MANCHESTER 1824
The University of Manchester

A novel approach to estimating information-theoretic measures for exploratory data analysis and explainable machine learning

A thesis submitted to the University of Manchester for the degree of
Doctor of Philosophy
in the Faculty of Science and Engineering

2021

Lisa K. Crow
Department of Physics and Astronomy

Contents

Word Count: 33684

List of figures

List of tables

List of publications

Watts, S. J., & Crow, L. (2019). Big variates — visualising and identifying key variables in a multivariate world. Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 940(December 2018), 441–447. https://doi.org/10.1016/j.nima.2019.06.060

Abstract

It is a common misconception in data analysis that more data equates to more knowledge. However, as data becomes bigger, the methods required to decipher and visualise critical information become an ever more cumbersome task. A central problem in data analysis is identifying and understanding relationships in complex systems. Mutual information has proven valuable in this regard and is already a crucial measure in many data analysis and machine learning tasks. Nearest neighbour techniques are a classic approach in non-parametric statistics and have proven effective in entropy estimation. Unfortunately, it is well established that estimating entropy is fraught with difficulties, especially for discrete-continuous mixed cases.

In this thesis, we develop an ensemble method to estimate information-theoretic measures using a novel noisy resampling technique. The method is empirically shown to be asymptotically unbiased and consistent. Moreover, through artificial and real-world experiments, we show that the approach repeatedly outperforms the current leading k-nearest-neighbour methods - the Kozachenko-Leonenko estimator and the KSG estimator - to achieve a more accurate and robust estimate for discrete and continuous random variables alike. This ability is essential in classification problems, where the class variable is often discrete, significantly widening the applicability of mutual information measures in data analysis. In real-world domains, the proposed method successfully identifies key variables supported by machine learning results.

New algorithms are implemented in an exploratory analysis tool for multivariate data. We investigate the visualisations currently used for multi-dimensional data and introduce *DataViewer*, a visualisation software package for exploring patterns in modern data sets. We propose a *variable interaction diagram* for illustrating variable correlations with significant mutual information. The software is designed to aid interpretation of complex data structures, which in turn motivates intelligent feature selection. The techniques are illustrated by application to several real world data sets.

Declaration of originality

I hereby confirm that no portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright statement

i The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the "Copyright") and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

ii Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made *only* in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

iii The ownership of certain Copyright, patents, designs, trademarks and other intellectual property (the "Intellectual Property") and any reproductions of copyright works in the thesis, for example graphs and tables ("Reproductions"), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

iv Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see `http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=24420`), in any relevant Thesis restriction declarations deposited in the University Library, The University Library's regulations (see `http://www.library.manchester.ac.uk/about/regulations/`) and in The University's policy on Presentation of Theses.

Acknowledgements

I feel incredibly fortunate to have worked with my supervisor, Professor Stephen Watts, without whom I would not have completed this thesis. I am extremely grateful for his patience, guidance, and support, and our weekly catch-ups throughout the pandemic were a welcome respite in my week. His passion and enthusiasm continue to inspire me.

I would also like to thank my family and friends who encouraged me when my confidence wavered and never stopped believing. Those who checked in on me in the depths of my writing, even if I had ignored their last five messages. Or posting care packages through my door when I hadn't left the house in days. Their unconditional support did not go unnoticed.

A special thanks must go to my partner for putting up with my late nights in the office. They have provided nothing but comfort, motivation, and endless cups of tea and biscuits through the highs and lows of my studies.

Chapter 1

Introduction

1.1 Motivation

In the world of "Big data," the goal of quantifying interactions in complex data structures is essential. Information-theoretic tools are becoming increasingly prevalent in data analysis and machine learning due to their sensitivity to non-linear dependencies [1]. Several information-theoretic quantities, such as entropy, mutual information, and Kullback-Leibler divergence, have proven their effectiveness in data analysis as measures of *relevancy* [1] and *redundancy* [2]. Mutual information, the most popular measure, plays a core role in many machine learning applications. The measure quantifies the amount of information that two variables share. This has led to its widespread use in machine learning including topics such as goodness-of-fit tests [2], clustering [3]–[6], parameter estimation [7] and feature selection [8]–[12]. All of which heavily rely on the successful identification of key variables to avoid over-fitting and sub-optimal learning rates.

It is crucial to reliably estimate the necessary information quantities before applying information-theoretic tools to data analysis and machine learning. Despite the widespread use of these measures, it is well established that a robust entropy estimate from empirical samples is fraught with difficulties. Consequently, entropy estimation is still an active research topic [13]–[21]. In addition, entropy is only well defined for discrete random variables and continuous random variables independently. In other words, the mathematical definition of entropy varies depending on the data type. Thus, the two are dissonant. Therefore, most estimators focus on either purely continuous or discrete cases, which are scarce in real-world applications, particularly in classification problems that involve a discrete class variable.

Furthermore, in real-world problems, probability distributions are generally unknown, calling for non-parametric techniques. For discrete random variables, the most common and intuitive non-parametric method to estimate entropy is to count the frequencies of the observed values. However, the problem is more complex for continuous random variables, as the number of possible outcomes appears infinite. Previous attempts to work around this problem typically involve quantising, or binning, continuous variables into discrete bins and applying a discrete entropy

[1] A variable is *relevant* when it is informative of a target variable, typically a class variable for classification.
[2] A variable is *redundant* if the *relevant* information it contains about a target variable is also possessed by other variables.

estimator. However, "good" quantisation is complicated, and the unavoidable dependence on the bin width introduces a bias into the entropy estimate, as well as a vulnerability to statistical fluctuations [22]. Alternatively, one can add noise to discrete samples and apply continuous methods. However, this is yet to be done successfully in the literature [23]. Instead, more complex and accurate methods exist that are only applicable to continuous random variables. This thesis focuses on the popular k-nearest-neighbour techniques.

This research aimed to develop a robust entropy estimation method that applies to continuous and discrete random variables. By broadening the applicability of information estimators, the benefits of information theory can be more widely appreciated. In the era of big data the ability to handle a variety of data types and successfully identify hidden structures from empirical samples is of increasing importance.

1.2 Thesis Overview

Real-world systems consist of a variety of data types, and a widely applicable entropy estimator is necessary to successfully deal with these cases. In this thesis, we investigate the estimation of information-theoretic measures, with the aim to create a robust entropy estimator that can be used for continuous and discrete variables alike. We demonstrate that our proposed estimator successfully deals with abnormal distributions, for example distributions with a large kurtosis, that common differential entropy estimators struggle with. The algorithm combines discrete and continuous methods using the Kozachenko-Leonenko estimator as a base to apply a novel noisy resampling technique.

In chapter 2 we provide an overview of information-theoretic concepts relevant for data analysis, including mutual information as a measure for variable dependence. In particular, we discuss the differences between Shannon entropy and differential entropy for discrete and continuous variables. We review current entropy estimation techniques in chapter 3. In the first half we compare fixed-width and adaptive partitioning methods for discrete variables. In the second half, we discuss the more accurate k-nearest-neighbour methods for entropy and mutual information of continuous variables. We consider the problems associated with each approach.

We begin chapter 4 by discussing the quantisation of random variables in terms of Shannon entropy and contrast it to the traditional bin width-centred view. Next, we discuss the difficulties using randomised discrete variables. In doing so, we derive a formula to explain the resultant biases observed in the literature, and therefore present an alternative to correct this. Finally, we combine these concepts and propose our novel noisy resampling algorithm for entropy estimation, which we empirically show is asymptotically unbiased and consistent. For artificial data, we compare the proposed method to popular k-nearest-neighbour estimators for continuous, discrete and mixed cases, adding the correct noise where appropriate.

The first half of chapter 5 evaluates the proposed method on real-world data. Correlated variables are identified via the mutual information estimates, and the results are analysed using correlation measures and machine learning models. These are compared with those obtained from popular k-nearest-neighbour estimators to demonstrate the superior performance of the proposed method. In the second half, the Kullback-Leibler divergence, another information measure, is discussed. We show how our method can be adapted to a k-nearest-neighbour divergence estimator. We use synthetic and real-word data examples to discuss the Kullback-Leibler divergence as a measure for predictivity in classification problems. We empirically show that for finite samples, there are statistical limits inherent with the estimate of the Kullback-Leibler divergence of $\mathcal{O}(\log(N))$ and relate it to the machine learning statistic Cohen's *kappa*.

Beyond designing algorithms for information-theoretic measures, we want to apply the ideas discussed in this thesis to real-word data to extract valuable information. In chapter 6 we discuss visualisations and the role they play in the human understanding of data structures. We illustrate a novel visualisation software package, DataViewer that implements the algorithms presented in this thesis. Here we explore the use of mutual information and the Kullback-Leibler divergence in exploratory data analysis and machine learning, using visualisation to present the structures detected.

References

[1] I. Kojadinovic, "On the use of mutual information in data analysis: an overview," *Proceedings of 11th International Symposium on Applied Stochastic Models and Data Analysis (ASMDA'05)*, pp. 738–747, 2005.

[2] H. Alizadeh Noughabi, "A new estimator of kullback–leibler information and its application in goodness of fit tests," *Journal of Statistical Computation and Simulation*, vol. 89, no. 10, pp. 1914–1934, 2019.

[3] A. Kraskov, H. Stögbauer, R. G. Andrzejak, and P. Grassberger, "Hierarchical clustering using mutual information," *Europhysics Letters*, vol. 70, no. 2, pp. 278–284, 2005, ISSN: 02955075. DOI: 10.1209/epl/i2004-10483-y. arXiv: 0311037 [q-bio].

[4] H. Qin, X. Ma, T. Herawan, and J. M. Zain, "Mgr: An information theory based hierarchical divisive clustering algorithm for categorical data," *Knowledge-Based Systems*, vol. 67, pp. 401–411, 2014, ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2014.03.013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705114001026.

[5] H. Liu, J. Zou, and N. Ravishanker, "Clustering high-frequency financial time series based on information theory," *Applied Stochastic Models in Business and Industry*, 2021.

[6] F. Rashidi, S. Nejatian, H. Parvin, and V. Rezaie, "Diversity based cluster weighting in cluster ensemble: An information theory approach," *Artificial Intelligence Review*, vol. 52, no. 2, pp. 1341–1368, 2019.

[7] J. Chao, E. S. Ward, and R. J. Ober, "Fisher information theory for parameter estimation in single molecule microscopy: Tutorial," *J. Opt. Soc. Am. A*, vol. 33, no. 7, B36–B57, Jul. 2016. DOI: 10.1364/JOSAA.33.000B36. [Online]. Available: http://www.osapublishing.org/josaa/abstract.cfm?URI=josaa-33-7-B36.

[8] H. Liu, J. Sun, L. Liu, and H. Zhang, "Feature selection with dynamic mutual information," *Pattern Recognition*, vol. 42, no. 7, pp. 1330–1339, 2009, ISSN: 00313203. DOI: 10.1016/j.patcog.2008.10.028.

[9] J. R. Vergara and P. A. Estévez, "A review of feature selection methods based on mutual information," *Neural Computing and Applications*, vol. 24, no. 1, pp. 175–186, 2014, ISSN: 09410643. DOI: 10.1007/s00521-013-1368-0.

[10] S. Xiangchenyang and L. Fang, "A Filter Approach to Feature Selection Based on Survival Cauchy-Schwartz Mutual Information," *Proceedings - 20th International Conference on High Performance Computing and Communications, 16th International Conference on Smart City and 4th International Conference on Data Science and Systems, HPCC/SmartCity/DSS 2018*, pp. 1381–1386, 2019. DOI: 10.1109/HPCC/SmartCity/DSS.2018.00228.

[11] M. Bennasar, Y. Hicks, and R. Setchi, "Feature selection using Joint Mutual Information Maximisation," *Expert Systems with Applications*, vol. 42, no. 22, pp. 8520–8532, 2015, ISSN: 09574174. DOI: 10.1016/j.eswa.2015.07.007. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2015.07.007.

[12] E. Hancer, B. Xue, and M. Zhang, "Differential evolution for filter feature selection based on information theory and feature ranking," *Knowledge-Based Systems*, vol. 140, pp. 103–119, 2018.

[13] L. Kozachenko and N. Leononko, "Sample Estimate of the Entropy of a Random Vector," *Probl. Peredachi Inf.*, vol. 23, no. 2, pp. 9–16, 1987.

[14] G. A. Darbellay and I. Vajda, "Estimation of the information by an adaptive partitioning of the observation space," *IEEE Transactions on Information Theory*, vol. 45, no. 4, pp. 1315–1321, 1999, ISSN: 00189448. DOI: 10.1109/18.761290.

[15] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, vol. 69, no. 6, p. 16, 2004, ISSN: 1063651X. DOI: 10.1103/PhysRevE.69.066138. arXiv: 0305641 [cond-mat].

[16] Q. Wang, S. R. Kulkarni, and S. Verdú, "Divergence estimation of continuous distributions based on data-dependent partitions," *IEEE Transactions on Information Theory*, vol. 51, no. 9, pp. 3064–3074, 2005, ISSN: 00189448. DOI: 10.1109/TIT.2005.853314.

[17] A. Hacine-Gharbi, M. Deriche, P. Ravier, R. Harba, and T. Mohamadi, "A new histogram-based estimation technique of entropy and mutual information using mean squared error minimization," *Computers and Electrical Engineering*, vol. 39, no. 3, pp. 918–933, 2013, ISSN: 00457906. DOI: 10.1016/j.compeleceng.2013.02.010. [Online]. Available: http://dx.doi.org/10.1016/j.compeleceng.2013.02.010.

[18] A. Kolchinsky and B. D. Tracey, "Estimating mixture entropy with pairwise distances," *Entropy*, vol. 19, no. 7, pp. 1–17, 2017, ISSN: 10994300. DOI: 10.3390/e19070361. arXiv: 1706.02419.

[19] K. R. Moon, K. Sricharan, and A. O. Hero, "Ensemble estimation of mutual information," *IEEE International Symposium on Information Theory - Proceedings*, pp. 3030–3034, 2017, ISSN: 21578095. DOI: 10.1109/ISIT.2017.8007086.

[20] G. Ariel and Y. Louzoun, "Estimating differential entropy using recursive copula splitting," *Entropy*, vol. 22, no. 2, 2020, ISSN: 10994300. DOI: 10.3390/e22020236. arXiv: 1911.06204.

[21] R. Spring and A. Shrivastava, "Mutual information estimation using LSH sampling," *IJCAI International Joint Conference on Artificial Intelligence*, vol. 2021-January, pp. 2807–2815, 2020, ISSN: 10450823. DOI: 10.24963/ijcai.2020/389.

[22] J. D. Victor, "Binless strategies for estimation of information from neural data," *Physical Review E*, no. November, 2002. DOI: 10.1103/PhysRevE.66.051903.

[23] W. Gao, S. Kannan, S. Oh, and P. Viswanath, "Estimating mutual information for discrete-continuous mixtures," *Advances in Neural Information Processing Systems*, vol. 2017-Decem, no. Nips, pp. 5987–5998, 2017, ISSN: 10495258. arXiv: 1709.06212.

Chapter 2

Basic Concepts in Information Theory

Information theory, proposed by Claude Shannon in 1948, is intended to quantitatively describe information transfer limits in communication systems. One can consider the input and outputs of a communication system as random variables in a data set. The efficiency of information transferred across channels is analogous to the correlation between those variables. For an ideal channel, there is a one-to-one mapping between the input and output corresponding to highly dependent variables. If the channel is significantly noisy the input and output would be independent. Note that the direction of transmission is irrelevant, as knowing either the input or output would provide information about the other. Thus, the efficiency is an attribute of the channel or relationship, rather than the variables [1].

Since its creation, information theory has gained momentum in various fields, from multimedia processing to data analysis [2]–[6]. However, the multidisciplinary success of information theory is due to the foundations in probability rather than describing physical systems by communication channels. The result is a generalised theory of information content and correlation.

2.1 Shannon entropy and Mutual Information

Information theory is built on the premise of entropy which can be interpreted as a measure of the average information or uncertainty associated with a random variable. Let $P(X)$ denote the probability of the random variable X with q discrete outcomes. The Shannon entropy of $P(X)$ is defined as:

$$H(X) \equiv -\sum_{x \epsilon X} P(x) \log_2(P(x))^1 \tag{2.1}$$

Where the base of the logarithm gives the units. A base of 2 expresses Shannon entropy in terms of bits. Another common unit is nats corresponding to natural logarithms. Note that it is convention that X represents its probability distribution, $H(X) = H(P(X))$ and this notation is used throughout.

By construction Shannon entropy is a non-negative quantity: $H(X) \geq 0$ in order to satisfy

[1] The symbol $\equiv$ is used to make explicit that this is a definition.

the necessary criteria for an information measure. In the case that $P(x) = 0$ by convention $0\log(0) = 0$ as an event that does not occur contains no information. By the same token, a highly probabilistic event is not very surprising and therefore contains less information than a lower probability event. The average information output of a system is maximised when all possible outcomes are equally probable, $P(X) = 1/q \; \forall X$, therefore $H(X) \leq \log_2(q)$ with equality *iff* X is uniform. This forms a concave function of the distribution of X, as seen in figure 2.1 for a distribution of probabilities of a two-outcome system, such as a biased coin [7], [8].

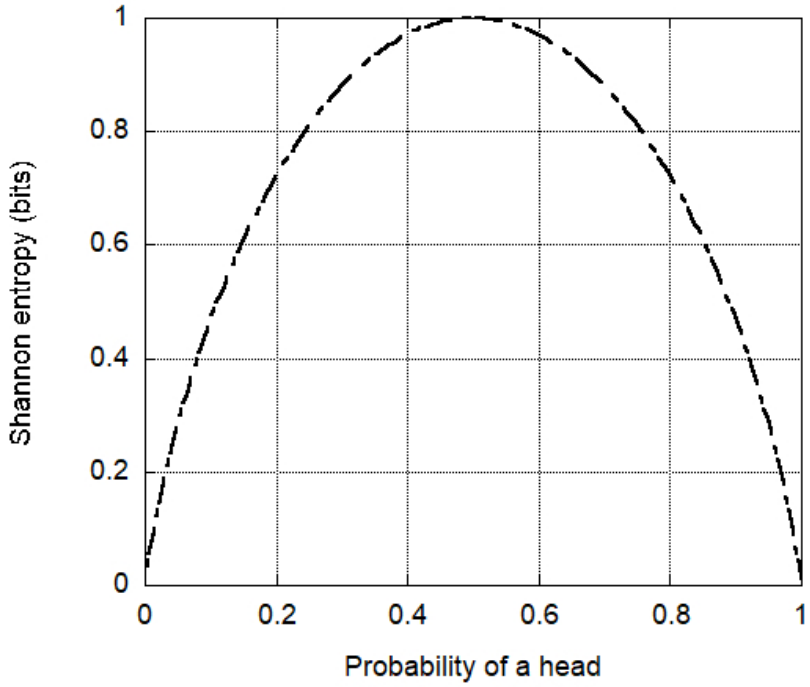

Figure 2.1: A graph to show the concave function of Shannon entropy as a function of probability for a two-outcome system, such as getting a head on a biased coin.

The entropy of a joint distribution for a pair of discrete random variables (X, Y) is a straightforward generalisation of equation (2.1),

$$H(X,Y) = -\sum_{x \epsilon X} \sum_{y \epsilon Y} P(x.y) \log_2(P(x.y)) \tag{2.2}$$

The joint entropy shares the same properties as with a single variable. In addition to these properties, the joint entropy is sub-additive $H(X,Y) \leq H(X) + H(Y)$, where the equality occurs *iff* X and Y are independent. The properties of Shannon entropy can be better understood via the illustration in figure 2.2 demonstrating the relationships between information measures in a two-dimensional system. $H(X|Y)$ is the conditional entropy quantifying the uncertainty surrounding X when given the knowledge of Y and vice-versa. $I(X,Y)$, corresponding to the intersection of $H(X)$ and $H(Y)$, quantifies the information shared by the random variables X

and Y, called the mutual information.

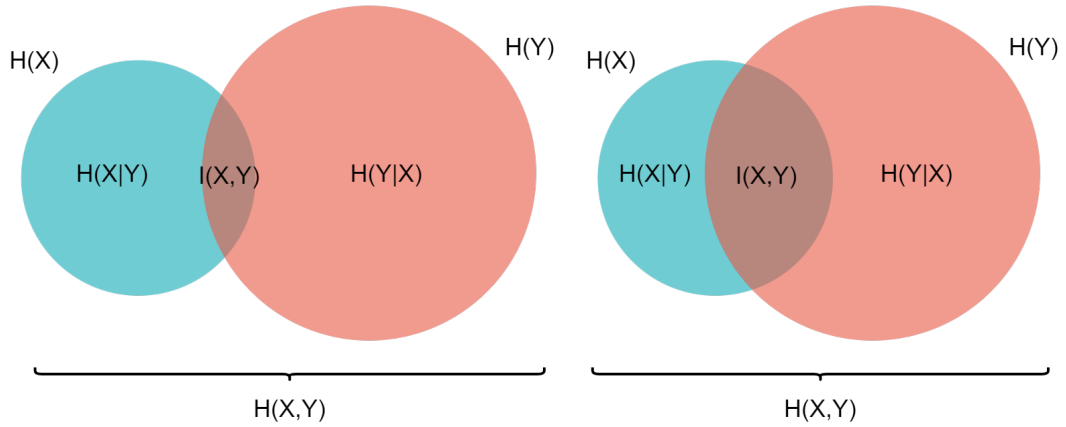

Figure 2.2: A Venn-diagram representation to illustrate the relationships between information measures of discrete variables X and Y. Each region is labelled to indicate the information measure they represented, where the area of the regions is proportional to the magnitude of the value they represent. Note that $H(X)$, $H(Y)$ and $H(X,Y)$ remain constant in both diagrams. However, the intersection is larger for the diagram on the right, illustrating a larger mutual information and a stronger relationship.

Mutual information is defined as the difference between the uncertainty of Y before we observe X and the uncertainty of Y after observing X. Therefore, it is the reduction in uncertainty of Y caused by observing X and similarly it is the amount of information gained about Y after observing X. The same is true when X and Y are interchanged as mutual information is a symmetric measure $I(X,Y) = I(Y,X)$. This can better understood by considering the mutual information as a measure of the difference in entropy of the joint distribution and the joint distribution assuming X and Y were independent. These are mathematically equivalent via the chain rule.

$$\begin{aligned} I(X,Y) &\equiv H(Y) - H(Y|X) \\ &= H(Y) + H(X) - H(X,Y) \end{aligned} \tag{2.3}$$

Mutual information is non-negative, $I(X,Y) \geq 0$, and the magnitude determines the strength of the dependence, such that $I(X,Y) = 0$ can only mean that X and Y are statistically independent. In addition, two variables cannot share more information than the smallest individual entropy $I(X,Y) \leq \text{Min}[H(X), H(Y)]$, this can be visualised using the Venn diagram-like depiction in figure 2.2.

2.2 Differential entropy and Mutual Information

By construction, the number of possible outcomes for the Shannon entropy is finite. However, for many physical problems the distributions in question are continuous, such that $q \to \infty$. To

calculate the Shannon entropy of a continuous probability distribution, $p(x)$ a discrete estimate of the probability density function (pdf) can be constructed by segmenting the distribution into q intervals of width Δ, such that $P_i = p(x_i)\Delta$. Allowing the Shannon entropy of a quantised estimate of the pdf, $H(X^\Delta)$ to be written as:

$$\begin{aligned} H(X^\Delta) &= -\sum_{i=1}^{q} p(x_i)\Delta \log_2(p(x_i)\Delta) \\ &= -\sum_{i=1}^{q} p(x_i)\Delta\Big(\log_2(p(x_i)) + \log_2(\Delta)\Big) \\ &= -\sum_{i=1}^{q} p(x_i)\Delta \log_2(p(x_i)) - \log_2(\Delta)\sum_{i=1}^{q} P_i \\ &\approx -\int_{-\infty}^{\infty} p(x)\log_2(p(x))dx - \infty \end{aligned} \tag{2.4}$$

In the limit of $\Delta \to 0$, $q \to \infty$ and $p(x_i) \to p(x)$, such that the first term becomes an integral over all space. In that same limit, the second term tends to infinity $\lim_{\Delta\to 0} \log_2(\Delta) \to \infty$ [9], suggesting that the Shannon entropy, and consequently the uncertainty of a continuous distribution, is infinite for all pdfs. Therefore, this quantity is not helpful, as the infinite term washes away any interesting information contained in the integral. Thus, a "differential entropy" quantity is defined, which ignores this infinite term.

$$h(X) = -\int_{-\infty}^{\infty} p(x)\log_2(p(x))dx \tag{2.5}$$

Although the differential entropy seems a natural extension to the Shannon entropy, they do not possess all the same properties, nor can they be interpreted in the same manner. An unusual property of differential entropy is that a continuous distribution can have a zero or negative differential entropy, unlike its discrete counterpart. For example, a continuous uniform distribution $\mathcal{U}(0,1)$ has a differential entropy of $h(X) = 0$ bits. Similarly, $\mathcal{U}(0,0.5)$, $h(X) = -1$ bits. Therefore, the interpretation of the differential entropy, when considered in isolation, is not well understood. Because of this property, the Venn diagram representation shown in figure 2.2, breaks down, as it cannot illustrate negative information.

Another unusual property of differential entropy is that it is not scale-invariant. A continuous variable measured in millimetres would have $\log_2(1000)$ bits more differential entropy than the same variable in meters. More generally, multiplying a continuous distribution X by a constant such that $Y = cX$ gives a differential entropy of $h(Y) = h(X) + \log_2|c|$. Although X completely describes Y, Y contains more information. A pragmatic explanation is that increasing or decreasing the range of the variable equivalently affects the number of fixed-width bins needed to describe the pdf reliably. Contrary to Shannon entropy, which only depends on the outcome probability rather than the value of the outcome.

This property explains why the Shannon entropy of a pdf is theoretically infinite. A pdf has infinite unique outcomes, and each outcome can convey infinite amounts of information. By continuously increasing the sample size and precision of the outcomes, the Shannon entropy of an estimated pdf would also continuously increase. In practice, however, we are limited to a finite sample with finite precision. Therefore, if we quantise an empirical sample of a continuous distribution, as $\Delta \to 0$ each segment will contain either 0 ($p(x_i) = 0$) or 1 ($p(x_i) = 1/N$) observation. As segments with zero probability do not contribute, the maximum number of unique contributing bins is N and thus the maximum Shannon entropy for any pdf is limited to $H(X^\Delta) \leq \log_2(N)$ bits. This is consistent with the maximum Shannon entropy of a discrete probability distribution with q unique outcomes $H(X) \leq \log_2(q)$.

It is evident from the differences in properties that the Shannon entropy and the differential entropy are not the same measure and are not interchangeable. Note the capitalisation of the Shannon entropy, $H(X)$ and the lower-case $h(X)$, which will only refer to the differential entropy for all future uses. This capitalisation is the convention, but the literature is fraught with inconsistencies, and the reader should be wary of this. From equation 2.4 the Shannon and differential entropy can be related via

$$H(X) \approx h(X) - \log_2(\Delta) \tag{2.6}$$

This relationship is approximate however we have dropped the superscript Δ, which previously indicated that X^Δ was a discrete approximation of the pdf X.

For the joint differential entropy with the joint pdf $p(x, y)$ and marginal pdfs $p(x)$ and $p(y)$:

$$h(X, Y) = -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x, y)\log_2(p(x, y))dxdy \tag{2.7}$$

The same relationship between $H(X, Y)$ and $h(X, Y)$ can be derived. Starting from an estimation for the joint differential entropy, where $p(x, y)$ is approximated as $P_{ij} = p(x_i, y_j)\Delta_x\Delta_y$. As before, the probability of a given point (x_i, y_j) being in bin (i, j) is the proportion of the volume of bin (i, j) to the total histogram. Thus, the joint Shannon entropy of two continuous random variables is similarly related to the differential entropy via

$$H(X, Y) = h(X, Y) - \log_2(\Delta_x\Delta_y) \tag{2.8}$$

In line with the joint Shannon entropy, $h(x, y) \leq h(x) + h(y)$ still holds, with equality *iff* X and Y are independent [8].

Similarly, the mutual information between two continuous random variables has the same prop-

erties as for the discrete case, and is defined as

$$\begin{aligned} I(X,Y) &= \int p(x,y) \log_2 \left(\frac{p(x,y)}{p(x)p(y)} \right) dxdy \\ &= h(X) + h(Y) - h(X,Y) \end{aligned} \tag{2.9}$$

Note that a lower case, to indicate continuous distributions, is unnecessary for the mutual information because it is equivalent for two continuous variables and their quantised approximations, as we show here.

$$\begin{aligned} I(X,Y) &= h(X) + h(Y) - h(X,Y) \\ &= H(X) + \log_2(\Delta_x) + H(Y) + \log_2(\Delta_y) - H(X,Y) - \log_2(\Delta_x \Delta_y) \\ &= H(X) + H(Y) - H(X,Y) \end{aligned} \tag{2.10}$$

A noteworthy property of mutual information is that it is invariant under transforms of the marginal variables. This property is true provided the original values are retrievable from the reparametrisation. Consequently, the mathematical complexity of the correlation between two variables has no bearing on the mutual information value; rather, it measures the strength of the association. Consider the mutual information between the variables X and Y is 1 bit. Now suppose we transformed $X' = e^X$ and $Y' = Y^3$ the mutual information is still 1 bit despite the increased complexity of the relationship [9]. These properties of the mutual information make it a useful quantity for measuring and comparing linear and nonlinear variable correlations necessary in data analysis.

References

[1] W. J. McGill, "Multivariate information transmission," *IRE Professional Group on Information Theory*, vol. 4, no. 4, pp. 93–111, 1954, ISSN: 21682704. DOI: 10.1109/TIT.1954.1057469.

[2] S. D. Chen, "A new image quality measure for assessment of histogram equalization-based contrast enhancement techniques," *Digital Signal Processing: A Review Journal*, vol. 22, no. 4, pp. 640–647, 2012, ISSN: 10512004. DOI: 10.1016/j.dsp.2012.04.002. [Online]. Available: http://dx.doi.org/10.1016/j.dsp.2012.04.002.

[3] P. J. Moreno, P. P. Ho, and N. Vasconcelos, "A Kullback-Leibler divergence based kernel for SVM classification in multimedia applications," *Advances in Neural Information Processing Systems*, vol. 16, 2004, ISSN: 10495258.

[4] H. Dilpazir, H. Mahmood, Z. Muhammad, and H. Malik, "Face recognition: A multivariate mutual information based approach," *Proceedings - 2015 IEEE 2nd International Conference on Cybernetics, CYBCONF 2015*, pp. 467–471, 2015. DOI: 10.1109/CYBConf.2015.7175979.

[5] K. Hlaváčková-Schindler, M. Paluš, M. Vejmelka, and J. Bhattacharya, "Causality detection based on information-theoretic approaches in time series analysis," *Physics Reports*, vol. 441, no. 1, pp. 1–46, 2007, ISSN: 03701573. DOI: 10.1016/j.physrep.2006.12.004.

[6] W. Wang, C. Liu, and D. Zhao, "How Much Data is Enough? A Statistical Approach with Case Study on Longitudinal Driving Behavior," *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 2, pp. 1–1, 2017, ISSN: 2379-8858. DOI: 10.1109/tiv.2017.2720459. arXiv: 1706.07637.

[7] C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, 1948, ISSN: 15387305. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[8] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. 2005, ISBN: 9780471241959. DOI: 10.1002/047174882X.

[9] J. V. Stone, "Information theory: A tutorial introduction," 2015.

Chapter 3

Entropy Estimation

Since Shannon's ground-breaking work in information theory, it has repeatedly demonstrated its practical applications in several statistical procedures. Among the information measures, mutual information is the most commonly used in data analysis to evaluate the statistical dependence between two variables. Unlike measures of linear dependence, or other rank correlation coefficients, the mutual information is of particular interest due to its sensitivity to dependencies that do not manifest themselves in the covariance [1]. See [2] for an overview of applications.

A principal problem in any of these applications is estimating the information content from observed samples. In most cases, the underlying probability distribution is unknown. Instead, the data is available in the form of N independent and identically distributed (i.i.d) observations that approximately represent the domain. It has become increasingly appreciated that a robust estimate of entropy from empirical data is fraught with difficulties and despite its widespread use, entropy estimation is still an active research topic [3]. The most straightforward and intuitive, non-parametric entropy estimation method is the plug-in approach. The estimated probabilities are directly substituted into the Shannon entropy formula using the relative frequencies. This is simple for discrete distributions. For continuous variables, however, the technique suffers from an unavoidable dependence on the quantisation, leading to serious bias problems [4].

There are several families of more complex differential entropy estimators that have improved upon the quantisation technique. One example is the kernel density estimator. Kernel density estimates offer advantages over the histogram in terms of improved convergence rate, an insensitivity to the origin choice, and the ability to specify sophisticated window shapes. However, in ref. [5] Kernel density estimates provided no additional insight to variable relationships compared to linear correlation coefficients, making the increased computation complexity unjustifiable. Kraskov supported this conclusion [6]. Alternatively, k-nearest-neighbour methods, originally proposed by Kozachenko and Leonenko [7] in 1987, have spiked in popularity in recent years. This resurgence is due to computational advances, making the approach more practical and efficient to implement.

In this section, we review entropy estimation using partitioning and k-nearest-neighbour techniques. These topics lay the groundwork for the proposed entropy estimator in chapter 4.

3.1 Partitioning Methods

For continuous random variables, the number of potential outcomes is infinite. Therefore, it is impossible to obtain probability estimates for each distinct outcome, especially given the limited amount of data typically attainable for real-world problems. However, partitioning the data to extract the relative frequencies reduces an infinite-dimensional problem into an achievable finite-dimensional problem. Thus, any entropy estimation via partitioning methods is unavoidably entwined with the problem of quantisation.

In quantisation, the true probability distribution, $p(x)$ is approximated using the relative frequencies of the occurrence of data samples. If $n_x(i)$ is the number of observations for which $X = x_i$ then one can approximate $p(x_i) \approx n_x(i)/N$ and similarly for $p(y_j)$. The joint entropy is then approximated using the product of the quantised states, $p(x_i, y_j) \approx n_{xy}(i, j)/N$.

A quantisation approach is the most straightforward method for estimating entropy, but it is not without difficulties. Here we will give a brief overview of fixed and adaptive partitioning methods. A more thorough discussion on partitioning techniques for entropy estimation can be found in [4].

3.1.1 Fixed partitioning

Fixed partitioning or "equal width binning" is the most widely used quantisation technique for classical histogram methods. The technique divides the data into q equally spaced discrete bins of width Δ. The relative frequencies are then directly substituted into 2.1. The technique is used in [8] to estimate the mutual information between two genes in RNA. From this analysis, the authors found significant mutual information values for non-correlated variables. This result is likely because a robust entropy estimate via fixed partitioning is complicated, as the accuracy of the estimate depends on the long-standing histogram question: How many bins are needed to construct a reliable depiction of the true probability distribution?

Many rules of thumb exist to determine the number of bins. For example, some use the rule that between 5 and 20 bins are typically sufficient [9]. Thus, constructing a histogram is subjective, as the eye fine-tunes the choice of bins. The user makes an instinctive trade-off between fewer bins to reduce noise or more bins to extract detail in the underlying probability distribution. This subjectivity surrounding the choice of q inputs bias into the density estimate, propagating through the mutual information. Instead, researchers have derived formulas in an attempt to objectively answer the question.

Scott's rule in [10] is perhaps the most well known and widely used fixed partitioning method. For a sample of N i.i.d events with a standard deviation σ, Scott's rule aims to minimise the

mean integrated squared error (MISE) via

$$\Delta = \left(\frac{6}{N \int p'(x)^2 dx} \right)^{\frac{1}{3}} \tag{3.1}$$

where $p'(x)$ is the first derivative. Unfortunately, Scott's rule requires prior knowledge of the underlying probability distribution and a non-zero first derivative, otherwise the formula fails. As this is often not viable, it is common to assume a normal distribution for a non-parametric version of Scott's rule: $\Delta = 3.5\sigma/N^{1/3}$ [10]. Popular fixed-width alternatives include Sturges' formula $q = 1 + \log_2(N)$, derived from the Binomial distribution [11] and Freedman and Diaconis rule where $\Delta = 2\ \text{IQR}(X)/N^{1/3}$ and IQR is the interquartile range [12]. In reality, there is no optimal number of bins that suits all problems, and many of these methods make strong, unwarranted assumptions about the underlying distributions [9].

More recently, the drive for improved entropy estimation techniques in partitioning has provided new methods to these classic rules. In [13], the authors derive a formula for the number of bins to minimise the mean squared error (MSE) on the Shannon entropy estimate, where $\text{MSE}(\hat{X}) = \text{Var}(\hat{X}) + \text{Bias}(\hat{X}, X)^2$. To bypass the parametric nature of this solution, the authors, like Scott, formalised the bias and variance for a normal distribution with an uncorrelated joint distribution. Despite the authors' similar assumptions, they empirically demonstrated that the estimator outperformed Scott's, Sturges' and the Freedman-Diaconis rules for estimating entropy using fixed partitioning methods.

Nevertheless, all non-parametric fixed partitioning methods require strong assumptions about the shape of the underlying probability distribution and blindly allocate bins with no real consideration for the data described. As a result, regions of high density are over smoothed, and the technique is poor at identifying sharp peaks [14].

Once quantised, the relative frequencies replace the discrete probabilities in 2.1. However, doing so tends to result in a systematic bias, independent of the probability distribution, which stems from the fact that the entropy is a nonlinear function of an unknown probability distribution. The bias manifests itself in the mutual information estimate as a systematic overestimation. The only remedy is to directly subtract the bias [4], [5].

There have been several attempts to quantify the systematic bias with varying degrees of rigour [15]–[17] with most arriving at comparable answers. The partitioning bias arises from two sources: the finite resolution and the finite sample size. Both [15] and [16] derive the same bias correction, with [16] arriving at the answer by modelling p_i using a Binomial distribution and a Taylor expansion around n_i. The analysis shows that finite data introduces an underestimate of $1/2N$ for each contribution to the entropy sum. However, independent of the quantisation method used, there is no universal rate at which the error goes to zero [4]. Nevertheless, due to its simplicity and low computational cost, fixed partitioning methods are still popular for

estimating the entropy of a random variable.

3.1.2 Adaptive partitioning

Adaptive partitioning or "data-dependent partitions" were developed to improve upon the traditional fixed-width methods. Instead of having equal bin widths, it is common in entropy estimation to partition the data based on statistical criteria. The simplest and most common adaptive partitioning method is "equiquantisation", which segments the marginal space into q bins, such that each bin contains approximately the same number of observations. Equiquantisation effectively applies a linear transform to each variable. As previously discussed in chapter 2 mutual information is invariant under linear transforms of the marginal space [18]–[20].

Unlike fixed partitioning, the relative frequencies do not preserve the pdf. Instead, equiquantisation forces the frequency distribution of the individual variables to be uniform, but not the joint distribution. As we are only interested in the variable interactions, not the individual variables, preserving the marginal distributions is unnecessary. In fact, insisting on preservation is considered disadvantageous. By forcing the marginal distributions to be uniform regions where $P(x)$ and $P(y)$ are dense, results in more closely spaced bins, and regions where the data points are sparse, have wider bins. Consequently, data points close to the support [1] contribute more equally to the entropy estimate. The effect is that it increases sensitivity to the distribution, amplifying data structures. The non-uniformity of the joint distribution then reveals the structural dependence between the two variables X and Y. A uniform $P(x, y)$ implies independence, as $P(x) = P(y) = 1/q$, and a non-uniform $P(x, y)$ implies a dependence. Maximising information in this way exhibits better statistical properties than fixed-width binning [18], [21].

Regardless of its advantages over fixed partitioning, the number of bins remains a crucial and complicated choice in adaptive partitioning, and there have been many attempts at formulating partitions. In [19] Paluš used a equiprobable box-counting method and proposed an upper limit on the number of bins of $q \leq \sqrt[d+1]{N}$ for d variables, which was reported to prevent heavily biased entropy estimates. Alternatively, Darbellay and Vajda in [22] iteratively segment the full parameter space of the d variables such that each bin is approximately uniform. The decision to partition a bin is based on a χ^2 statistical test to achieve conditional independence. Although $\sqrt{N}$-consistent and asymptotically unbiased, this approach severely underestimates the mutual information for some distributions attributed to the statistical test used. More recently, Wang *et al.* uses adaptive partitioning to estimate the Kullback-Leibler divergence. Wang *et al.* proves strong convergence and outperforms the Darbellay and Vajda approach through additional bias correction terms [20].

Independent of the partitioning method used, there will always be limitations to partitioning techniques due to a loss of information from the quantisation of the continuous variable. The

[1] The support of a random variable is the set of values which are not mapped to zero for a real-valued pdf.

amount of information lost depends on the quality and size of the initial sample, and the quantisation method.

A small data sample restricts the accuracy of any information estimate, as it is insufficient to reliably estimate the density. What constitutes a sufficient sample size, however, is complicated and depends on the complexity of the distribution being estimated. Even for relatively simple distributions, statistical fluctuations in small samples introduce both statistical and systematic biases to the density estimate that propagate through to the mutual information [1], [23].

Unfortunately, there is no one q or Δ that reliably constructs the pdf for all distributions and sample sizes.

3.2 k-Nearest-Neighbor Entropy Estimators

For continuous variables the mutual information estimate is susceptible to biases and statistical fluctuations when partitioning methods are applied. The more promising k-nearest-neighbour methods are a family of well-known techniques that have empirically proven effective at density estimation. They are highly popular, as the approach is non-parametric, with few assumptions about the pdf and a number of other reasons that make them a practical and efficient choice [24]. The success of these techniques in density estimation naturally lends itself to applications in entropy estimation, and the topic has been highly researched [25]–[31].

Variations of k-nearest-neighbour estimators have been derived frequently in the literature using geometrical arguments [6], [7], [27], [32], [33]. These methods utilise the Euclidean geometry of the sample space around a test point. Let $\lambda_{k,i}^d$ be the d-dimensional L^2 distance from x_i^d to its k^{th} nearest neighbour x_j^d, where k is a small, fixed-positive integer. Each nearest-neighbour distance provides a local view of the density distribution around a single, test point in the data sample. The concept of nearest-neighbour distances is illustrated in figure 3.1. By considering the density of the d-dimensional sphere a relationship between the distribution and the number of data points contained within the sphere is apparent.

The exemplar k-nearest-neighbour estimator was initially developed by Kozachenko and Leonenko in 1987 [7]. Wherein, Kozachenko and Leonenko proved the mean square consistency under mild conditions for $k = 1$ and general d. Since then, the Kozachenko-Leonenko (KL) estimator has been studied with great interest in the literature with many variations presented. In [34] Tsybakov and Van der Meulen considered a truncated version of the KL estimator and showed $\sqrt{N}$-rate of convergence when $d = 1$ for a class of one-dimensional densities with unbounded support and exponentially decreasing tails. In [27] Singh *et al.* generalised to $k \geq 1$. The revision is almost equivalent to the original when $k = 1$, with the only difference being the use of $\log(N)$ in place of $\log(N - 1)$. The proposed estimator, given in bits, is derived in [27]

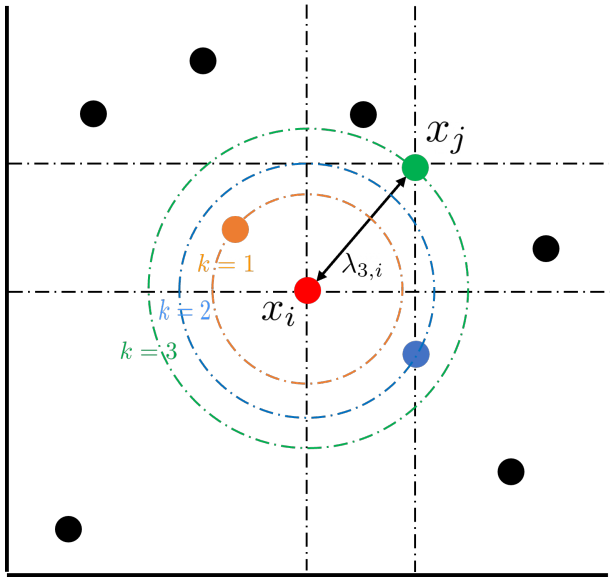

Figure 3.1: Schematic of k nearest-neighbour distance method shown for a two-dimensional sample. Centered around the test point shown in red, the first three nearest-neighbours are indicated by concentric circles. For $k = 3$ the two-dimensional k-nearest neighbour distance $\lambda_{3,i}$ is indicated by the black arrow.

and defined as

$$\hat{h}_k(X) = \frac{d}{N}\sum_{i=1}^{N}\log_2(\lambda_{k,i}^d) + \log_2(S_d) + \log_2(N) + \frac{\gamma - L_{k-1}}{\ln(2)} \tag{3.2}$$

$$L_{k-1} = \sum_{p=1}^{k-1}\frac{1}{p} \tag{3.3}$$

where $\gamma(= 0.577216...)$ is the Euler-Mascheroni constant, L_{k-1} is the harmonic number of $k-1$ and S_d is the volume of a unit-sphere $S_d = \pi^{d/2}/\Gamma\left(\frac{d}{2}+1\right)$ with $\Gamma(.)$ as the Gamma function. [27] uses heuristic arguments to show the estimator is asymptotically unbiased and consistent for fixed k and general d. Very recently Devroye and Gyorfi showed that for bounded X, without any smoothness conditions the KL estimator is strongly consistent *iff* $\mathbb{E}\{\log[||X|| + 1)\} < \infty$ [31].

It has been established in [27], [29], [30], [33] that the variance of the KL estimator decays at a rate of $\mathcal{O}(1/N)$ under various assumptions. [27] established for $d = 1$ and general fixed k that the variance is a decreasing function of k:

$$Var[\hat{h}_k(X)] = \frac{1}{N}\left(Var[\log(f(x))] + \Psi_1(k)\right) \tag{3.4}$$

where Ψ_1 is the trigamma function. [28] extends this to general d and k showing that the variance of the KL estimator is bounded by $O\left((Nk)^{-1}\right)$.

Therefore, under the criterion of the mean squared error, larger values of k yield a smaller error

for all distributions. For practical purposes, however, larger values of k are computationally expensive and can degrade the entropy estimate if too large. Thus, the choice of k is an important issue, and the bias-variance trade-off must be considered. There are two approaches to the choice of k, one where k grows with the sample size, and the other where k is a small constant [26], [35]. Most literature favours the latter and benefits from the smaller computational demands and optimal convergence rates [29]. Based on Monte Carlo simulations, Singh *et al.* suggests a blanket rule of $k = 4$, stating that the gain from higher values is insubstantial. However, these results are only shown for small samples sizes, < 50, limiting the applicability of this proposed rule-of-thumb.

