.

[HP91] Marron J.S. Hall P., Sheather S.J. Jones M.C. An optimal data-based bandwidth selection in kernel density estimation. *Biometrica*, 2:263–269, 1991.

[HSPVB07] Katerina Hlavackova-Schindler, Milan Palus, Martin Vejmelka, and Joydeep Bhattacharya. Causality detection based on information-theoretic approaches in time series analysis. *Physics Reports*, 441(1):1–46, March 2007.

[Jak03] Aleks Jakulin. *Learning Based on Attribute Interactions.* PhD thesis, Faculty of Computer and Information Science, University in Ljubljana, 2003.

[JB04] Aleks Jakulin and Ivan Bratko. Quantifying and visualizing attribute interactions, Mar 2004.

[JK72] N. L. Johnson and S. Kotz. *Distributions in Statistics: Continuous Multivariate Distributions*. John Wiley, New York, NY, 1972.

[Kau71] S. Kauffman. Gene regulation networks: a theory for their global structure and behaviors. *Curr Top Dev Biol*, 6(6):145–182, 1971.

[KDB04] C. Knüpfer, P. Dittrich, and C. Beckstein. Artificial gene regulation: a data source for validation of reverse bioengineering. In *Proceedings of the 6th German Workshop on Artificial Life*, pages 66–75. Akademische Verlagsgesellschaft Aka, 2004.

[KGCH05] Kevin H. Knuth, Anthony Gotera, Charles T. Curry, and Karen A. Huyser. Revealing relationships among relevant climate variables with information theory. In *Earth-Sun System Technology Conference 2005*, 2005.

[KK51] J. F. Kenney and E. S. Keeping. *Mathematics of Statistics, Part 2, 2nd ed.* Van Nostrand, Princeton, NJ, 1951.

[Koz87] N.N. Kozachenko, L.F.; Leonenko. Sample estimate of the entropy of a random vector. *Problemy Peredachi Informatsii*, 23(2):9–16, 1987.

[KS61] A. Kaiser and T. Schreiber. Information transfer in continuous processes, 2002/6/1.

[KSG04] Alexander Kraskov, Harald Stoegbauer, and Peter Grassberger. Estimating mutual information. *Physical Review E*, 69:066138, 2004.

[Lan61] R. Landauer. Irreversibility and heat generation in the computing process. *IBM Journal of Research and Development*, 5(??):183–191, 1961.

[LFS98] Shoudan Liang, Stefanie Fuhrmann, and Roland Somogyi. Reveal: A general reverse engineering algorithm inference of genetic networks. *Pacific Symposium on Biocomputing (PSB)*, 3:18–29, 1998.

[LW08] Kuo-Ching Liang and Xiaodong Wang. Gene regulatory network reconstruction using conditional mutual information. *EURASIP Journal on Bioinformatics and Systems Biology*, 2008, 2008.

[Mac03] D.J.C. MacKay. *Information Theory, Inference, and Learning Algorithms.* Cambridge University Press, 2003.

[Mat99] Hiroyuki Matsuda. Physical nature of higher-order mutual information: Intrinsic correlations and frustration. *Phys. Rev. E*, 62(3):3096–3102, Sep 1999.

[McG54] William McGill. Multivariate information transmission. *Psychometrika*, 19(2):97–116, June 1954.

[Men93] P. Mendes. Gepasi: a software package for modelling the dynamics, steady states and control of biochemical and other systems. *Comput Appl Biosci*, 9(5):563–571, October 1993.

[Mit97] Tom M. Mitchell. *Machine Learning.* McGraw-Hill, New York, 1997.

[MKD+04] Hong-Wu Ma, Bharani Kumar, Uta Ditges, Florian Gunzer, Jan Buer, and An-Ping Zeng. An extended transcriptional regulatory network of escherichia coli and analysis of its hierarchical structure and network motifs. *Nucl. Acids Res.*, 32(22):6643–6649, December 2004.

[MKLB07] Patrick E. Meyer, Kevin Kontos, Frederic Lafitte, and Gianluca Bontempi. Information-theoretic inference of large transcriptional regulatory networks. *EURASIP J. Bioinformatics Syst. Biol.*, 2007(1):8–8, 2007.