3.2.1 The Bias of k-Nearest-Neighbour Methods

The problem with k-nearest-neighbour methods is an extension of the overarching problem of the nonlinear dependence of entropy on the probability distribution. Meaning the KL estimator exhibits substantial biases when the density is small [36], [37]. Establishing the convergence rate of the bias is challenging. [38] showed that the bias vanishes asymptotically, however they neglected the bias incurring at the boundaries. For data points located near the boundaries of the variable support the k-nearest-neighbour radius (neighbourhood) for a data point exceeds the support. The result is a truncated sphere and an overestimation bias on the entropy. The larger the value of k, the more data points have truncated neighbourhoods. Resulting in a bias-variance trade-off for increasing k. As $k/N \to 0$, the bias decays to zero in the interior of the density. At the support, however, the bias is much larger $\mathcal{O}(1)$, compared to $\mathcal{O}((k/N)^{2/d})$, and does not converge to zero. For higher dimensions, the boundary effect is more significant [39], [40].

In the literature, there are many correction methods [39], [40]. For example, [40] replaces the density estimate of a boundary point with that from a nearby point without a truncated neighbourhood. On the other hand, [39] base their correction term on the logarithm of the proportion of volume inside the support compared to the whole neighbourhood volume. [39] shows this is more effective than the approach in [40].

For one-dimensional random variables, however, the estimator can be considered unbiased even for relatively small sample sizes. For higher dimensions, the KL estimator converges relatively slowly to the true entropy, resulting in a significant bias even for large sample sizes [39]. Thus, one would need an exponentially larger sample for an equivalent bias at higher dimensions. This bias is not equal for all distributions, however, and the KL estimator is known to perform well on multivariate normal distributions [40].

Despite these difficulties, in general k-nearest-neighbour methods exhibit faster convergence and smaller errors compared to partitioning methods, making them the favourable approach [41].

3.3 k-nearest-neighbour Estimators for Mutual Information

Calculating entropy in data analysis is a stepping stone in the calculation of more interesting information-theoretic terms. The mutual information is the most commonly used measure in information theory to determine the dependence between two random variables.

In [6] Kraskov *et. al.* argued that the biases on each of the three KL entropy estimates, as discussed in section 3.2.1, do not cancel out due to the different distance scales in the joint and marginal spaces for different distributions. Kraskov consequently developed a k-nearest-neighbour method for directly calculating the mutual information based of the work of Kozachenko-Leonenko. It uses the distance to the joint k^{th} nearest-neighbour and counts the number of instances, n_x and n_y, bounded by this distance in the marginal spaces separately. This ensures that the distance scales cancel out. The mutual information estimator, referred to as the KSG estimator, is thus

$$\hat{I}(X,Y) = \Psi(k) + \Psi(N) - \frac{1}{N}\sum_{i=1}^{N}\left[\Psi(n_x+1) + \Psi(n_y+1)\right] \tag{3.5}$$

where $\Psi(...)$ is the digamma function. Kraskov empirically showed that this method outperformed the KL estimator when estimating mutual information, which we will refer to as the 3H-KL estimator.

Although the KSG estimator is prevalent throughout the literature, the theoretical properties remain unknown. Even the original reference [6] states there are distributions for which the estimator does not converge. More recently, [35] establishes consistency and identifies an upper bound on the rate of convergence of the bias and [36] empirically demonstrates that the variance $\mathcal{O}(1/N)$.

Despite the practical performance of the KSG estimator, it similarly, suffers from the same problems as the KL estimator due to its foundations in k-nearest-neighbour techniques. Just as with the KL estimator, the KSG estimator suffers from the boundary effect, requiring exponentially larger samples for the equivalent bias. [3] showed k-nearest-neighbour estimators in general tend to systematically underestimate the mutual information for strongly dependent variables and small sample sizes.

3.4 Discussion

The question of variable dependence is a topic that has been studied for decades. However, big data has presented us with a growing problem. Data sets with many variables and even more potential relationships are now widespread. Examining each relationship is no longer practical.

Instead, we need an objective and reliable way to identify significant correlations. Mutual information provides an obvious basis for comparing variable interactions. While non-parametric mutual information estimators have been extensively studied, researchers have typically avoided mixed cases due to the difficulties surrounding them. Thus, the field is largely unexplored.

The most common attempts to overcome mixed data has been to quantise continuous variables and apply discrete methods, discussed in section 3.1 with adaptive partitioning approaches proving most effective. However, partitioning methods are vulnerable to bias and statistical fluctuations, making nearest neighbour methods more desirable. Alternatively, one can add a small independent noise distribution to a discrete variable to artificially add distance between repeating points and apply a continuous estimator. However, the literature suggests that adding noise causes the estimate to degrade and is unstable with respect to the noise added, but this is not the case if the noise distribution is correctly selected [37]. In chapter 4 we address these issues.

References

[1] F. Emmert-Streib and M. Dehmer, *Information theory and statistical learning*. 2009, ISBN: 9780387848150. DOI: 10.1007/978-0-387-84816-7.

[2] J. Beirlant, E. Dudewicz, L. Györfi, and I. Dénes, "Nonparametric entropy estimation. An overview," *International Journal of Mathematical and Statistical Sciences*, vol. 6, no. 1, pp. 17–39, 1997.

[3] S. Gao, G. Ver Steeg, and A. Galstyan, "Efficient estimation of mutual information for strongly dependent variables," in *Artificial intelligence and statistics*, PMLR, 2015, pp. 277–286.

[4] L. Paninski, "Estimation of entropy and mutual information," *Neural Computation*, vol. 15, no. 6, pp. 1191–1253, 2003, ISSN: 08997667. DOI: 10.1162/089976603321780272.

[5] R. Steuer, J. Kurths, C. Daub, J. Weise, and J. Selbig, "The mutual information: Detecting and evaluating dependencies between variables," *Bioinformatics (Oxford, England)*, vol. 18 Suppl 2, S231–40, Feb. 2002. DOI: 10.1093/bioinformatics/18.suppl_2.S231.

[6] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, vol. 69, no. 6, p. 16, 2004, ISSN: 1063651X. DOI: 10.1103/PhysRevE.69.066138. arXiv: 0305641 [cond-mat].

[7] L. Kozachenko and N. Leononko, "Sample Estimate of the Entropy of a Random Vector," *Probl. Peredachi Inf.*, vol. 23, no. 2, pp. 9–16, 1987.

[8] A. J. Butte and I. S. Kohane, "Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements.," *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, vol. 426, pp. 418–429, 2000, ISSN: 2335-6928.

[9] P. Sulewski, "Equal-bin-width histogram versus equal-bin-count histogram," *Journal of Applied Statistics*, vol. 48, no. 12, pp. 2092–2111, 2021, ISSN: 13600532. DOI: 10.1080/02664763.2020.1784853. [Online]. Available: https://doi.org/10.1080/02664763.2020.1784853.

[10] D. W. Scott, "On optimal and data-based histograms," *Biometrika*, 1979, ISSN: 00063444. DOI: 10.1093/biomet/66.3.605.

[11] H. A. Sturges, *The Choice of a Class Interval*, 1926. DOI: 10.1080/01621459.1926.10502161.

[12] D. Freedman and P. Diaconis, "On the histogram as a density estimator:L2 theory," *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 1981, ISSN: 00443719. DOI: 10.1007/BF01025868.

[13] A. Hacine-Gharbi, M. Deriche, P. Ravier, R. Harba, and T. Mohamadi, "A new histogram-based estimation technique of entropy and mutual information using mean squared error minimization," *Computers and Electrical Engineering*, vol. 39, no. 3, pp. 918–933, 2013, ISSN: 00457906. DOI: 10.1016/j.compeleceng.2013.02.010. [Online]. Available: http://dx.doi.org/10.1016/j.compeleceng.2013.02.010.

[14] L. Denby and C. Mallows, "Variations on the histogram," *Journal of Computational and Graphical Statistics*, vol. 18, no. 1, pp. 21–31, 2009, ISSN: 10618600. DOI: 10.1198/jcgs.2009.0002. [Online]. Available: https://doi.org/10.1198/jcgs.2009.0002.

[15] G. Miller, "Note on the bias of information estimates," *Information theory in psychology*, pp. 95–100, 1955.

[16] H. Herzel, A. O. Schmitt, and W. Ebeling, "Finite sample effects in sequence analysis," *Chaos, Solitons and Fractals*, vol. 4, no. 1, pp. 97–113, 1994, ISSN: 09600779. DOI: 10.1016/0960-0779(94)90020-5.

[17] B. Efron and C. Stein, "The Jackknife Estimate of Variance," *The Annals of Statistics*, 1981, ISSN: 0090-5364. DOI: 10.1214/aos/1176345462.

[18] R. E. Valdes-Perez and R. C. Conant, "Information Loss Due to Data Quantization in Reconstructability Analysis," *International Journal of General Systems*, vol. 9, no. 4, pp. 235–247, 1983, ISSN: 15635104. DOI: 10.1080/03081078308960824.

[19] M. Paluš, "Testing for nonlinearity using redundancies: quantitative and qualitative aspects," *Physica D: Nonlinear Phenomena*, vol. 80, no. 1-2, pp. 186–205, 1995, ISSN: 01672789. DOI: 10.1016/0167-2789(95)90079-9.

[20] Q. Wang, S. R. Kulkarni, and S. Verdú, "Divergence estimation of continuous distributions based on data-dependent partitions," *IEEE Transactions on Information Theory*, vol. 51, no. 9, pp. 3064–3074, 2005, ISSN: 00189448. DOI: 10.1109/TIT.2005.853314.

[21] M. Vejmelka, "Quantifying interactions between complex oscillatory systems: a topic in time series analysis," Ph.D. dissertation, 2008.

[22] G. A. Darbellay and I. Vajda, "Estimation of the information by an adaptive partitioning of the observation space," *IEEE Transactions on Information Theory*, vol. 45, no. 4, pp. 1315–1321, 1999, ISSN: 00189448. DOI: 10.1109/18.761290.

[23] T. Schürmann, "Bias analysis in entropy estimation," *Journal of Physics A: Mathematical and General*, vol. 37, no. 27, 2004, ISSN: 03054470. DOI: 10.1088/0305-4470/37/27/L02. arXiv: 0403192 [cond-mat].

[24] G. H. Chen and D. Shah, *Explaining the success of nearest neighbor methods in prediction*, 5-6. 2018, vol. 10, pp. 337–588, ISBN: 9781680833683. DOI: 10.1561/2200000064.

[25] T. B. Berrett, "Modern k-Nearest Neighbour Methods in Entropy Estimation, Independence Testing and Classification," no. July, 2017. [Online]. Available: http://www.statslab.cam.ac.uk/$%5Csim$tbb26/thesis.pdf.

[26] F. Pérez-Cruz, "Estimation of information theoretic measures for continuous random variables," *Advances in Neural Information Processing Systems 21 - Proceedings of the 2008 Conference*, pp. 1257–1264, 2009.

[27] H. Singh, N. Misra, V. Hnizdo, A. Fedorowicz, and E. Demchuk, "Nearest neighbor estimates of entropy," *American Journal of Mathematical and Management Sciences*, vol. 23, no. 3-4, pp. 301–321, 2003, ISSN: 01966324. DOI: 10.1080/01966324.2003.10737616.

[28] S. Singh and B. Póczos, "Analysis of k-Nearest Neighbor Distances with Application to Entropy Estimation," 2016. arXiv: 1603.08578. [Online]. Available: http://arxiv.org/abs/1603.08578.

[29] ——, "Finite-sample analysis of fixed-K Nearest neighbor density functional estimators," *Advances in Neural Information Processing Systems*, no. Nips, pp. 1225–1233, 2016, ISSN: 10495258. arXiv: 1606.01554.

[30] S. Delattre and N. Fournier, "On the kozachenko-leonenko entropy estimator," no. 1, pp. 1–30, 2016. arXiv: arXiv:1602.07440v1.

[31] L. Devroye and L. Györfi, "On the consistency of the Kozachenko-Leonenko entropy estimate," *arXiv e-prints*, arXiv:2102.12952, 2021. arXiv: 2102.12952 [math.ST].

[32] J. D. Victor, "Binless strategies for estimation of information from neural data," *Physical Review E*, no. November, 2002. DOI: 10.1103/PhysRevE.66.051903.

[33] S. Li, R. M. Mnatsakanov, and M. E. Andrew, "k-Nearest Neighbor Based Consistent Entropy Estimation for Hyperspherical Distributions," *entropy*, vol. 13, no. 1, pp. 650–667, 2011. DOI: 10.3390/e13030650.

[34] A. B. Tsybakov and E. C. van der Meulen, "Root-n Consistent Estimators of Entropy for Densities with Unbounded Support," *Scandinavian Journal of Statistics*, vol. 23, no. 1, pp. 75–83, 1996.

[35] W. Gao, S. Oh, and P. Viswanath, "Demystifying Fixed κ-Nearest Neighbor Information Estimators," *IEEE Transactions on Information Theory*, vol. 64, no. 8, pp. 5629–5661, 2018, ISSN: 00189448. DOI: 10.1109/TIT.2018.2807481. arXiv: 1604.03006.

[36] C. M. Holmes and I. Nemenman, "Estimation of mutual information for real-valued data with error bars and controlled bias," *arXiv*, 2019. DOI: 10.1101/589929.

[37] W. Gao, "Information theory meets big data: Theory, Algorithms and Applications to Deep Learning," Ph.D. dissertation, University of Illinois, 2019.

[38] G. Biau and L. Devroye, *Springer Series in the Data Sciences*. 2015, vol. 1, ISBN: 978-3-319-25386-2. DOI: 10.1007/978-3-319-25388-6.

[39] A. Charzýnska and A. Gambin, "Improvement of the k-NN entropy estimator with applications in systems biology," *Entropy*, vol. 18, no. 1, pp. 1–19, 2016, ISSN: 10994300. DOI: 10.3390/e18010013.

[40] K. Sricharan, R. Raich, and A. O. Hero III, "Boundary Compensated k-NN Graphs," *2010 IEEE International Workshop on Machine Learning for Signal Processing*, pp. 277–282, 2010. DOI: 10.1109/MLSP.2010.5589237.

[41] Q. Wang, S. R. Kulkarni, and S. Verdú, "A nearest-neighbor approach to estimating divergence between continuous random vectors," *IEEE International Symposium on Information Theory - Proceedings*, no. 5, pp. 242–246, 2006, ISSN: 21578101. DOI: 10.1109/ISIT.2006.261842.

Chapter 4

Noisy Resampling Entropy Estimation Method

The difficulties associated with estimating information measures from finite empirical samples has motivated researchers to explore more versatile methods. Estimating parameters from empirical samples is a problem faced in a number of scientific fields, and as a result techniques such as resampling and aggregated estimators, which average the result, have been developed to improve upon empirical estimates. For our entropy estimator, we take inspiration from ensemble methods. Ensemble methods determine an empirical parameter through combining techniques and repeating applications to form an aggregate estimator. Such techniques have been shown to improve upon a single estimate. Randomised ensemble methods are premised on a base estimator, then an ensemble of results is formed by repeating the base estimator on independent and random perturbations of the data sample. A common randomised ensemble example is bagging (**b**ootstrap **agg**regat**ing**). Bagging uses the bootstrap resampling method and averages over the repeats. This has the added advantage of estimating sampling variability by measuring the standard error on a single estimate [1]. Resampling is a valuable technique used in parameter estimation to improve or determine the statistics of an estimator or model. Resampling within the context of information estimates, however, can be dangerous, as it is a nonlinear quantity.

Due to the inherent difficulties associated with estimating entropy and the problem of mixed systems we propose a new ensemble method to improve on the reliability and accuracy of an information estimate. Similar to bagging, the proposed noisy resampling entropy estimation method is an ensemble technique to generate multiple samples from a single data set in order to find an average. The repeat samples are generated through a two-stage process of quantising (binning) and randomising (adding noise) to produce continuous approximations of the original sample, see figure 4.1 for a flow chart of the methodology

This method combines the two approaches typically implemented for mixed systems. Being able to quantise the continuous variables reliably means that the same entropy estimation technique can be applied to both continuous and discrete variables, which is crucial for consistency when estimating mutual information. The KL estimator is then applied to each randomised sample.

Each generated sample consists of N instances and is effectively i.i.d. This method is shown to improve on a single 3H-KL estimate of mutual information and outperforms or is equivalent to the KSG estimator for all tested mutual information estimates. Having reliable methods for mixed systems significantly increases the applicability of mutual information in machine learning.

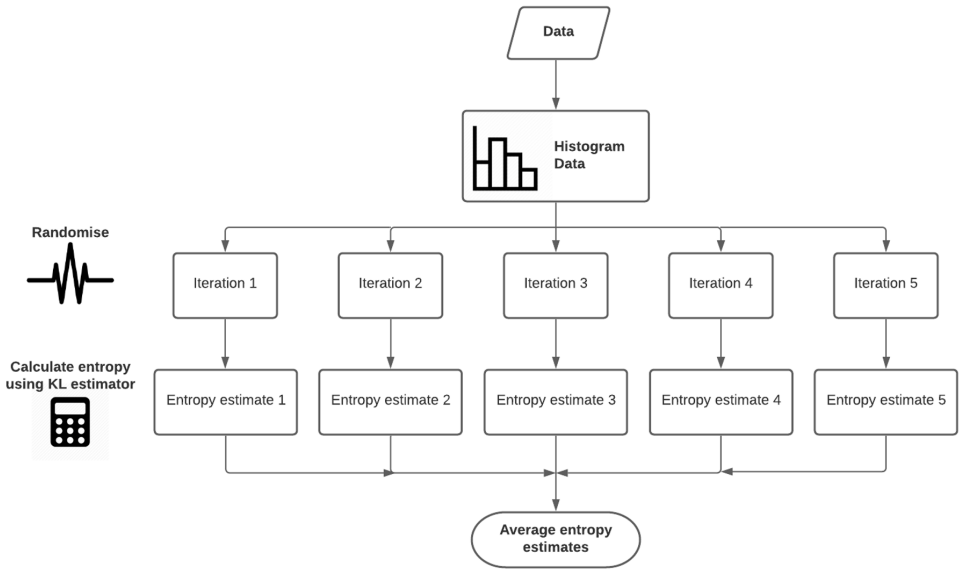

Figure 4.1: A flow chart depicting the steps involved in the new two-stage noisy resampling entropy estimation method. The method applies both quantisation (histogramming) and randomisation (adding noise) procedures. Note that the data is already quantised for discrete random variables, removing the need for this step. The example depicted generates 5 randomised "iterations" or repeats of the data sample. We estimate the entropy using the KL entropy estimator for each iteration, and the result is averaged.

4.1 Quantisation

4.1.1 What is the Entropy of a Histogram?

Optimised binning has been an active area of research for many decades with various applications. It is difficult, however, for any method that uses fixed-partitioning to return a reliable entropy estimate due to statistical fluctuations for small bins and a bias if the bins are too large. There are many methods within the literature that aim to minimise statistical criteria to justify the choice in binning. Nevertheless, the practice seems problematic, as Shannon entropy is sensitive to the bin size [2]. This provokes the question: What is the Shannon entropy of a histogram?

To answer this, we first consider the limits on the Shannon entropy of a finite sample. Theoretically, the Shannon entropy of a continuous pdf is infinite. Although, we noted in section 2.2 that for a finite sample of a continuous distribution, the Shannon entropy is limited by the

number of unique outcomes. While the precision of the measurements can govern the number of unique outcomes, it cannot exceed the sample size: $H(X) \leq \log_2(N)$ bits. This Shannon entropy would be an extreme case of over-binning, and all knowledge of the underlying probability distribution would be lost. Nevertheless, it sets an upper limit on the Shannon entropy of an empirical sample of a continuous distribution. From this, we know that $H(X)$ of any continuous sample is bounded by $0 \leq H(X) \leq \log_2(N)$, representing the extremes of under- and over-binning, respectively. Therefore, Shannon entropy for any finite sample, independent of the probability distribution, converges to $\log_2(N)$ bits as the number of bins, q increases and the bin width, Δ decreases. However, as differential entropy is well-defined for a pdf, there is a value of $H(X)$ and a corresponding bin width according to equation 2.6, which most accurately represents the true pdf.

We note the importance of a distribution-independent limit on the Shannon entropy of a finite sample. This universal limit suggests a common $\log_2(N)$ dependence for all probability distributions. To determine the form of this dependence, we consider the bin width that minimises the MISE, using the parametric form of Scott's formula for different pdfs with known differential entropy. From this bin width and the known differential entropy, we can determine the theoretical form of the Shannon entropy for a continuous distribution. This calculation has been done for several distributions in table 4.1, where, for simplicity, the results are given in nats.

Distribution	$h(x)$ (nats)	Δ	$H(X)$ (nats)
Normal	$\frac{1}{2}\ln[2\pi e\sigma^2]$	$2 \times 3^{1/3}\pi^{1/6}\sigma N^{-1/3}$	$\frac{1}{3}\ln\left[N\frac{\pi e^{3/2}}{3\times 2^{3/2}}\right] = \frac{1}{3}\ln[1.66N]$
Exponential	$1 - \ln[\lambda]$	$12^{1/3}\lambda^{-1}N^{-1/3}$	$\frac{1}{3}\ln\left[N\frac{e^3}{12}\right] = \frac{1}{3}\ln[1.67N]$
Maxwell-Boltzmann	$\frac{1}{2}\ln\left[\frac{\pi}{\beta}\right] + \gamma - \frac{1}{2}$	$\left(\frac{6\times 2^{3/2}\sqrt{\pi}}{7\beta^{3/2}}\right)^{1/3} N^{-1/3}$	$\frac{1}{3}\ln\left[N\frac{\pi e^{3(\gamma-0.5)}}{2.42}\right] = \frac{1}{3}\ln[1.64N]$
Gamma ($\alpha = 2$, $\beta = 1$)	$1 + \gamma$	$24^{1/3}N^{-1/3}$	$\frac{1}{3}\ln\left[N\frac{e^{3(\gamma+1)}}{24}\right] = \frac{1}{3}\ln[4.73N]$

Table 4.1: A table to demonstrate the theoretical form of Shannon entropy according to Scott's formula for minimum MISE for a selection of continuous pdfs. The bin widths were determined using the parametric form of Scott's formula and the Shannon entropy calculated via $H(X) = h(x) - \ln(\Delta)$. All results are given in nats for conciseness.

Table 4.1 demonstrates that the theoretical Shannon entropy for a minimum MISE histogram has a definite form. This is true for all distributions tested. In fact, for any of the common fixed-width binning rules, the resulting histogram has a Shannon entropy of the form: $H = \frac{1}{M}\log_2(\beta N)$ bits, where M and β are constants. Note that for $H \leq \log_2(N)$ to be true $M \geq 1$ and for $N = 1$, we would expect the Shannon entropy to go to zero. Thus, we set $\beta = 1$, such that

$$H = \frac{1}{M}\log_2(N) \tag{4.1}$$

However, setting $\beta = 1$ removes any distribution dependence in the Shannon entropy for continuous random variables. To verify that the Shannon entropy of a continuous variable is indeed independent of the distribution, we consider the MSE of the differential entropy estimate for dif-

ferent quantisations. To do this, we simulated samples for several one-dimensional probability distributions with known differential entropy. The distributions used are described in table 4.2 and were chosen to encompass standard comparisons, as well as various characteristics. For each simulated sample, we quantised the data points into q bins. We varied q between $2 \leq q \leq N/5$, and extracted the bin width and Shannon entropy via $\Delta = Range/q$ and the relative frequencies, respectively. For each q and N value the process was repeated 500 times for i.i.d trials. The MSE was thus determined using the known differential entropy and $h = H + \log_2(\Delta)$. To confirm the Shannon entropy of a finite sample is distribution independent, we examined the MSE of the differential entropy as a function of Shannon entropy for different sample sizes and distributions. If the Shannon entropy of a continuous distribution is independent of the probability distribution it describes, then the minimum MSE for each distribution will coincide. The results from this investigation are shown on the right of figure 4.2. For comparison, we also illustrate the behaviour of the MSE of the differential entropy for the same data as a function of bin width. This is shown on the left of figure 4.2.

Distribution	Mean	Std dev	Support	Skewness	Kurtosis	h (bits)
Normal, $\mathcal{N}(0,1)$	0.00	1.00	$(-\infty, \infty)$	0.00	3.00	2.05
Uniform, $\mathcal{U}[0,1)$	0.50	0.30	$[0,1)$	0.00	-1.20	0.00
Exponential, Exp(1)	1.00	1.00	$[0,\infty)$	2.00	6.00	1.45
Log Normal, Log $\mathcal{N}(0,1)$	1.65	2.16	$(0,\infty)$	6.18	111	2.05
Gamma, $\Gamma(2,1)$	2.00	1.41	$(0,\infty)$	1.41	3.0	2.28

Table 4.2: A table of attributes for several one-dimensional pdfs that we use throughout our simulated experiments. We include the attributes; skewness - a measure of the asymmetry of a distribution - and kurtosis - a measure of the tail, where high kurtosis indicates a substantial deviation from the mean.

The behaviour of the MSE of differential entropy is comparable for all distributions. We initially observe an abrupt decrease in the MSE as Δ increases. This is due to Poisson fluctuations. Poisson statistics apply to discrete and independent processes and is a fundamental form of noise in a histogram that arises due to sampling randomness. For high average counts the Poisson fluctuations tends to a Gaussian, however, for low counts the Poisson distribution is asymmetric. For example, if one expected 1 event per bin, then 36.8% of the time one would observe zero events, 36.8% of the time one would observe 1 event and 18.4% of the time one would observe 2 events. The zero-probability bins, due to the Poisson fluctuations, subsequently cause the Shannon entropy to be underestimated. As Δ increases further, a bias arises due to the loss of fine-scale structure in the distributions. For the uniform distribution, however, this is not the case. Instead, the MSE continues to decrease as $q \to 1$, as a uniform distribution cannot be under-binned.

When considering the MSE of the differential entropy as a function of the bin width, as known from previous work in fixed partitioning, there is no one Δ value suitable for all distributions. In contrast, for Shannon entropy, the behaviour of each distribution is more coherent, highlighting the drawbacks of considering quantisation in terms of bin width.

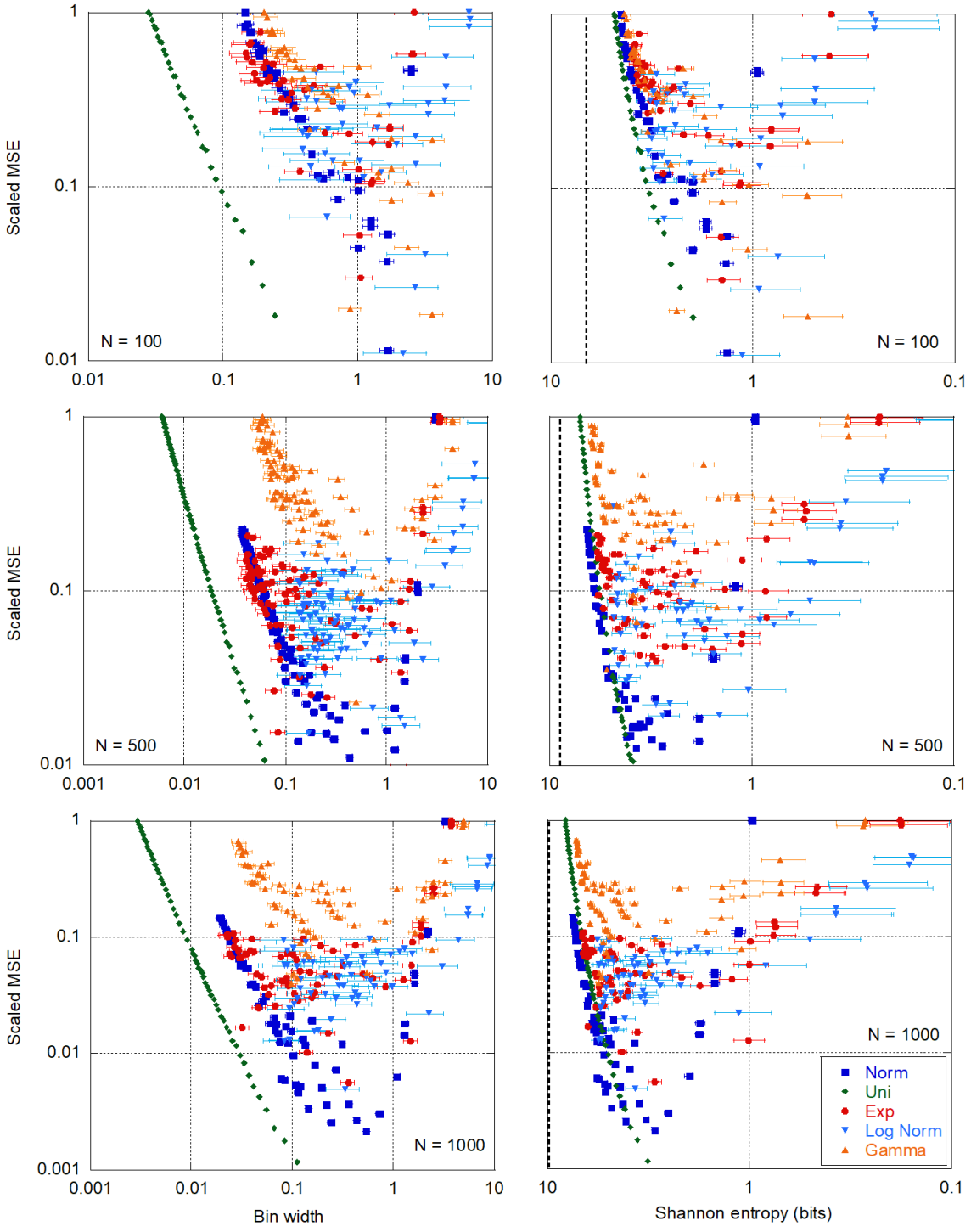

Figure 4.2: Graphs comparing the behaviour of the MSE of the differential entropy as a function of bin width (left) and Shannon entropy (right) for one-dimensional pdfs. The MSE was calculated from the differential entropy of 500 i.i.d trials, via $h = H + \log(\Delta)$. The MSE on the y-axis has been scaled with the minimum value calculated scaled to zero and the maximum value set to one to allow comparison of the MSE curves. The error bars on the x-axis represent ± 1 standard deviation of the bin width or Shannon entropy obtained from the independent trials. Note that the x-axis for the Shannon entropy plots are reversed, such that the maximum Shannon entropy ($\log(N)$), indicated by the black dashed line, is on the left-hand side of the Shannon entropy plots.

Firstly, for Shannon entropy, all distributions tend to a common maximum value, indicated by the black dashed line on the left-hand side of the Shannon entropy plots, illustrating $\log_2(N)$. For bin widths, however, the maximum bin width (i.e. when there is only one bin) depends on the range of the finite sample, which in turn depends on the distribution—leading to misaligned curves. This disjointedness is most evident for the uniform distribution, which is significantly translated along the x-axis for bin width compared to the other distributions. However, in terms of the Shannon entropy, the uniform distribution instead signifies an upper bound, corresponding to the uniform maximum entropy distribution for bounded samples.

Secondly, the different ranges for each of the distributions inevitably results in varied optimal bin widths in order to capture the features of the variable. However, when the MSE is considered as a function of Shannon entropy the minimum is observed as independent of the distribution, instead only depending on the sample size. This result is phenomenal and has repercussions in the area of quantisation and all surrounding topics that utilise it. This analysis reinforces our distribution-independent ansatz: $H = \log(N)/M$ and confirms the result obtained from table 4.1.

The fact that the optimal Shannon entropy of a finite sample is distribution-independent has immediate implications. One implication is that if a continuous sample was quantised via current fixed-partitioning methods, using the relative frequencies to estimate the Shannon entropy would be nonsensical. This is because the sample size predetermines the optimal Shannon entropy for a continuous sample.

Furthermore, the current treatment of random variables causes substantial biases in the Shannon entropy estimates. Standardising the Shannon entropy of each random variable minimises the bias and statistical fluctuations in the mutual information estimate.

4.1.2 Value of M

Now that we know the optimal Shannon entropy of a continuous finite sample, we can turn the idea of fixed-partitioning entropy estimation on its head. Instead of binning the data to find the Shannon entropy, it is possible to use the Shannon entropy in 4.1 to determine the optimal binning by substituting it into 2.6.

$$\Delta = \frac{2^h}{N^{1/M}} \tag{4.2}$$

However, to determine the optimal Δ, one first needs to know the differential entropy. To do this, we will use the asymptotically unbiased and consistent KL estimator. By combining these methods and using them to fine-tune the binning, it allows for more consistent and objective histograms. The steps necessary to implement the proposed binning procedure are detailed in algorithm 1. Note that Δ is adjusted to utilise the marginal space and avoid the end bin exceeding

the support during implementation.

Algorithm 1: The first step in the two-stage noisy resampling entropy estimator

Input:

- $\{x_i\}_{i=1}^{N} \in \mathbb{R}$: the sample
- $k \in \mathbb{Z}^{+}$: the number of nearest neighbours
- $M \in \mathbb{R}$: the N dependence in the initial Shannon entropy

Quantise:
for $i \leftarrow 1$ **to** N **do**
  $\lambda_{k,i} \leftarrow \text{Min}\{||x_i - x_j||\}\ j \neq i$: the k^{th} smallest distance
end
$\hat{h}_{KL} \leftarrow \frac{1}{N}\sum_{i=1}^{N} \log(\lambda_{k,i}) + \log(S_1) + \log(N) + \gamma - L_{k-1}$
$\Delta \leftarrow |\frac{2^h}{N^{1/M}}|$
$q \leftarrow \text{Floor}\left[\frac{\text{Range}}{\Delta}\right]$
$\Delta \leftarrow \frac{\text{Range}}{q}$: adjusting Δ to utilise marginal space
for $i \leftarrow 1$ **to** N **do**
  $X_i = \text{Floor}[\frac{x_i - x_{min}}{\Delta}]$
end

Output:

- $\{X_i\}_{i=1}^{N}$: the quantised instances

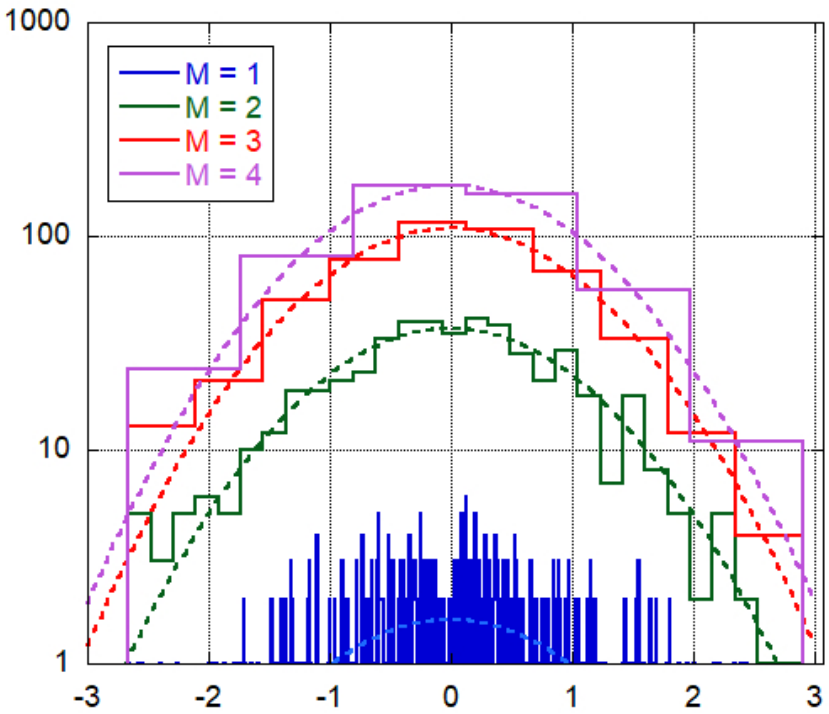

Figure 4.3: Histograms of the same normally distributed sample, with 500 data points, binned with algorithm 1 for different values of the algorithm parameter M. The histograms are plotted over one another and shown on a log scale to better illustrate the over-binning for $M = 1$. Note that the counts are given, not the pdf estimates, to allow the viewer to compare shape more easily.

Figure 4.3 shows how the proposed method performs on a normal distribution for $M = \{1, 2, 3, 4\}$ and $h_{true} = 2.05$ bits. The bin widths for these values of M were calculated to be $\Delta = \{0.008, 0.19, 0.56, 0.93\}$ respectively. These corresponded to differential entropy values of $h = \{1.11, 2.03, 2.04, 2.06\}$ bits. With the exception of $M = 1$, the resultant estimate corresponds well with the true entropy. As supported by the N dependence for the minimum MISE, $M = 3$ achieved one of the best estimates. Conversely, $M = 1$ demonstrated extreme over binning with $q = 690$ compared to the 500 instances in the sample, demonstrating the importance of the choice of M. For the same data, the bin widths for Scott's formula and Hacine-Gharbi's method were similarly $\Delta = 0.44$ for both corresponding to $h = 2.02$ bits. Which interestingly, despite the Δ value falling between those for $M = 2$ and $M = 3$, Scott's and Hacine-Gharbi's methods perform slightly worse than $M = 2$. This begs the question what value of M should be used.

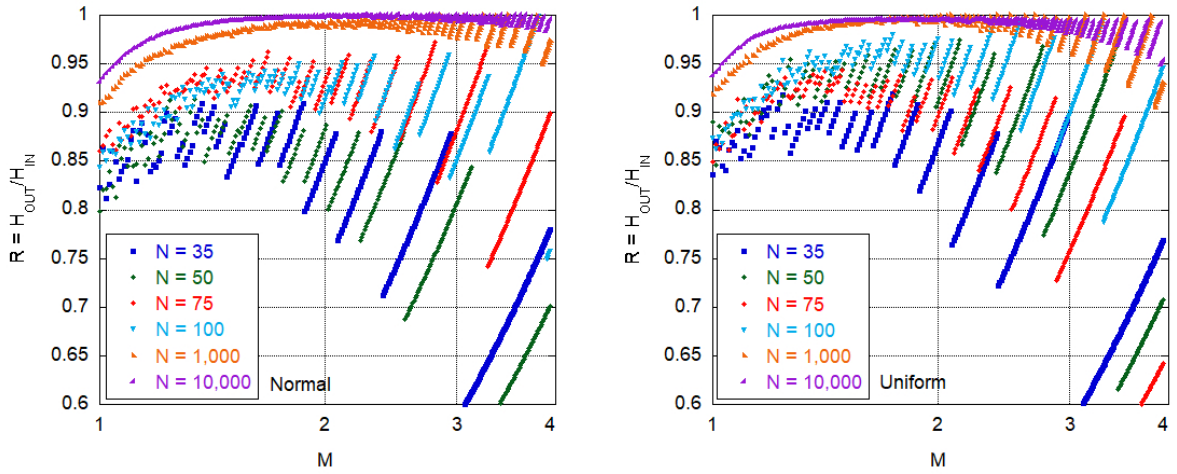

Figure 4.4: A plot of the ratio $R = H_{OUT}/H_{IN}$ as a function of M for different sample sizes on a $\log(M)$ scale. This is shown for normal (left) and uniform (right) distributions.

Equation 4.1 expresses our estimate of the Shannon entropy of a finite sample. Therefore, the Shannon entropy, estimated from the resulting histogram's frequencies, should be approximately equal to the initial estimate, from equation 4.1. For figure 4.3, consider the example $M = 3$. We determined the bin width using the Shannon entropy $H_{IN} = \log(N)/3 = 2.99$ bits, while the Shannon entropy calculated from the resulting histogram was $H_{OUT} = 2.88$ bits. Although similar in value, we ask if there is a value of M for which $H_{IN} = H_{OUT}$.

Consequently, we define the variable $R = H_{OUT}/H_{IN}$. Therefore, as $H_{OUT} \to H_{IN}$, $R \to 1$. The behaviour of R is shown in figure 4.4 for normal (left) and uniform (right) samples over the range $1 \leq M \leq 4$. On the left, normal samples of varying sizes are generated with zero mean and unit standard deviation. Then, the samples are quantised according to algorithm 1, and H_{OUT} is calculated from the resultant frequencies. Each point represents the average R value determined from 500 i.i.d trials. On the right, we repeat the experiment for uniform samples with $\mathcal{U}[0, 1)$.

The behaviour of R is comparable for all tested distributions and can be derived for the uniform distribution [3] by considering the Poisson statistics of the mean entries per bin.

$$R = \frac{H_{OUT}}{H_{IN}} = \frac{M}{\log_2(N)} \sum_{j=1}^{j=N} \frac{j}{j!} \log_2 \left(\frac{N}{j} \right) N^{(1-\frac{1}{M})(j-1)} \exp(-N^{1-\frac{1}{M}}) \tag{4.3}$$

Generically, as M increases, we observe R increasing before plateauing at some maximum value dependent on the sample size and then decreasing. All sample sizes reach their maximum R value between approximately $1.5 \leq M \leq 2.0$, after which R plateaus. This indicates that the statistical fluctuations have dissipated by $M = 2$. Further increases in M result in the deterioration of R, the point at which this occurs increases with sample size. The decline of R suggests an optimal region of M with a degradation of the binning for lower and higher values. The deterioration of R coincides with fluctuations characterised by diagonal lines. The fluctuations are attributed to the change in quantisation for increasing M. As M increases, the bin width Δ increases and there are fewer bins. When there are only a few bins, the data is under-binned, and increasing M does not significantly change the relative frequencies. Therefore, $H_{IN} = \log(N)/M$ decreases, while H_{OUT} remains the same. Resulting in the gradual rise of the ratio R. Eventually, M increases sufficiently to decrease the number of bins. Thus changing the quantisation and H_{OUT} decreases accordingly. Despite H_{IN} only decreasing marginally, the decrease in H_{OUT} is instantaneous and large. This sudden change to H_{OUT} causes the observed jump in the value of R. This process of a gradual rise followed by a sudden drop continues until the value of M corresponds to $q = 1$.

For smaller sample sizes ($N \leq 100$), R peaks for $M \approx 1.5$ and quickly degrades. Whereas, for larger sample sizes ($N > 100$), R approaches one between $1.5 \leq M \leq 2$ and plateaus for a range of M values before degrading. This difference between the sample sizes is unsurprising, as smaller values of the number of bins, q more accurately represent the distribution when the sample size is small. The degradation of R for $M \approx 3$ contrasts Scott's formula, which suggests a value of $M = 3$. However, even for reasonable sample sizes, R begins to deteriorate for $M < 3$, indicating Scott's formula under-bins the data.

To demonstrate that $M = 3$ is under-binned, we consider the χ^2_{red} for a normal least-squares fit on the histograms in figure 4.3. When fitted to the normalised relative frequencies, a reduced $\chi^2_{red} = 1$ indicates that the histogram is representative of the underlying distribution. Therefore, the reduced χ^2_{red} give an indication of which M corresponds to the best representation of the true pdf. The pdf estimates for $M = \{1, 2, 3, 4\}$ achieved $\chi^2_{red} = \{0.43, 0.85, 0.61, 0.41\}$, respectively. These results support the conclusion that $M = 3$ would result in an inferior density estimate, instead suggesting $M = 2$ would be more suitable.

Cost function

In the previous experiment, the behaviour of R did not tell us about the quality of the resulting histogram. A recent study by Shimazaki and Shinomoto [4], similar to Scott, uses a MISE minimisation procedure to derive a cost function for a histogram bin width based on the Poisson sampling in time series analysis.

$$Cost = \frac{2\overline{n} - \sigma^2}{\Delta^2} \tag{4.4}$$

Where $\overline{n}$ and σ are respectively the mean and standard deviation of the number of instances per bin. Similar to the MSE, equation 4.4 identifies and penalises increased variance and bias. One of the overwhelming benefits of this approach is that the cost function does not require prior knowledge of the pdf. We can thus use this function to determine the optimal value of M for any empirical sample.

Using the derived cost function, they find two solutions to the minimum MISE bin width. This is determined from the autocorrelation function of the pdf $\phi(\tau) = \int p(x)p(x-\tau)$, where the integral is over the relevant range of $p(x)$. In the case where the autocorrelation function is smooth, resulting in $\phi'(0) = 0$ due to the symmetry, then the optimal bin width is

$$\Delta \approx \left(-\frac{6\mu}{\phi''(0)N} \right)^{1/3} \tag{4.5}$$

Where μ is the mean spike rate. Interestingly, Shimazaki and Shinomoto's solution in 4.5 is consistent with Scott's formula for a non-time series histogram i.e. when $\mu = 1$. For the proof see appendix A where we use a Taylor expansion to show that $\phi''(\tau) \approx \int p'(x)^2 dx$.

The second solution requires the autocorrelation function to have a cusp at $\tau = 0$ ($\phi'(0+) < 0$), for which the optimal bin width is

$$\Delta \approx \left(-\frac{3\mu}{\phi'(0)N} \right)^{1/2} \tag{4.6}$$

Equation 4.6 for the cost function additionally provides a solution to the uniform distribution, which has a cusp at $\tau = 0$. This is an advantage over Scott's formula, which requires the first derivative to be non-zero.

These solutions suggest that to remove statistical fluctuations and avoid under binning the N dependence of H_{IN} $2 \leq M \leq 3$ is a reasonable rule of thumb.

Using Shimazaki and Shinomoto's cost function, we explore the values of M, which simultaneously minimise the bias and variance for the one-dimensional distributions described in table 4.2. We simulate samples of sizes $N \in \{50, 100, 200, 500, 1000, 5000\}$ for each distribution, where the data was quantised using the proposed method set out in 1. The differential entropy in

equation 4.2 was estimated using the KL estimator with $k = 4$, and M was varied between 1.0 and 6.0 in increments of 0.1. The results are shown in figure 4.5. Note that the cost is scaled between zero and one for easier comparisons between the distributions. It is apparent from figure 4.5 that the cost function reflects the behaviour observed for the MSE of the differential entropy in figure 4.2. However, here we consider these features for single samples in terms of M. Thus, one can deduce an appropriate value for M independent of sample size.

Independent of the sample size or distribution, the cost rapidly decreases for $M > 1$ due to statistical fluctuations dissipating. For $M > 3$, the cost rises slowly due to under-binning, with the exception of the uniform distribution. As we previously observed with the MSE, the characteristic bias from under-binning does not manifest in the uniform distribution as it is flat. We also note the step-like features for $M > 3$, analogous to the diagonal lines observed in figure 4.4 due to under-binning. For non-uniform distributions, the steps-like feature corresponds with an increasing cost. The decrease in statistical fluctuations combined with the rising bias leaves a minimised cost region between $M = 2$ and $M = 3$, corresponding to the two solutions found by Shimazaki and Shinomoto [4]. However, the minimum cost does not depend on the distribution, as [4] suggested. Instead, it is easy to see that the behaviour of the normalised cost function, for $M < 2$, is consistent between distributions. This can be proven mathematically [5].

Efficiency

Recall the maximum Shannon entropy distribution is a uniform distribution, where $H = \log(q)$. A uniform distribution is the most efficient way H bits of information can be stored. The more the distribution deviates from uniformity, the less efficient the use of marginal space. As only a perfectly uniform distribution has an efficiency of one, all other distributions have limited efficiency. Nevertheless, the efficiency of the binning can be compared for different values of M. The efficiency will plateau for some M value when it has reached the maximum for its underlying density. Figure 4.6 shows the bin allocation efficiency for different values for M using the same data as in figure 4.4 with the results from the normal distribution on the left and the uniform on the right.

$$\epsilon = \frac{2^H}{q} \tag{4.7}$$

Figure 4.6 illustrates the improvement of the use of the marginal space for increasing M. For the normal distribution we observe that the larger the sample size the lower the maximum efficiency. This is because, for distributions with tails, the difference in densities between the peak and the edges of the tail increases with N and the shape becomes less and less uniform. For the uniform distribution the opposite is true as the density is better estimated with increasing N. Eventually, for all distributions, the efficiency plateaus at $M \approx 2$ and once again we observe step like jumps indicating that the shape of the underlying distribution is being lost as the bins become more and more equiprobable. The larger the sample size, the larger the value of M is before the shape of

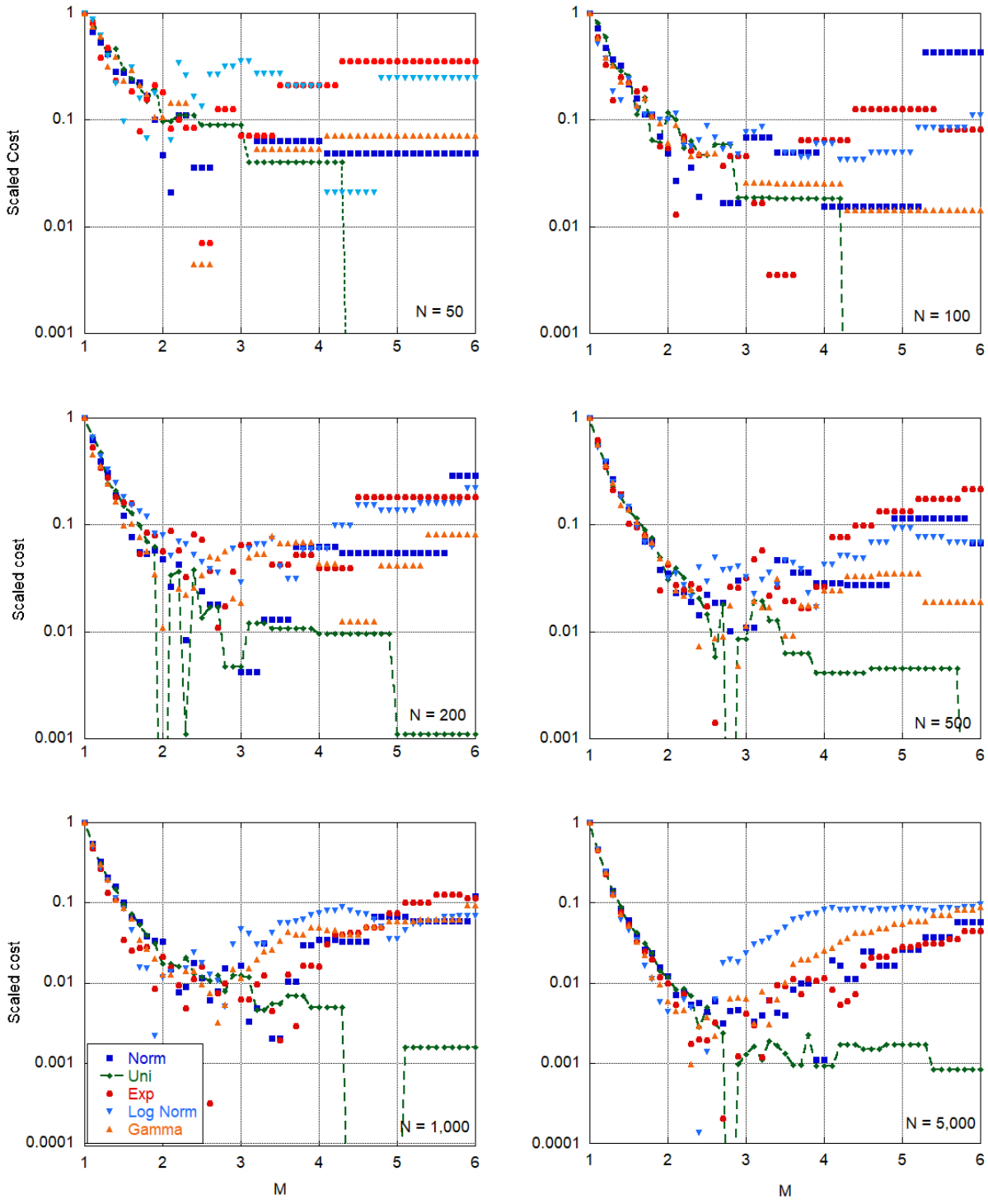

Figure 4.5: The histogram cost function in [4] as a function of the histogramming parameter M in algorithm 1, where increasing M corresponds to fewer bins. Each data point represents the binning cost of a single sample of size N, as indicated on each plot. The data points for the uniform distribution are connected to highlight that, unlike the other distributions, it continues to decrease for large M. Note also that the cost is scaled so that the minimum cost of each histogram is zero and the maximum cost is one.

the distribution begins to degrade. For $N = 35$ this happens as low as $M = 1.5$, whereas for $N = 1,000$ this occurs at $M \approx 3$ and for $N = 10,000$ this is not observed in the given range. For the small samples ($N \leq 100$) these step like features are observed shortly after the efficiency

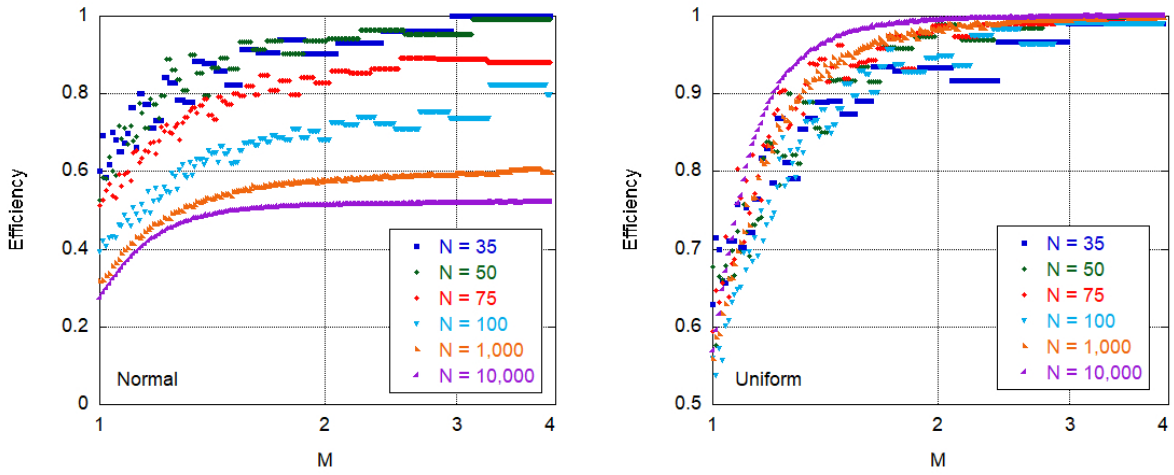

Figure 4.6: A plot of the binning efficiency ($= 2^H/q$) of algorithm 1 as a function of M. Demonstrated for different sample sizes on a $\log(M)$ scale. This is shown for normal (left) and uniform (right) distributions.

plateaus. Increasing M beyond this would result in a degraded density estimate. Therefore, for small sample sizes M should be greater than 1.5 and for large sample sizes M should be less than 3.

Examples

In figure 4.7 we illustrate how the method performs for different values of M in the reduced range $M = \{1.5, 2.0, 2.5, 3.0\}$ on artificial samples of normal, uniform, exponential and gamma distributions, as detailed in table 4.2. It is clear, even for $M = 1.5$, the extent of the over-binning which was previously identified by the cost function. This is apparent for all the probability distributions analysed indicated by the large fluctuations in the bin heights. As M increases the histogram becomes more stable and provides a better estimate for the pdf.

For a log normal distribution we also fit the histogram using a standard least-squares fit, as shown in figure 4.8. The resulting fitting parameters can be found in table 4.3. The reduced χ^2 is closest to one for $M = 2$ indicating that the data is best represented for this value of M.

M	Δ	q	H (bits)	bias (bits)	μ	error	σ	error	χ^2_{Red}
1.5	0.07	225	5.74	0.144	0.03	0.01	1.09	0.07	0.62
2.0	0.19	80	4.35	0.093	0.02	0.02	1.00	0.06	0.95
2.5	0.35	43	3.51	0.052	0.03	0.02	0.98	0.05	0.88
3.0	0.53	28	2.92	0.043	0.06	0.02	1.00	0.06	0.71

Table 4.3: Histogram attributes and the corresponding least-squared fit parameters for the Log Normal density estimate in figure 4.8 for different values of the binning parameter M in algorithm 1.

Unfortunately, there is not one value of M that is ideal for all situations. Instead, there is a range of appropriate values which a user can fine tune, within reason, for a given data set. For general

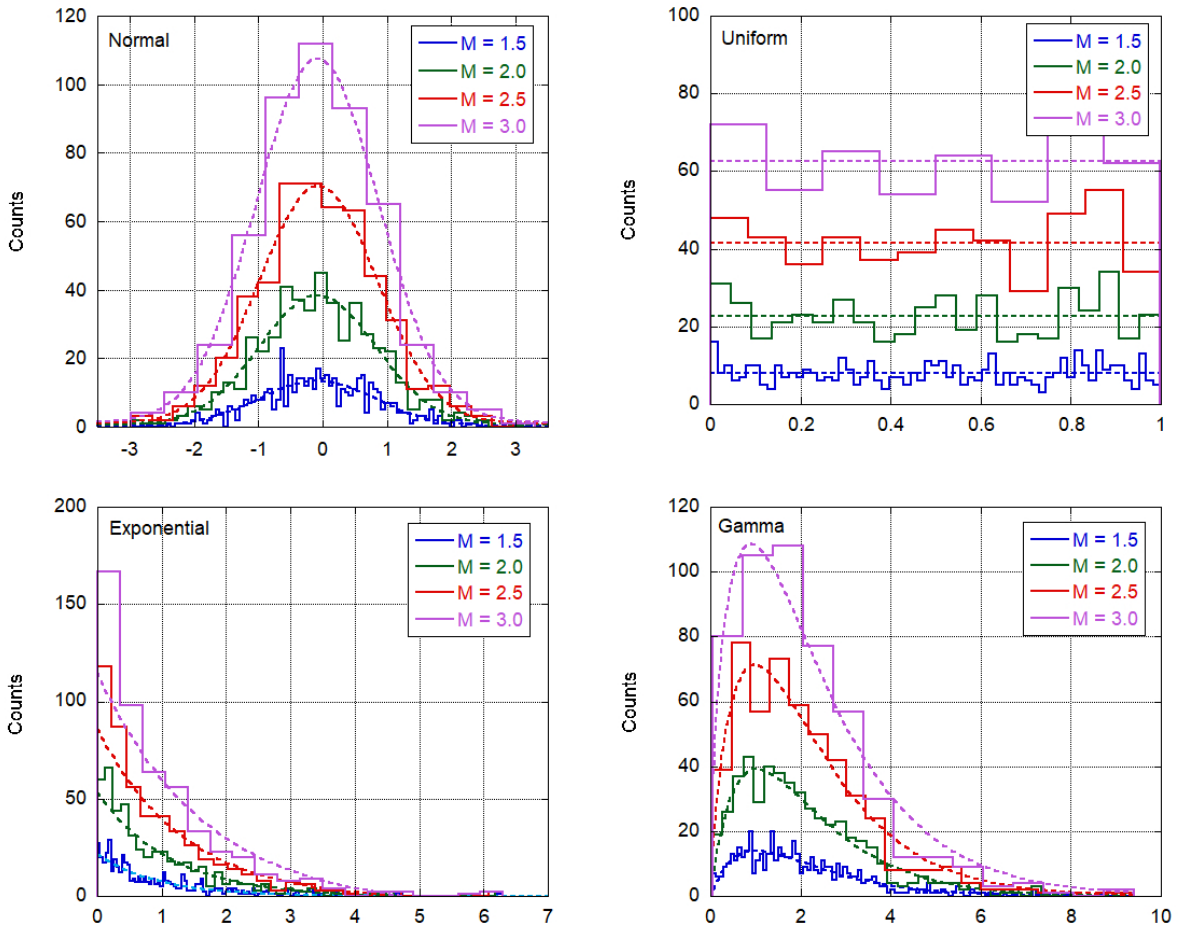

Figure 4.7: Histograms of the several pdfs. For each distribution, a sample of 500 data points was binned with algorithm 1 for different values of the algorithm parameter M. Note that these have not been normalised and therefore cannot be used for pdf fitting. The fits shown are for illustration purposes only.

distributions a value of between 2 and 3 is recommended, with $M = 2$ giving the maximum entropy without introducing statistical fluctuations. The most important thing, however, is that M is consistent for all variables. It is the act of setting the Shannon entropy such that it is equal for all variables that enables an objective comparison of information content between variables.

This is a data driven approach to quantising samples that can be used on any continuous random variable. The binning algorithm makes no assumptions about the underlying pdf and instead assumes an empirical Shannon entropy of $\log_2(N)/M$ bits. An ansatz based on the remarkable result that the Shannon entropy of a continuous random variable is only dependent on the sample size. For the remainder of this work we will use the parameter $M = 2$ unless otherwise specified.

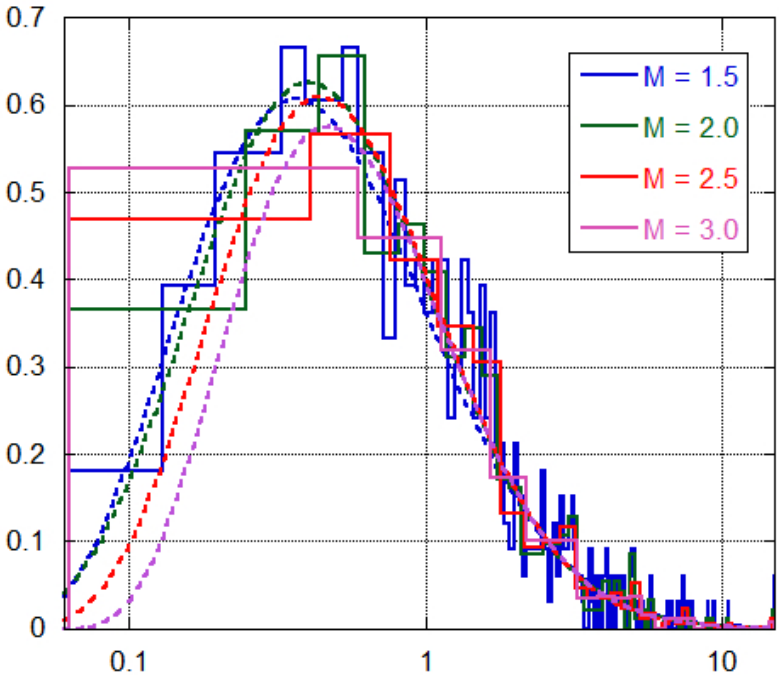

Figure 4.8: The evolution of a Log Normal distribution ($\mu = 0$ and $\sigma = 1$) for fixed $N = 500$ and increasing M. The sample range is plotted using a log scale to allow a better visual comparison of the different binning and fittings. The sample was fitted with a log normal distribution using least squares fitting for each value of M. The corresponding histogram attributes, fit parameters and χ^2_{Red} can be found in table 4.3.

4.2 Adding noise

Now that we have a reliable approach for qauntising continuous variables, it might be instinctual to simply estimate the entropies or mutual information via the conversion equation $h = H + \log(\Delta)$, however this would be ill-informed. As previously discussed all partitioning methods are not without their limitations due to the loss of information inevitable from the quantisation, and we remind the reader that this was not the intention of the quantising approach proposed. Continuous estimators, such as the nearest-neighbour KL and KSG estimators, perform comparatively better than quantising methods where the relative frequencies are plugged into the equation for Shannon entropy. These nearest-neighbour estimators, however, can only be applied to purely continuous variables as well as having a number of known problems for edge cases.

We aim to build on the quantising method described in section 4.1 which lays the foundations for an ensemble entropy estimator for both discrete and continuous variables alike using the superior nearest-neighbour methods. We will also address some of the known nearest-neighbour limitations and show that entropy estimates are made more robust to sample fluctuations due to the process of quantisation and additive noise.

4.2.1 Noise distribution

For mixed samples with both continuous and discrete variables, one approach would be to quantise the continuous variables and apply purely discrete techniques, or vice versa for discrete to continuous. The core idea of our approach is to convert all continuous variables to discrete through the quantisation in section 4.1, and then convert all variables (discrete and quantised) to continuous by the addition of noise, ϵ.

$$x_i = X_i + \epsilon \tag{4.8}$$

Where X_i and ϵ are respectively independent sequences of i.i.d variables.

Adding noise to a discrete sample is a technique used to enable the application of more accurate continuous methods. Typically these approaches add continuous normal noise, which is regarded as advantageous due to its symmetry and smoothing properties. However, note that a poorly chosen noise distribution will distort the features and add information to the entropy estimate, perceived as bias. This introduced bias has been repeatedly identified in the literature [6], [7], resulting in the rejection of this approach as the research has thus far deemed it inappropriate. Here, however, we show that the observed bias is due to the noise distribution having an entropy itself, which combines with the entropy of the sample. Using this insight we show how the addition of noise can be applied successfully to improve the estimate of both discrete and quantised variables.

To substantiate this let $u(x)$ be a continuous pdf of a finite sample, which is only non-zero over a fixed range of x, [0,Δ). Now consider $u(x)$ repeated q times along the x-axis, each translated by Δ so that there is no overlap between consecutive distributions. $u(x + (i-1)\Delta)$ represents the pdf of the continuous random number restricted to a single bin: $[(i-1)\Delta, i\Delta]$. Figure 4.9 (left) illustrates this for $i = 1, 2, 3$, where $u(x + (i-1)\Delta)$ has been shortened to u_{i-1}, and the original non-translated pdf is $u(x) = u_1$.

Now define a new pdf, $g(x)$, which describes the original $u(x)$ plus its translations up to q.

$$g(x) = \sum_{i=1}^{q} p_i u_{i-1} \tag{4.9}$$

By definition $\int_{-\infty}^{\infty} g(x)dx = 1$ and the probability coefficients, p_i are positive and $\sum_{i=1}^{q} p_i = 1$.

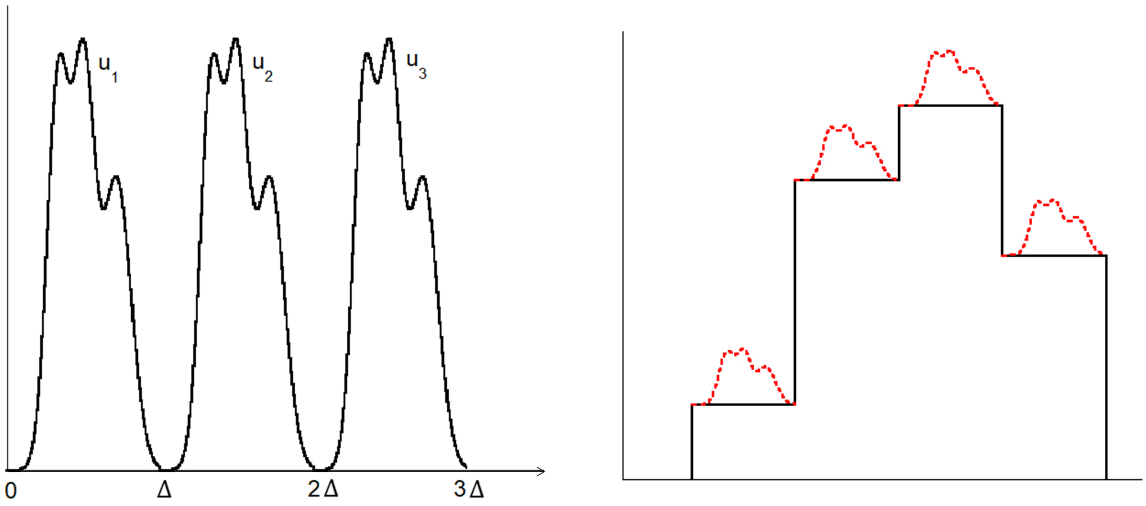

Figure 4.9: Schematics to illustrate the idea of adding noise to discrete data. On the left, we visualise non-overlapping noise distributions, where each repeat is only non-zero in the range $(i\Delta, (i+1)\Delta)$. On the right, we visualise the concept of each bin in a histogram with a noise distribution that does not overlap into neighbouring bins.

The differential entropy for $g(x)$ is therefore

$$\begin{aligned} h(g) &= -\int_{-\infty}^{\infty} g(x)\log_2[g(x)]dx \\ &= -\int_{-\infty}^{\infty} \sum_{i=1}^{q} p_i u_{i-1} \log_2 \left[\sum_{i=1}^{q} p_i u_{i-1}\right] dx \end{aligned} \tag{4.10}$$

By construction, the consecutive u_{i-1} do not overlap, meaning only one is ever non-zero for a given interval of Δ. This non-overlapping characteristic allows the integral to be written as a sum of its constituent parts.

$$\begin{aligned} h(g) &= -\sum_{i=1}^{q} \int_{(i-1)\Delta}^{i\Delta} p_i u_{i-1} \log_2[p_i u_{i-1}]dx \\ &= -\sum_{i=1}^{q} \int_{(i-1)\Delta}^{i\Delta} p_i u_{i-1} \big(\log_2[p_i] + \log_2[u_{i-1}]\big)dx \\ &= -\sum_{i=1}^{q} p_i \log_2[p_i] \int_{(i-1)\Delta}^{i\Delta} u_{i-1}dx - \sum_{i=1}^{q} p_i \int_{(i-1)\Delta}^{i\Delta} u_{i-1}\log_2[u_{i-1}]dx \\ &= H(p) + h(u) \end{aligned} \tag{4.11}$$

Therefore, the differential entropy of $g(x) = \sum_{i=1}^{q} p_i u_{i-1}$ is a sum of the Shannon entropy of the probability coefficients, p_i and the differential entropy of a single $u(x)$ distribution. By considering $u(x)$ the noise distribution and p_i, the normalised relative frequencies of a quantised sample, this set up describes the entropy of a discrete variable with added noise. The diagram in figure 4.9 (right) illustrates this idea.

We can verify the above result using simulations of distributions with known differential entropy for the underlying and noise distribution. According to equation 4.11, the expected entropy will equal the sum of the Shannon entropy of the underlying distribution and the differential entropy of the noise distribution used. As by the derivation, subsequent $u(x)$ cannot overlap. Therefore, the selected distribution parameters restrict the majority of the non-zero density to the boundaries of the bin to limits the complications of overlapping noise distributions. Furthermore, we consider symmetric and non-symmetric noise distributions, some of which fill the bin, and others only partial fill it. This variety of noise distributions ensures a range of features of which to test equation 4.11 against.

We quantised one-dimensional normal distributions using algorithm 1 and add noise to the binned instances. Note that the binned instances are integer values corresponding to the bin they belong to, i such that the bin boundaries are always $[i, i + 1)$. The entropy was then measured using the KL entropy estimator. This experiment is shown in figure 4.10, where the height of the bars indicate the theoretical differential entropy of the noise distributions shown on the x-axis. This is the expected divergence from the true value of the underlying normal pdf. For

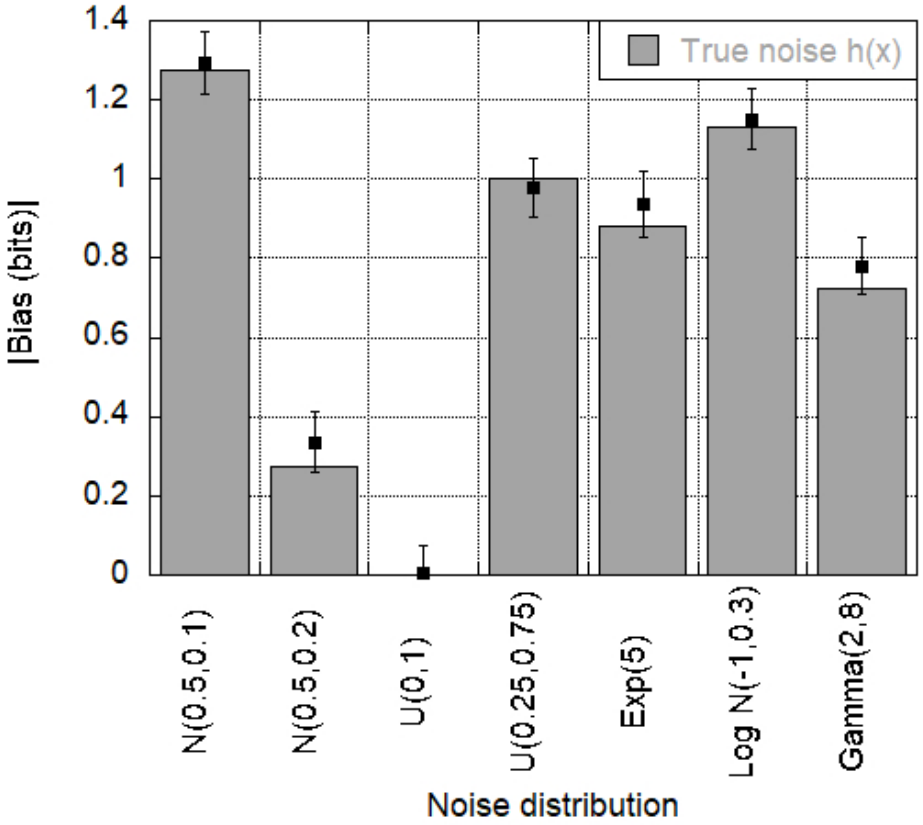

Figure 4.10: The measured bias of the entropy estimate for a quantised normal distribution with zero mean and unit variance for different noise distributions. Each data point is the average deviation or "bias" for the entropy estimate of the underlying pdf from 500 i.i.d trials, each consisting of $1,000$ instances. Note that all values are given as their absolute value. The error bars indicate ± 1 standard deviation obtained from the ensemble of estimates.

all distributions, the observed bias is consistent with the theory that $h(g) = H(P) + h(u)$. This result explains the failures for noise distributions in previous work, as the noise entropy was unaccounted for.

The noise distributions used were chosen to limit the majority of the non-zero density to the boundaries of the bins, as the derivation does not hold for overlapping noise distributions. We demonstrate the consequences of the support of the noise distribution in appendix B by considering normal, uniform and exponential noise distributions with varying parameters. It may be intuitive to think that adding noise that exceeds the bin boundaries may somewhat counteract the bias in histograms that result from statistical fluctuations of instances entering neighbouring bins. However, we briefly compare the results of a uniform and normal noise distribution in figure 4.11 to illustrate the advantages of a noise distribution that is well confined within the bin. For a discrete uniform distribution $\mathcal{U}[0, 15)$, we add a continuous uniform noise distribution ($\mathcal{U}[0, 1)$), and a normal noise distribution ($\mathcal{N}(0.5, 0.3)$). By definition the uniform noise is perfectly confined to the bin of width $\Delta = 1$ and $h_{true} = 0.00$ bits. Whereas, the normal noise, centered in the middle of the bin, has a standard deviation of $\sigma = 0.3$ and $h_{true} = 0.310$ bits. Although most of the non-zero density is confined within the bin for these parameters, the tails do somewhat exceed the bin boundaries. Figure 4.11 shows the bias of the entropy estimate from the entropy of the underlying $\mathcal{U}[0, 15)$ distribution. For the uniform distribution, which is perfectly bounded and has zero differential entropy, the bias is small and decreases asymptotically as expected. For the normal noise distribution, which has tails exceeding the bin boundaries, there is an observed bias, as expected. However, the bias is significantly less than the entropy of the noise distribution, so it cannot be directly subtracted, and it does not vanish asymptotically.

As the continuous uniform noise distribution $\mathcal{U}(0, 1)$ has a differential entropy of zero and is well constrained by the bin boundaries, the noise does not distort the underlying distribution, and the entropy estimate is unaffected. This is beneficial because it removes the need to subtract off the noise distribution entropy and avoids unpredictable effects from the noise distribution exceeding the bin boundaries. Interestingly, this is the foundation of histograms as density estimators. In the limit of $\Delta \to 0$, the spread of instances within a bin is approximately uniform. Therefore, we propose that a continuous random sample can be approximately reconstructed from its histogram by the addition of continuous uniform noise $\mathcal{U}[0, 1)$, denoted by u_j.

$$\hat{x}_j = X_j + u_j \tag{4.12}$$

Thus, generating an approximation of the original data of equal size to the original sample. We refer to this as randomising. An example is shown in figure 4.12 for a bivariate normal distribution that has undergone the quantisation in algorithm 1 and randomisation with uniform noise. The noisy sample on the right of figure 4.12 replicates the original sample reasonable well.

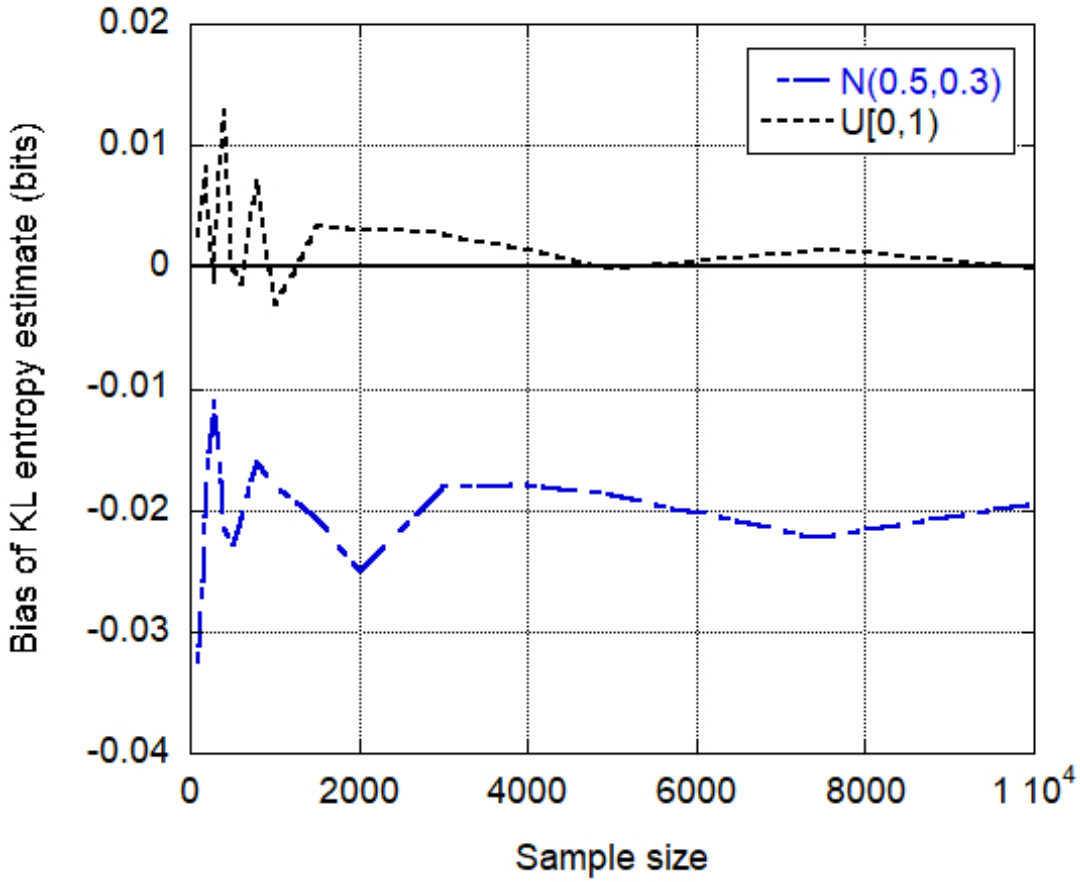

Figure 4.11: The comparison of a uniform noise distribution perfectly bounded within the bin and a normal noise distribution with tails that exceed the bin boundaries. The average observed bias of the resulting entropy estimates for 250 i.i.d trials of a discrete $\mathcal{U}[0, 15)$ distribution is given as a function of sample size to illustrate the effects.

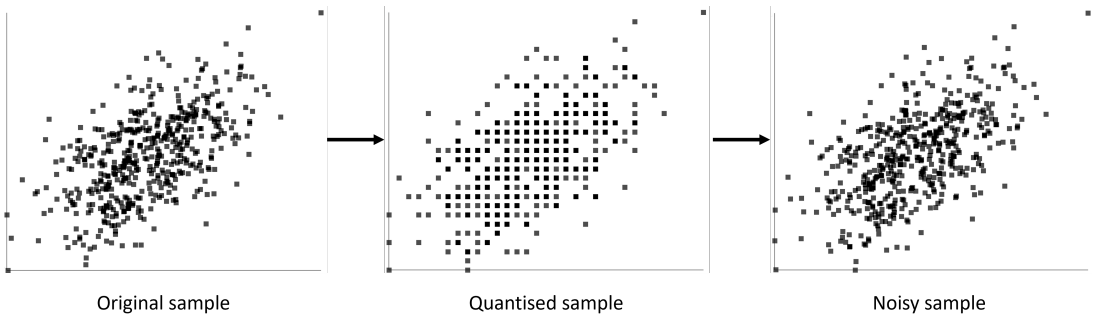

Figure 4.12: A flow chart illustrating the process of quantising and randomising a continuous sample for a bivariate normal distribution with zero mean, unit variance and $\rho = 0.6$. On the left is the scatter plot for the original sample. The sample was then quantised via algorithm 1 to obtain the scatter plot shown in the middle. On the right, we show the scatter plot for the randomised sample with uniform noise ($\mathcal{U}[0, 1)$).

If one was only interested in the mutual information, theoretically, any noise distribution could be used. Although the individual entropy estimates would be affected, the entropy of the noise would cancel out in the mutual information, provided the noise distributions were uncorrelated. We show this for the 3H-KL and KSG estimators for a discrete uniform distribution $\mathcal{U}[0, 15)$ in figure 4.13. For all noise distributions, the MSE of the mutual information decreases asymptotically, as expected. Figure 4.13 thus demonstrates that any uncorrelated and reasonably non-overlapping noise distribution can be used to estimate the mutual information. However, we note that the 3H-KL estimator cannot produce an estimate of the mutual information without noise, as there are no non-zero nearest-neighbour distances in the original discrete sample. We also note that although the KSG estimator can estimate the mutual information of a discrete sample, the result has an asymptotically increasing bias. We illustrate this in figure 4.14 for discrete $\mathcal{U}[0, 15)$ distribution.

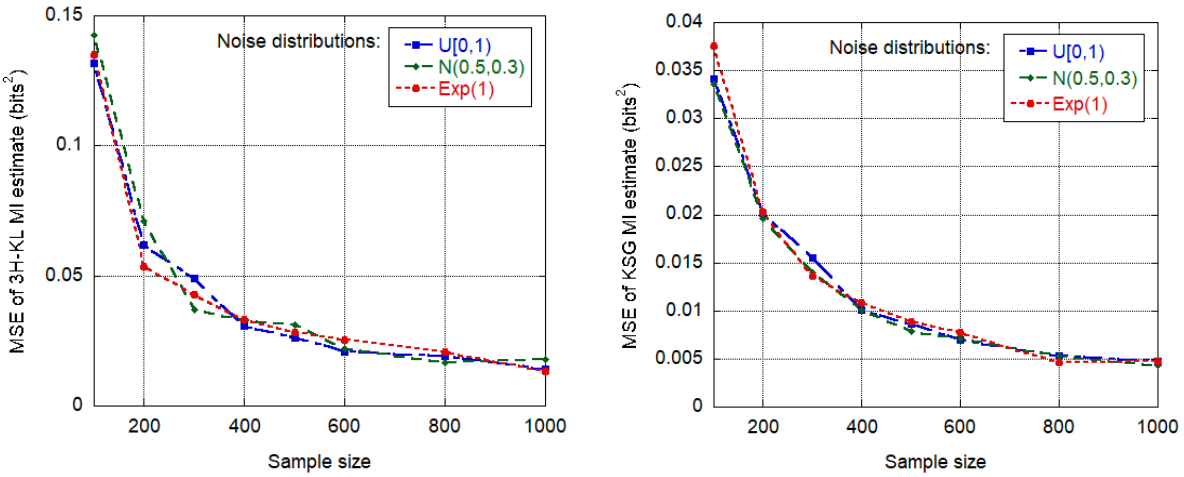

Figure 4.13: Comparing the MSE of the mutual information from 250 i.i.d trials of a discrete uniform distribution $\mathcal{U}[0, 15)$, for different noise distributions, as a function of sample size. The results for the 3H-KL estimator are shown on the left, and the results for the KSG estimator on the right.

4.2.2 Randomisation

This randomisation process can be applied to both discrete and quantised random variables. Doing so allows the application of continuous k-nearest-neighbour estimators to both data types. For example, suppose the noise was added to the discrete variable for the Shannon entropy ($h(g) = H(P) + h(u)$), and the KL estimator applied to the raw (non-quantised and non-noisy) continuous variable for the differential entropy. In that case, these cannot necessarily be combined to get any meaningful value for the mutual information. However, by quantising the continuous variable and adding noise, the KL estimator gives the Shannon entropy of the continuous distribution, making the combination compatible.

This new approach calculates the Shannon entropy via the KL estimator because the quantisation

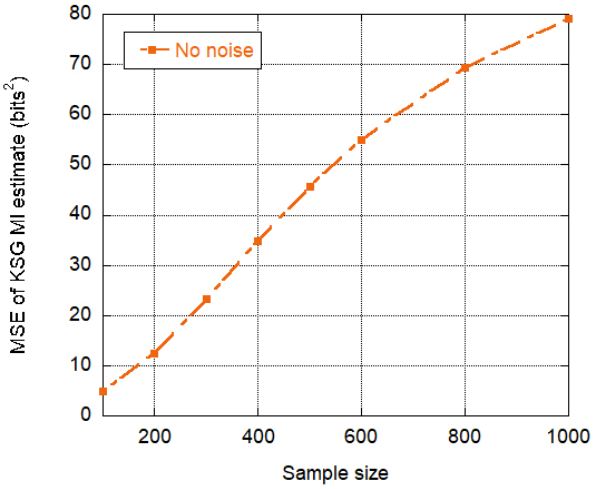

Figure 4.14: The MSE of the mutual information estimate, obtained from 250 i.i.d trials, as a function of sample size, showing the poor behaviour of the KSG estimator for a discrete uniform distribution $\mathcal{U}[0, 15)$.

effectively re-scales the distribution so that $\Delta = 1$, meaning $h(X) = H(X)$. We can also see this if we consider how the quantising affects the nearest neighbour's distances between the noisy data points, $\hat{x}_j$. Using the definition for a quantised data point

$$X_j = \text{Floor}\left[\frac{x_j - x_{min}}{\Delta}\right] \tag{4.13}$$

For a quantised sample in one dimension, the marginal space is transformed by a factor of Δ^{-1} compared to the original data sample. The added noise for the approximation of the sample: $\hat{x}_j = X_j + u_j$, does not change the order of magnitude of the marginal space once quantised. Thus, the spacing for the noisy sample points is also transformed by a factor of Δ^{-1}. When compared to the original sample, this transformation changes the value of the entropy estimation as follows:

$$\Lambda_{k,j} = \sqrt{(\hat{x}_j - \hat{x}_{k,l})^2} \approx \frac{1}{\Delta}\sqrt{(x_j - x_{k,l})^2} = \frac{1}{\Delta}\lambda_{k,j} \tag{4.14}$$

To distinguish between the KL estimator applied to the raw data and the KL estimator applied to samples of the proposed method we have denoted the latter as $\hat{\zeta}$.

$$\begin{aligned}
\hat{\zeta} &= \frac{1}{N}\sum_{i=1}^{N} \log[\Lambda_{k,j}] + \log[S_1] + \log[N] + \gamma - L_{k-1} \\
&= \frac{1}{N}\sum_{i=1}^{N} \log\left[\frac{1}{\Delta}\lambda_{k,j}\right] + \log[S_1] + \log[N] + \gamma - L_{k-1} \\
&= \hat{h}_{KL} - \log(\Delta) \\
&= \hat{H}
\end{aligned} \tag{4.15}$$

Therefore, by quantising the sample we force the differential KL entropy estimator to become

a Shannon entropy estimator. When referring to these in the future we will maintain the convention that for the original sample with k-NN distances $\lambda_{k,j}$, the KL estimator determines the differential entropy $\hat{h}_{KL}$ and for the proposed quantised and noisy sample, with k-NN distances $\Lambda_{k,j}$ the KL estimator determines the Shannon entropy $\hat{H}_W$. The derivation can easily be extended to two dimensions with the assumption that $\Delta_x = \Delta_y$. If $\Delta_x \neq \Delta_y$, however, we can still empirically show that the resultant entropy estimate in two dimensions is equivalent to the Shannon entropy and does not give rise to biases.

Consider a continuous uniform distribution, $\mathcal{U}[0, 1)$ and a normal distribution with zero mean and $\sigma = 10$. These distributions have been selected due to the large difference in the bin width calculated via algorithm 1. For example, for $N = 100$ and averaged over 500 i.i.d trials $\Delta_{\mathcal{U}} = 0.108 \pm 0.009$ and $\Delta_{\mathcal{N}} = 4.3 \pm 0.4$ for the uniform and normal distributions respectively. If the variables are independent then the theoretical joint differential entropy of these distributions is $h_{xy} = 5.369$ bits. By substituting $\hat{\zeta}_{xy} = H_{xy}$ into equation $h_{xy} = H_{xy} + \log_2(\Delta_x \Delta y)$ and using the average bin width from algorithm 1, we can estimate the joint differential entropy. If $\hat{\zeta}_{xy}$ is equal to the Shannon entropy of the system when $\Delta_x \neq \Delta_y$ then this should reasonably estimate the theoretical joint differential entropy. This experiment is shown in figure 4.15. For comparison the KL estimator applied to the raw data is also shown.

As can be seen from the plot the 2D entropy estimate from the proposed method is consistent with the theoretical value for all sample sizes. For $N = 100$ the bias of the joint distribution, as estimated via the proposed method, was 0.06 ± 0.03, this is shown to decrease, and for $N = 10,000$ the bias was calculated to be 0.0079 ± 0.0003 bits. This not only shows that using quantised nearest-neighbour distances in the KL estimator results in estimates of the Shannon entropy, but also demonstrates the effectiveness of the algorithm. The performance of the KL estimator applied to the raw data is inferior in comparison.

Whether one uses the Shannon entropy or the differential entropy to calculate the mutual information is inconsequential, as the value is the same for both. See section 2. The preferential treatment in the literature for the differential entropies solely lies in the superior estimation methods. The same methods that we have exploited here. However, properties of the Shannon entropy are advantageous as unlike the differential entropy the Shannon entropy is non-negative meaning that the interpretation of its value is unambiguous.

Note that in figure 4.15 the uniform and normal distribution were quantised and the noise added separately to each one-dimensional variable ie. $\Delta_{\mathcal{U}} = 2^{h_{KL}(\mathcal{U})}/\sqrt{N}$ and $\Delta_{\mathcal{N}} = 2^{h_{KL}(\mathcal{N})}/\sqrt{N}$ respectively. It might seem a natural progression to quantise the joint distribution for the two-dimensional entropy, such that $\Delta_{xy} = 2^{h_{xy}}/N^{1/M}$, where $\Delta_{xy} = \Delta_x \Delta y$. This, however, presents the problem of how many bins are needed for each variable. Which ultimately is the quantisation problem. Therefore, this is only suitable in cases where the bin widths needed to describe each of the variable distributions are of similar values. As if one was to naively use the same bin width

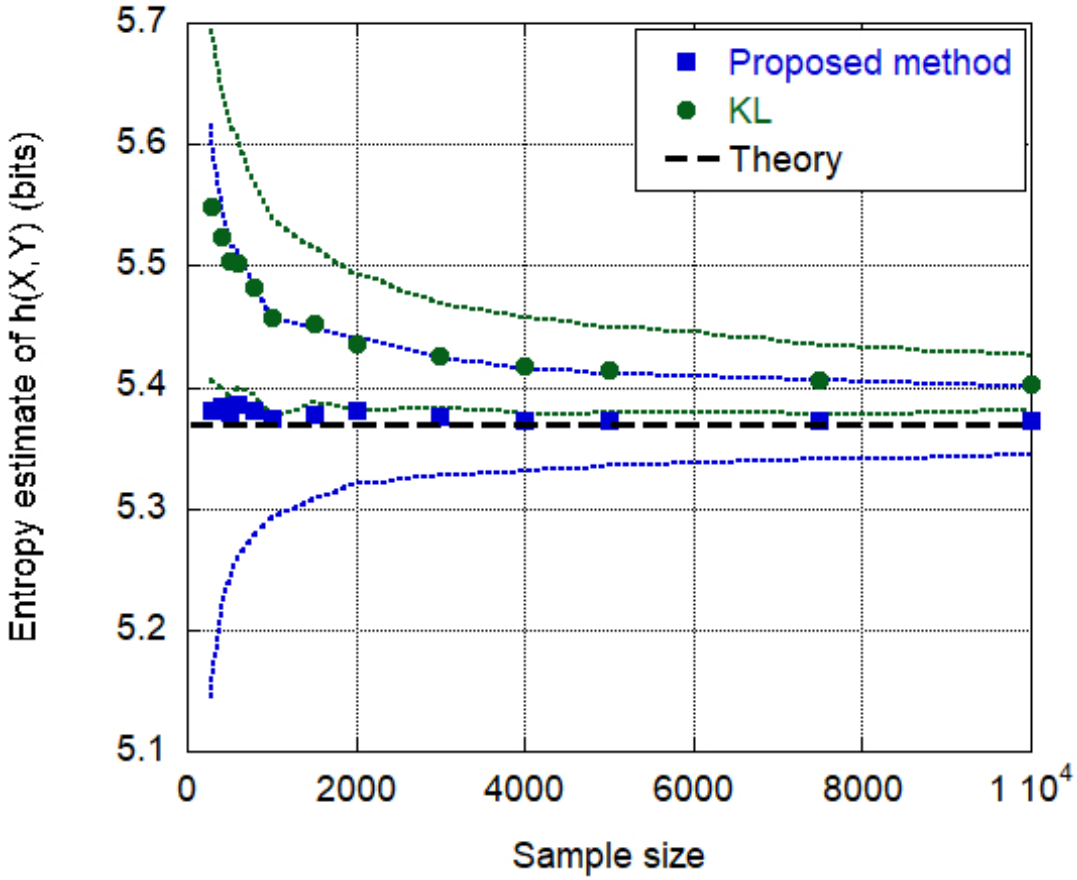

Figure 4.15: The bias of the 2D entropy estimate from the proposed algorithm as a function of sample size. The black horizontal line indicates the theoretical differential entropy at $h_{xy} = 5.369$ bits. The dashed lines indicate ± 1 standard deviation estimated from 500 i.i.d trials.

for both ie. $\Delta_x = \Delta_y = \sqrt{\Delta_{xy}}$ then the result would be two sub optimally binned distributions. In addition, the KL estimator requires a greater sample size in order to achieve the same accuracy for higher dimensions, therefore, binning the variables in this way is less accurate and more time consuming than considering each individually.