[MLB07] Patrick E. Meyer, Frederic Lafitte, and Gianluca Bontempi. *minet: Mutual Information Network Inference*, 2007. R package version 1.1.6.

[MNB+04] Adam A. Margolin, Ilya Nemenman, Katia Basso, Ulf Klein, Chris Wiggins, Gustavo Stolovitzky, Riccardo Dalla Favera, and Andrea Califano. Aracne: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context, 2004.

[Mod86] R. Moddemeijer. On estimation of entropy and mutual information of continuous distributions. *EUSIPCO-86*, pages 1033–1035, September 2-5 1986.

[Mod99] R. Moddemeijer. A statistic to estimate the variance of the histogram based mutual information estimator based on dependent pairs of observations. *Signal Processing*, 75(1):51–63, 1999.

[MRL95] Young-Il Moon, Balaji Rajagopalan, and Upmanu Lall. Estimation of mutual information using kernel density estimators. *Phys. Rev. E*, 52(3):2318–2321, Sep 1995.

[MSK03] P. Mendes, W. Sha, and Y. Keying. Artificial gene networks for objective comparison of analysis algorithms. *Bioinformatics*, 19(Suppl. 2), 2003.

[NBdRvS04] Ilya Nemenman, William Bialek, and Rob de Ruyter van Steveninck. Entropy and information in neural spike trains: Progress on the sampling problem. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 69(5):056111, 2004.

[Ng04] See-Kiong Ng. Molecular biology for the practical bioinformatician. In Limsoon Wong, editor, *The Practical Bioinformatician*, chapter 1, pages 193–215. World Scientific, 5 Toh Tuck Link, Singapore 596224, 2004.

[NZ03] S. Nadarajah and K. Zografos. Formulas for rényi information and related measures for univariate distributions. *Inf. Sci.*, 155(1-2):119–138, 2003.

[Pan03] Liam Paninski. Estimation of entropy and mutual information. *Neural Comp.*, 15(6):1191–1253, June 2003.

[PcvS03] Milan Paluš and Aneta Stefanovska. Direction of coupling from phases of interacting oscillators: An information-theoretic approach. *Phys. Rev. E*, 67(5):055201, May 2003.

[Pea00] Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, March 2000.

[PK98] Foster Provost and Ron Kohavi. On applied research in machine learning. In *Machine learning*, pages 127–132, 1998.

[PQB05] Ernesto Pereda, Rodrigo Quian Quiroga, and Joydeep Bhattacharya. Nonlinear multivariate analysis of neurophysiological signals, 2005.

[Pri95] Dean Prichard. Generalized redundancies for time series analysis. *Physica D*, 84:476–493, 1995.

[Pro98] Foster Provost. The case against accuracy estimation for comparing induction algorithms. In *In Proceedings of the Fifteenth International Conference on Machine Learning*, pages 445–453. Morgan Kaufmann, 1998.

[PV07] Milan Palus and Martin Vejmelka. Directionality of coupling from bivariate time series: How to avoid false causalities and missed connections. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 75(5):056211, 2007.

[R D08] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008. ISBN 3-900051-07-0.

[RG05] S. Rogers and M. Girolami. A bayesian regression approach to the inference of regulatory networks from gene expression data. *Bioinformatics*, 21(14):3131–3137, July 2005.

[Rot52] J. Rothstein. Organization and Entropy. *Journal of Applied Physics*, 23:1281–1282, November 1952.

[RTS05] J. J. Rice, Y. Tu, and G. Stolovitzky. Reconstructing biological networks using conditional correlation analysis. *Bioinformatics*, 21(6):765–773, March 2005.

[RV94] B. D. Ripley and W. N. Venables. *Modern applied statistics with S-Plus*. Springer-Verlag, New York, NY, 1994.

[Sch96] M. Schena. Genome analysis with gene expression microarrays. *Bioessays*, pages 427–431, 1996.

[Sch00] Thomas Schreiber. Measuring information transfer. *Phys. Rev. Lett.*, 85(2):461–464, Jul 2000.

[SH78] S. Satterthwaite and T. Hutchinson. A generalisation of gumbel's bivariate logistic distribution. *Metrika*, 25(1):163–170, December 1978.

[Sha48] Claude E. Shannon. *A Mathematical Theory of Communication*, volume 27. Bell Systems Technical Journal, July & October 1948.