4.2.3 Resampling

The randomisation procedure generates an independent sample with the same pdf as the original data. We can repeat this procedure and then estimate the entropy for each randomised i.i.d sample via the KL estimator to obtain a distribution of estimates. The idea is that the average of the ensemble is a stable entropy estimate from a single sample. This approach is similar in concept to the ensemble method of Bagging. Bagging generates repeated samples using a bootstrap resampling method. It then applies an estimator to each sample generated and takes the average. Bagging has been shown to improve the accuracy of the estimate, especially for unstable estimators [8], [9]. Our two-stage quantisation and randomisation process is not, however, the same as Bagging. This distinction is important, as resampling using the bootstrap method

within the realm of entropy estimation is inappropriate. Bootstrap resampling is only applicable to quantities that are linear in the underlying pdf, which entropy is not. This requirement is because Bagging uses what is known as "sampling with replacement", where the same instances can appear multiple times in a single sample. k-nearest-neighbour entropy estimators interpret non-unique instances in a sample as fine-scale structures, which increases the information content [10]. The method we propose does not encounter this issue, as it is unlikely that any two instances in a sample are identical through the addition of noise. We should note that our repeated randomisation is not strictly the same as traditional resampling. Resampling typically selects instances from the original sample using a random or systematic method, treating the original sample as an alternative for the complete domain. On the other hand, we generate new samples according to the relative frequencies of the original pdf estimate. Note that prior to this point, all examples of the proposed method were implemented with $N_I = 1$.

We can generate any number of repeats by applying this procedure to each variable in a data set. Allowing the user to improve the entropy estimate without the need to collect more data. We therefore propose the generation of N_I i.i.d samples generated from the quantisation in algorithm 1 and the randomisation in algorithm 2. These algorithms combine to form what shall be referred to as "algorithm W", the proposed entropy estimator. Each approximation of the original data sample constitutes an "iteration." The KL entropy estimator can then be applied to each sample generated. The pseudo-code for this part of the algorithm is set out in algorithm 2, where $\hat{H}_W$, the output of algorithm 2, is the Shannon entropy calculated using the KL formula for differential entropy. Note that the noise is added separately to each variable for higher dimensions, such as the joint entropy, but the nearest neighbour distances are for the joint space.

Algorithm 2: The second step in the two-stage noisy KL entropy estimator

Input:

- $\{X_i\}_{i=1}^{N} \in \mathbb{R}$: the quantised sample
- $k \in \mathbb{Z}^+$: the number of nearest neighbours
- $N_I \in \mathbb{Z}^+$: the number of iterations

Randomisation:
for $i \leftarrow 1$ **to** N_I **do**
  for $j \leftarrow 1$ **to** N **do**
    $\hat{x}_j \leftarrow X_j + u_j$ where u_j is a random number from $\mathcal{U}[0, 1)$
  end
  for $j \leftarrow 1$ **to** N **do**
    $\Lambda_{k,j} \leftarrow \min \{(\hat{x}_j - \hat{x}_{k,l})^2\}\ \forall j \neq l$: The smallest k^{th} nearest-neighbour distance
  end
  $\hat{\zeta}_i \leftarrow \frac{1}{N} \sum_{i=1}^{N} \log[\Lambda_{k,j}] + \log[S_1] + \log[N] + \gamma - L_{k-1}$
end
Output:

- $\hat{H}_W = \frac{1}{N_I} \sum_{i=1}^{N_I} \hat{\zeta} i$

We now validate our technique through several synthetic experiments for continuous and discrete distributions. First, we discuss the application of our estimator to one-dimensional distributions from table 4.2, as well as the discrete distributions in 4.4. For each distribution, we implement algorithm W on 500 i.i.d trials with $k = 1$. For each sample, we estimate $H(X)$, and determine the MSE from the distribution of estimates. We illustrate that the MSE decreases as the number of iterations increases for both continuous and discrete distributions. See figure 4.16. This reduction occurs due to the reduced estimation error from repeated applications of the KL estimator, as observed in resampling techniques.

Distribution	Mean	Std dev	Support	Skewness	Kurtosis	H (bits)
Bernoulli ($p = 0.25$)	0.25	0.188	$\{0, 1\}$	1.155	0.167	0.811
Uniform $\mathcal{U}[1.6]$	3.5	2.92	$\{1, 6\}$	0.00	-1.27	2.58
Binomial $\mathcal{B}(100, 0.5)$	50	0.25	$\{0, 100\}$	0.00	-0.02	4.37
Geometric ($p = 0.95$)	1.05	0.06	$\{1, 2, 3, ...\}$	4.70	24.05	0.30

Table 4.4: Attributes of one-dimensional discrete distributions used in simulated experiments.

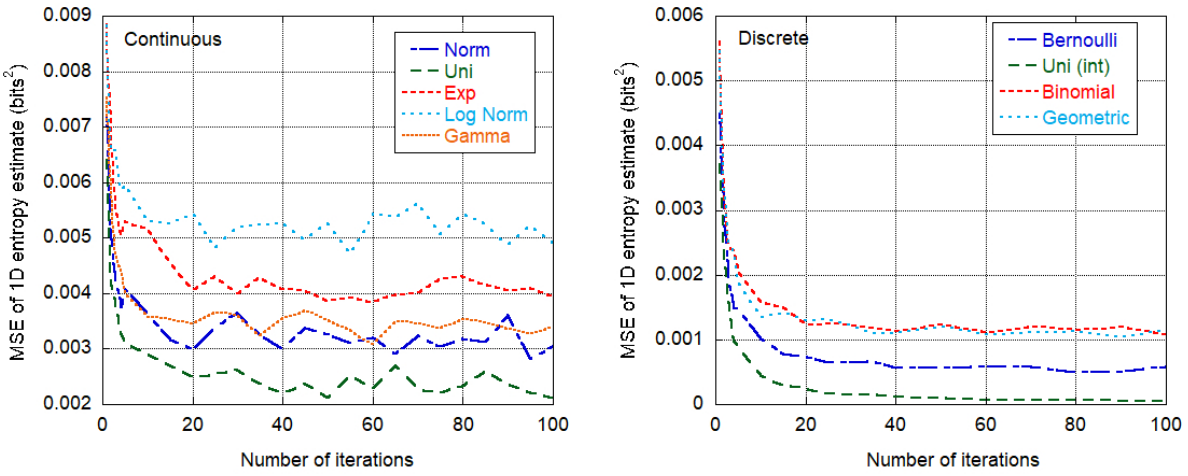

Figure 4.16: The MSE as a function of the number of iterations for samples of size $N = 1,000$. The MSE was obtained from 500 i.i.d estimates of the one-dimensional entropy. On the left this is shown for continuous distributions which were quantised and randomised according to algorithm W. On the right is the same for the discrete distributions in table 4.4. For discrete distributions there is no need to undergo the quantisation and instead only step 2 in algorithm 2 is applied. For both the continuous and discrete distributions the parameter $k = 1$ was used.

In addition, the bias decreases as a function of N_I. In figure 4.17, we implemented algorithm W on a independent bivariate normal distribution ($\rho = 0$ and $I_{true} = 0$ bits). Each graph shows the estimate of $H(X, Y)$ for a single data sample, as a function of the number of iterations applied. The cumulative average of the estimate from algorithm W is plotted for up to 200 iterations to assess the convergence. As the number of iterations increases, we observe that the estimate stabilises for $N_I \approx 50$ to a value close the true value. The KL estimator was equally applied to the same data samples for comparison. These are shown in figure 4.17 in green. The theoretical value for the two-dimensional entropy is indicated by a solid black line ($h(X, Y)_{true} = 4.094$ bits). For all samples tested, algorithm W achieved a smaller bias than

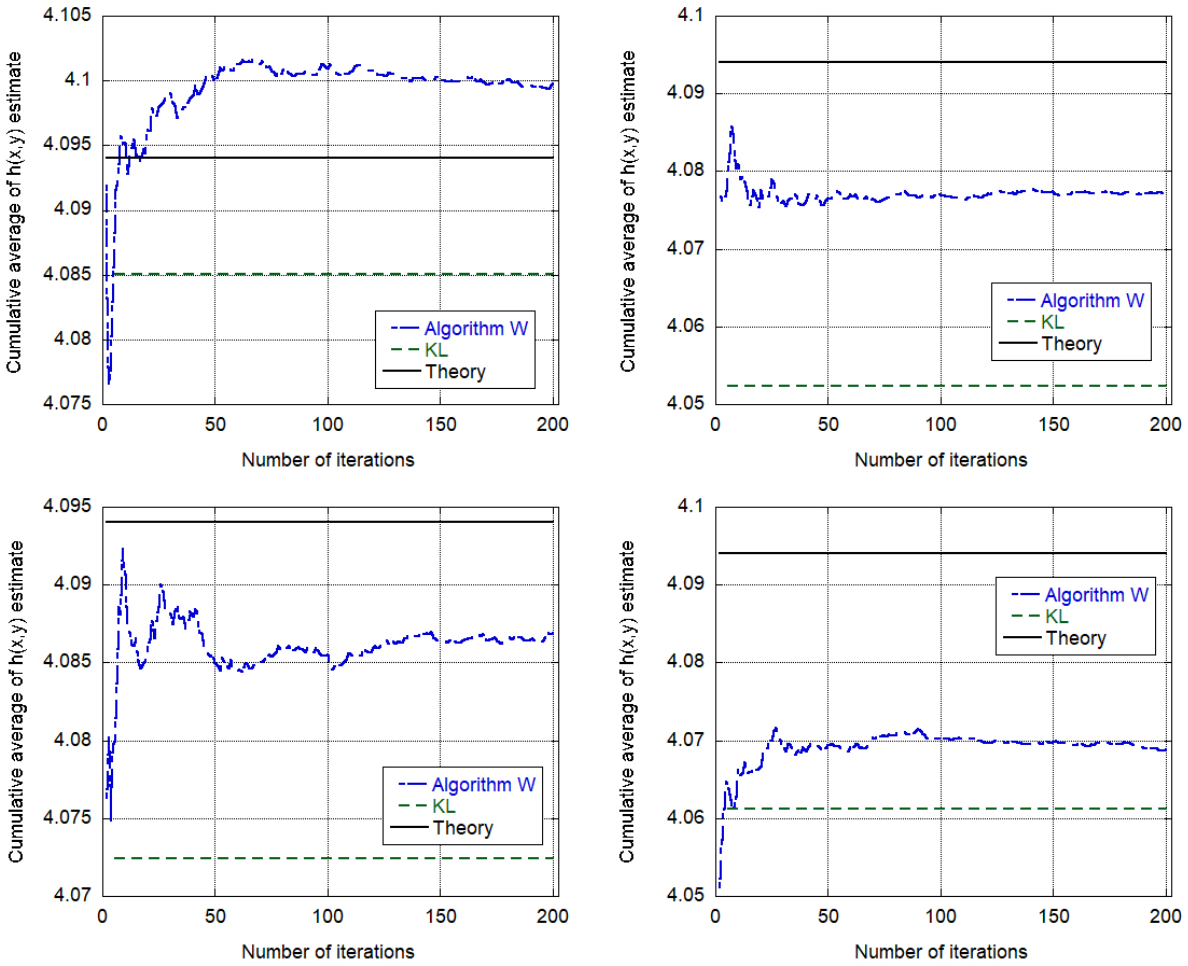

Figure 4.17: Four examples of the convergence of the algorithm W estimate for single samples of an independent joint normal distribution. For each example, the KL estimate for the same single data sample is shown in green, and the theoretical value is shown in black.

the KL estimator for the same sample. The results here are conducive to a resampling process that improves the accuracy of the KL entropy estimator and reduces the estimation error.

To illustrate that this resampling method is suitable for nonlinear functions of the pdf, we generate 500 i.i.d trials of bivariate normal distributions. Each sample is of size $N = 5,000$ and with covariance $\rho = 0.6$. For each trial, we estimate the mutual information by applying the $3H - KL$ estimator and similarly applying algorithm W for $N_I = 1$. Thus, producing sampling distributions for the bivariate normal distribution for the 3H-KL and Algorithm W, shown in figure 4.18. For a typical system, however, multiple samples are not always feasible. Repeating the calculation on randomised data allows us to estimate the sampling distribution. This is also shown in figure 4.18 for $N_I = 500$ for a single trial.

Figure 4.18 shows that the randomisation procedure mimics the sampling distribution for i.i.d trials. The results were consistent with the theoretical mutual information value, 0.3219 bits, indicated by the black dashed line. The resampling in algorithm W does not indicate any skew and accurately replicates the true probability distribution of estimates. Thus, unlike, in boot-

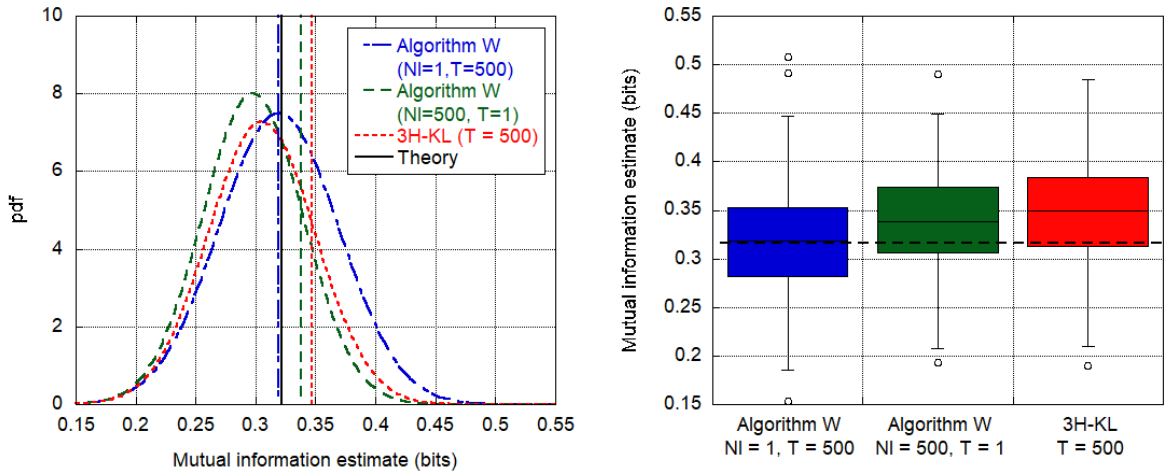

Figure 4.18: Comparison of the sampling distribution for algorithm W and the 3H-KL estimator, for N_I iterations and T i.i.d trials. On the left are the Least-squares normal fits to a histogram of the results. The vertical coloured lines indicate the μ for each distribution. On the right are distribution summaries in the form of box plots, where the mean of each is indicated by a solid black line and open circles indicate outliers.

strapping this type of resampling approach can be used with entropy estimators despite being nonlinear in the underlying pdf.

4.3 Bias and Consistency

As with most common entropy estimators, the KL estimator is asymptotically unbiased for sufficiently regular probability distributions as $N \to \infty$ [11] i.e.

$$\lim_{N\to\infty} E\left[\hat{h}(X)\right] = h(x) \tag{4.16}$$

However for higher dimensions the KL estimator requires exponentially greater sample sizes for the equivalent bias in one dimension. As a result, the bias is sample size dependent.

Due to the proposed algorithms foundations on the KL estimator it is expected to share many of its properties. To confirm the asymptotic unbiased nature of algorithm W we will consider the behaviour of the bias for increasing sample sizes. If there is a systematic drift of the estimated entropy with changing N and the drift is larger than the standard deviation this would be considered bias. However, if algorithm W is asymptotically unbiased then the the estimate will tend to the true value as N increases. Figure 4.19 shows the estimated entropy for a normal distribution in one (left) and two (right) dimensions. The theoretical values are shown in black corresponding to $h_{true} = 2.0471$ bits for one dimension and in two dimensions with $\rho = 0.6$, $h_{true} = 3.772$ bits. As the bias is inherently related to the choice of k the entropy estimate is shown as a function of N for $k = 1, 4, 10$. For smaller values of k the nearest-neighbour approach will better detect fine-scale structure, however, if too small it can falsely identify struc-

tures from statistical fluctuations. On the other hand, if k is too large the fine-scale structures are missed and the entropy estimate can be underestimated.

The average entropy estimate from 500 i.i.d samples is shown as the solid line while the dashed outer lines, with the corresponding colour, indicate ± 1 standard deviation. By considering the

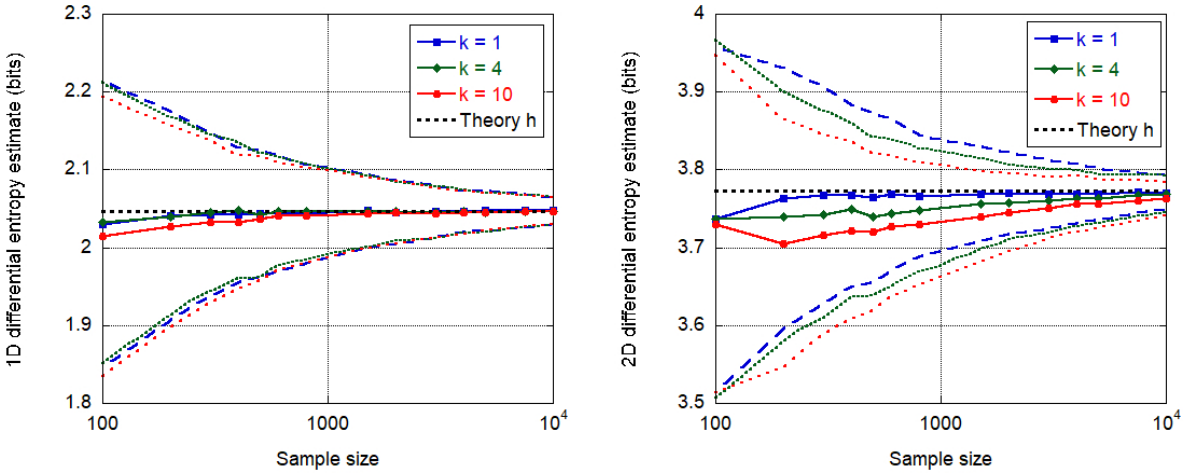

Figure 4.19: The entropy estimate as calculate using algorithm W as a function of N for $k = 1, 4, 10$. Starting with $N = 100$, 500 i.i.d samples where generated from a marginal (left) and bivariate (right) normal distribution with $\rho = 0.6$. The data was quantised according to the proposed method and re-sampled with $N_I = 50$.

estimation error there is no observed sample size dependent drift for any value of k shown. We observe that all mutual information estimates, for $k = 1, 4$ and 10, converge to the true mutual information as $N \to \infty$. Of the three values of k shown in figure 4.19 the convergence of $k = 10$ is the slowest in one dimension, due to a large portion of the nearest neighbour distances that are truncated due to being close the edge of the support for small sample sizes. There is minimal differences, however, between the convergence of $k = 1$ and $k = 4$ in one dimension. Whereas, for two dimensions $k = 1$ converges significantly faster than the other values. Although overall, the convergence is slower for all k values for two dimensions compared to one. This is not unexpected due to the known increasing bias with the number of dimensions. As observed, we note that the entropy value is dependent on k.

Any function of a empirical sample will itself be a random variable that can take on many values due to fluctuations in a sample. This gives rise to a distribution of estimates indicating the quality of the estimator. We will therefore also demonstrate the suitability of the proposed algorithm from the characteristics of the distribution of its estimates.

For this we will conduct a series of experiments for each of the standard distributions. In each experiment an ensemble of 500 mutual information estimates were calculated from i.i.d trials. A histogram of the values was constructed for $N = \{500, 1000, 5000, 10000\}$ and fitted to a normal distribution using least-squares fitting. For an estimator to be categorised as asymptotically unbiased and consistent the expected value will tend to the true value and its accuracy will

improve as the sample size increases. Therefore, as N increases the mean on the normal fit will tend to the theoretical value and the standard deviation will decrease.

Figure 4.20 shows a series of plots for independent 2D distributions of the pdfs described in table 4.2. Each plot shows the histogram of the distribution of mutual information estimates for $N = 500$, as well as the fits obtained from the histograms for all of the sample sizes. The fit parameters and the reduced χ^2 values are given in table 4.5.

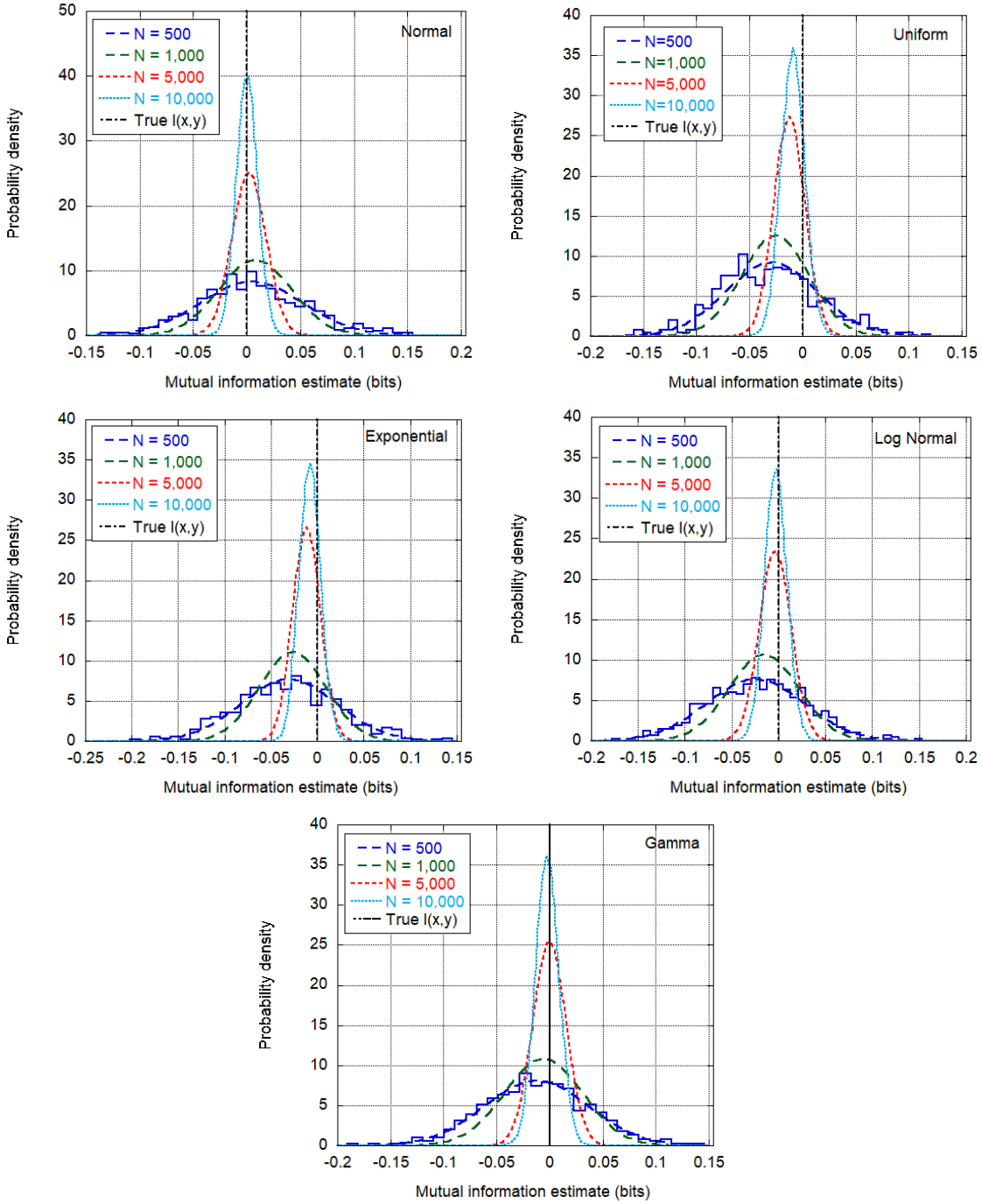

Figure 4.20: The distribution of mutual information values for different sample sizes and pdfs. Each distribution is constructed from 500 mutual information estimates calculated using the 3H-principal and algorithm W for $k = 1$ and $N_I = 50$. In each plot the black vertical line illustrates the theoretical value for the mutual information.

It is apparent from the fit parameters in table 4.5 that as sample size increases the mean tends to the true mutual information and the standard deviation decreases. This is similarly observed in figure 4.20 which shows the distribution of estimates becoming increasingly concentrated near the true value for larger sample sizes, indicating that the proposed estimator is consistent. Unsurprisingly, due to the asymptotically unbiased nature of the KL estimator, the distribution fits for $N = 500$ tend to be centered at marginally negative values compared to the true mutual information. This bias decreases at N increases, demonstrating the asymptotic unbiased properties of the proposed estimator. Therefore, algorithm W is asymptotically unbiased and consistent.

N		Normal	Uniform	Exponential	Log Normal	Gamma
500	mean	0.005 ± 0.002	-0.029 ± 0.002	-0.030 ± 0.002	-0.025 ± 0.002	-0.012 ± 0.0002
	std dev	0.048 ± 0.002	0.043 ± 0.002	0.052 ± 0.002	0.052 ± 0.002	0.049 ± 0.002
	Red χ^2	1.22	1.66	1.26	0.57	0.39
$1,000$	mean	0.008 ± 0.002	-0.026 ± 0.001	-0.026 ± 0.002	-0.016 ± 0.002	-0.006 ± 0.002
	std dev	0.034 ± 0.001	0.032 ± 0.001	0.036 ± 0.001	0.037 ± 0.001	0.037 ± 0.001
	Red χ^2	0.66	1.39	0.60	0.86	1.00
$5,000$	mean	0.0020 ± 0.0007	-0.0124 ± 0.0006	-0.0116 ± 0.0007	-0.0037 ± 0.0008	-0.0000 ± 0.0007
	std dev	0.0159 ± 0.0006	0.0146 ± 0.0005	0.0150 ± 0.0005	0.0170 ± 0.0006	0.0157 ± 0.0005
	Red χ^2	0.79	2.11	0.7205	1.19	1.76
$10,000$	mean	0.0006 ± 0.0004	-0.0086 ± 0.0005	-0.0078 ± 0.0005	-0.0025 ± 0.0005	-0.0018 ± 0.0005
	std dev	0.0099 ± 0.0003	0.0111 ± 0.0004	0.0115 ± 0.0004	0.0119 ± 0.0004	0.0111 ± 0.0004
	Red χ^2	0.76	0.82	1.18	0.70	1.37

Table 4.5: The normal least-squared fit parameters for figure 4.20 to demonstrate that algorithm W is asymptotically unbiased and consistency.

4.4 Simulations

In this section we evaluate the performance of the proposed estimator in a variety of artificial experiments where the true mutual information is known. For comparison we will also evaluate the 3H-KL estimator and the KSG estimator. For fairness of comparison the parameter k is fixed at $k = 1$ for all estimators and $N_I = 50$ throughout. For each experiment we will perform 250 i.i.d trials and consider the MSE of the mutual information estimate as a function of sample size.

Experiment 1. We will first consider the performance of our estimator for a bivariate normal distribution with $\rho = 0.6$. The normal distribution often constitutes a base line test due to its prevalence in real world systems. Both the KL and KSG estimators have repeatedly demonstrated their affinity to the normal distribution, with the KSG estimator known to empirically outperform the 3H-KL. An exemplar joint distribution is shown in the scatter plot on the left of figure 4.21 for $N = 1,000$. Each Gaussian is of zero mean and unit variance and consequently $\Delta_x \approx \Delta_y$.

The results of experiment 1 are shown on the right of figure 4.21. In this experiment our proposed estimator outperforms the other methods with a comparable smaller MSE particularly at small sample sizes. As expected the KSG estimator significantly outperforms the 3H-KL. The comparatively bad performance of the 3H-KL estimator can be attributed to bias that arises due

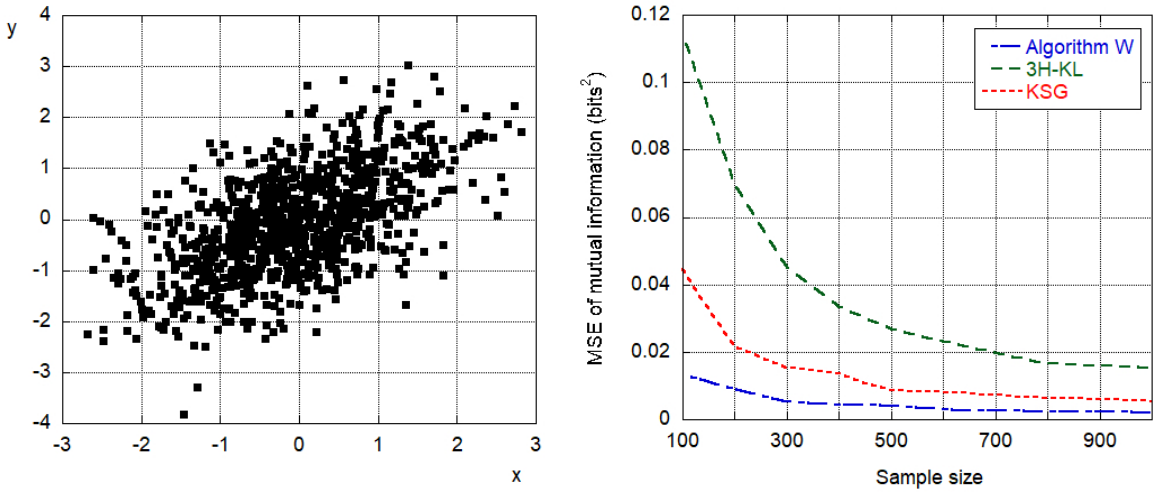

Figure 4.21: Experiment 1. Left: Scatter plot for a bivariate normal distribution with $\rho = 0.6$. Right: MSE of the mutual information as a function the sample size for 250 i.i.d trials for the corresponding scatter plot.

to the different distance scales in the joint and marginal spaces. As for a given k the nearest neighbour distances in the joint space will be larger than in the marginal spaces for the same k value. Due to this non-uniformity in the density the biases in each of the entropy estimates for the 3H-KL estimator do not cancel when considering the mutual information [12]. This was the grounds for the KSG estimator which aims to reduce this effect. It is due to the quantisation in algorithm W that reduces the scale dependence seen in the KL estimator, as the quantised marginal distributions tend to have scales of the same, or similar, orders of magnitude.

Experiment 2. Next we evaluate the performance for a case where the scales of the marginal distributions vary significantly. Consider the joint independent uniform-normal distribution in figure 4.15, where the uniform distribution has a continuous range between 0 and 1 and the normal distribution is of zero mean and $\sigma = 10$. The joint distribution is shown on the left of figure 4.22. We previously saw how the quantisation was able to account for the different marginal spaces when compared to the KL estimator for the joint entropy. We now consider the MSE of the mutual information, this time with $N_I > 1$. The 3H-KL estimator, as before, suffers from significant biases. Despite the aims of the KSG estimator to eradicate the effect the proposed estimator still outperforms the others.

Experiment 3. Variable X is a discrete Bernoulli distribution that can take the value 0 or 1. For a given value of X, Y is a continuous uniform distribution over the range $[X, X + 2)$. The true mutual information of this system can be determined analytically, with $I_{true} = -(p \log_2(p) + (1 - p) \log_2(1 - p))/2$ bits, where p is the probability of success. We choose $p = 0.25$ and a scatter plot of a sample data set is shown on the left of figure 4.23 for $N = 1,000$.

Due to the discrete variable X the KL and KSG estimator require noise to be added to the system in order to be implemented. We will use the same principal for the noise distribution as

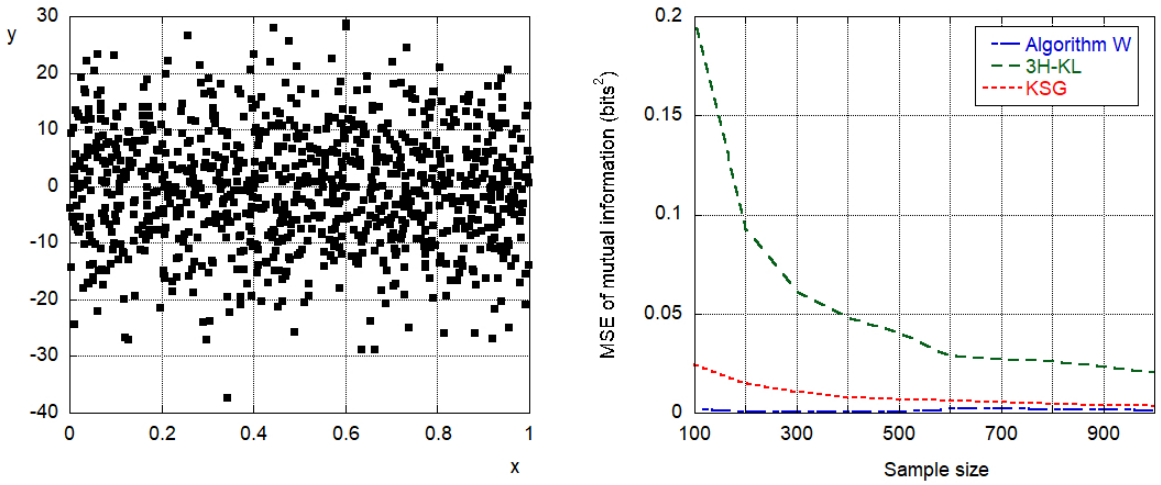

Figure 4.22: Experiment 2. Left: Scatter plot for a joint uniform-normal distribution. Right: MSE of the mutual information as a function the sample size for 250 i.i.d trials for the corresponding scatter plot.

used in our method. Therefore, we will add uniformly distributed noise to any discrete samples $\mathcal{U}[0, 1)$ in order to maintain the entropy of the system. This will be done for all discrete variables considered throughout to create the noisy KL and noisy KSG estimators.

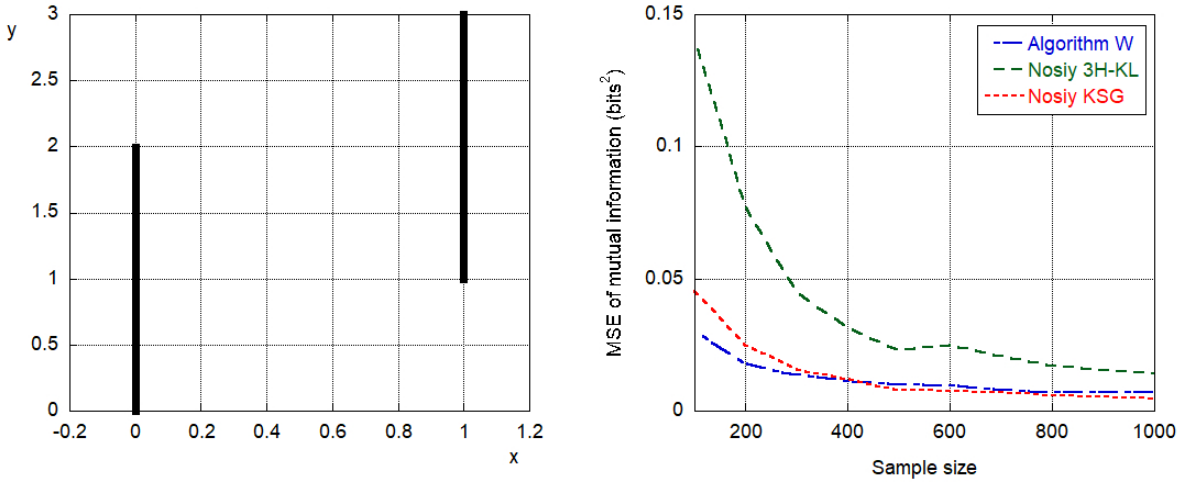

Figure 4.23: Experiment 3. Left: Scatter plot for a discrete-continuous mixed Bernoulli-Uniform distribution. Right: MSE of the mutual information as a function the sample size for 250 i.i.d trials for the corresponding scatter plot.

The addition of uniform noise allows the other estimators to converge with increasing sample size. The noisy KSG and the proposed method are comparable. For small N our estimator performs marginally better while the KSG performing marginally better for large N.

Experiment 4. Here we replicate the mixed-distribution experiment in [6], where variable Y is a standard exponential distribution $Exp(1)$, and X is a discrete "zero-inflated Poissonization" of Y, i.e. $X = Poisson(Y)$. The distribution for $N = 1,000$ is shown on the left of figure 4.24. The theoretical mutual information is $I_{true} \approx 0.4345$ bits.

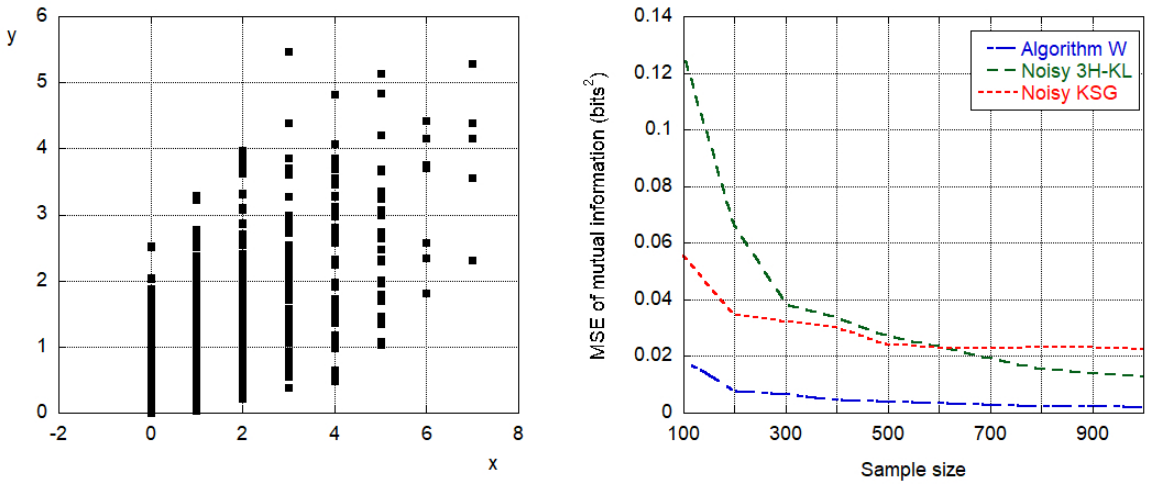

Figure 4.24: Experiment 4. Left: Scatter plot for a discrete-continuous mixed Poisson-Exponential distribution. Right: MSE of the mutual information as a function the sample size for 250 i.i.d trials for the corresponding scatter plot.

4.5 The effect of precision on k

For a continuous distribution the number of possible distinct values is infinite. However, for a random variable the precision of a measurement is limited which drastically reduces the number of distinct values in the domain and consequently reduces the entropy. This is intuitive, if you knew the result of some random variable to ± 0.1 rather than to ± 1 then the first would be more informative and the uncertainty surrounding the outcome would be less.

For nearest-neighbour estimators this is a known problem as they heavily rely on unique values in order to have non-zero nearest neighbour distances. As the precision decreases there is an increase in the size of the nearest neighbour distances, this causes an over-estimate for the entropy. As the precision decreases further the variable is effectively quantised and clusters of repeating values appear which, if k is smaller than the cluster size, reduces the number of non-zero nearest-neighbour distances. Thus, a limited precision results in a slow decline in the estimators abilities until the variable is effectively discrete and without the addition of noise is not applicable.

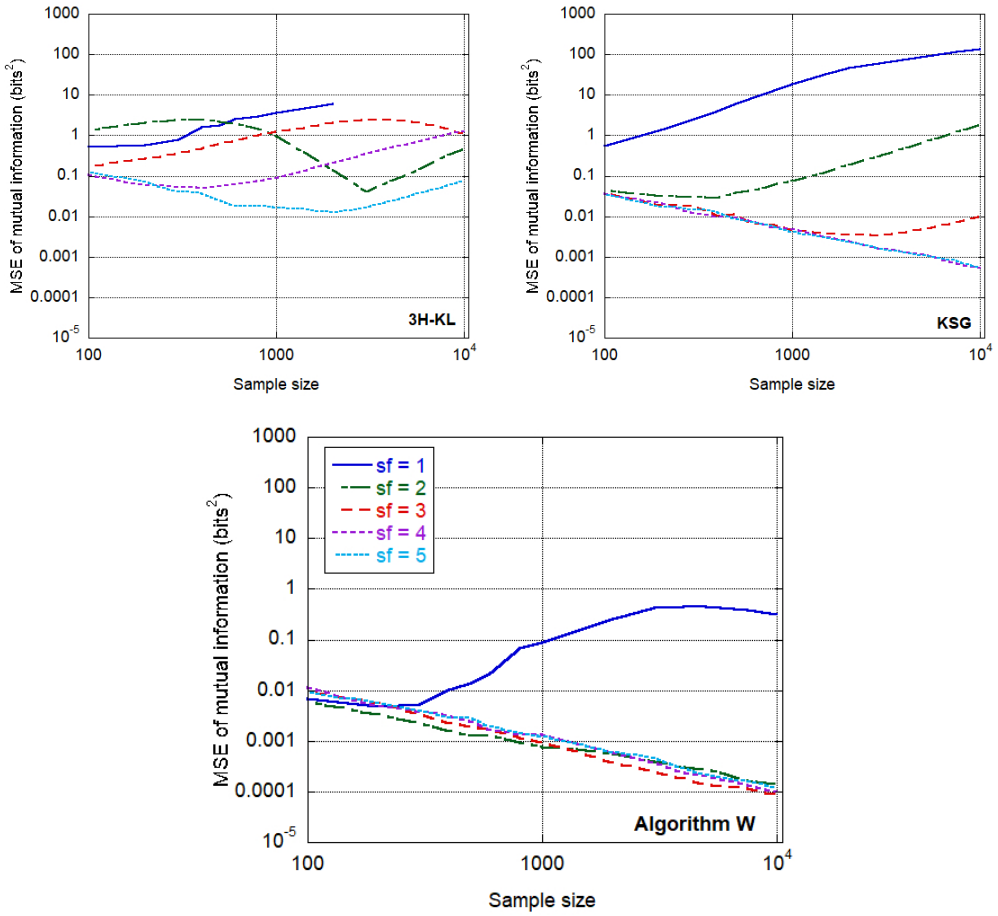

Figure 4.25: The MSE of the 3H-KL, KSG and algorithm W for mutual information estimates. Here we used the parameters $k = 1$ for all the estimators and $N_I = 50$ for algorithm W. The MSE was determined from 250 i.i.d trials of an uncorrelated bivariate normal distribution. We repeated the experiment for different numbers of significant figures.

In figure 4.25 we demonstrate this for algorithm W, the 3H-KL and KSG estimators for mutual information of a bivariate normal distribution. The precision of the samples in limited by the number of significant figures (sf), where in this experiment sf $= \{1, 2, 3, 4, 5\}$ and the MSE of the mutual information is plotted as a function of sample size.

When the precision is limited a sample size dependent bias arises. This is because for a given precision as N increases the number of duplicate instances increases and thus the number of non-zero nearest-neighbour distances decreases, degrading the estimate. The same is not observed for algorithm W for sf > 1. Although the initial quantising is based on of the result of the KL estimator, the subsequent steps reduces this nearest-neighbour problem. In the case of limited precision the binning would be of a reduced quality, however, the addition of noise and the resampling almost removes the chance of zero-valued nearest-neighbour distances which, in turn, reduces the dependence of the estimator on the number of unique values in a sample.

Increasing k such that $k >$ number of duplicates would reduce the number of zero-valued nearest-neighbour distances, improving the estimate. However, increasing k too much comes

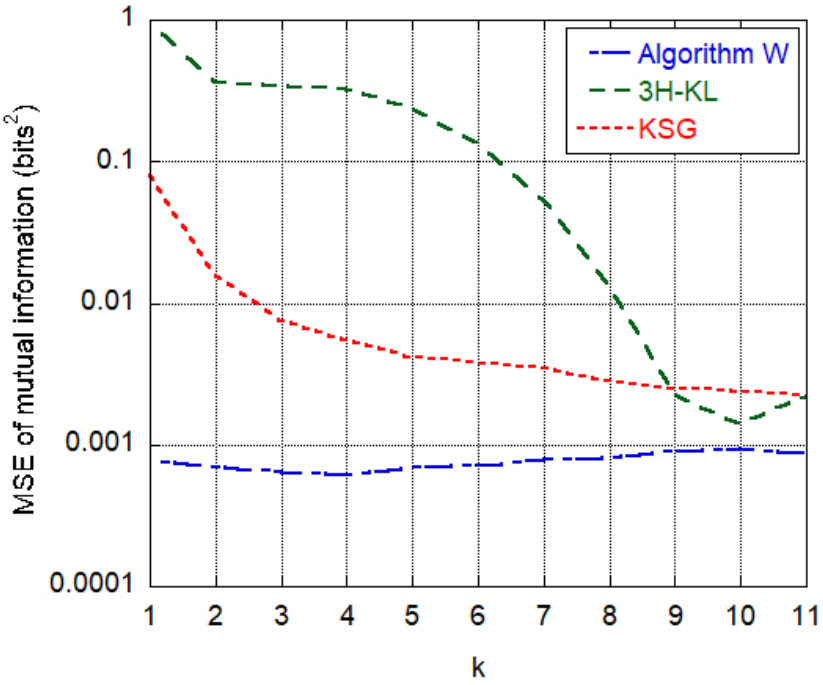

Figure 4.26: The MSE of the mutual information for the proposed algorithm, the 3H-KL estimator and the KSG estimator. The MSE was averaged over 250 i.i.d samples of a bivariate normal distribution of size $N = 1,000$ and sf $= 2$.

with the trade-off of additional biases for a reduced error. For a sample size of $N = 1,000$ with a precision of 2 the average number of duplicates for each unique value is 10 ± 1. Therefore, increasing k up to this value will improve on the estimate for the 3H-KL and KSG estimator. On the other hand, increasing k much beyond this will see the bias begin to rise due to missing fine scale structures in the data. The MSE as a function of k can be seen for the $N = 1,000$ and sf $= 2$ example in figure 4.26. For the 3H-KL and KSG estimators the MSE of the mutual information decreases as k increases up to ≈ 10. However, for algorithm W the estimate is stable, even for low values of k.

Unfortunately, there is no value of k which is optimal for all samples sizes, data precision and distributions. Unless specified otherwise we will thus use $k = 1$ for all estimators for consistency and easy comparisons.

4.6 Error Analysis

The variance of an estimator is crucial in determining the reliability of the result, as the error is necessary to know the number of relevant significant figures. However, calculating the error on an entropy estimate is not simple, see [13]. Due to the complex nature of this problem, very little work has been done to assess the sampling errors on many commonly used information estimators. However, when it is not possible to derive the errors, information about the error distribution can be drawn from repeated measurements.

This section will discuss the errors on the entropy and the mutual information estimates for algorithm W. To estimate the variance, we simulate i.i.d trials of pdfs and estimate the entropy and mutual information for each. The variance is then estimated as the sample variance. Finally, we then fit a model to these values by varying the dependent parameters, using $\sqrt{2/T}$ as the error on the variance estimate for T trials [14]. We then compare the sample variance for the 3H-KL and KSG estimators with algorithm W.

It is expected that the variance of algorithm W scales as $1/N$. However, other factors such as the choice of M, k and N_I can also affect the variability. We have previously shown that $h(g) = h(u) + H(P)$, where $h(u)$ is the differential entropy of the continuous uniform noise distribution ($h(u) = 0$ bits), and $H(P)$ is the Shannon entropy of the empirical sample. Thus, we propose the error on $h(g)$, the entropy estimate from algorithm W, is comprised of two contributions: an error due to the binning of the data ($H(P)$), which is referred to in the literature as a *quantisation error*, and an error due to the KL estimator.

$$Var[h(g)] = Var[h(u)] + Var[H(P)] \tag{4.17}$$

As the quantised data sample does not change between iterations, the $Var[H(P)]$ term is constant with regard to N_I. It can be considered an error due to residual statistical fluctuations on the histogram. Conversely, the $Var[h(u)]$ term is analogous to repeated applications of the KL estimator to N_I i.i.d samples. Thus, the first term is expected to $\approx Var[\hat{h}_{KL}(X)]/N_I$, where $Var[\hat{h}_{KL}(X)]$ is the variance on the KL estimator. The variance on the KL entropy estimator is well known and approximated by $Var[\hat{h}_{KL}(X)] = (Var[\ln(p(X))] + \Psi^1(k))/N$, where distribution dependent fluctuations arise due to sample noise in the nearest-neighbour distances. For example, for the uniform distribution $Var[\ln(p(X))] = 0$ and thus $Var[\hat{h}_{KL}(X)] = \Psi^1(k)/N$. In the case of the proposed estimator, independent of the underlying distribution, uniform continuous random numbers are added to the quantised data, resulting in uniform k-nearest-neighbour statistics for all probability distributions. Therefore, we propose that this removes the distribution dependence observed for the KL variance and gives us a $\Psi^1(k)$ dependence. Therefore, the expected form the variance on algorithm W for a entropy estimate is

$$Var[\hat{H}(X)] = \frac{1}{N}\left(\frac{A\Psi^{(1)}(k)}{N_I} + B\right) \tag{4.18}$$

Where A is constant, and B depends on the quality of the binning, i.e. the choice of M. For simplicity, we will keep $M = 2$ throughout this thesis, unless specified otherwise. Note, also, that equation 4.18 is in the units nats and needs to be divided by $\ln(2)^2$ to obtain bits.

Firstly, we want to confirm the distribution independence and the $1/N$ scaling for the variance. To do this, we estimate the variance for several one-dimensional distributions from 250 i.i.d trials. For each distribution, we estimated the entropy via algorithm W for $N_I = 50$, $k = 1$ and

$M = 2$. This is shown in figure 4.27 as a function of sample size. Thus, from this plot, we confirm the variance is indeed distribution independent and of $\mathcal{O}(N^{-1})$.

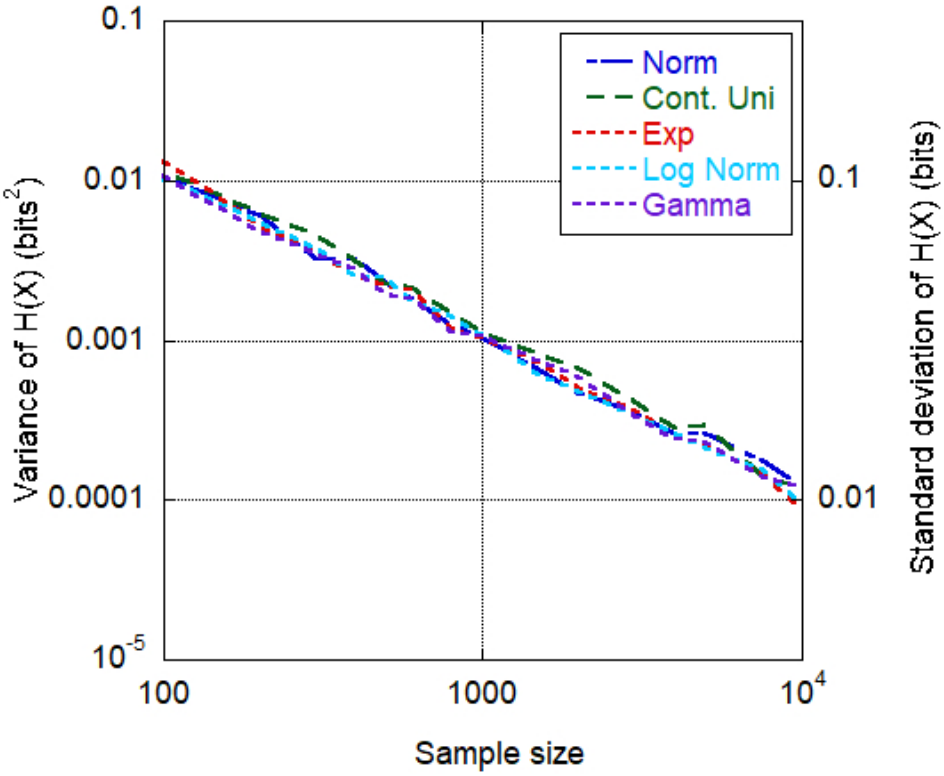

Figure 4.27: The sample variance of the entropy estimate of 250 i.i.d trials for several one-dimensional distributions on a log-log scale. On the right-hand axis we also give the standard deviation, σ, where $Var[\hat{H}] = \sigma^2$.

Next, we want to verify the dependence on k and N_I. For this we consider the variance of a one-dimensional normal distribution as a function of N_I, and different values of k. The results from this, for 250 i.i.d trials, is shown in figure 4.28. We conduct this experiment for $k = \{1, 2, 4\}$ for a sample of $N = 1,000$. For each value of k we fitted a least-squares fit according to equation 4.18. The corresponding fit parameters are shown in table 4.6. We observed that A and B are consistent for all samples tested with $A \approx 1.5$ and $B \approx 0.5$. The reduced χ^2 confirm that this is a good fit for the data. Thus, for $N = 500$, the estimation error on an entropy estimate is ≈ 0.05 bits.

k	M	N_I	A	error	B	error	χ^2_{red}
1	2.0	50	1.5	0.2	0.51	0.02	0.77
2	2.0	50	1.6	0.3	0.45	0.02	1.53
4	2.0	50	1.3	0.4	0.47	0.02	0.75

Table 4.6: The error model least-squared fit parameters for the variance on the entropy estimate in figure 4.28, using the formula in equation 4.18.

Finally, we consider the variance on the mutual information. Here we compare the sample variance as a function of N for different numbers of iterations and different values of k. This is

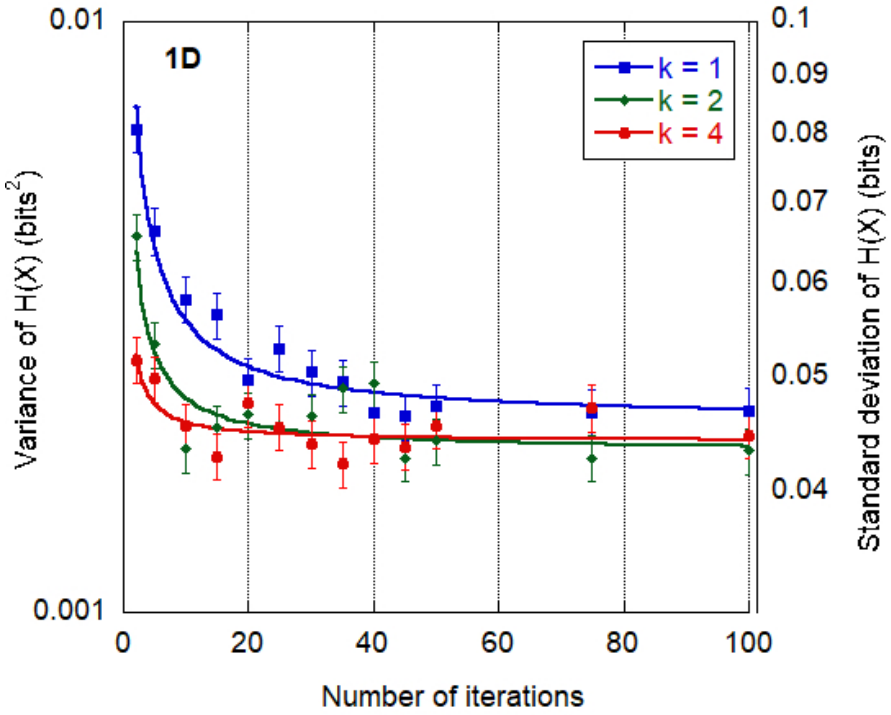

Figure 4.28: The variance of the entropy of a one-dimension normal distribution with zero mean and unit standard deviation. The variance was estimated from 250 i.i.d trials of samples with $N = 1,000$ and $k = \{1, 2, 4\}$.

shown for the uncorrelated joint normal distribution in figure 4.29 and 4.30 for N_I and k respectively. The variance for the mutual information is similarly expected to follow the same form as for the entropy variance, with different values of A and B. We fit this to the data in both figures and give the fit parameters in table 4.7. This demonstrates that the variance on the mutual information is consistent with the predicted formula for different iterations and confirms the dependence on k.

k	M	N_I	A	error	B	error	χ^2_{red}
1	2	10	2.93	0.01	0.571	0.001	1.13
1	2	25	2.94	0.01	0.496	0.001	0.83
1	2	50	2.30	0.01	0.477	0.001	2.54
1	2	-	3.0	0.3	0.50	0.02	0.52
2	2	-	3.1	0.3	0.31	0.01	0.90
4	2	-	3.6	0.4	0.194	0.007	1.30

Table 4.7: The error model least-squared fit parameters for the mutual information estimate in figures 4.29 and 4.30, using the formula in equation 4.18. The $-$ for the N_I value in some of the rows indicates that this was the parameter that variance was a function of, thus does not have a single values attributed to the fit. Note that the A and B in this table are different from those in table 4.6 for the entropy estimate error model.

As shown in [15] there is also a ρ dependence in the quantisation error when estimating mutual information. Thus the full error model for the mutual information estimate for $k = 1$ ($\Psi^1(k =$

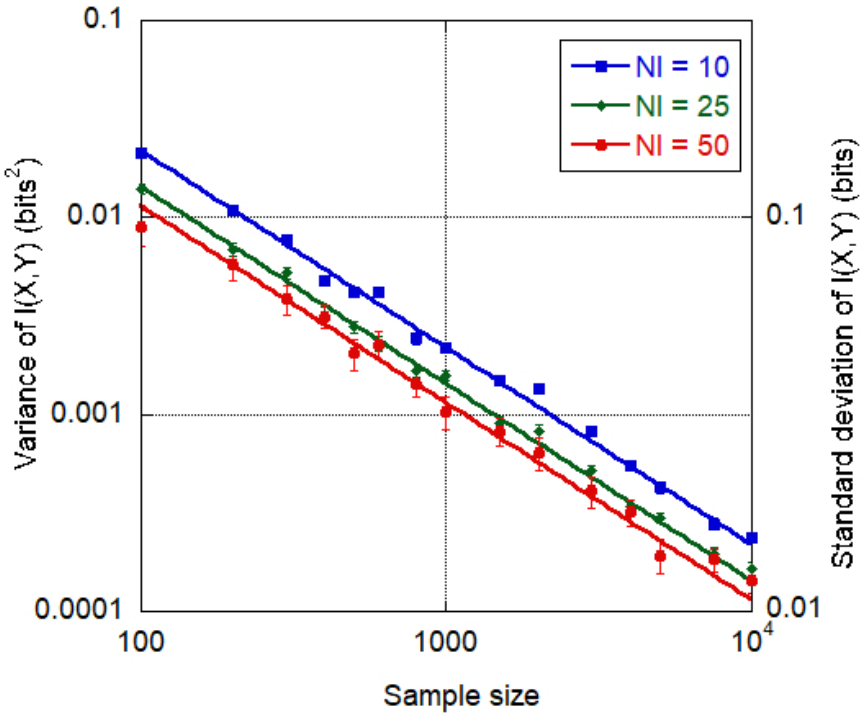

Figure 4.29: The variance of the mutual information of 250 i.i.d trials for $N_I = \{10, 25, 50\}$ and $k = 1$ for an uncorrelated bivariate normal distribution.

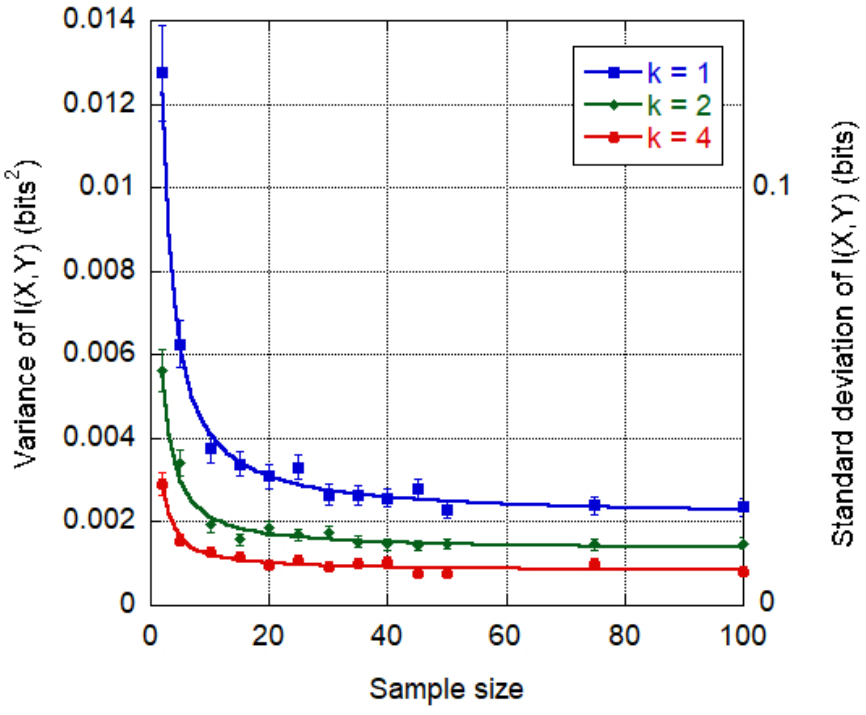

Figure 4.30: The variance of the mutual information of 250 i.i.d trials for $N_I = \{1, 2, 4\}$ and $k = 1$ for an uncorrelated bivariate normal distribution.

$1) = 1.645$) and $M = 2$ in algorithm W is expected to follow

$$Var[\hat{I}(X,Y)] = \frac{1}{N}\left(\frac{3 \times 1.645}{N_I} + \frac{1+\rho^2}{2}\right) \tag{4.19}$$

Which is given in nats. We verify this model for different sample sizes of a bivariate normal distribution, shown in figure 4.31. Here, we fit the theoretical error model in equation 4.19 to the variance for $N = 500, 1000, 5000$, with no free parameters. The reduced χ^2 were thus $1.34, 1.33$ and 3.37 for each of the sample sizes respectively. Figure 4.31 demonstrates that the ρ^2 dependence is correct. We similarly validate our model for the mutual information variance

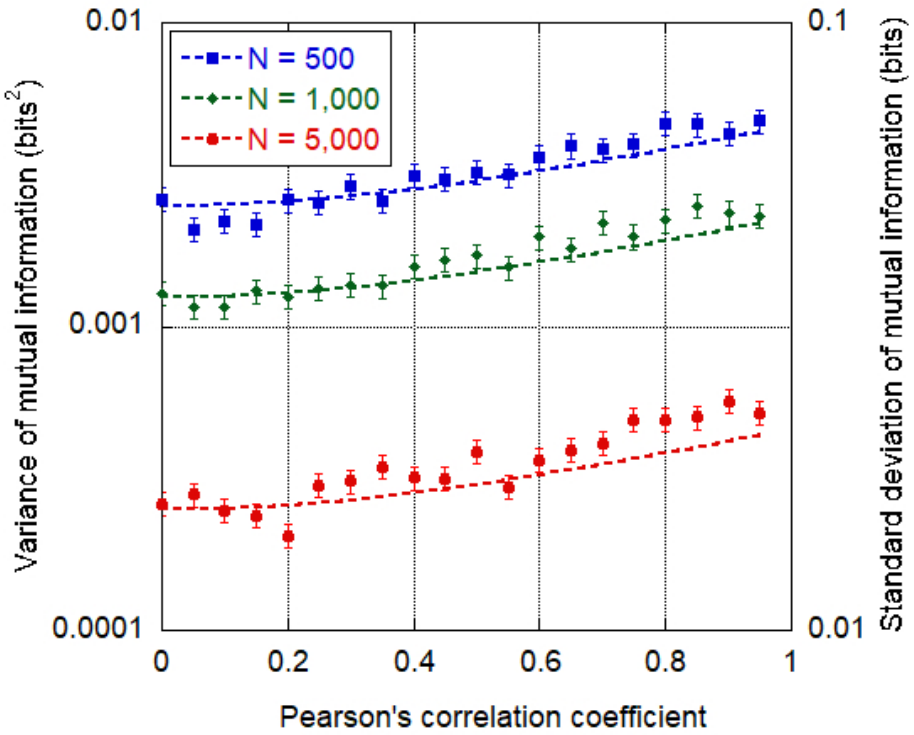

Figure 4.31: The variance on the mutual information estimate ($k = 1$ and $N_I = 50$) for 250 i.i.d trials of a bivariate normal distribution, shown as a function of the the covariance, rho.

of algorithm W via a reduced χ^2 goodness-of-fit test for bivariate normally distributed samples. A normal distribution allows the errors to be easily assessed as the true mutual information is known and related to the Pearson's correlation coefficient via $I(x, y) = -\frac{1}{2}\log_2[1 - \rho^2]$. The results from the experiment are illustrated in figure 4.32. The mutual information values are obtained from algorithm W with a single sample for $N = 500$, 1000 and 5000. The error bars were determined from equation 4.19 using the measured Pearson's correlation coefficient. The model achieved a reduced $\chi^2 = 0.88, 0.5$ and 0.59 for $N = 500, 1000$ and 5000 respectively. These values are suitably close to one, and therefore we conclude that the model is sufficient.

However, it is apparent from the fit parameters in table 4.7 for figure 4.28, that B also depends on k which has, thus far, not been discussed. This dependence can be attributed to the quality of the binning that comes from the initial k-nearest-neighbour estimate in $\Delta = 2^{\hat{h}_{KL}}/\sqrt{N}$. In reality, B will also be some function of M, which is minimum when the binning is optimal. The exact function of M requires extensive work to establish, however, a sufficient rule of thumb of the variance is $\mathcal{O}((NN_I)^{-1} + N^{-1})$. Therefore, for a sample size of $N = 1,000$, the estimation error on the mutual information will be between $0.03 - 0.06$ bits for $k = 1$. The error can be reduced

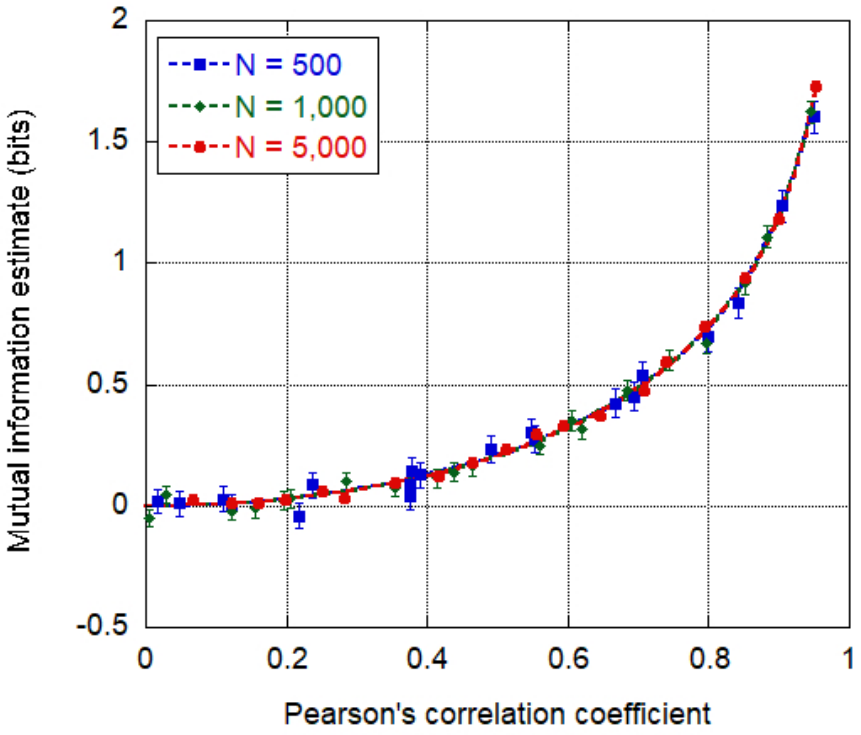

Figure 4.32: Reduced χ^2 goodness-of-fit test on variance model for mutual information conducted with bivariate normally distributed samples for $k = 1$ and $N_I = 50$.

by increasing k, although we note that as k increases, the bias on the estimate will increase. High precision measurements of mutual information are especially important in classification with a discrete class, as the class typically contains $\mathcal{O}(1)$ bit. Thus, seemingly small values of mutual information can play crucial roles in determining the class output.

Consequently, we compare the sample variance of algorithm W with the 3H-KL and KSG estimators using simulated data. For each experiment, we calculate the average mutual information for each estimator from 500 i.i.d trials. As before, the sample variance from these trials constitutes the variance estimate. This approach is necessary for the KSG estimator, as there has been little work on determining its variance despite its prevalence in the literature. The difficulty in determining the error on the KSG estimator stems from its distribution dependence, which makes it difficult to devise an expression for general probability distributions. The same is true for the KL estimator in higher dimensions. As algorithm W has no distribution dependencies the same error model can be applied to all distributions, thus an general error model is possible, as shown.

To investigate how the errors compare between distributions and estimators we calculate the sample variance for a selection of distributions for $k = 4$ for all estimators. This experiment is illustrated in figure 4.33. Each plot shows the variance on the estimators as a function of sample size for a particular distribution: 3H-KL (blue), KSG (green) and algorithm W (red).

For all distributions, the 3H-KL estimator has the largest variance out of the estimators tested.

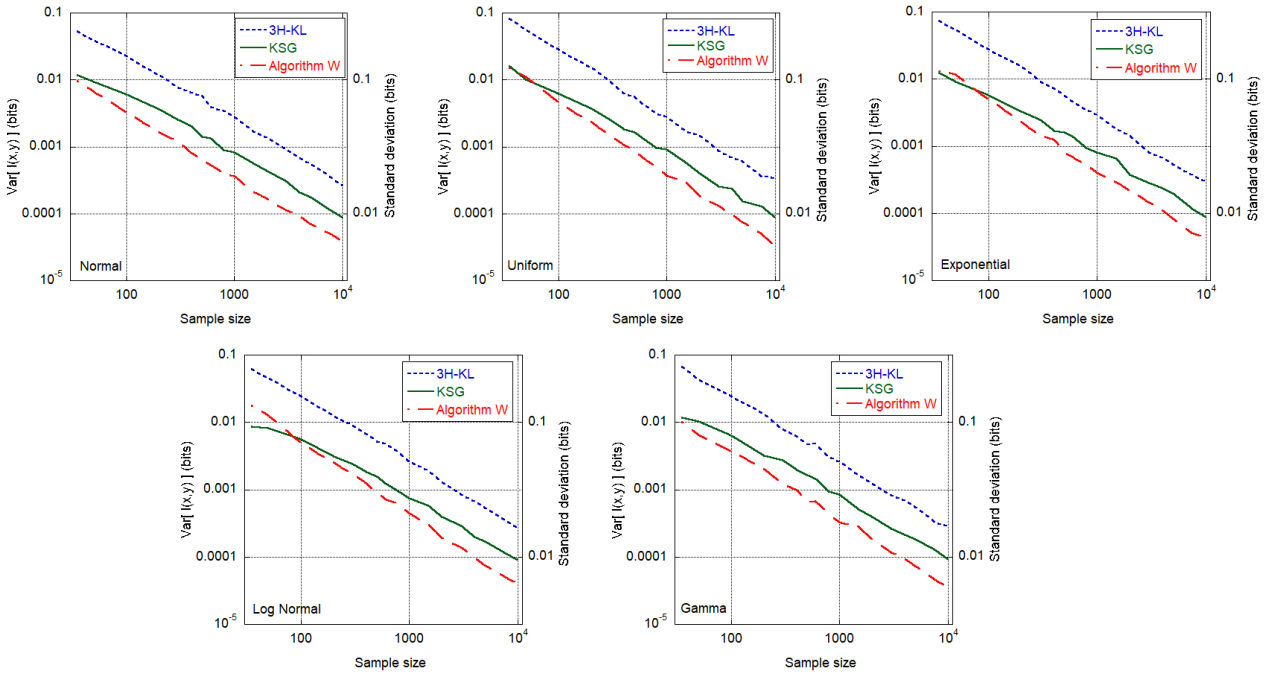

Figure 4.33: Comparison of variance for different mutual information estimators and probability distributions. The probability distributions simulated were (top row - left to right) normal $\mathcal{N}(0,1)$, uniform $\mathcal{U}[0,1]$, exponential ($\lambda = 1$), (bottom row - left to right) log normal ($\mu = 0$, $\sigma = 1$) and gamma ($\alpha = 2$, $\beta = 1$).

In addition, for small sample sizes, the variance on the KSG estimator diverges from the $1/N$. This divergence is due to the $1/N$ scaling of the variance being valid only for large N. Due to this phenomenon, the variance on the KSG estimator and our proposed estimator compete as to which one best performs for small sample sizes. However, for $N \geq 100$, algorithm W consistently demonstrates a lower variance for all distributions tested. Thus, despite the additional quantisation error, the proposed estimator has a smaller variance than the 3H-KL estimator, and is more stable than the KSG estimator.

Unfortunately, it is impossible to determine the exact error for real data, where we only have one sample. We can estimate the error on the mutual information via the equation 4.19. However, estimating the errors in this way cannot tell the user if anything unusual is occurring in the calculations. Alternatively, one can use the resampling itself to tell us about the error. The standard deviation of the ensemble of N_I mutual information estimates, denoted $\hat{\sigma}_e$, tells us about the error on one iteration. For small k, the first A term dominates the error. Thus, although we cannot estimate the error exactly from the resampling, we can use the ensemble variance to estimate that the true error will be between

$$\frac{\hat{\sigma}_e^2}{N_I} \leq \sigma_I^2 \leq \frac{\hat{\sigma}_e^2}{N_I} + \frac{1+\rho^2}{2N} \tag{4.20}$$

This approach is useful as it is based on the real data.

4.7 Discussion

In this section, we have proposed quantisation and noisy resampling techniques that, when combined, produce the foundations for a robust entropy estimation algorithm, referred to as algorithm W. In doing so, we have shown a distribution-independent form of Shannon entropy for continuous variables, $H(X) = \log(N)/M$. A unified Shannon entropy is advantageous because it avoids assumptions about the underlying distribution for quantisation. While the choice of M is complicated, experiments show a clear region of acceptable values between $2 \leq M \leq 3$. In this region, the statistical fluctuations have sufficiently dissipated, while the bias due to underbinning is not yet significant.

The technique of adding noise to an empirical sample has previously been used in various applications with benefits. However, in entropy estimation, an arbitrary noise function directly and negatively affects the estimate. We derive a formula to show that the entropy of the noise distribution is added to the entropy of the empirical sample when randomising discrete or quantised samples — illustrating the importance of choosing and accounting for a given noise distribution. Furthermore, we demonstrate the benefits of implementing an uncorrelated uniform distribution bounded between zero and one to ensure that the entropy of the noise distribution is zero and does not exceed the bin boundaries.

Repeating the randomisation technique to mimic resampling, to our knowledge, has not previously been used. In fact, entropy estimation as a whole has greatly avoided resampling. However, when combined with the quantisation, the resampling has advantageous properties that mitigate some known nearest-neighbour problems, such as the number of unique values (precision) and different marginal scales. We empirically show that algorithm W is asymptotically unbiased and consistent. We present various simulated experiments to illustrate the effectiveness of algorithm W for discrete, continuous and mixed random variables. Simultaneously, we show that the approach either outperforms or is comparable to the KSG estimators for a number of simulated cases.

We also demonstrate that the variance of algorithm W is composed of two parts: a $\mathcal{O}(N^{-1})$ term associated with the quantisation of a sample, and a term that scales as $\mathcal{O}((NN_I)^{-1})$ associated with the base KL estimator. Despite the additional quantisation error, the resampling allows us to reduce the error on an information estimate without the need for larger samples. Thus, algorithm W has a smaller variance compared to the 3H-KL and KSG estimators, allowing the result to be known to higher precision. Furthermore, we show that the expressions derived for the variances generalise to all tested distributions. This generalisation is unlike the variance for the KL and KSG estimators, which exhibit distribution dependencies, making generalised error equations complicated to formulate for these estimators. The ability to estimate an error for any pdf is therefore an important and distinctive quality of algorithm W for assessing the reliability

of the result. In addition, the small estimation errors allows us to use the mutual information as a precision tool to study variable interactions.

The increased robustness of the algorithm W estimate for increasing iterations, however, should be weighed against the computational time required for N_I repeated applications of the 3H-KL estimator. This comparative time can be dramatically reduced by using multi-threading, where each iteration is ran in parallel, at which point the computational time is dependent on the specifications of the computer used. For each available thread, the computational time of algorithm W is approximately a factor of $N_I/n_{threads}$ times that of the 3H-KL estimator. Both the 3H-KL and algorithm W scale linearly with N, unlike the KSG estimator, which has approximately a $\mathcal{O}(N^{3/2})$ dependence. Therefore, in comparison to the KSG estimator, algorithm W prevails particularly for large data samples, $> 4,000$. Whereas, in comparison to the 3H-KL estimator multi-threading becomes an indispensable commodity. Although the increased applicability and reliability of the algorithm W estimate outweighs this computational demand.

References

[1] J. Sexton and P. Laake, "Standard errors for bagged and random forest estimators," *Computational Statistics and Data Analysis*, vol. 53, no. 3, pp. 801–811, 2009, ISSN: 01679473. DOI: 10.1016/j.csda.2008.08.007. [Online]. Available: http://dx.doi.org/10.1016/j.csda.2008.08.007.

[2] C. Li, V. P. Singh, and A. K. Mishra, "Entropy theory-based criterion for hydrometric network evaluation and design: Maximum information minimum redundancy," *Water Resources Research*, vol. 48, no. 5, 2012.

[3] S. Watts, "Calculating the discrete entropy of a histogram," *Private communications*, 2017.

[4] H. Shimazaki and S. Shinomoto, "A method for selecting the bin size of a time histogram," *Neural Computation*, vol. 19, pp. 1503–1527, 2007.

[5] S. Watts, "Relationship between the cost function and M," *Private communications*, 2017.

[6] W. Gao, S. Kannan, S. Oh, and P. Viswanath, "Estimating mutual information for discrete-continuous mixtures," *Advances in Neural Information Processing Systems*, vol. 2017-Decem, no. Nips, pp. 5987–5998, 2017, ISSN: 10495258. arXiv: 1709.06212.

[7] W. Alghamdi and F. P. Calmon, "Mutual Information as a Function of Moments," *IEEE International Symposium on Information Theory - Proceedings*, vol. 2019-July, pp. 3122–3126, 2019, ISSN: 21578095. DOI: 10.1109/ISIT.2019.8849815.

[8] L. Breiman, "Bagging predictors: Technical Report No. 421," *Department of Statistics University of California*, no. 2, p. 19, 1994.

[9] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, Second Edition. Springer, New York, 2009, ISBN: 978-0-387-84857-0.

[10] C. M. Holmes and I. Nemenman, "Estimation of mutual information for real-valued data with error bars and controlled bias," *arXiv*, 2019. DOI: 10.1101/589929.

[11] H. Singh, N. Misra, V. Hnizdo, A. Fedorowicz, and E. Demchuk, "Nearest neighbor estimates of entropy," *American Journal of Mathematical and Management Sciences*, vol. 23, no. 3-4, pp. 301–321, 2003, ISSN: 01966324. DOI: 10.1080/01966324.2003.10737616.

[12] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, vol. 69, no. 6, p. 16, 2004, ISSN: 1063651X. DOI: 10.1103/PhysRevE.69.066138. arXiv: 0305641 [cond-mat].

[13] L. Paninski, "Estimation of entropy and mutual information," *Neural Computation*, vol. 15, no. 6, pp. 1191–1253, 2003, ISSN: 08997667. DOI: 10.1162/089976603321780272.

[14] S. Watts, "Confidence interval for the error - 'The error on the error'," *Private communications*, 2020.

[15] R. Moddemeijer, "On estimation of entropy and mutual information of continuous distributions," vol. 16, pp. 233–248, 1989.

Chapter 5

Information Measures and Machine Learning

We previously demonstrated the effectiveness of algorithm W for estimating the entropy and mutual information of simulated data. It is also necessary to test the algorithm's performance on real-world data to establish its applicability to real-world data analysis tasks. The advantage of simulated data is that the entropy and mutual information values are known. However, with real-world data sets, mutual information cannot be known in advance. Therefore, we must measure the algorithm's performance in alternative ways. We use several approaches to achieve this and compare the proposed algorithm's performance against the 3H-KL and KSG mutual information estimators.

Firstly, in section 5.1, we introduce the real-world data sets that will be used in experiments throughout the remainder of this thesis. We then employ the methods described below to verify the superior performance of algorithm W compared to the popular 3H-KL and KSG estimators in a real-world setting.

- For purely discrete data sets, there is no question of quantisation methods. Therefore, mutual information is easily calculated using plug-in methods of the relative frequencies. If applied correctly, noisy k-nearest-neighbour methods should obtain the same result as the plug-in method. We therefore compare mutual information estimates from algorithm W, the noisy 3H-KL and the noisy KSG estimator against the plug-in mutual information values for a purely discrete real-world data set.
- The Pearson's correlation coefficient, ρ, is a popular measure of linear dependence, which has a known relationship with the mutual information, assuming the distributions are normal. As mutual information is sensitive to nonlinear and linear dependencies, all correlations that have non-zero ρ should also have non-zero mutual mutual information. We compare the Pearson's correlation coefficient as a function of the mutual information for purely continuous and mixed cases alike.

- Finally, using the mutual information as a measure of relevancy we compare the mutual information values for variable-class interactions for several real-world data sets. Using classification algorithms as an alternative evaluation of relevancy, we compare the classification accuracy with the mutual information estimate.

Finally, in section 5.3, we consider an alternative information measure; the Kullback-Leibler divergence. The Kullback-Leibler divergence measures the difference between two distributions, making it a natural metric for many machine learning tasks. The Kullback-Leibler divergence similarly has a k-nearest-neighbour estimator, which, when used in conjunction with the key elements of algorithm W, improves the convergence and accuracy of the estimator. We discuss the limit on any Kullback-Leibler estimate, similar to what we observed for entropy in the previous chapter. In addition, we evaluate the predictive abilities of variable subsets for supervised classification problems by estimating their Kullback-Leibler divergence. Finally, we relate it to the *kappa* statistic, a measure of model performance, and thus use it as a non-parametric method for assessing the machine learning capabilities of real-world data sets.

5.1 Real-World Data

In this section, we describe four real-world data sets and one Monte Carlo sample. The data sets were chosen due to their predominance in the literature and the various data types and patterns exhibited. We primarily consider classification problems, which are reflected in the data sets chosen.

Each data set is evaluated using a classification algorithm. The algorithm implemented for each was chosen based on the individual problem and its performance for the full data set. This is because no one algorithm is applicable to all data patterns - no free lunch theorem [1]. For each data set, we give the machine learning statistics: *kappa* and the *accuracy estimate*-a measure of the percentage of instances correctly classified using 10-fold cross-validation-for the full data set. All learning algorithms are conducted using the data mining software Weka [2], an open-source collection of machine learning techniques, including feature selection and classification algorithms. A brief description of each of the classifiers used in this thesis can be found in appendix C.

The **Wisconsin breast cancer diagnostic (WBCD)** data set (https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)) dates from 1992 and consists of features collected from 569 patients - 357 of whom had benign breast tumours, and 212 with malignant cases [3]. It is a common classification data set in machine learning, which has been analysed many times over [4]–[7]. From digitised images, 10 characteristics were extracted from a cluster of cells. The features consisted of cell radius, perimeter, area, compactness, smoothness, concavity,

concave points, symmetry, fractal dimension [1] and texture. As a cluster of cells was analysed, the average value, the most extreme/worst value, and the standard error was measured for each characteristic, collating to 30 continuous variables. A binary classification variable indicated if the tumour was benign (0) or malignant (1). The simple logistic regression algorithm performed best on this data set, building a model from 10 of the variables, which achieved a $\kappa = 0.9546$ and a 97.891% accuracy.

The **Qualitative Bankruptcy** data set (https://archive.ics.uci.edu/ml/datasets/qualitative_bankruptcy) consists of 250 Korean business bank loan cases collected between $2001 - 2002$ in ref [8]. Experienced loan officers evaluated six bankruptcy risk factors and assigned risk levels for each case: negative (0), average (1) or positive (2). In addition, a binary class variable indicates if the business did (1) or did not (0) go bankrupt over the time period. All the variables in this data set are discrete, adding a challenge to the analysis. The purpose of this data set in [8] was to generate a set of classification rules to mimic the expert's decision making. Therefore, we will use the PART decision tree algorithm to mimic the decision-making process. The model achieves a $\kappa = 0.9918$ and a 99.6% accuracy.

The **Particle** data set, unlike the others, is a Monte Carlo sample for K_s production in electron-positron interactions [9]. The Monte Carlo was generated using the JETSET [10] and EvtGen [11] simulation packages. It consists of $5,000$ instances, $1,264$ signal instances, and $3,736$ background instances. Each instance consists of 8 variables and a classification label to specify signal or background. Due to its superior performance, we chose the PART decision tree algorithm for this data set, which achieved a $\kappa = 0.8975$ and a 96.18% accuracy.

The **Coronary Heart Disease (CHD)** risk-factor study in [12] was carried out in three rural areas of Western Cape, South Africa. This data set was obtained from [13]. The sample is made up of 160 cases of CHD and 302 control cases, indicated by the class variable. A value of 1 shows the individual has coronary heart disease, and a value of 0 indicates they were from the control group. The data set consists of 8 continuous variables and 1 binary variable to describe each instance; systolic blood pressure (*sbp*), cumulative tobacco consumption (*tobacco*), low-density lipoprotein (*ldl*), abdominal fat (*adiposity*), family history (*famhist*), psychosocial stress (*type A*), BMI value (*obesity*), *alcohol*, and *age*. The logistic regression model performed the best on this data set, achieving a $\kappa = 0.3359$ and a 71.2121% accuracy.

The **Prostate** data was obtained from [13], collected from a 1989 study in [14] that examined the correlation between the level of prostate specific antigens (PSA) and numerous clinical measures in 97 men about to receive a radial prostatectomy. From this data [14] aimed to predict the PSA level, a continuous variable, from 8 variables log cancer volume (*lcavol*), log prostate weight (*lweight*), *age*, log of the amount of benign prostatic hyperplasia (*lbph*), seminal vesicle invasion (*svi*), log of capsular penetration (*lcp*), Gleason score (*gleason*), and percent of Gleason scores

[1] The negative of the gradient of the perimeter and the scale on a log graph.

4 or 5 (*pgg45*).

5.2 Comparing algorithm W with the 3H-KL and KSG estimators

The 3H-KL and KSG estimators are known for performing well on normal distributions. However, real data is often more complex in shape. This set of experiments consists of comparing the performance of the proposed estimator with the 3H-KL and KSG estimators for mutual information on real data sets. Here they will be used to estimate the mutual information between pairs of variables. Where appropriate, noise is added to discrete variables for the 3H-KL and KSG estimators to make them applicable. All estimators use the first nearest-neighbour for a direct comparison

First, we consider the Bankruptcy data set, which consists of six discrete variables and one binary class variable. Purely discrete data, such as this one, does not need to be quantised, removing the problems associated with choosing a bin width. Instead, the mutual information is easily calculated from the relative frequencies using a plug-in method. Although there is still an error associated with this estimate, the value should correspond to the value obtained from continuous estimators, as mutual information is equal for discrete and continuous estimates of identical probability distributions. Therefore, we will compare the plug-in mutual information estimates with those obtained from algorithm W, the noisy 3H-KL and noisy KSG estimators. Figure 5.1 shows the comparisons of the three k-nearest-neighbour methods with the discrete plug-in method. Each plot has a least-squares linear fit to aid interpretation. All plots show a positive trend between the estimates, however, the variance around the linear trend differs significantly. It is clear from figure 5.1 that algorithm W surpasses the other estimators, with the plug-in and k-nearest-neighbour estimates almost perfectly corresponding. Not only does this demonstrate the performance of the algorithm W, but it also emphasises the importance of averaging over many randomised samples. For the noisy 3H-KL and the noisy KSG estimators, where the sample was only randomised once, there is considerable variation around the linear trend—demonstrating that without an aggregated estimate, a naive noisy k-nearest-neighbour mutual information estimator is inadequate at producing a reliable and robust estimate for discrete samples.

Note that the algorithm W mutual information estimate is consistently less than the plug-in method. This is because the Bankruptcy data set is small, with only 250 instances, and the method is asymptotically unbiased. Which, as we saw in section 4.3, tends to underestimate for small sample sizes.

Next, we consider mutual information in terms of Pearson's correlation coefficient. The Pearson's correlation coefficient, ρ, is a normalised measure of linear dependence with a value between -1 and 1. A pair of linearly correlated variables will have a non-zero Pearson's correlation coefficient and a non-zero mutual information. For two normal distributions, this falls on the

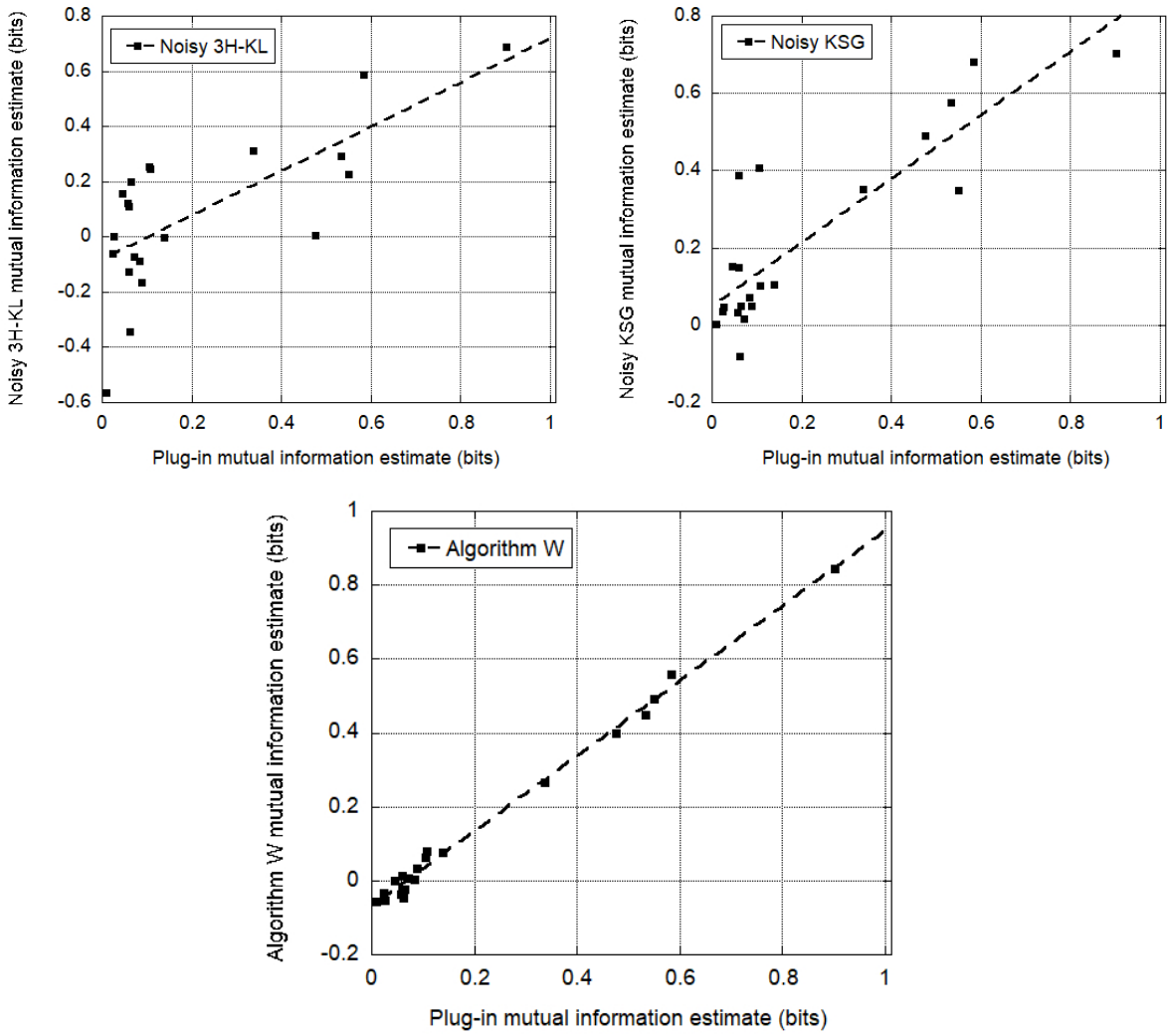

Figure 5.1: A comparison of the performance of algorithm W, the noisy 3H-KL and the noisy KSG estimators for mutual information. The methods were tested on the purely discrete Bankruptcy data set and contrasted with the estimates obtained from the plug-in approach for relative frequencies.

curve $I(x, y) = \log_2(1-\rho^2)/2$. A nonlinear correlation, on the other hand, will not be identified by Pearson's correlation coefficient but will be revealed by the non-zero mutual information. No pair should fall within the realm of small mutual information and large ρ, which would signify that the mutual information estimator missed a linear correlation.