[Sil86] B. W. Silverman. *Density estimation for statistics and data analysis*. Monographs on Statistics and Applied Probability, London: Chapman and Hall, 1986, 1986.

[SJH02] V. A. Smith, E. D. Jarvis, and A. J. Hartemink. Evaluating functional network inference using simulations of complex biological systems. *Bioinformatics*, 18 Suppl 1, 2002.

[SKD+02] R. Steuer, J. Kurths, C. O. Daub, J. Weise, and J. Selbig. The mutual information: detecting and evaluating dependencies between variables. *Bioinformatics*, 18 Suppl 2, 2002.

[Som96] R. Somogyi. Genetic network inference in computational models and applications to large-scale gene expression data, 1996.

[SOMMA02] Shai S. Shen-Orr, Ron Milo, Shmoolik Mangan, and Uri Alon. Network motifs in the transcriptional regulation network of escherichia coli. *Nat Genet*, 31(1):64–68, May 2002.

[Soo92] Nader Ebrahimi; Mohamed Habibullah; Ehsan S. Soofi. Testing exponentiality based on kullback-leibler information. *Journal of the Royal Statistical Society. Series B (Methodological*, 54(3):739–748, 1992.

[SS] Hideaki Shimazaki and Shigeru Shinomoto. Histogram bin size selection.

[SS05a] Juliane Schäfer and Korbinian Strimmer. An empirical Bayes approach to inferring large-scale gene association networks. *Bioinformatics*, 21(6):754–64, Mar 2005.

[SS05b] Juliane Schäfer and Korbinian Strimmer. A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology*, 4(1), 2005.

[SS07] Hideaki Shimazaki and Shigeru Shinomoto. A recipe for optimizing a time-histogram. *Advances in Neural Information Processing Systems 19*, 2007.

[SYV98] M. Studený, Y, and J. Vejnarová. The multi-information function as a tool for measuring stochastic dependence. In *Learning in Graphical Models*, pages 261–298. Kluwer Academic Publishers, 1998.

[TFMF01] G.D. Tourassi, E.D. Frederick, M.K. Markey, and C.E. Floyd. Application of the mutual information criterion for feature selection in computer-aided diagnosis. *Medical Physics*, pages 2394–2402, 2001.

[THS01] A. Sharma T.I. Harrold and S. Sheather. Selection of kernel bandwidth for measuring dependence in hydrologic time series using the mutual information. *Stochastic environmental research and risk assessment*, 15(4):310–324, 2001.

[Vap99] V. N. Vapnik. An overview of statistical learning theory. *Neural Networks, IEEE Transactions on*, 10(5):988–999, 1999.

[VdBVLN+06] T. Van den Bulcke, K. Van Leemput, B. Naudts, P. van Remortel, H. Ma, A. Verschoren, B. De Moor, and K. Marchal. Syntren: a generator of synthetic gene expression data for design and analysis of structure learning algorithms. *BMC Bioinformatics*, 7, 2006.

[Vic02] Jonathan D. Victor. Binless strategies for estimation of information from neural data. *Phys. Rev. E*, 66(5):051903, Nov 2002.

[WAG+06] Werhli, V. Adriano, Grzegorczyk, Marco, Husmeier, and Dirk. Comparative evaluation of reverse engineering gene regulatory networks with relevance networks, graphical gaussian models and bayesian networks. *Bioinformatics*, 22(20):2523–2531, October 2006.

[Wat60] Satosi Watanabe. Information theoretical analysis of multivariate correlation. *IBM Journal of Research and Development*, 4:66–82, 1960.

[WJ95] M. Wand and M. Jones. *Kernel Snoothing*. Chapman and Hall, London, 1995.

[YAP+04] Jing Yu, Anne, Paul, Alexander, and Erich. Advances to bayesian network inference for generating causal networks from observational biological data. *Bioinformatics*, July 2004.

[YTC02] M. K. Yeung, J. Tegnér, and J. J. Collins. Reverse engineering gene networks using singular value decomposition and robust regression. *Proc Natl Acad Sci U S A*, 99(9):6163–6168, April 2002.

[ZDS01] D. E. Zak, F. J. Doyle, and J. S. Schwaber. Simulation studies for the identification of genetic networks from cdna array and regulatory activity data. *Proceedings of the Second International Conference on Systems Biology*, 2001.