Figure 5.2 shows the mutual information estimates versus the Pearson's correlation coefficient for all pairs of variables in the Breast Cancer data set, as estimated via algorithm W, the 3H-KL and KSG estimators. From figure 5.2, we can quickly deduce that the 3H-KL estimator performs significantly worse than the other estimators. For the 3H-KL estimator, we observe a bias that shifts the base of the normal curve to non-zero (slightly positive). This bias incorrectly suggests independent variables are correlated, although weakly. The significantly negative mutual information values observed for the 3H-KL estimator are un-interpretable and therefore not identified as significant correlations. However, many of these variable pairs have a significant Pearson's

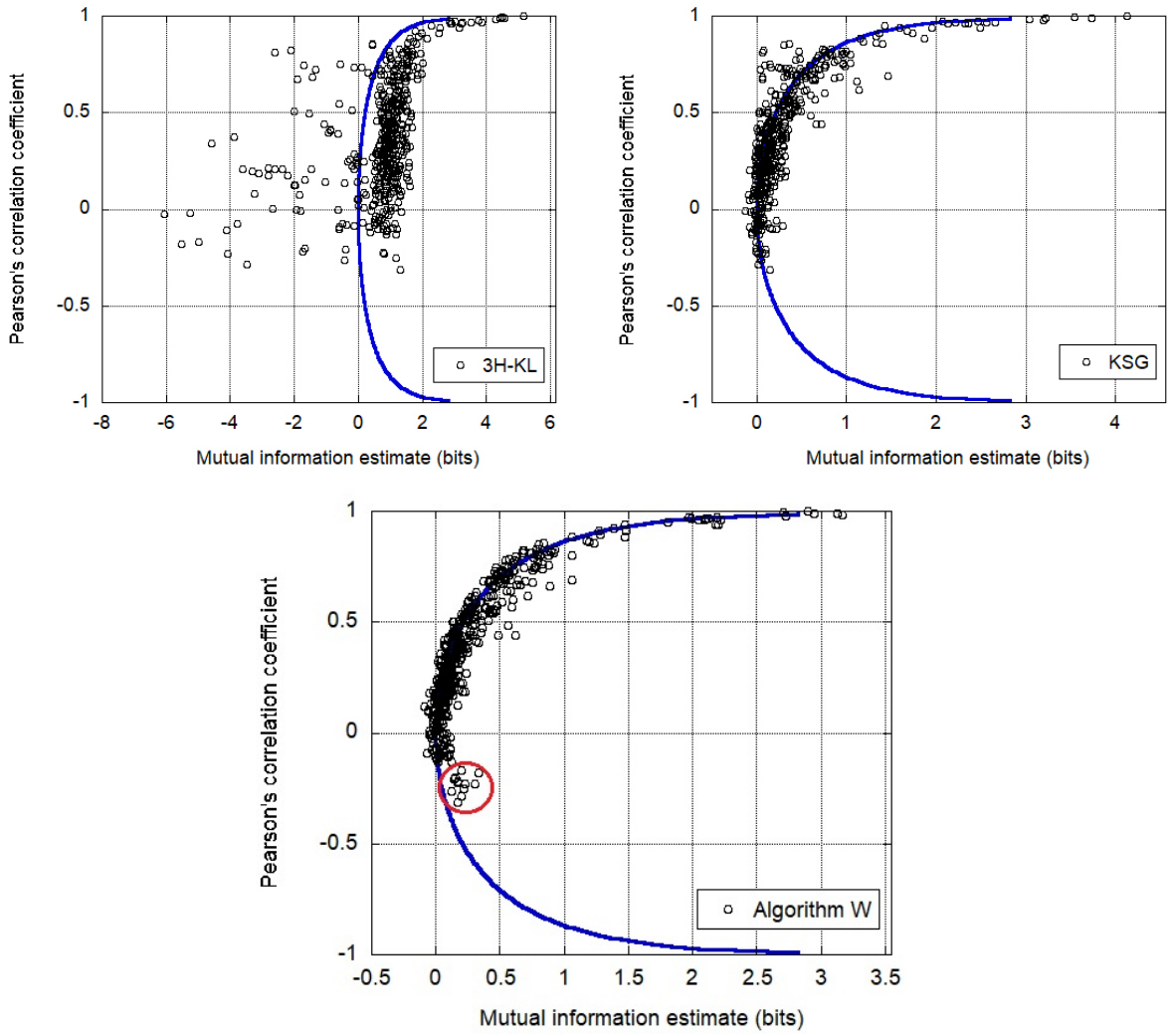

Figure 5.2: A comparison of the proposed estimator (algorithm W) with the 3H-KL and KSG mutual information estimators. The mutual information was estimated for pairwise correlations in the WBCD data set and plotted as a function of the Pearson's correlation coefficient. The blue line indicates the theoretical relationship between ρ and $I(X, Y)$ for a joint normal distribution.

correlation coefficient, indicating a linear relationship that the 3H-KL estimator has missed. These relationships correspond to the variable pairs that contain *area mean* or *area worst*. This is likely due to the difference in scales between the cell feature area and the remainder of the Breast Cancer data set variables. Similarly, the same pairs manifest themselves in the KSG plot, where we observe a cluster of points above the expected curve. Although the mutual information values for these pairs are greater than zero for the KSG estimator, they are still considerably less (≤ 0.1) than expected, given the corresponding ρ value. We observed the same effect for the 3H-KL and KSG estimators in the artificial uniform-normal distribution in section 4.4, which also had different marginal distribution scales. The proposed estimator, on the other hand, does not display these undesirable characteristics.

Nonlinear dependencies typically diverge from the normal distribution relationship between ρ and $I(X, Y)$ and are contained within the curve. The pairs that demonstrate this are the same for

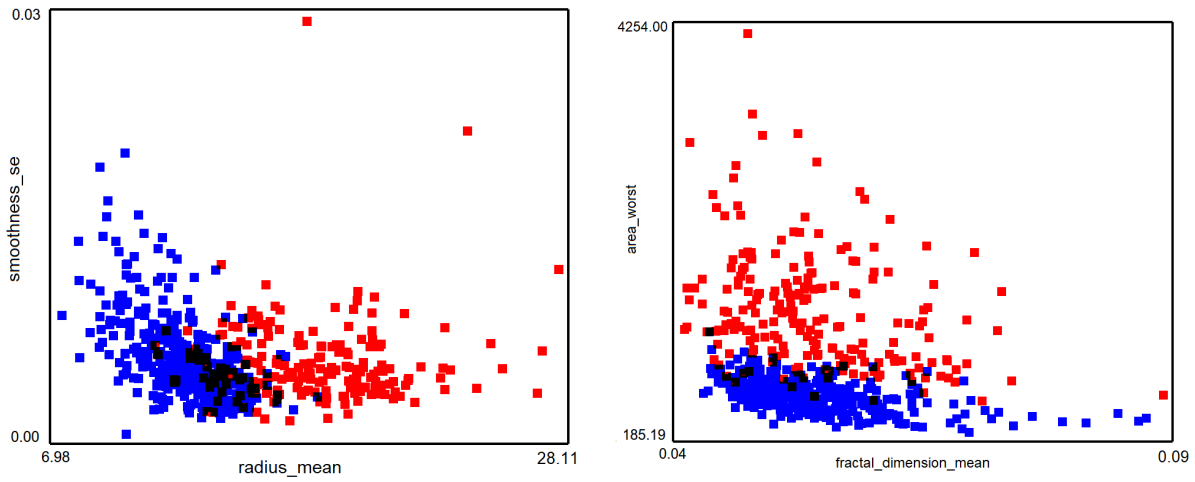

Figure 5.3: Scatter plots for the WBCD data set to illustrate nonlinear correlations which measure a non-zero Pearson's correlation coefficient. The data has been brushed to illustrate the two classes, where the black indicates an overlap.

both algorithm W and the KSG estimator, with the exception of the correlations circled in red for algorithm W. These data points can be attributed to nonlinear dependencies that have a linear component, thus Pearson's correlation coefficient is non-zero. The correlations that have been circled exhibit similar dependencies to each other. Examples of this dependence are illustrated in figure 5.3 for *radius mean* vs *smoothness se* and *fractal dimension mean* vs *area worst*. Figure 5.3 shows that the Pearson's correlation coefficient has identified a small negative linear correlation, however, from observing the correlations there is clearly a non-linear relationship with a "L" shape dependence, which explains the positioning of these points.

Finally, we evaluate the effectiveness of each of the mutual information estimators at measuring the correlation or "discriminating ability" of variables with a class. This is a classic question in supervised machine learning. Figure 5.4 compares the mutual information estimates from algorithm W with the 3H-KL and KSG estimators as a function of the accuracy estimate, a machine learning metric. The mutual information was calculated for each variable-class correlation in the WBCD, Bankruptcy, Particle and CHD data sets. Note that the Prostate data was not included as the class is continuous and therefore an accuracy estimate cannot be defined. The accuracy estimates were obtained from the the respective classification algorithm in section 5.1.

As we have seen throughout these experiments the 3H-KL estimator performs poorly on many real world distributions. This is particularly apparent for discrete variables. Although using an appropriate noise distribution does improve on the estimate the process needs to be repeated in order to obtain a reliable estimate. Both algorithm W and the noisy KSG estimator perform reasonable well for the majority of pairs.

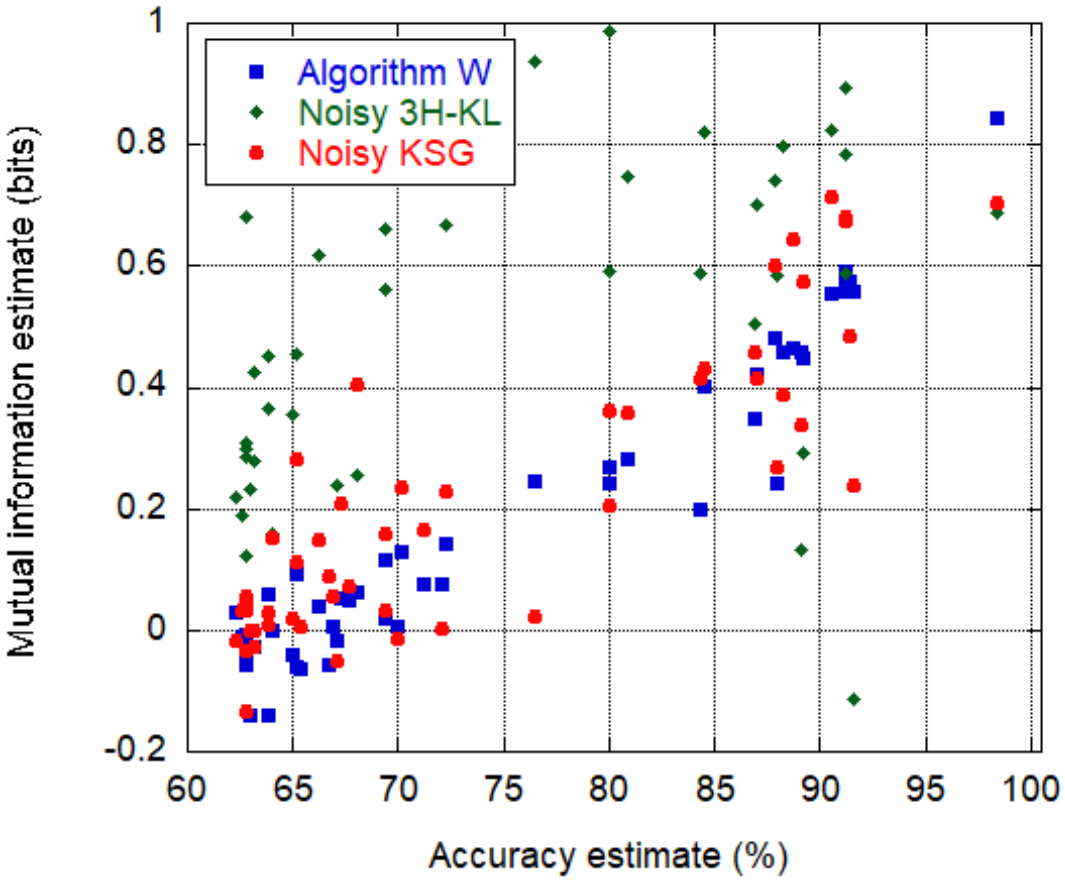

Figure 5.4: A comparison of the mutual information estimators for pairwise variable-class correlations as function of the accuracy estimate from classification learning models.

5.3 Kullback-Leibler Divergence

There are many quantities in information theory that have shown to be effective in data analysis. Among these metrics is the Kullback-Leibler divergence, also known as the "relative entropy," which measures the difference in entropy between two distributions. Consider two d-dimensional probability mass functions $P(X)$ and $Q(X)$, both functions of the random variable X. The Kullback-Leibler divergence is defined as

$$\begin{aligned} D(P|Q) &= \sum P(X) \log\left(\frac{P(X)}{Q(X)}\right) \\ &= H_X(P,Q) - H(P) \end{aligned} \tag{5.1}$$

where $H_X(P,Q)$ denotes the cross-entropy: $H_X(P,Q) = \sum P(X)\log(Q(X))$. Note that here we have used the probability distributions for $H(P) = H(P(X))$ rather than our previous notation $H(X) = H(P(X))$. This is to be explicit that the Kullback-Leibler divergence measures the difference in two distributions for the same random variable X. This extends, as expected,

to probability density functions $p(X)$ and $q(X)$:

$$\begin{aligned} d(p|q) &= \int p(X) \log \left(\frac{p(X)}{q(X)}\right) dx \\ &= h_X(p,q) - h(p) \end{aligned} \tag{5.2}$$

As before, upper and lower cases indicate the corresponding discrete and continuous measures. For both the discrete and continuous cases, the Kullback-Leibler divergence is non-negative. A zero-valued divergence, i.e. when there is no difference in entropy between the distributions, can only mean that P and Q are identical. The greater the divergence value, the more the distributions differ. Although the examples given use the probability mass functions, all the properties are analogous to probability density functions and are interchangeable with p and q, respectively.

The measure is also often referred to as the Kullback-Leibler *distance*, this terminology, however, is misleading as it is an asymmetric measure ($D(P|Q) \neq D(Q|P)$), that does not satisfy the triangle inequality. Instead, the difference in entropy between the distributions is in reference to one of the distributions. $D(P|Q)$ is more accurately described as quantifying the amount of information lost by approximating a reference distribution P, by a second "model" distribution Q. Interestingly, the mutual information is a special case of the Kullback-Leibler divergence where the reference distribution is the joint distribution $P = P(X,Y)$ and the model distribution is the assumption that the two are independent $Q = P(x)P(y)$: $I(X,Y) = D(P(x,y)|P(x)P(y))$. Thus, the mutual information gives the entropy lost by assuming that $p(x)$ and $p(y)$ are independent.

Akin to entropy the corresponding discrete and continuous measures are not necessarily mathematically equivalent. Consider the quantised continuous distributions P^Δ and Q^Δ, using the estimates $P_i = p(x_i)\Delta_P$ and $Q_i = q(x_i)\Delta_Q$ the relationship between the discrete and continuous Kullback-Leibler divergence can be derived.

$$d(P|Q) \approx D(P|Q) + \log_2 \left(\frac{\Delta_Q}{\Delta_P}\right) \tag{5.3}$$

Note that the bin width terms do not cancel out as they did for the mutual information. For the Kullback-Leibler divergence, equivalence only occurs when the two quantised distributions have the same bin width. Although, as the Kullback-Leibler divergence quantifies a change in entropy, it is not unexpected that the conversion between discrete and continuous distributions follows that of the entropy rather than the mutual information. However, if the two distributions were binned separately the relative position of each distribution would need to be preserved in order to correctly calculate the "distance" between them. Therefore, in practise $d(P|Q) = D(P|Q)$.

Quantifying the amount by which two distributions differ is of particular interest to statisticians in a number of data analysis tasks. The Kullback-Leibler divergence has proven to be effective

and popular choice in replacing standard statistical methods as a similarity measure [15]. This lends itself to a variety of uses in classification tasks [16] as well as clustering problems between unknown distributions [17], [18]. There are also a number of publications showing the effectiveness of the Kullback-Leibler divergence as a cost function for model selection [19]–[22]. More recently, in [23] the Kullback-Leibler divergence was used to answer the question "how much data is enough?" for natural driving data. They measured the divergence of distributions for increasing sample sizes to see if adding more data changed the model. [23] used the Kullback-Leibler divergence to compare the distributions for different sample sizes. Taking a value of zero to indicate that additional data did not provide more information about the density function and a large value to indicates that more data improved the density estimate.

The Kullback-Leibler divergence can similarly be used for quantifying the discriminating abilities of a variable or subset of variables for the classic two-class problem. By measuring how similar the two d-dimensional class distributions are we can quantify how easy it is to distinguish between the classes using that subset of variables. A large Kullback-Leibler divergence value means that the classes have different distributions in these variables, whereas a small value means that the distributions are indistinguishable and therefore have a low discriminating power. This measure is different from the mutual information. Rather than comparing the correlation between variables the Kullback-Leibler divergence compares two d-dimensional distributions. The mutual information is in fact a special case of the Kullback-Leibler divergence where the model distribution is the assumption that the two variable are independent. This is a useful measure when the sample has two classes, however, the Kullback-Leibler divergence does not expand to multiple classes. We could compare the discriminating power for each pair of classes present. For example, for a data set with three or more classes the number of comparisons required would be $C!/2(C-2)!$, where C is the number of classes. However, this quickly becomes complicated.

Despite the number of uses of the Kullback-Leibler divergence its asymmetry is problematic. When comparing distributions it is desirable to treat each distribution equally. This is particularly so for classification problems. [24] uses a symmetric Kullback-Leibler divergence consisting of $d(p,q) = d(p|q) + d(q|p)$, also known as the *J*-divergence. However, its relationship to the classifiers performance is more tenuous [25]. Instead [25] suggests a symmetric *resistor-average* Kullback-Leibler measure defined as:

$$\frac{1}{R(p,q)} = \frac{1}{D(p|q)} + \frac{1}{D(q|p)} \tag{5.4}$$

This resistor-average metric is better suited for classification as it was found to be less arbitrary than other types of means. When used in the context of evaluating variables predictive power of a class we refer to the quantity $R(p,q)$ as the *Class Distance Resistance* (CDR).

Much like entropy, estimating the Kullback-Leibler divergence is fraught with difficulties. Similar approaches have been employed to derive an estimator. In [26], an adaptive partitioning

estimator is proposed based on equiquantitised relative frequencies. The accuracy of which deteriorates with sample size. More recently, a handful of k-nearest-neighbour estimators were developed [27]–[29]. In [27], Wang *et al.* proposes a first nearest-neighbour estimator for the Kullback-Leibler divergence, which is extended to k-nearest-neighbour in [28]. Let $\{x_1, ..., x_N\}^d$ be N d-dimensional i.i.d samples drawn from the probability distribution p and similarly let $\{y_1, ..., y_M\}^d$ be M d-dimensional i.i.d samples drawn from the probability distribution q. For the Kozachenko-Leonenko entropy estimator, the nearest-neighbour distances provide a local view of the density surrounding a test point. The Kullback-Leibler divergence, however, compares the densities of the two distributions. Wang *et al.* achieves this by using the samples from one of the distributions as the test points. The k-nearest-neighbour distances of the test points are found for both distributions and the local densities compared. The L^2 distance of a d-dimensional test point x_i^d to its k-nearest-neighbour in $\{x_j^d\}_{j\neq i}$ is defined as

$$\lambda_{k,i,d}^{pp} = \text{Min}||x_i^d - x_j^d||_2 j \neq i \tag{5.5}$$

where $||\dots||_2$ denotes the Euclidean L_2 norm. For example, in three dimensions the k-nearest-neighbour distance from the test point $\{x_i^1, x_i^2, x_i^3\}$ is

$$\lambda_{k,i,3}^{pp} = \text{Min}\left[\sqrt{(x_i^1 - x_j^1)^2 + (x_i^2 - x_j^2)^2 + (x_i^3 - x_j^3)^2}\right] \tag{5.6}$$

Then $\lambda_{k,i,d}^{pq}$ denotes the distance of the same test point x_i^d, in the reference distribution, to its k-nearest-neighbour in the model distribution, $\{y_j^d\}$:

$$\lambda_{k,i,d}^{pq} = \text{Min}||x_i^d - y_j^d||_2 \tag{5.7}$$

The Kullback-Leibler divergence is then estimated via the following formula:

$$d(p|q) = \frac{d}{n}\sum_{i=1}^{N}\log_2\left(\frac{\lambda_{k,i,d}^{pq}}{\lambda_{k,i,d}^{pp}}\right) + \log_2\left(\frac{M}{N-1}\right) \tag{5.8}$$

where the second term is a correction term for unbalanced classes. They prove that the estimator is asymptotically unbiased and mean-square consistent. Almost sure convergence was later proven in [30] for fixed k. The convergence rate for a number of k values was tested and found that the convergence was fastest for $k = 1$ at the expense of a higher variance. Estimating the Kullback-Leibler divergence via k-nearest-neighbour distances has the same associated problems as the k-nearest-neighbour estimators for entropy and mutual information.

We can use the same key principals implemented in algorithm W, of quantising and randomising the variables, to calculate the Kullback-Leibler divergence. We simply substitute the entropy estimator with equation 5.8. Doing so drastically improves the convergence rate of the Kullback-Leibler estimator, as seen in figure 5.5. Where equation 5.8 was applied to a single sample

consisting of two balanced classes each distributed normally with $\sigma = 1$ and $\mu = 0$, which has a theoretical divergence of zero. This synthetic experiment was repeated for algorithm W, with $N_I = 50$. It was observed that algorithm W converges much faster in comparison to the application of equation 5.8 to the raw data.

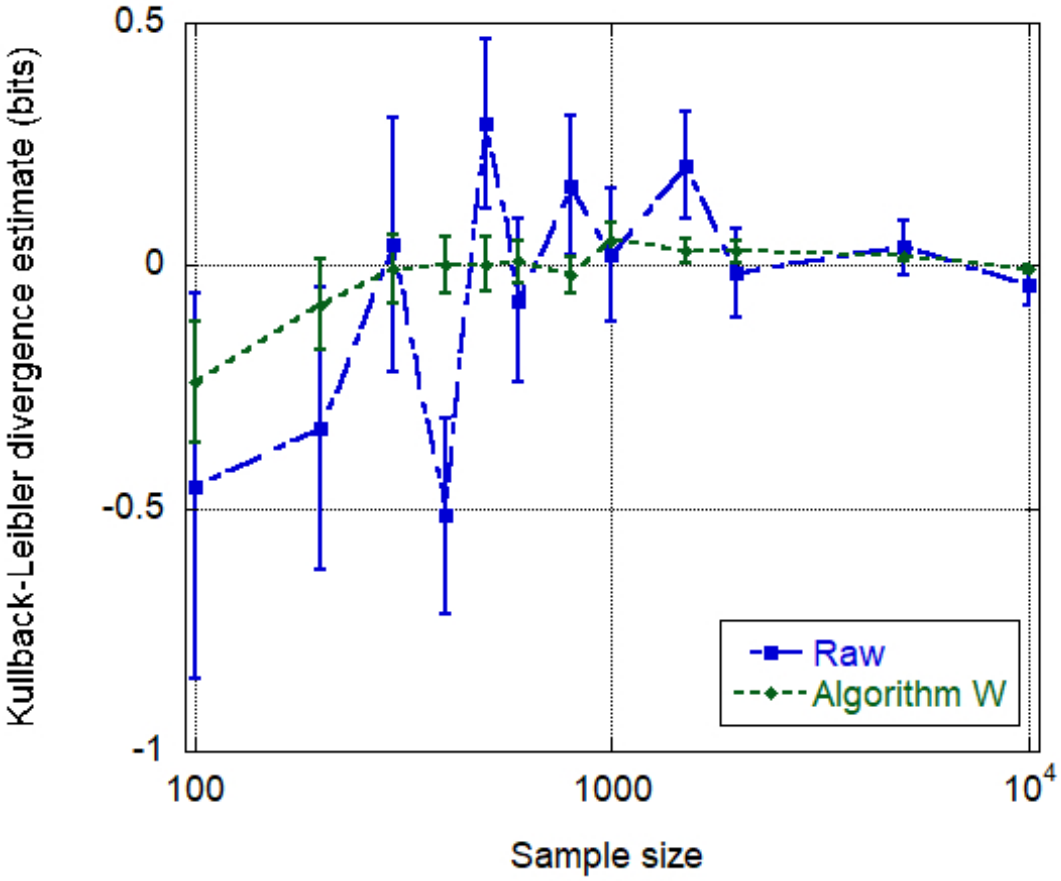

Figure 5.5: Comparison of convergence rates of algorithm W, with $N_I = 50$, and the application of equation 5.8 to a singular raw data sample. Both methods used the first nearest-neighbour and the error bars indicate one standard deviation when repeated on 250 independent trials.

As the mutual information is a special case of the Kullback-Leibler divergence, it is expected that the error model is similar in form as the one determined for the mutual information in section 4.6. For example, for $N = 1,000$ we expect an estimation error of $\mathcal{O}((2N \ln(2)^2)^{-1}) \approx 0.03$ bits, this is consistent with the estimation error obtained from the i.i.d trials. However, further work is needed to to confirm the error model.

As two distributions become increasingly separable, then theoretically, the Kullback-Leibler divergence would increase indefinitely. This is comparable to the conclusion that the theoretical Shannon entropy of a continuous distribution is infinite. However, the information content of a finite sample is limited to $\leq \log_2(N)$ bits. Therefore the loss of entropy defined by the Kullback-Leibler divergence is also limited. To demonstrate this, consider two normally distributed samples with the same standard deviation. The theoretical Kullback-Leibler divergence for two normal distributions with the same σ is $\frac{1}{2\sigma^2}(\mu_1 - \mu_2)^2$, in nats, where μ_i refers to the mean

of each of the distributions. If $\mu_1 = \mu_2 = 0$, then Kullback-Leibler divergence would be zero and the two distributions would be identical. Suppose we translate one of the normal distributions along the x-axis so that the means are offset. In that case, it becomes increasingly easier to discriminate between the two distributions, i.e. the Kullback-Leibler divergence increases. We illustrate the idea of offset normal distributions in figure 5.6. However, we know that a finite data sample has a maximum entropy. Thus, a limited information content simultaneously limits the Kullback-Leibler divergence for a finite sample. In figure 5.7 we simulate the aforementioned case of two increasingly offset normal distributions for $N = 50, 500, 5000$. We observe that the Kullback-Leibler divergence estimate saturates for a value that increases with sample size. To explain this, we point to the Chernoff-Stein lemma, which states that the probability of an error is $2^{-d(p,q)}$, which assumes an infinitely large sample [31]. If N is finite, on the other hand, the probability of an error is bounded by $1/N$. Thus, the maximum Kullback-Leibler divergence for a finite sample is $d(p|q) \leq \log_2(N)$ bits. This value is indicated in figure 5.7 by the horizontal dashed lines, the colour of which corresponds to the colour of the data points for that sample size. Recently, this finite limit was also shown in [32].

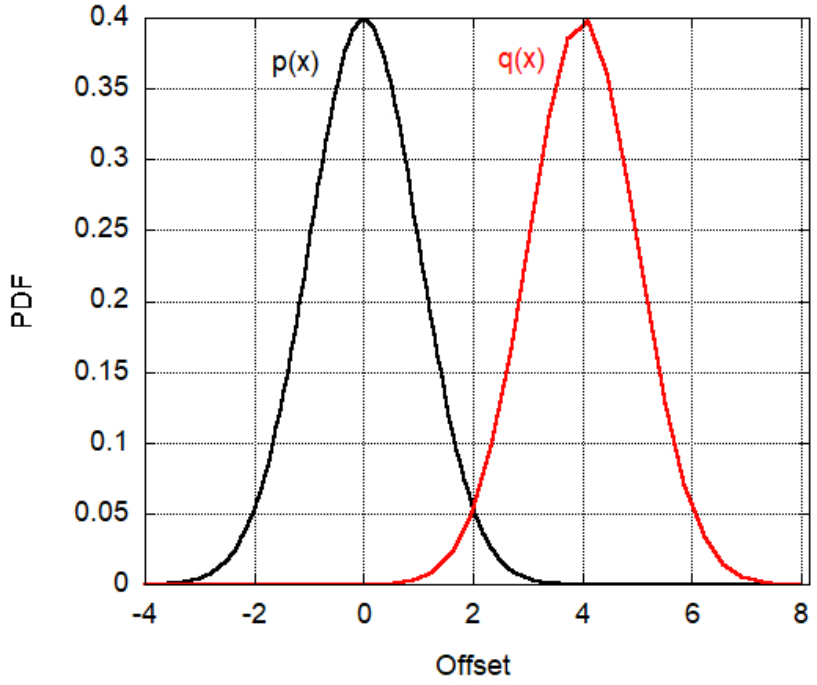

Figure 5.6: An schematic of an offset normal distribution. A reference pdf $p(X)$ with zero mean and σ variance is shown in black. A second identically distributed pdf, $q(X)$, in red, is centered at $\mu = 4\sigma$. The distance between the distribution means is referred to as the *offset* and is given as a multiple of σ. Here $q(X)$ is offset from $p(x)$ by 4.

This saturation for limited data has interesting consequences for data analysis. In machine learning more data samples improves the statistics and the performance of models. In essence, this is what the limit on the Kullback-Leibler divergence refers to. Two distributions in a finite sample may be separable with infinite data, however if the data is insufficient the information is simply

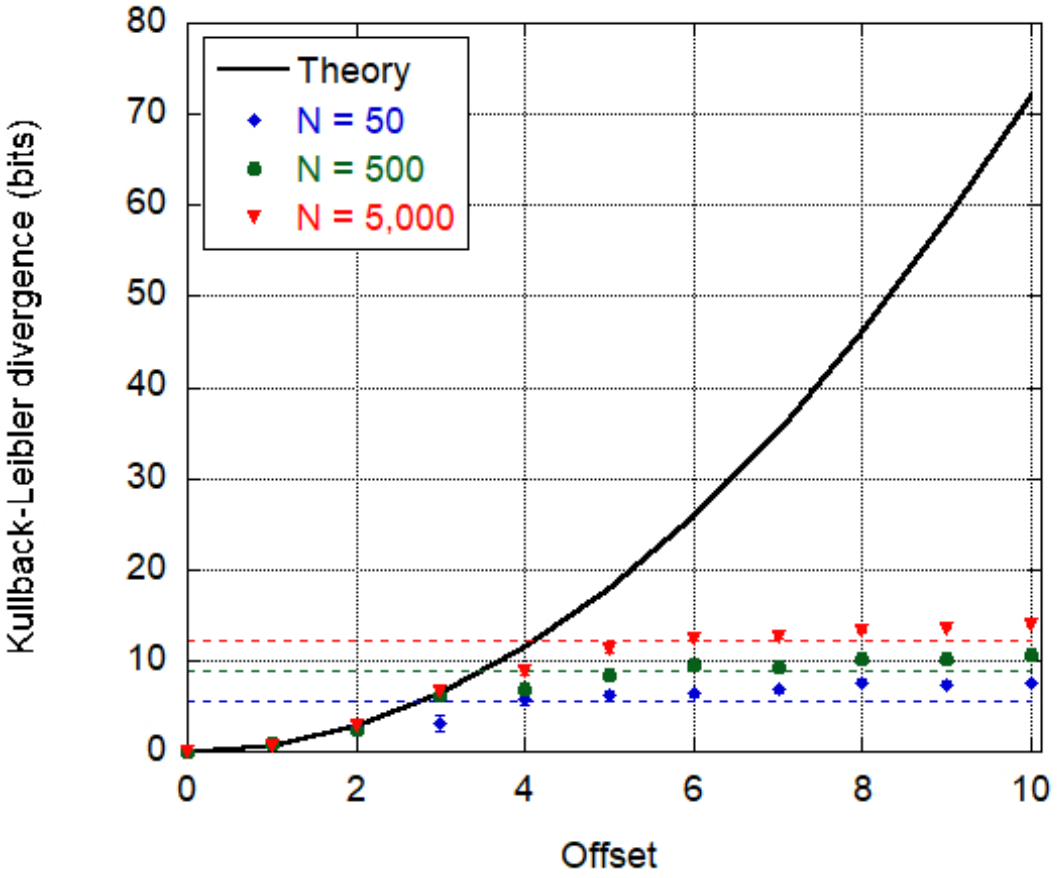

Figure 5.7: Standard deviation for each normal distribution was one, the offset refers to the number of standard deviations the two distributions have been translated along the x-axis ie an offset of 2 means the mean of each distribution is 2σ separation. $N_I = 50, 250$ Trials and $k = 1$. The horizontal lines illustrate $\log_2(N)$ the fundamental limit on any Kullback-Leibler divergence estimate.

not attainable. [23] asked "How much data is enough?", but we can flip this on its head and ask "How well will this data do?". For this we must first discuss the common machine learning statistic *kappa*.

5.4 Cohen's *kappa*

The *kappa* statistic was first introduced by Cohen in [33] and is a common measure used in model performance. Cohen's *kappa* measures how well the predictive output of a classifier agrees with the true class labels. It defines the measure of agreement correcting for the possibility that the classifiers match by chance. The chance of agreement depends on the percentages of correct labels in each class and it reduces as the number of classes increases. For a data sample with C classes, where in class i there are n_i instances, let m_i be the number of instances that the classifier assigns the label i and m_{Ti} be the number of instances that are correctly classified as class i, where "T" denotes "*True*". Similarly, m_{Fi} is the number of instances that are incorrectly classified as class i, where "F" denotes "*False*". This is visualised for the two-class case in

table D.1. Thus, the observed agreement is the probability that any output label assigned by the

	class 1	class 2	total
class 1	m_{T1}	m_{F1}	m_1
class 2	m_{F2}	m_{T2}	m_2
total	n_1	n_2	N

Table 5.1: A confusion matrix, used to describe the performance for a classifier, is shown to visualise the notation for correctly and incorrectly labelled instances for a two-class system.

classifier is correct:

$$P_o = \frac{\sum_{i=1}^{C} m_{Ti}}{N} \tag{5.9}$$

When $P_o = 1$ all the instances have been correctly labelled and when $P_o = 0$ all the instances are labelled incorrectly. Then, let P_e be the expected agreement defined by the probability that an instance was classified correctly simply by chance

$$P_e = \sum_{i=1}^{C} \left(\frac{n_i}{N} \frac{m_i}{N} \right) \tag{5.10}$$

The *kappa* statistic is thus defined as

$$\kappa = \frac{P_o - P_e}{1 - P_e} \tag{5.11}$$

A *kappa* value of zero means that the classifier achieves no better than randomly assigning the class labels and a *kappa* value of one means the classifier perfectly predicts the class for every instance. Cohen's *kappa* is always less than or equal to one. It can be negative, however, this indicates that the classifier is useless [34], [35].

kappa is particularly useful for multi-class classification problems or imbalanced classes as it captures information about the model that other measures such as the accuracy do not. The *kappa* statistic is larger for solutions which perform well for all classes. For example, consider a data sample with 200 instances and two classes where class 1 has 150 instances and class 2 has 50 instances. It may be the case that classifying all of the instances as class 1 achieves the model with the highest accuracy estimate. However, the solution is myopic and ultimately pointless. *kappa* attempts to avoid such a solution, instead preferring models which achieve balanced accuracy in each of the classes. Key to the *kappa* statistics is the confusion matrix. A confusion matrix is commonly used in supervised machine learning to summarise the performance of a classification algorithm. For the traditional two-class case with the confusion matrix

$$\begin{bmatrix} m_{T1} & m_{F1} \\ m_{F2} & m_{T2} \end{bmatrix} \tag{5.12}$$

kappa can be calculated via

$$\kappa = \frac{2(m_{T1}m_{T2} - m_{F2}m_{F1})}{(m_{T1} + m_{F1})(m_{F1} + m_{T2}) + (m_{T1} + m_{F2})(m_{F2} + m_{T2})} \tag{5.13}$$

The larger the value of *kappa* the more reliable the classification model. Therefore, statisticians aim to find the classification algorithm that maximises *kappa*. However, the quality of the data will limit the performance of any algorithm. As the accuracy estimate and *kappa* usually have the same trend, "bad" quality data will have a limited accuracy, and thus limited *kappa*, that it can achieve [34].

5.5 The Kullback-Leibler divergence and Cohen's *kappa*

As observed in figure 5.7 the Kullback-Leibler estimate plateaus when the data is no longer sufficient to increase the discriminating power of the sample. This is a fundamental limit of a finite data sample and alludes to the quality of the data as a predictive tool. The exact value of *kappa* is dependent on the suitability of the learning algorithm implemented, however, the Kullback-Leibler divergence can be used to estimate the expected *kappa* for a given variable subset. Define

$$d_{12} = d(p_1|p_2) \text{ and } d_{21} = d(p_2|p_1) \tag{5.14}$$

where 1 and 2 refer to the probability distributions corresponding to classes 1 and 2. After some algebra the *kappa* statistic for the classic two-class case can be estimated via

$$\kappa = \frac{2n_1(1 - 2^{-d_{12}}) + 2n_2(1 - 2^{-d_{21}})}{n_1(2 + 2^{-d_{21}} - 2^{-d_{12}}) + n_2(2 + 2^{-d_{12}} - 2^{-d_{21}})} \tag{5.15}$$

The proof of equation 5.15 can be found in appendix D using the Chernoff-Stein lemma [31]. Note the dependence of *kappa* on n_1 and n_2. This characteristic is the foundation to the case that *kappa* is only useful when the classes are balanced, $n_1 = n_2$ [35]. This dependence disappears if $d_{12} = d_{21}$.

There are some special cases which simplify equation 5.15.

- If the classes are balanced $n_1 = n_2 = N/2$

$$\kappa = 1 - \frac{1}{2}(2^{-d_{12}} + 2^{-d_{21}}) \tag{5.16}$$

- If $d_{12} = d_{21} = d$

$$\kappa = 1 - 2^{-d} \tag{5.17}$$

The second approximation gives the *kappa* statistic, a measure of success, as one minus the

Chernoff-Stein lemma for the probability of an error [31]. This approximation is particularly useful as it is often the case that $d_{12} \approx d_{21}$.

The benefits of being able to estimate the expected *kappa* directly from the data is two fold. Firstly, the CDR values can be used for feature selection by evaluating the suitability of variable subsets. Secondly, if the data set is limited in its abilities to discriminate then this will be identified. Preventing the need to endlessly search learning algorithms for a negligible improvements to the model. A rule-of-thumb used to indicate what a *kappa* value would mean for a classification model is given in table 5.2, along with the corresponding CDR values.

CDR (bits)	*kappa*	Agreement
≤ 0.32	< 0.2	Poor
$0.33 - 0.74$	$0.21 - 0.40$	Fair
$0.75 - 1.32$	$0.41 - 0.60$	Moderate
$1.33 - 2.32$	$0.61 - 0.80$	Good
≥ 2.33	$0.81 - 1.00$	Very Good

Table 5.2: An interpretation of the success of a machine learning application using the *kappa* statistic where the last two columns are taken from [35] and the CDR is approximated as CDR $= -\log_2(1-\kappa)$.

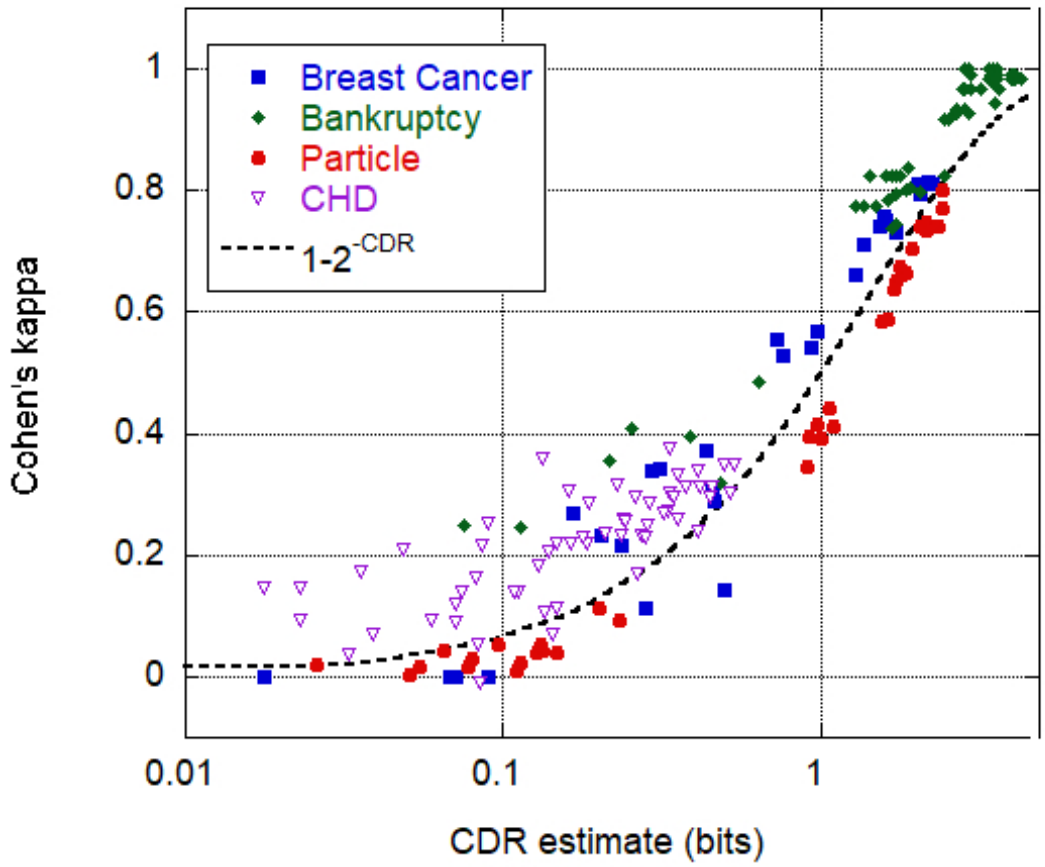

Figure 5.8: Cohen's *kappa* as a function of the CDR estimate for pairwise correlations as estimated via the algorithm W.

For the data sets described in section 5.1 we can evaluate the quality of the data by calculating the CDR values for various subsets of variables. The true *kappa* values, obtained from classification algorithms in Weka, are then plotted as a function of the CDR. The approximation that $\kappa = 1 - 2^{-\text{CDR}}$ is shown for reference.

As expected, the data sets tend to follow the $1 - 2^{-\text{CDR}}$ trend displayed in figure 5.8 supporting the relationship between the Kullback-Leibler divergence and *kappa*. Note the log of the CDR value on the x-axis. From figure 5.8 we can make various conclusions about the quality of the model possible from each of the data sets. The Bankruptcy data set achieves the most accurate models, indicated by the large *kappa*, this is reflected in the CDR values. In contrast, the CHD data set performs the worst. For small CDR values the *kappa* value tends to be underestimated. This is apparent for the CHD data set which performs marginally better than expected. The Particle and Breast Cancer data sets fall somewhere in the middle, producing reasonable models for certain variable subsets.

References

[1] D. W. Corne and J. D. Knowles, "No free lunch and free leftovers theorems for multiobjective optimisation problems," in *International Conference on Evolutionary Multi-Criterion Optimization*, Springer, 2003, pp. 327–341.

[2] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining, Fourth Edition: Practical Machine Learning Tools and Techniques*, 4th. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2016, ISBN: 0128042915.

[3] W. N. Street, W. H. Wolberg, and O. Mangasarian, "Nuclear Feature Extraction For Breast Tumor Diagnosis," *International Symposium on Electronic Imaging: Science and Technology*, vol. 1905, pp. 861–870, 1993.

[4] O. L. Mangasarian, W. N. Street, and W. H. Wolberg, "Breast Cancer Diagnosis and Prognosis via Linear Programming," *INFORMS*, vol. 43, no. 4, pp. 570–577, 1995.

[5] O. L. Mangasarian, "Unsupervised classification via convex absolute value inequalities," *Optimization*, vol. 64, no. 1, pp. 81–86, 2015, ISSN: 10294945. [Online]. Available: `http://dx.doi.org/10.1080/02331934.2014.947501`.

[6] I. Maglogiannis, E. Zafiropoulos, and I. Anagnostopoulos, "An intelligent system for automated breast cancer diagnosis and prognosis using SVM based classifiers," *Applied Intelligence*, vol. 30, no. 1, pp. 24–36, 2009. DOI: `10.1007/s10489-007-0073-z`.

[7] S. C. Bagui, S. Bagui, K. Pal, and N. R. Pal, "Breast cancer detection using rank nearest neighbor classification rules," *Pattern Recognition*, vol. 36, no. 1, pp. 25–34, 2003, ISSN: 00313203. DOI: 10.1016/S0031-3203(02)00044-4.

[8] M. J. Kim and I. Han, "The discovery of experts' decision rules from qualitative bankruptcy data using genetic algorithms," *Expert Systems with Applications*, vol. 25, no. 4, pp. 637–646, 2003, ISSN: 09574174. DOI: 10.1016/S0957-4174(03)00102-7.

[9] L. Teodorescu, "Gene expression programming approach to event selection in high energy physics," *IEEE Transactions on Nuclear Science*, vol. 53, no. 4, pp. 2221–2227, 2006, ISSN: 00189499. DOI: 10.1109/TNS.2006.878571.

[10] T. Sjostrand, "High-energy-physics event generation with PYTHIA 5.7 and JETSET 7.4," *Computer Physics Communications*, vol. 82, pp. 74–89, 1994.

[11] D. Lange, "The EvtGen particle decay simulation package," *Nuclear Instrumental Methods Phys. Res. A*, vol. A462, pp. 152–155, 2001.

[12] J. E. Rossouw, J. P. du Plessis, A. J. Benade, P. C. Jordaan, J. P. Kotzé, P. L. Jooste, and J. J. Ferreira, "Coronary risk factor screening in three rural communities. The CORIS baseline study," *South African Medical Journal*, vol. 64, no. 12, pp. 430–436, 1983, ISSN: 00382469.

[13] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, Second Edition. Springer, New York, 2009, ISBN: 978-0-387-84857-0.

[14] T. A. Stamey, J. N. Kabalin, J. E. McNeal, I. M. Johnstone, F. Freiha, E. A. Redwine, and N. Yang, "Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate. ii. radical prostatectomy treated patients," *The Journal of urology*, vol. 141, no. 5, pp. 1076–1083, 1989.

[15] S. Kullback and R. Leibler, "On Information and Sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 1951. DOI: 10.1214/aoms/1177729694. arXiv: 1706.01538. [Online]. Available: http://arxiv.org/abs/1706.01538.

[16] S. Fekri-Ershad, "Gender classification in human face images for smart phone applications based on local texture information and evaluated Kullback-Leibler divergence," *Traitement du Signal*, vol. 36, no. 6, pp. 507–514, 2019, ISSN: 19585608. DOI: 10.18280/ts.360605.

[17] J. Kasturi, R. Acharya, and M. Ramanathan, "An information theoretic approach for analyzing temporal patterns of gene expression," *Bioinformatics*, vol. 19, no. 4, pp. 449–458, 2003, ISSN: 13674803. DOI: 10.1093/bioinformatics/btg020.

[18] M. M. De Queiroz, R. W. Silva, and R. H. Loschi, "Shannon entropy and Kullback-Leibler divergence in multivariate log fundamental skew-normal and related distributions," *Canadian Journal of Statistics*, vol. 44, no. 2, pp. 219–237, 2016, ISSN: 1708945X. DOI: 10.1002/cjs.11285. arXiv: 1408.4755.

[19] A. Jakulin, "Attribute interactions in machine learning," *Master's thesis University of Ljubljana*, no. February, 2003.

[20] A. Smith, P. A. Naik, and C. L. Tsai, "Markov-switching model selection using Kullback-Leibler divergence," *Journal of Econometrics*, vol. 134, no. 2, pp. 553–577, 2006, ISSN: 03044076. DOI: 10.1016/j.jeconom.2005.07.005.

[21] P. G. Bissiri and S. G. Walker, "Converting information into probability measures with the Kullback-Leibler divergence," *Annals of the Institute of Statistical Mathematics*, vol. 64, no. 6, pp. 1139–1160, 2012, ISSN: 00203157. DOI: 10.1007/s10463-012-0350-4.

[22] O. M. Cliff, M. Prokopenko, and R. Fitch, "Minimising the kullback-leibler divergence for model selection in distributed nonlinear systems," *Entropy*, 2018, ISSN: 10994300. DOI: 10.3390/e20020051.

[23] W. Wang, C. Liu, and D. Zhao, "How Much Data is Enough? A Statistical Approach with Case Study on Longitudinal Driving Behavior," *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 2, pp. 1–1, 2017, ISSN: 2379-8858. DOI: 10.1109/tiv.2017.2720459. arXiv: 1706.07637.

[24] P. J. Moreno, P. P. Ho, and N. Vasconcelos, "A Kullback-Leibler divergence based kernel for SVM classification in multimedia applications," *Advances in Neural Information Processing Systems*, vol. 16, 2004, ISSN: 10495258.

[25] S. Sinanović and D. H. Johnson, "Toward a theory of information processing," *Signal Processing*, vol. 87, no. 6, pp. 1326–1344, 2007, ISSN: 01651684. DOI: 10.1016/j.sigpro.2006.11.005.

[26] Q. Wang, S. R. Kulkarni, and S. Verdú, "Divergence estimation of continuous distributions based on data-dependent partitions," *IEEE Transactions on Information Theory*, vol. 51, no. 9, pp. 3064–3074, 2005, ISSN: 00189448. DOI: 10.1109/TIT.2005.853314.

[27] ——, "A nearest-neighbor approach to estimating divergence between continuous random vectors," *IEEE International Symposium on Information Theory - Proceedings*, no. 5, pp. 242–246, 2006, ISSN: 21578101. DOI: 10.1109/ISIT.2006.261842.

[28] ——, "Divergence estimation for multidimensional densities via κ-nearest-neighbor distances," *IEEE Transactions on Information Theory*, vol. 55, no. 5, pp. 2392–2405, 2009, ISSN: 00189448. DOI: 10.1109/TIT.2009.2016060.

[29] N. Leonenko, L. Pronzato, and V. Savani, "A class of rényi information estimators for multidimensional densities," *Annals of Statistics*, vol. 36, no. 5, pp. 2153–2182, 2008, ISSN: 00905364. DOI: 10.1214/07-AOS539.

[30] F. Pérez-Cruz, "Estimation of information theoretic measures for continuous random variables," *Advances in Neural Information Processing Systems 21 - Proceedings of the 2008 Conference*, pp. 1257–1264, 2009.

[31] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. 2005, ISBN: 9780471241959. DOI: 10.1002/047174882X.

[32] D. McAllester and K. Stratos, "Formal Limitations on the Estimation of Mutual Information," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, vol. 108, Palermo, Italy, 2020. DOI: 10.3390/proceedings2019033031. arXiv: 1910.00365.

[33] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, no. 1, pp. 37–46, 1960. DOI: 10.1177/001316446002000104. eprint: https://doi.org/10.1177/001316446002000104. [Online]. Available: https://doi.org/10.1177/001316446002000104.

[34] F. Emmert-Streib and M. Dehmer, *Information theory and statistical learning*. 2009, ISBN: 9780387848150. DOI: 10.1007/978-0-387-84816-7.

[35] L. Flight and S. A. Julious, "The disagreeable behaviour of the kappa statistic," *Pharmaceutical Statistics*, vol. 14, no. 1, pp. 74–78, 2015, ISSN: 15391612. DOI: 10.1002/pst.1659.

Chapter 6

Exploratory Data Analysis and Explainable Machine Learning

Identifying interesting associations between random variables is paramount to understanding a system. Continuously advancing computational abilities means more methods than ever are available for extracting this information from a sample. As a result, modern data analysis techniques and machine learning algorithms are now at the forefront of scientific advancements. The use of these techniques in any data analysis task can improve the knowledge and understanding of the system. Such promises fueling the never-ending competition for better results.

Consequently, there is an eagerness to acquire as much data as possible to maximize the knowledge gained. Recent advances in computational infrastructure and accessible storage have facilitated this growth in data size in both the number of instances and the number of variables, giving rise to the field of "big data". With many believing, the more, the better. Naturally, big data brings about new challenges for querying, exploring, and analysing these ever-growing data sets. In the age of machine learning, it is all too easy to overload algorithms with too much information, which slows down learning and can cause the algorithm to overfit. While it is true that more instances do offer greater statistical power, higher-dimensional data sets often lead to unreliable results due to increased computational complexity. Instead, the performance of a machine learning algorithm is better improved by applying the most appropriate algorithm to a subset of the most informative variables.

To achieve an optimal subset, it is helpful to examine the data prior to more complex data analysis to identify and remove unwanted variables, preventing them from confusing the algorithm. Removing these unwanted variables is often part of a pre-processing step before applying any machine learning and is referred to as feature selection. Traditionally, supervised learning tasks use feature selection. Where supervised learning uses pre-labeled instances to train a classification algorithm that can predict the labels on unseen data. Although the approach is not always employed, the benefits are well established.

Feature selection methods aim to identify the predictive abilities of a variable, or set of variables. A variable is considered *relevant* if it correlated to the class variable and *irrelevant* if not [1].

Feature selection should not be confused with the process of feature extraction, which reduces the dimensionality of the data sample by compressing multiple variables into a single variable [2]. Feature selection, instead, returns a ranking or an optimised subset of the original variables that are highly predictive of the class. The internal relationships (variable-variable) in a subset are significant for successful supervised learning. If the data set consists of several strongly relevant variables, there will likely be overlapping information about the class. That is to say, the variables themselves have non-zero mutual information with one another. Therefore, using two highly relevant variables in a classification model may be no more predictive than using one. Information is *redundant* when the variables in a subset contains reoccurring information. Ideally, an optimised subset consists of a few relevant variables that are uncorrelated to one another. Meaning that variables that appear to have a weak correlation with the class can contain information that no other variable does, making it highly valuable in a variable subset [3].

Feature selection approaches are separated into two categories: univariate and multivariate methods. Univariate methods evaluate the predictive abilities of a single variable with the class, such as *attribute ranking*. This approach gravitates itself to a subset with only highly relevant variables and not necessarily an optimised subset. In comparison, multivariate methods evaluate the predictive abilities of subsets of variables as a whole, comparing their performance to other subsets to identify the optimal selection [4].

It is all well and good that machines can run millions of calculations in a fraction of a second to select an optimised subset; however it is not conducive of *learning* about a system. As we move towards machine learning governing decisions in our everyday lives, such as mortgage applications or medical procedures, it is increasingly vital that we shift our attitudes away from a black-box mentality towards explainable machine learning. Humans have an innate need to understand decision making; therefore, for machine learning models to have a future in real-world applications, we must be able to comprehend their predictions and decisions. Blindly applying machine learning is not sustainable. Explaining complex results, however, requires an understanding of the variables deemed to be beneficial. Thus, the identification of relevant variables and variable-variable interactions is a valuable commodity in modern data analysis. Here we put forward a guided approach to feature selection which aids in the comprehension of machine learning models and how well it performs [5], [6].

6.1 Exploratory Data Analysis and Visualisations

Exploratory data analysis describes an investigation-centred approach to data analysis. Rather than seeking to answer a pre-determined question, the analyst familiarises themselves with the data to understand its characteristics without necessarily having prior knowledge of the underlying mechanisms. Using an exploratory approach, one can use the improved insight of the data

to explain machine learning results. A greater understanding of the data leads to a better understanding of the outcomes. The process utilises as many methods as deemed necessary, often employing data visualisations, an unmatched tool in exploratory data analysis. Although multivariate data analysis is typical in many scientific fields, traditional visualisation techniques are unsuitable for more than two dimensions, tending to display only one or two dimensions at a time, i.e. histograms and scatter plots. The increase in data collection calls for an evolution in visualisation methods to tackle high dimensional data sets [6].

There are a number of multivariate visualisation tools available including three-dimensional scatter plots, scatter plot matrices [7], hyperboxes [8], star coordinates [9] and parallel coordinates [10]. Many of these method utilise recent advancements in graphical techniques to aid the visualisations for real-time data manipulations. Here, we review interactive parallel coordinates plots for multivariate visualisation. We expand upon the concept of hypergraphs to propose a *variable interaction diagram* for presenting pairwise inter-variable and variable-class data structures. Finally, we introduce *DataViewer*, a novel visualisation software package for exploratory data analysis. DataViewer provides a graphical user interface (GUI) for exploring variable correlations using information-theoretic measures. We leverage the power of *co-plots*, also known as *linked brushing*, to highlight patterns and guide the user in an exploratory analysis. We hope that these visualisations and algorithms will aid data scientists in finding key variable interactions and further the understanding of a data set.

6.1.1 Parallel Coordinate Plots

A parallel coordinates plot is a multivariate data visualisation tool that allows the viewer to visually detect structures and patterns beyond three-dimensions. Each variable is projected onto a one-dimensional axis aligned in parallel. An instance is represented as a *polyline* connecting all the axes. Each variable is normalised and the polyline intersects each axis at the point corresponding to the variable value. A schematic is shown in figure 6.1. At first, these plots can come across as complicated and "messy" due to visual clutter, such as polyline crossings. However, simple features, such as linear correlations and clustering, display a duality in Cartesian and parallel coordinates. As with conventional Cartesian plots, structures are easily identifiable with some developed intuition. Recognising relationships between variables is discussed in [11] once familiar parallel coordinates allow for straightforward pairwise comparison with adjacent variables.

However, the patterns revealed are highly dependent on the order in which the parallel axes are presented. For non-adjacent variables, a comparison can be challenging. Dimension management is therefore a recurring topic of interest in multivariate visualisations [10], [12]. [13] discussed the problems associated with understanding relations between non-adjacent dimensions and highlighted the importance of axis permutations. For a d-dimensional data set, $(d+1)/2$

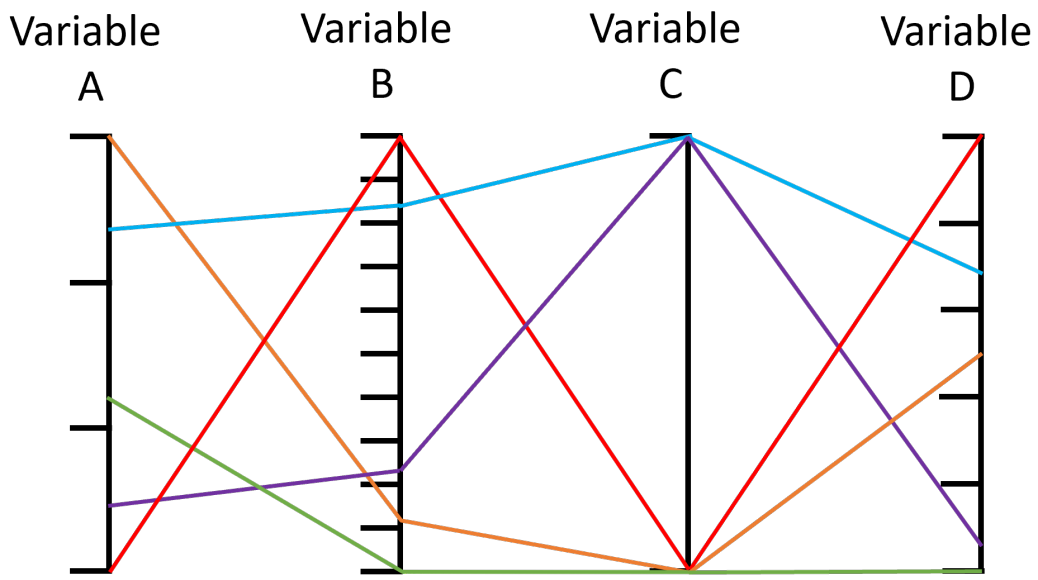

Figure 6.1: A simple schematic of a parallel coordinate plot where each instance has been brushed a different colour.

parallel coordinate plots are required to observe all possible pairwise combinations of adjacent axes [13]. This is considerably more efficient than viewing all $d!$ pairwise scatter plots, however, the topic of optimal axis ordering is still an active research topic [14], [15].

For this reason, interactive axis ordering and flipping features are paramount to benefit from the visualisation fully. The advancement of computer graphics has facilitated the practicality of parallel coordinates in modern analysis. These advancements have enabled the ability to append axis positions in real-time and use algorithms to establish an intelligent ordering, instigating a unique interactive angle. These approaches use optimising algorithms, that minimise a global cost parameter to solve the axes ordering problem. For instance, minimising the number of polyline crossings produces less "tangled" plots and reduces visual clutter [16].

In addition, interactive features, including pruning and brushing, are powerful tools for exploratory visualisation. Pruning unwanted items on the parallel coordinates plot reduces the number of instances or variables displayed, alleviating visual clutter. For example, the benefits of pruning include the ability to filter outliers from the data set and remove low-density regions. This ability isolates clusters or subsets of the data for an exploratory approach. However, one might not want to remove these points physically from the data set. Instead, a similar visual effect to pruning is achieved through brushing. Brushing adds colour or transparency to the data, highlighting clusters of instances or high-density regions, making them easier to track through the plot [17].

6.1.2 Visualising Variable Interactions

Visualising variable interactions for exploratory data analysis enhances the viewer's understanding of the relationships in a data set and how they tie into one another. Visualising how systems are connected is a topic of great interest, referred to as "graph theory". Hypergraphs, a category of visualisations in graph theory, communicate complex network systems through a series of nodes and lines.

In [18] Jakulin *et. al.* uses the premise of nodes and connections to visualise variable interactions. Jakulin uses the size of the node to represent uncertainty for variables and their connections. For a variable node this is the entropy of that variable, and for a connections this is the strength of the correlation. To attempt to decompose the various information measures, Jakulin also uses colour to indicate positive and negative values, where white circles indicate a positive information and grey circles indicate negative information such as redundancy. The joint entropy of the variables can be obtained by summing over all of the grey nodes and subtracting the white nodes. For a non-expert these graphics are difficult to comprehend as they require sufficient knowledge about information theory. Using the area to represent the magnitude of the information measures can also be difficult to visually interpret, especially when the differences are small.

We propose a visual representation of the variable interactions also inspired by hypergraphs referred to as the "variable interaction diagram" (VID). The VID visualises pairwise interactions between variables. Nodes, displayed as circles of fixed size, represent each variable, and the lines connecting them indicate a correlation. The nodes are labelled using the variable name and the colour and thickness of the lines correspond to the strength of the relationship. A thick line indicates a strong correlation, making highly correlated connections easy to identify, and a thin line indicates a weak correlation. Therefore, two variables unconnected by a line are not directly dependent, and an unconnected node indicates the variable is independent of all other variables in the system. As the mutual information is a symmetric quantity ($I(X, Y) = I(Y, X)$) the lines in a VID are undirected. A schematic of a VID for a 10 dimensional data set is given in figure 6.2.

The scale in figure 6.2 indicates the strength of the connections. It is divided into four equally-sized regions, the blue region for weakly correlated to the red for strongly correlated. Being able to compare correlation strengths is crucial for any analysis. However, the individual entropies limit the maximum amount of information that any two variables can share, i.e. a variable cannot share more information than it has. Therefore, a variable with a small entropy sharing all its information will have a smaller mutual information than a variable with a larger entropy sharing only some of its information. Algorithm W ensures that the individual entropies are the same for each continuous variable. However, for a discrete variable, the Shannon entropy is fixed. Thus, comparing the mutual information values for discrete-discrete and continuous-discrete

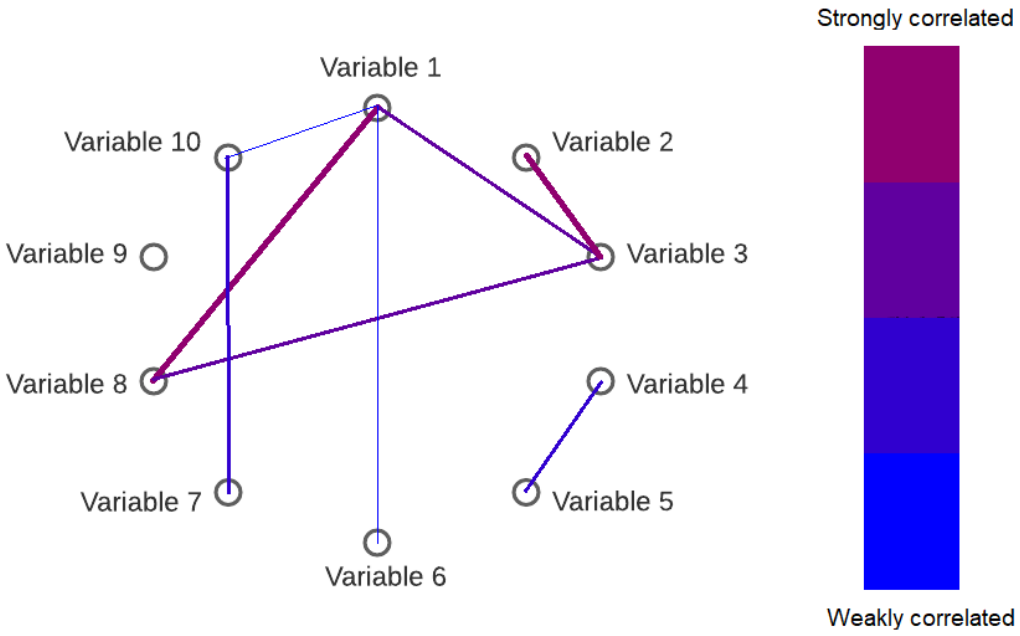

Figure 6.2: A schematic of a VID for a 10-dimensional system.

combinations would result in incorrect conclusions. The mutual information must be normalised to compare correlation strengths. Therefore, the mutual information is scaled between zero and one by dividing by the smallest entropy to give the similarity index (SI).

$$SI(x,y) = \frac{I(x,y)}{Min[H(x),H(y)]} \tag{6.1}$$

Normalising the mutual information is paramount for supervised problems, as the class is more often than not discrete.

In unsupervised learning, we are interested in variable-variable interactions to identify interesting correlations and predict the value of any variable with the knowledge of other variables. Whereas, in supervised learning, the objective is to predict the class using the variables. Thus, in supervised learning, we are particularly interested in correlations that involve the class. It may therefore be valuable to concentrate solely on the class-variable interactions. By positioning the class variable in the centre of the VID, the viewer can focus on these correlations alone. This reduces the visual clutter if there are otherwise many correlations or variables. A schematic for the supervised VID is shown in figure 6.3 with the class positioned in the centre of the diagram. In practice, however, any variable can be put in the central position to highlight that variable's relationships. The particular query of the analyst depends on which visualisation is the most suitable.

The VID aims to enable the viewer to identify pairs and clusters of variables that are correlated. For exploratory data analysis, this is useful to guide the user to interesting structures within the data. For classification problems, the VID, both supervised and unsupervised, is valuable for

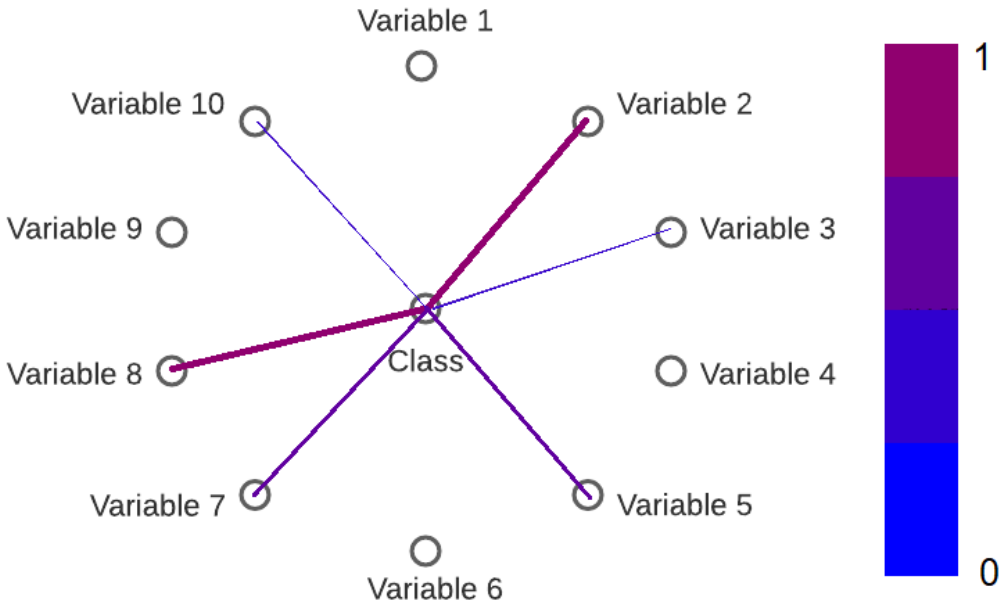

Figure 6.3: A schematic of the "supervised" VID for a 10-dimensional system with the class variable in the center.

feature selection.

6.2 Visualisation Software: *DataViewer*

In addition to the algorithms, we developed an application to implement the ideas presented in this thesis. The software was required to implement algorithm W, detailed in chapter 4, to calculate the mutual information and CDR values for pairwise relationships. The intention was for non-data scientists to use the application to identify critical correlations for exploratory data analysis and help decipher complex data structures. The application was to incorporate visualisations, including histograms, scatter plots, parallel coordinate plots, and the VID introduced in 6.1.2 to create a minimum viable product (MVP). An MVP is a basic version of the desired application, with the minimum functionality necessary for distribution and use.

To make the techniques accessible to non-data scientists, a Graphical User Interface (GUI) was necessary. The objective was to allow the user to upload their data via a CSV file, implement basic data editing, and apply the proposed algorithms. We also wanted to utilise interactive visualisations, which are increasingly popular due to their ability to allow real-time updates and manipulations of the data. These are particularly useful in exploratory data analysis, allowing the viewer to investigate queries as they arise.

6.2.1 The Code and External Libraries

DataViewer's code-base is in C++, chosen due to its fast performance and modular features, making it easy to add functionality throughout the development. In addition, C++ has several compatible external libraries, which were employed to increase the efficiency of some calculations. We discuss some of the external libraries used here.

The GUI has been developed in the Qt framework, a GUI software. Qt's foundations in C++, its ease of use and cross-platform accessibility made it an ideal choice for this project. In Qt, objects communicate using signal and slot messaging. Thus, a GUI widget can send event information from the user interacting with the screen (a signal) to special functions called slots. This simple yet effective construct enabled the development of a highly interactive interface with minimal impact on the code of the underlying algorithms.

The external library, the Open Graphics Library (OpenGL), was used to streamline some visualisations, as it enabled efficient displaying of large quantities of data via the computer's Graphics Processing Unit. In addition, OpenGL accelerates the rendering of QCustomPlot, a Qt widget for visualisations. This combination offers high performance visualisation for real-time manipulations.

The most computationally time-consuming aspect of algorithm W is the nearest neighbour search. Unfortunately, there are no exact algorithms for solving the nearest neighbour problem faster than the linear search approach. On the other hand, approximate algorithms can speed up the process for higher dimensions and large k, with only a minor loss in accuracy for $k > 1$. To reduce the computational time the Fast Library for Approximate Nearest-Neighbours (FLANN) was implemented. It is a C++ external library that performs approximate nearest neighbour searches in high dimensions [19].

The random numbers used for noise distributions were generated using the "Mersenne Twister" engine [20], which passes the *Diehard* statistical tests [21]. The seed for the engine is generated using a non-deterministic source e.g. a hardware device on the users computer to feed the generator. This generates 64-bit unsigned integers which is then used to create the desired distribution.

6.2.2 Minimum Viable Product

An academic version of the DataViewer is publicly available and has been distributed under an open-source license, the details of which can be found in appendix E. The executable was released so that no external libraries needed to be downloaded in order for the full functionality of the software. This required using features in Qt to automatically link the external libraries which

were included within the deployment file. The MVP release contained all the basic functionality necessary. The application has continued to undergo development since the MVP release. The next development being the addition of multi-threading programming for the iterative calculations to reduce calculation time. This has been tested on the algorithms externally from the application and has shown to be worthwhile.

6.2.3 Main Window

Once downloaded the software can be started by clicking on the executable in the deployment file, however, if the licensing documents are not present this will result in a failed execution informing the user of the problem. Once open data can be loaded, in the form of a CSV file, by selecting the 'File' menu via the main window. Upon successfully loading the data the parallel coordinates plot is automatically displayed in the main window. Each axis is labelled using the headers provided in the CSV file and the plot height is adjusted to account for the length of the longest variable name. Although this is automatically displayed in the horizontal format the option is available to the user to rotate the plot vertically in order to make best use of the screen space available. Figure 6.4 shows the application upon opening a data set for the first time. The top image shows the default orientation of the parallel coordinates plot and the bottom shows the alternative display, which can be selected via the menu on the left hand side of the window.

The axes are by default displayed in the ordering corresponding to the order they appeared in the uploaded data file. They can be manually moved by clicking and dragging the variable to its desired location.

6.2.4 Groups and Classes

By default upon opening the data all instances are assigned to a single group, which is an attribute of each of the instances. Via the group editing manager, seen in figure 6.5 new groups can be created. The instances the user wants to include in group can be selected by highlighting the desired area anywhere on the parallel coordinates plot by simply clicking and dragging. The selected instances are assigned to the new group and any unselected instance remains in the default group. The default group cannot be deleted unless all instances are assigned to alternative groups and similarly upon deleting a group the instances are automatically re-assigned to the default group.

The user can also create groups by assigning a class variable. This is best done with a discrete variable as a new group is automatically created for each distinct value in the variable. These groups then similarly appear in the group editor. The group and any assigned feature such as group name, colour or visibility becomes an attribute of that instance and is maintained in all visualisations. This is known as the idea of co-plots. Co-plots are multiple plots of the same

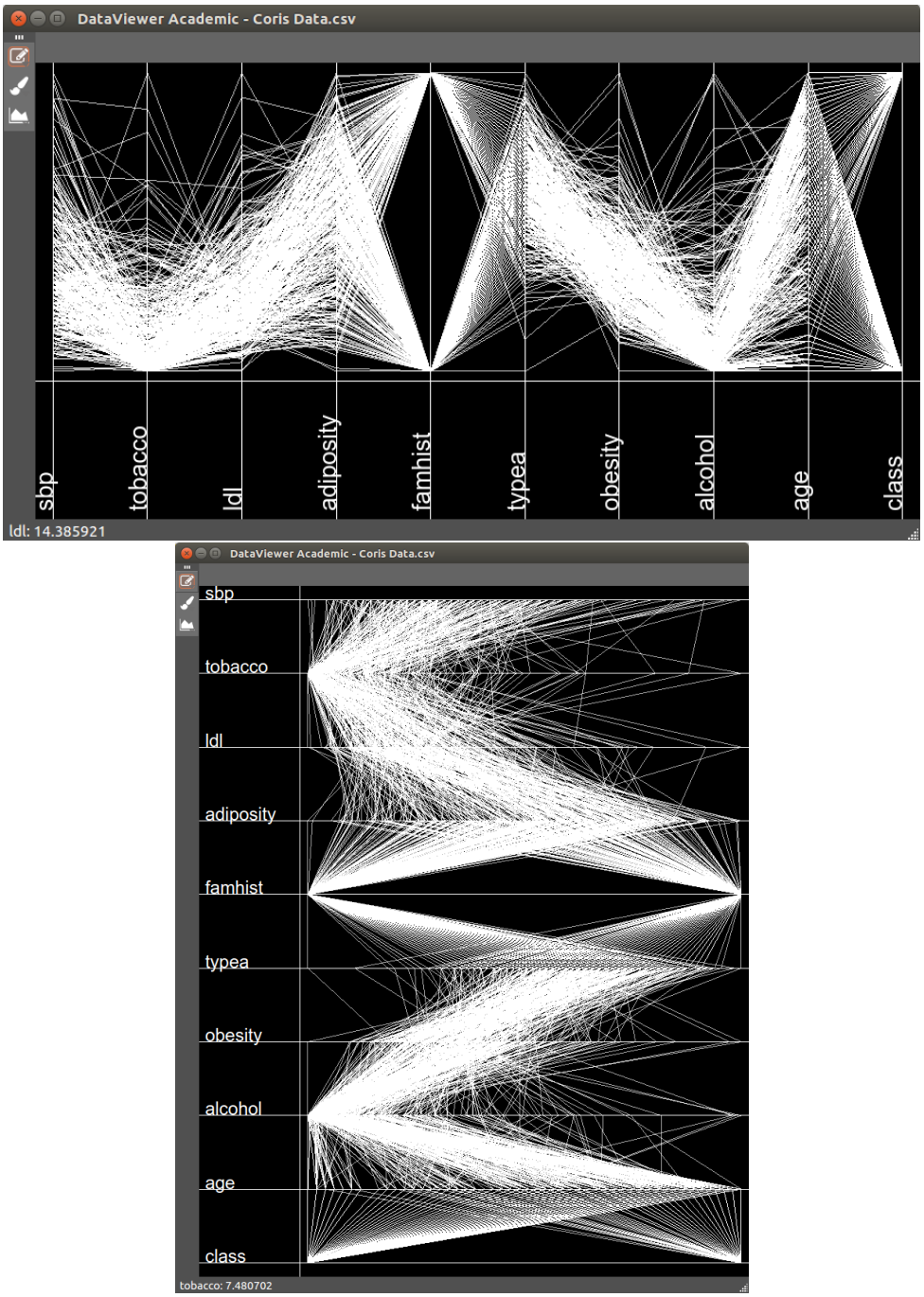

Figure 6.4: The parallel coordinates plot for the CHD data set in a horizontal and vertical orientation. On the left hand side is the manipulations menu. This is where the various visualisations and manipulations can be selected from. As the user hovers over the plot the variable name and axis value at the cursor location is displayed in the status bar at the bottom of the window.

data presented in different ways. Colour allows the analyst to keep track of subsets of points through each visualisation.

The application also includes a pruning feature, this can be used to remove instances from the data set entirely or to temporarily hide them. Similar to pruned instances, hidden items (instances or variables) will not appear in any of the visualisations or calculations. However, hidden items can be made visible again, this can be done via the group editor.

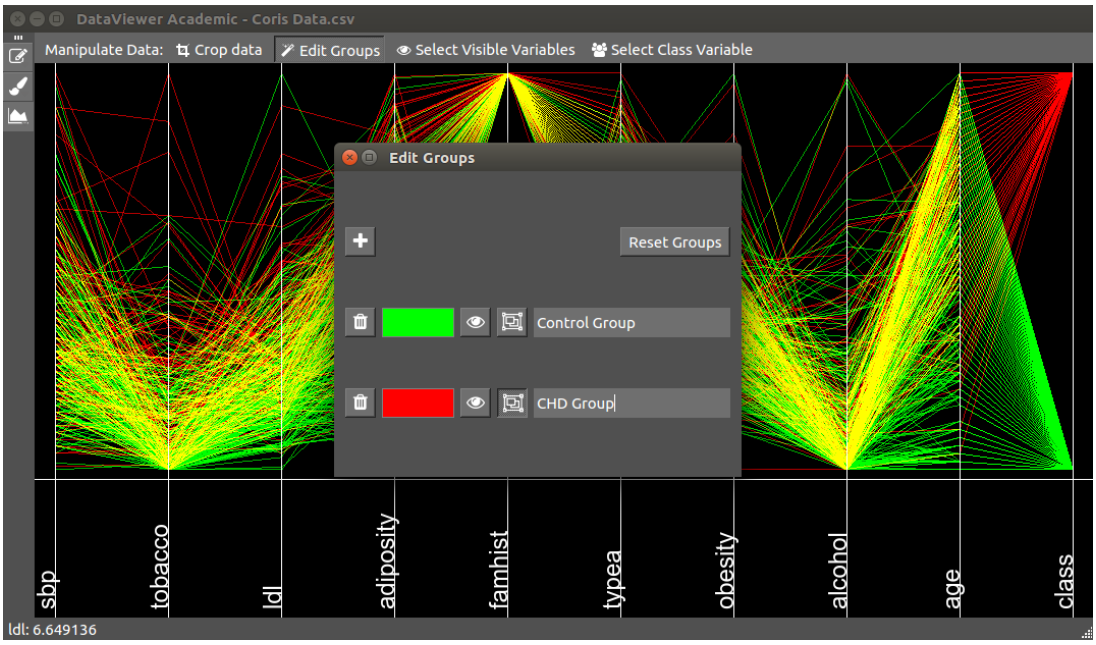

Figure 6.5: The Edit groups window in DataViewer. Any number of new groups can be created by clicking on the plus sign. Individual groups can then be coloured, hidden or deleted. While resetting the groups deletes all groups and creates and returns the data to its default grouping.

When points are overplotted the transparency of the polylines can be adjusted such that high density regions have a more prominent colour than low density regions. This also accounts for different coloured polylines which are overplotted, indicated by the mixing of the colours. The level of transparency can be adjusted by the user via the opacity slider accessed through the manipulation menu.

6.2.5 Histogram Widget

When the histogram visualisation is selected, a new window is opened and the data is automatically binned using the quantisation approach in section 4.1, where $\Delta = 2^h/\sqrt{N}$ using $k = 1$ and $M = 2$. However, the parameters k and M can be adjusted through the histogram window to achieve the desired quantisation. Once the user has achieved the desired binning, the application notifies them that the new values will be extended to all calculations. The bin width is calculated for the full variable, not for individual groups. However, the groups assigned are then binned separately. The user can scan through the variables using the drop-down menu. Recall that hidden variables will not be displayed.

Clicking on the histogram displays an information box that tells the user some basic features about the histogram. Similarly, the status bar at the bottom gives real-time information about the selected bin, such as the bin boundaries and bin height. An example of the histogram window is shown in figure 6.6.

The information contained in the histogram can be exported in several ways; as a png file of the

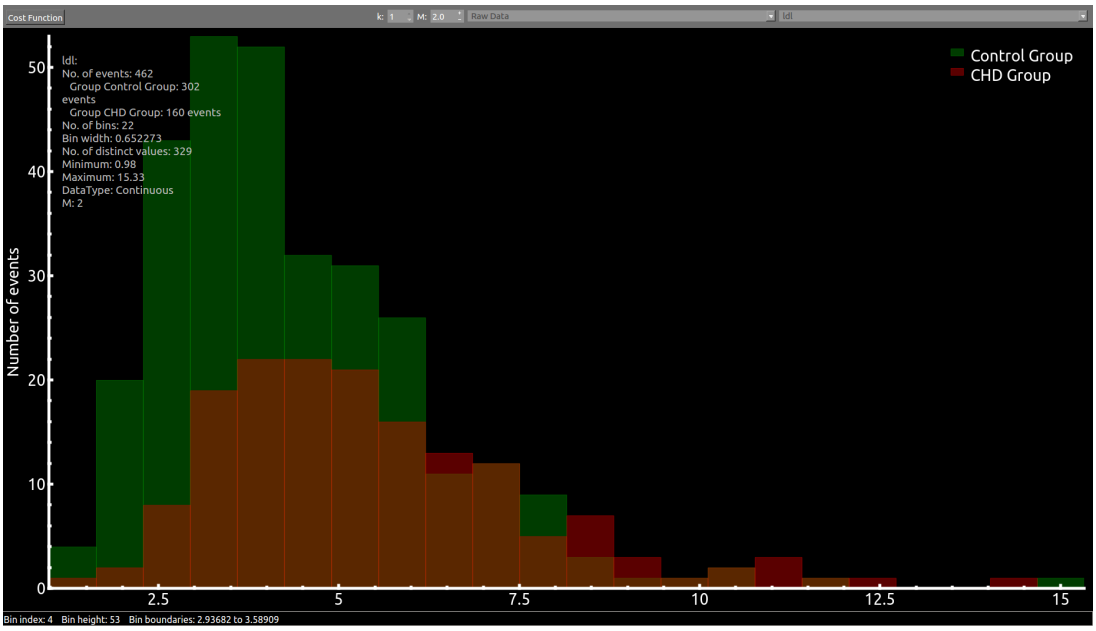

Figure 6.6: Fixed-width histogram for the variable *ldl* from the CHD data set, where the two groups have been binned separately and brushed such that green corresponds to the Control group and red to the CHD patients.

plot, as a CSV file of the binned instances and as a CSV file of the relative frequencies and bin boundaries.

Equiprobable binning

The most common quantisation technique for histograms is bins of fixed width. This method preserves the distribution in the relative frequencies but can be sensitive to extreme values. Although estimating the distribution is a fundamental data analysis technique, it is often a means to determine more interesting features, such as variable interactions. An alternative quantisation method, known as equiprobable quantisation, or "equiquantisation", does not preserve the distribution in the relative frequencies. However, partitioning the data in this way has been shown to be more responsive to variable interactions, which are of interest. It has been shown that doing so produces superior results for information-theoretic measures [22]. In addition, visualising equiquantisation is especially valuable when considering variable-class interactions [23] [24]. See section 3.1.2 for reference. The equiprobable histogram is uninteresting for non-grouped data sets; however, it is beneficial for distributions with a large kurtosis (tails). We will see how this can be used in section 6.3 for the Particle data set.

An example of an equiprobable histogram is shown in figure 6.7, where the classes have been coloured and histogrammed separately. DataViewer quantises continuous variable into $q = round[\sqrt{N}]$ bins, each with $n_i = round[N/q]$ instances. If N does not partition perfectly into q bins, such that $q * n_i - N \neq 0$, the bins are randomly chosen to have ± 1 instance depending on if there is an excess or deficit of instances. This ensures the distribution is as close to uniform

as possible, by preventing the final bin from having $n_q << round[N/q]$.

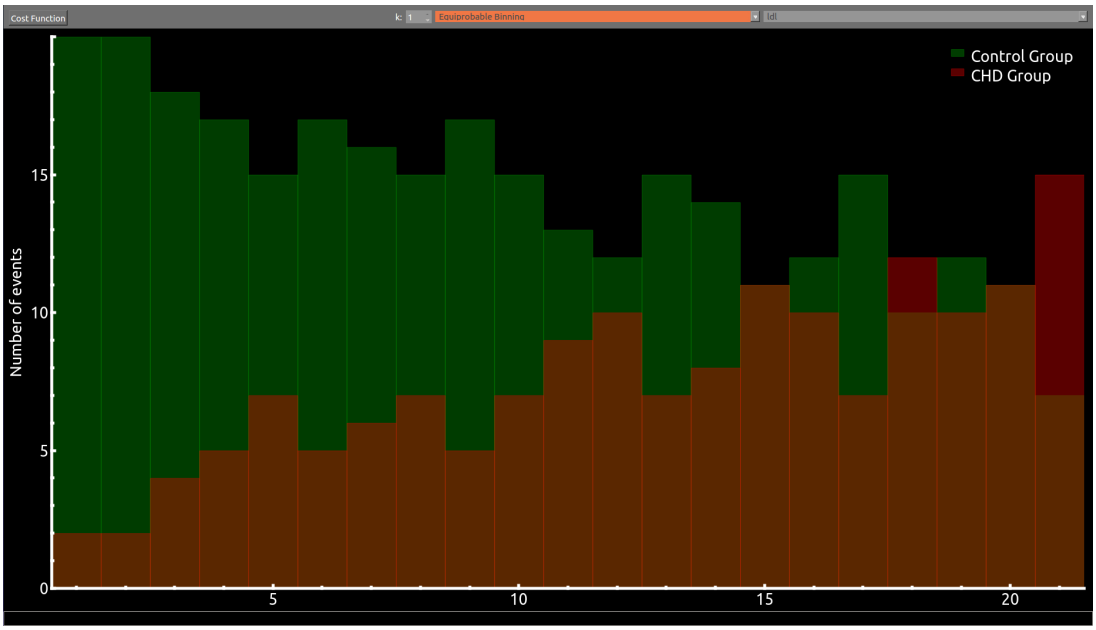

Figure 6.7: Equiprobable histogram for the variable *ldl* from the CHD data set.

The equiprobable binning feature is part of the histogram GUI. It can be accessed via selecting the setting on the 'Histogram type' drop down menu found at the top of the window. This feature can also be used to swap between the fixed-width histograms for the raw and normalised data ($n_i/(\Delta N)$).

Just as with the standard histogram GUI you can swap between variables, click on the graph to view relevant information about the histogram. You can also right-click on the window to either save the individual image of the current histogram as a jpeg or to output the binned indexes in a tab formatted CSV file. This enables the file to be read back into DataViewer and can subsequently be used to for viewing the equiprobable parallel coordinate plots.

6.2.6 Scatter Plots Widget

Similar to the histogram GUI, selecting the scatter plot visualisation option creates a separate window. This allows the user to view all plots simultaneously and have full control over the layout.

The scatter plot widget has two drop down menus positioned at the top of the window. These allow the user to select the variables on the X and Y axes, from the visible items. The coloration and transparency assigned in the main window is similarly applied here. Changing the colours or transparency in the main window automatically updates in the scatter plot GUI. The transparency can be turned on and off for the scatter plot using the "Opacity" tick-box. This can be useful when using high amounts of transparency in the parallel coordinates plot as with scatter plots the

points are typically smaller and there is less chance of overplotting. Nonetheless, transparency in scatter plots is equally useful to alleviate the problems associated with overplotting when it occurs. A screenshot of the scatter plot GUI can be seen in figure 6.8.

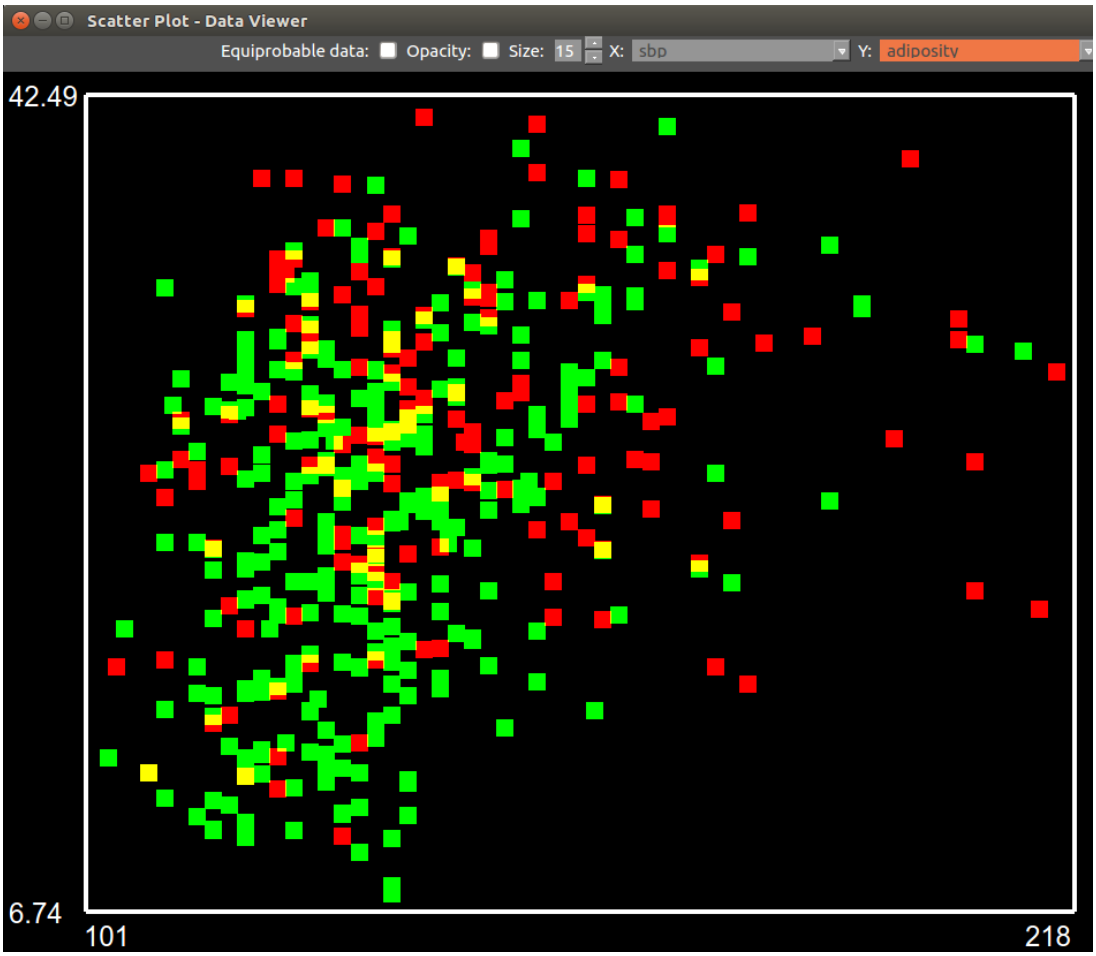

Figure 6.8: Scatter plot widget in DataViewer.

Other features include the ability to change the size of the points and swap between the original data and the equiquantised data. In future versions of the software, it will be beneficial for the pruning and brushing features, as available in the parallel coordinates plot, to be available in the scatter plot GUI. Allowing clusters of points to be more easily highlighted.

6.2.7 Variable Interaction Diagram

Selecting the VID from the Data Visualisation menu causes a pop-up box to appear with check-boxes with the options "Supervised" and "Normalised". Checking the "Supervised" QCheckBox gives the supervised VID, with the current class variable in the center. If a class variable has not been selected, clicking "OK" to continue will result in a warning pop-up informing the user this is a necessary requirement before aborting the visualisation. Checking the "Normalised" QCheckBox gives the SI values (normalised mutual information) instead of the mutual informa-

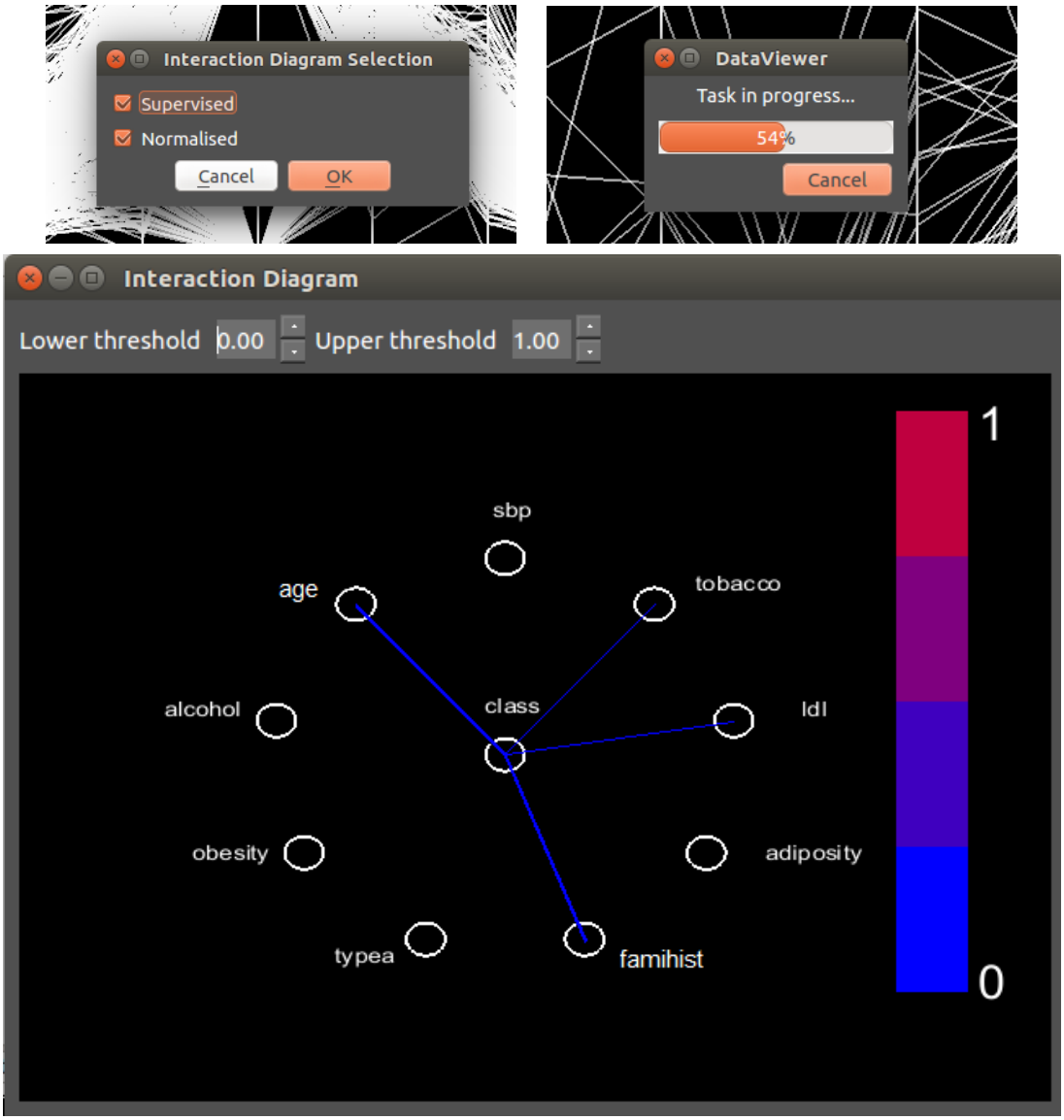

Figure 6.9: Screenshots of the VID GUI, illustrated using the supervised VID for the CHD data set.

tion. Once the setting have been selected the user clicks "OK" and algorithm W is implemented. Depending on the size and dimensionality of the sample a progress bar may appear to shows the progress of the calculation. The progress is determined from the percentage of iterations completed. For example, for 50 iterations, 1 iteration $= 2\%$, therefore for every iteration completed in the algorithm the progress bar will increase by 2%. The number of iterations can be chosen from the Options menu in the main window, these similarly apply to the CDR values. Once the calculates are completed the VID widget is displayed. Screenshots of the GUI are given in figure 6.9.

Each node in the VID can have up to $d - 1$ connections. For highly-correlated data sets the number of connections may cause the visualisation to become overcrowded. Similarly for large dimensional data. For this reason the user can adjust the upper and lower thresholds of the connections displayed. This allows the user to dive deeper into the variable interactions and

scan through the correlations in order of strength.

The SI values can be viewed in the application or exported through the "Results" menu at the top of the window. Exporting the data will result in the generation of a CSV file with the variable pairs and their mutual information values, and SI values if "Normalised" was selected, as well as their errors. Note that DataViewer estimates the errors on the mutual information using the resampling variance method discussed in section 4.6.

6.2.8 CDR

The latest version of DataViewer also includes the ability to calculate the Kullback-Leibler divergence and CDR values for any range of dimensions from 1 to $d-1$. This feature requires a discrete class variable to be selected. If there are more than two classes the CDR values are calculated for each pairwise class combination. Due to the indeterminable number of potential dimensions and classes there is currently not a visualisation to display this information. The numerical values are instead displayed in a widget and can be exported to a CSV file.

6.3 Application to Case Studies

The main goal of explainable machine learning is to make learning algorithms transparent and understandable. Many "black-box" techniques, such as Neural Networks, are too complex to be understood by humans as they can require thousands of parameters to come to a decision. That is not to say that interpretable models do not exist. For example, decision trees, which provide a series of classification rules, and linear regression models, which assign weights to the variables, are interpretable.

One way to increase the interpretability of an output model is through dimensionality reduction. The fewer dimensions, the less complicated the model, and the more interpretable it is. However, higher interpretability could lead to a trade-off of lower accuracy. Although, an interpretability-accuracy trade-off will not always be the case and in reality accuracy will increase for some data sets.

6.3.1 Experimental method

Here, we will take an empirical approach to demonstrate the ideas presented in this thesis on several classification problems. We will analyse the data sets, described in section 5.1, using the application DataViewer. Applying algorithm W in various settings demonstrates the multi-faceted uses of the method and shows how the results can guide machine learning and explain the output models.

First, we will use the SI and CDR values to identify relevant and redundant variables, and use the visualisations in DataViewer to guide an intelligent and efficient approach to machine learning. In doing so, we demonstrate the effectiveness of the parallel coordinates for multivariate visualisations. We also illustrate the use of equiprobable binning to explain otherwise unattainable information by visualising correlations that are not evident in traditional methods.

We compare the SI and CDR values with the accuracy estimates for individual variables to evaluate the measure's abilities to identify relevancy. Similarly, we examine the variables identified as irrelevant and investigate how removing them from the data set changes the learning algorithm performance.

Then, by considering the inter-correlations in the data set using the VID, we suggest variable combinations that will benefit the learning algorithm. We evaluate the conclusions made by considering the learning accuracy estimate and *kappa* values for these variable combinations. Next, we perform an exhaustive wrapper search on the data sets to find the optimal variable subset. An exhaustive wrapper search is when a classification algorithm is applied to all possible combinations of variables in all dimensions to determine the variable subset that achieves the best *kappa* statistic. Finally, using the visualisations, we attempt to explain these results.

6.3.2 Wisconsin Breast Cancer Data

In the original paper, an adaptation of the multi-surface method known as the MSM-Tree was used to classify benign and malignant tumours. The MSM-Tree iteratively adds multi-dimensional surfaces to separate the classes until an acceptable level of classification is achieved. In [25] the number of surfaces and variables was manually limited to one and three, respectively, to maintain generalisation and prevent over-fitting. In [26], a later paper by the same research group, these parameters were justified after an exhaustive search for all combinations up to four dimensions and two surfaces. They concluded that the three-dimensional single surface model was optimal. This restriction was imposed due to the computational cost of analysing further dimensional combinations.

Even restricting oneself to two dimensions results in $\frac{31!}{2!(31-2)!} = 465$ bivariate scatter plots. This would be extremely time-consuming to analyse by hand. The analysis could be made slightly easier by displaying the scatter plot matrix. However, this still involves searching through all combinations, and without a quantifying measure of a desired feature, this is a subjective method. Alternatively, we could display the multidimensional data set as a parallel coordinates plot, seen in figure 6.10, where the classes are coloured blue for benign and red for malignant. The axes ordering in figure 6.10 is a solution to the travelling salesmen problem, described in [27], based on the SI values calculated by DataViewer [1]. The solution has split the variables into two paths extending out from the class variable, *diagnosis*.

[1] This technique for ordering the axes is only mentioned and will not be discussed further as work is needed to complete this line of research.

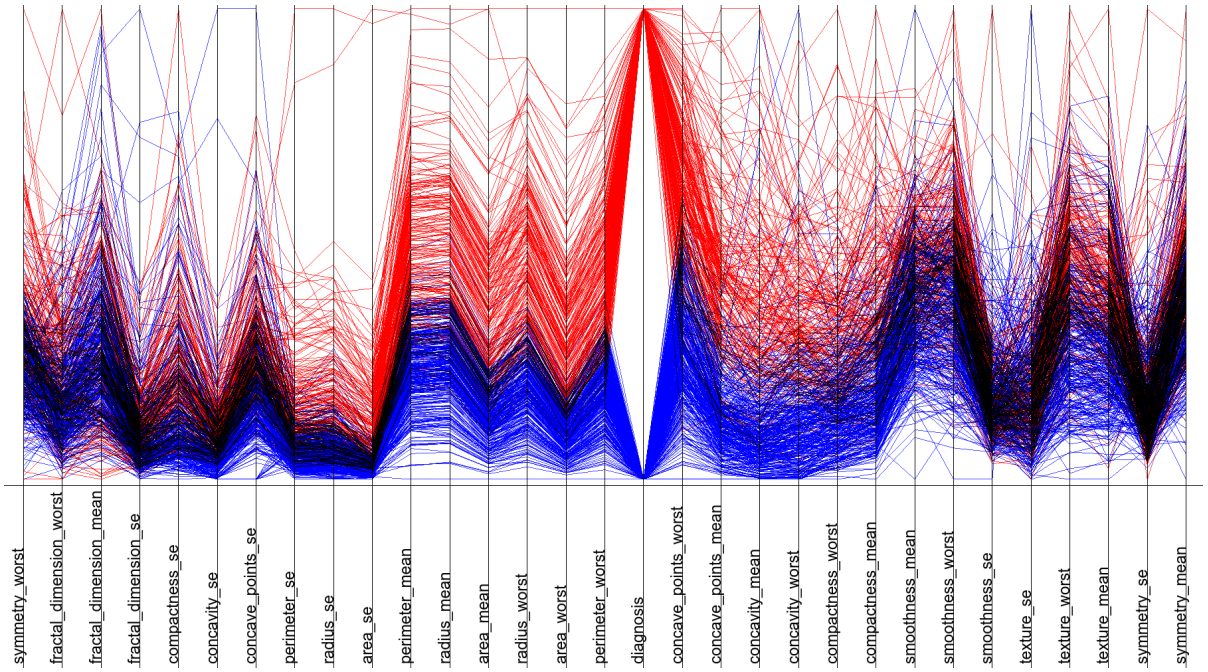

Figure 6.10: Ordered parallel coordinate plot of the WBCD data set.

The central variables of the parallel coordinates plot display distinct regions of the class colours, clearly demonstrating their ability to discriminate between the classes. It is also simple to identify correlated variables from the parallel coordinates plot. To the left of the class variable, the polylines run parallel, indicating that these variables are related. This signifies that these variables contains similar information, and it may not be beneficial to include them all in a subset for a classification solution. However, the variable subsets in the two paths leading from the class are less likely to be correlated. Therefore, we can conclude that having variables from both regions will benefit a machine learning algorithm.

In contrast, the variables with a small or no similarity index to the class variable are on the far sides of the plot. They are easy to discern as the class colours become increasingly mixed, indicated by the darker colouration. Note, however, the variable *symmetry worst* has distinct class regions despite being on the far edge of the parallel coordinates plot. Similarly for *texture worst* and *texture mean*. Being position far from the class does not strictly mean they are irrelevant to the system, but does suggest that they share less information with the central variables.

Many interesting features are not visible in detail in the parallel coordinates plot. To drill down deeper into the infrastructure of the data set we must illustrate the class-variable interactions in the supervised VID in figure 6.11. We can instantly visually identify several variables-*fractal dimension worst*, *smoothness mean*, *fractal dimension mean*, *texture se*, *smoothness se*, *compactness se*, *symmetry se* and *fractal dimension se*-that are not directly linked to the class, indicated by the lack of line or a fine line. This is unsurprising due to the darker colouration observed for these variables in the parallel coordinates plot.

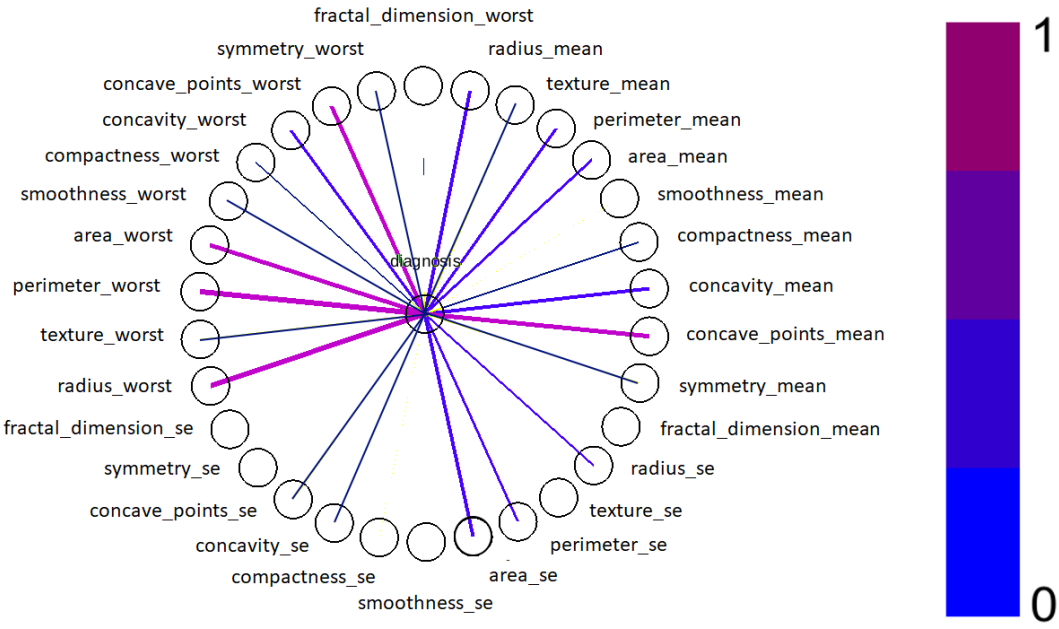

Figure 6.11: Supervised interaction diagram of the SI values between 0 and 1 for the WBCD.

Conversely, we can identify *concave point worst*, *radius worst*, *area worst*, *perimeter worst* and *concave points mean* to have the strongest relationships with the class, all of which have a SI value ≥ 0.50. There are also many other correlations with the class, indicating a highly correlated data set. It is therefore not unexpected that many of these variables share information. For example, it is likely that the average, worst, and standard error for each of the cell features will have some correlation, similarly, for the perimeter, radius and area for the cluster of cells. These correlations are intuitive, but they may not always be as simple to identify. Therefore, it is important to consider the inter-correlations.

In figures 6.12 and 6.13 we display the VIDs for correlations between variables in the WBCD data set. Note, these diagrams do not include the class variable. To reduce the complexity arising from many variable interactions we have restricted the range of SI values shown in each plot using the spin box feature in DataViewer. The ranges chosen were merely determined on the basis of visual clarity.

The connections in the VIDs agree with the expected inter-correlations. We observe correlations between the mean, worst, and standard error of several features and between clusters of variables such as perimeter, radius and area, and concave points, concavity and compactness. These clusters correspond to the two paths extending out from the class variable in the ordered parallel coordinates plot.

The sheer number of inter-correlations illustrates the redundancy within this data set and makes it difficult to identify potentially interesting relationships for exploratory analysis. For the supervised problem, including irrelevant variables and multiple variables with the same information will slow down and confuse a machine learning algorithm. Therefore, it is important to reduce

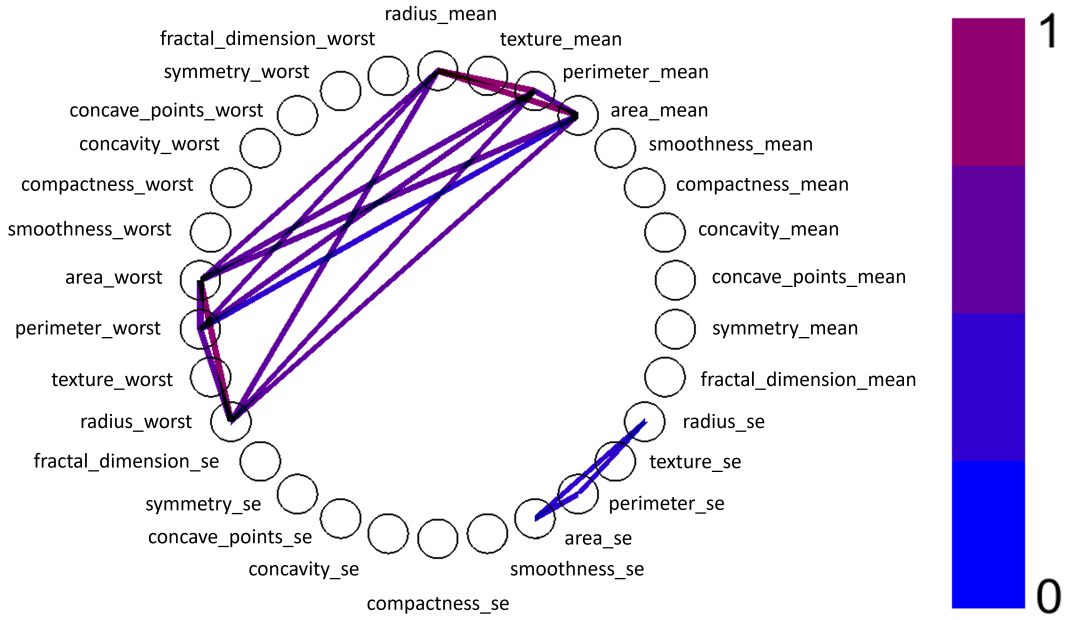

Figure 6.12: A variable interaction diagram showing the inter-correlations of the WBCD data for correlations restricted to $0.35 < SI \leq 1.00$ bits.

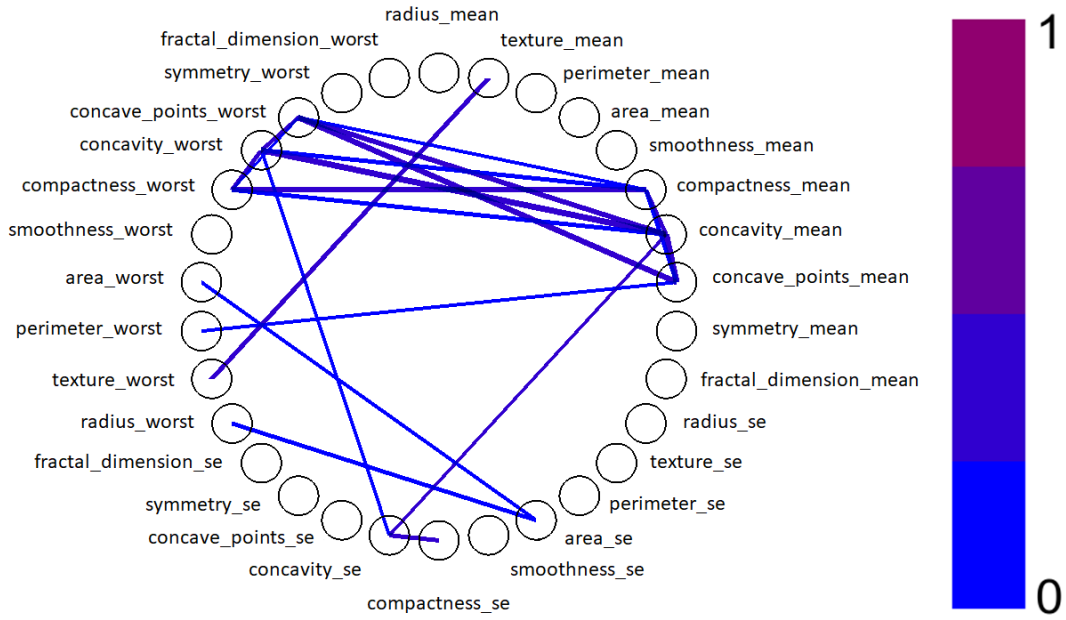

Figure 6.13: A variable interaction diagram showing the inter-correlations of the WBCD data for correlations restricted to $0.20 < SI \leq 0.35$ bits.

the dimensionality to improve the model performance and explainability. We want to remove any variables that do not directly or indirectly contribute to the class as well as relevant variables that are strongly related. The VID identifies these variables.

We observe that the variables not connected to the class in the supervised VID have no significant correlations with relevant variables. We can therefore conclude that the variables, *fractal dimension worst*, *smoothness mean*, *fractal dimension mean*, *texture se*, *smoothness se*, *compact-*

ness se, *symmetry se* and *fractal dimension se* are irrelevant. Removing these from the learning process only marginally affects the model performance compared to the full data set, achieving a accuracy of 97.715% and $\kappa = 0.951$.

We can further reduce the dimensionality by eliminating redundant variables identified by the variable clusters observed. We can simply pick the best variable from each cluster and remove the rest, drastically reducing the amount of redundant information present. In the case of the WBCD data set the variables *perimeter worst* and *concave points worst* are adjacent to the class in the parallel coordinates plot due to their significant SI value with *diagnosis*. Therefore, these variables will be kept in the subset, and those highly correlated to them removed. We will similarly remove other related variables, for example, we will remove *texture worst* and *texture se*, but keep *texture mean* based on the SI values. We continue to constantly pick those correlated to the class, but independent from other variables leaving us with the 5-dimensional subset *perimeter worst*, *concave points worst*, *symmetry worst*, *texture mean* and *smoothness worst*. Applying the simple logistic regression model to this subset makes rules on all but one variable, *symmetry worst*. This model achieves an accuracy of 97.364% and a $\kappa = 0.943$. Again, despite the removal of 17 variables, the model has only taken marginal setbacks.

In the original analysis, the three attributes selected in the final model were *area worst*, *smoothness worst* and *texture mean*, correctly classifying 97% of the test sample. It is easy to see that an equivalent subset can be deduced from the selected variables. As *area worst* and *perimeter worst* were highly correlated to both each other and the class, they are essentially interchangeable in the final subset. For comparison we apply the simple logistic regression to the 3-dimensional subset for both *area worst* and *perimeter worst* obtaining 96.84%, $\kappa = 0.932$ and 97.190% and $\kappa = 0.940$ respectively. *smoothness worst* and *texture mean* are desirable variables because they are not only correlated to class, but they share little information with any other variables in the data set. These can be identified in the VID by looking for variables with the least number of connections to other variables.

6.3.3 Qualitative Bankruptcy Data

The Bankruptcy data set consists of 6 discrete variables and a binary class. As all the variables are discrete this problem is challenging to investigate using classic visualisations. This can be seen for the parallel coordinates, in figure 6.14, where it is difficult to visually discern any data patterns.

The VID in figure 6.15 simultaneously shows the interactions with the class and the inter-correlations between the variables. This visualisation is more insightful to the data. We can instantly identify the relevant variables and how they are connected to one another. These correlations are supported by the CDR values in table 6.1.

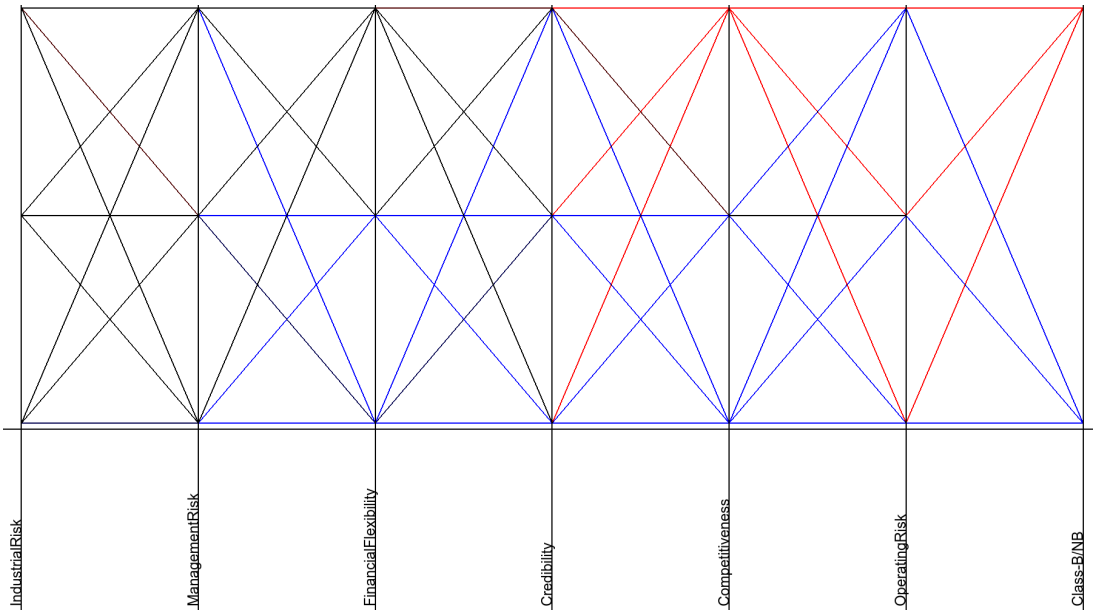

Figure 6.14: Parallel coordinates for the purely discrete Bankruptcy data set.

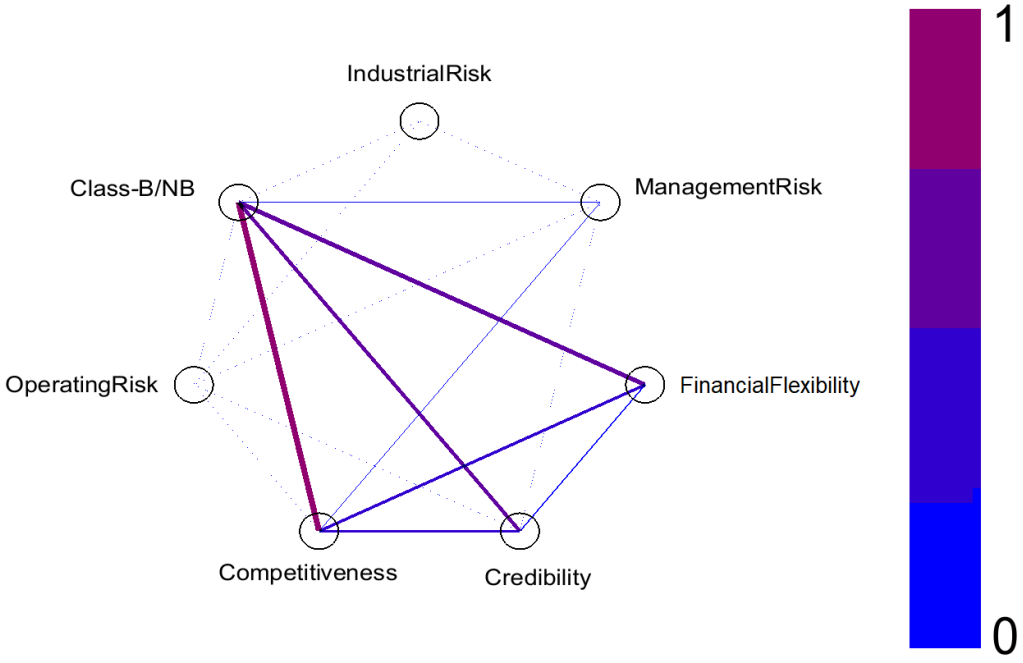

Figure 6.15: The VID for the Bankruptcy data set for SI values between 0 and 1.

For the Bankruptcy data set problem, the objective was to find a set of rules that mimicked the expert's decision making process. In [28] the performance of several learning algorithms were compared. The model which best matched the expert's classification was a genetic algorithm which had a consistency of 94.4% with the experts and a 94% accuracy. It used a composite fitness function to maximise the sum of the predictive accuracy and coverage (how well the condition is universally applied to all cases) to extract a set of 11 prediction rules. This was the least number of rules out of all the models. In the analysis there was no attempt to select the relevant variables, only to test if the prediction matched that of the experts. As a result

Variable	SI (bits)	CDR	Accuracy estimate (%)	*kappa*
Industrial Risk	0.00	0.075	64.0	0.249
Management Risk	0.06	0.217	68.0	0.356
Financial Flexibility	0.57	1.416	91.2	0.823
Credibility	0.46	1.358	89.2	0.776
Competitiveness	0.86	3.166	98.4	0.967
Operating Risk	0.00	0.114	62.8	0.247
Industrial Risk, Financial Flexibility	-	1.666	91.2	0.823
Industrial Risk, Competitiveness	-	2.840	98.4	0.967
Industrial Risk, Operating Risk	-	0.384	70.0	0.397
Credibility, Competitiveness	-	3.584	99.6	0.992
Industrial Risk, Financial Flexibility, Competitiveness	-	3.384	100.0	1.000
Industrial Risk, Competitiveness, Operating Risk	-	2.835	100.0	1.000
Management Risk, Credibility, Competitiveness	-	3.407	99.2	0.984
Financial Flexibility, Credibility, Competitiveness	-	4.040	99.6	0.992
Credibility, Competitiveness, Operating Risk	-	3.402	99.6	0.992

Table 6.1: Comparison of the SI and CDR values with the accuracy estimate and *kappa* statistic from learning algorithms for the Bankruptcy data set. All variables combinations are in relation to the class. Note that mutual information, and consequently SI, is not defined for more than two-dimensions, indicated by the '-'.

all variables were considered and included in the final model, with the suggestion that more variables would improve on the result. This is not necessarily true and could in fact have the opposite effect. As, despite the fact that there are only 6 variables, a learning algorithm can still be affected by the dimensionality.

Despite the fact that all variables appeared in the final rules in [28], three variables (*Financial Flexibility*, *Credibility* and *Competitiveness*) appeared considerably more often than the other variables. From the VID in 6.15 this is unsurprising as the individual variables have the strongest correlations with the class. In addition, this subset had the largest CDR value for three dimensions, see table 6.1. However, it is also apparent from the VID that these variables are correlated with one another. In fact, removing *Financial Flexibility* from this subset does not affect the machine learning model, despite the two-dimensional combination having a smaller CDR value. This is due to the information that *Financial Flexibility* shares with *Credibility* and with *Competitiveness*.

In fact, there is no variable that can be combined with *Competitiveness* and *Credibility* that will improve the performance of the two-dimensional model. However, when the pair is combined with *Management Risk*, a variable correlated to *Competitiveness*, the model degrades. This is the only variable, other than *Credibility* and *Financial Flexibility*, that has a significant correlations with *Competitiveness* as well as being the only variable which degrades the otherwise optimal two-dimensional model.

More accurate three-dimensional models can be obtained using multi-variable interactions. For example, *Industrial Risk* is only beneficial to a model when combined with *Operating Risk* or both *Financial Flexibility* and *Competitiveness*. These variable combinations are apparent from the CDR values. Although the CDR does not directly indicate the optimal combination, it does

direct the user to possible combinations.

Many of these two and three dimensional combinations achieve a better performing model that the full the 6 variable data set. This demonstrates the inappropriate infatuation with more variables which degrade the model due to redundant information.

6.3.4 Particle Data

In [29], Teodorescu applied a genetic algorithm to the Particle data set, which uses the accuracy estimate as a fitness function. The intention was to create an automated process for optimising cuts on variables. With this in mind, we have used the PART decision tree algorithm, as it provides interpretable rules equivalent to cuts. We will use this algorithm throughout the Particle analysis.

Figure 6.16 shows the parallel coordinates plot for the Particle data set. The data is brushed so that the signal instances are green and the background red. From the plot, we can already make some observations about the data. For the variable *Fsig* we can distinguish between the signal and background colours on the one-dimensional projection, suggesting that this variable can discriminate between the classes. Many variables also demonstrate extremely high and low-

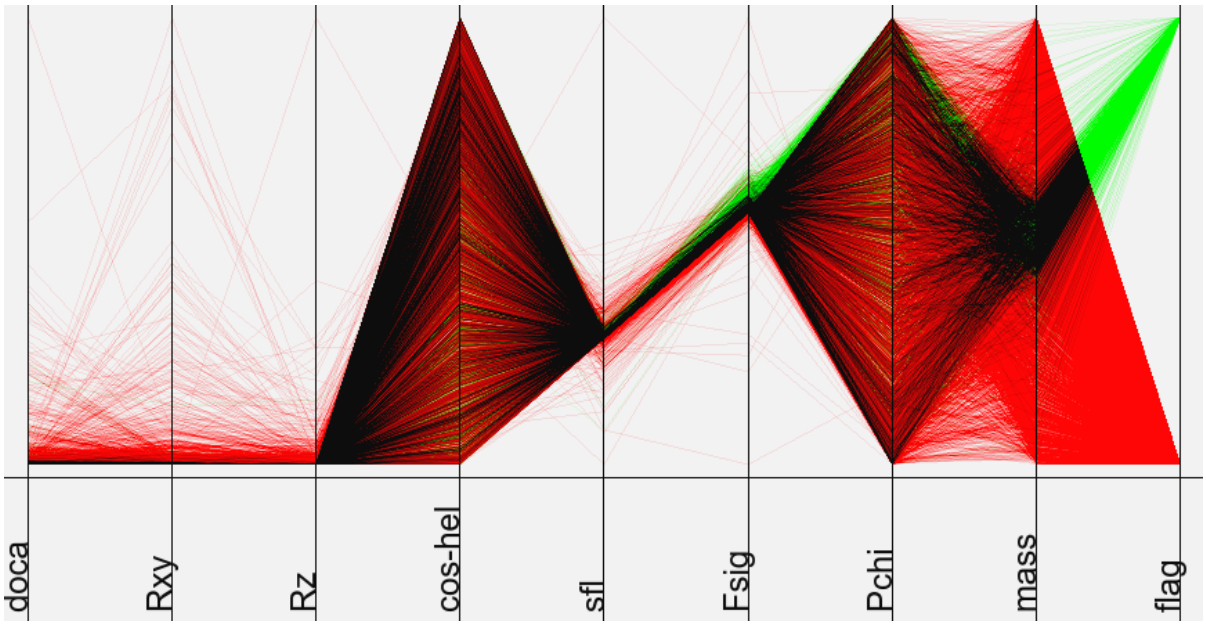

Figure 6.16: Parallel coordinates plot for the Particle data set produced by DataViewer.

density regions, going from thousands of instances to just a few over a small space. This can be seen in the variables *doca*, *Rxy*, *Rz*, *sfl* and *Fsig*, indicative of a distribution with a large kurtosis (tail). Quantising a distribution with a large kurtosis comes with a cost. Narrow bins are necessary to maintain the detail in the sharp, narrow peak. However, this means there are many zero-probability bins in the large tails. This is demonstrated using the fixed-width histogram for the variable *sfl* in figure 6.17 (left). The histogram is not only visually unpleasing, but it is also

difficult to visually extract information when exploring a data set. Alternatively, such variables can be viewed using equiprobable histograms, see section 3.1.2. The equiprobable histogram for the variable *sfl* is shown on the right of figure 6.17. This alternative visualisation is more visually pleasing and insightful for the viewer, and enables them to identify predictive variables easily, when not otherwise possible.

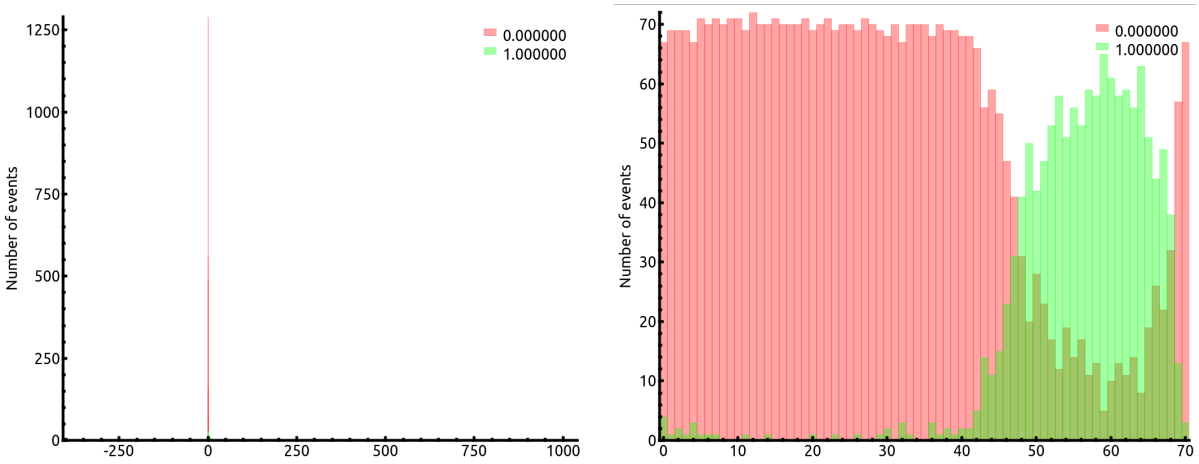

Figure 6.17: The fixed-width histogram (left) and the equiprobable histogram (right) for the Particle data set variable, *sfl* as plotted by DataViewer with the same colouration as in corresponding parallel coordinates plot.

This extends to scatter plots. Plotting the pairwise equiquantised instances can reveal new information and further the understanding of the data set. Consider the scatter plot matrix in figure 6.18. On the left diagonal, the scatter plots are of the original data. Due to the large kurtosis of many variables, the data structures do not fully utilise the space. This is inefficient and visually difficult to identify how well two variables distinguish between the classes. On the right diagonal of figure 6.18 are the scatter plots for the equiquantised instances. Comparatively, these are significantly more insightful to the viewer. Upon observing the equiquantised scatter plots, the discriminating ability of the variable combinations is apparent. However, there is still a need to quantify the discriminating power to avoid subjectivity.

Applying algorithm W to the original or equiquantised data is analogous, as the mutual information is invariant with respect to linear transformations. Some argue that equiquantising is, in fact, more beneficial [30], [31]. Although it is vital that the same method is applied to all continuous variables in the data set. In the case of distributions with a large kurtosis, we recommend that the distributions are equiquantised to avoid regions of sparse density which can otherwise cause the mutual information to be underestimated.

Therefore, the variable interaction diagram for the equiquantised data is shown in figure 6.19. Here, the variable-class and variable-variable correlations are shown on the same plot. This is visually accessible when the data set does not contain too many variables, and there is not an excessive number of correlations to cause visual clutter.

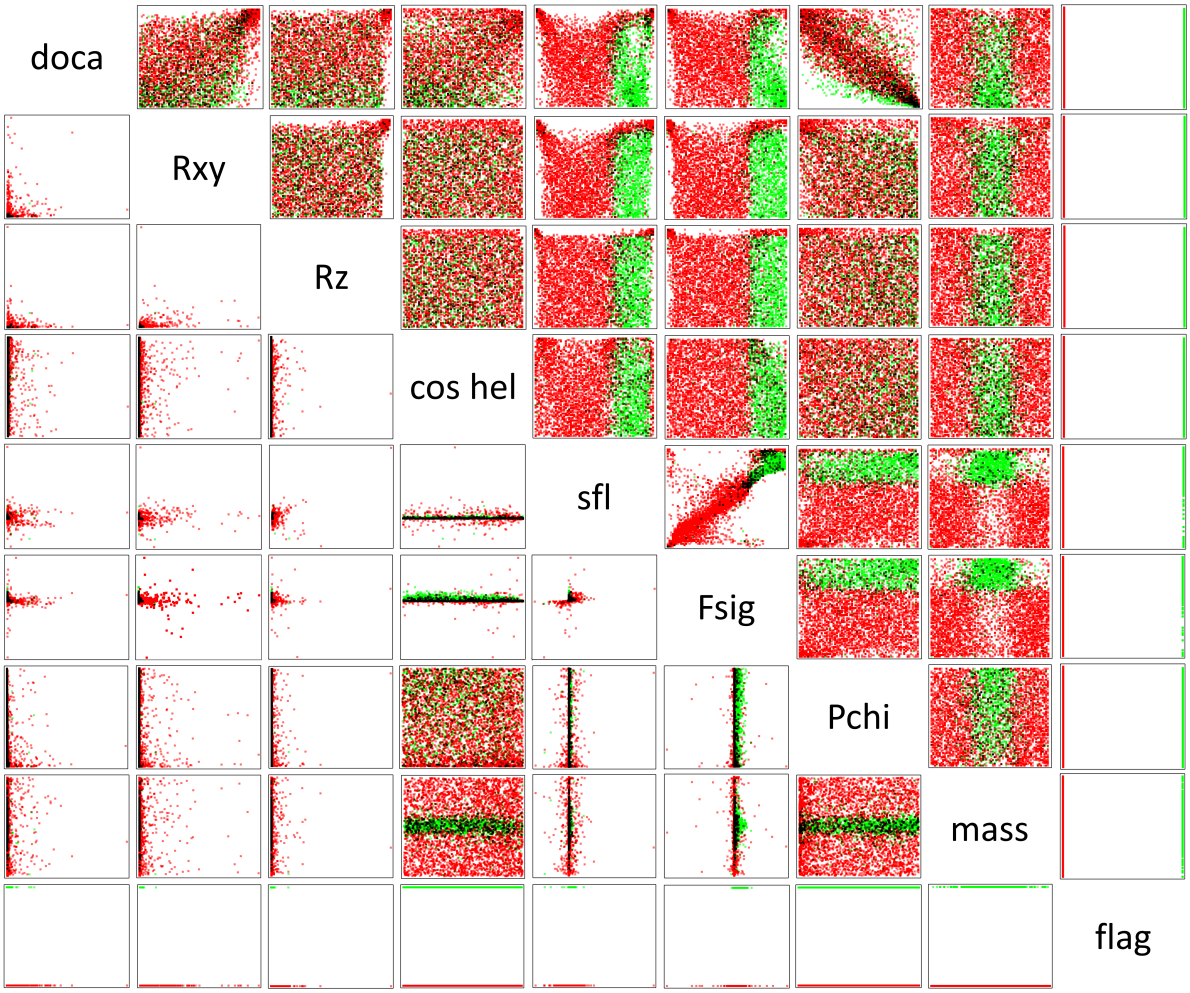

Figure 6.18: Hybrid scatter plot matrix for the Particle data set. On the left diagonal side are the scatter plots for the original data and on the right diagonal side the scatter plots for the equiquantised data.

We can easily identify that the variables *mass*, *Fsig* and *sfl* are the most relevant to the class variable *flag*. This is corroborated by the CDR values for these variables, given in table 6.2. For the subset of variables, {*mass*, *Fsig*, *sfl*} the CDR value is 2.945 bits, corresponding to a $\kappa \approx 1 - 2^{-2.945} = 0.87$. This prediction is consistent with the model obtained using the PART decision tree, which had a $\kappa = 0.86$ and a 94.72% accuracy. Although this is a decrease in both κ and accuracy compared to the full data set, suggesting other conditionally dependent variables are related to the class.

By considering the SI and CDR values for pairwise combinations, we can identify which ones are of interest. Consider the variable combinations {*cos-hel*, *Pchi*} and {*sfl*, *Pchi*}, both of which have zero SI and thus independent. However, unlike {*cos-hel*, *Pchi*}, {*sfl*, *Pchi*} has a non-zero CDR value of 1.61 bits. Therefore, {*sfl*, *Pchi*} can discriminate between the classes. This is confirmed by the scatter plots, which can be found in figure 6.18. For {*cos-hel*, *Pchi*} there are no structural patterns in either the original or the equiprobable scatter plot. However, for {*sfl*, *Pchi*} there is clear structure and distinct regions of the signal and background. Meaning that

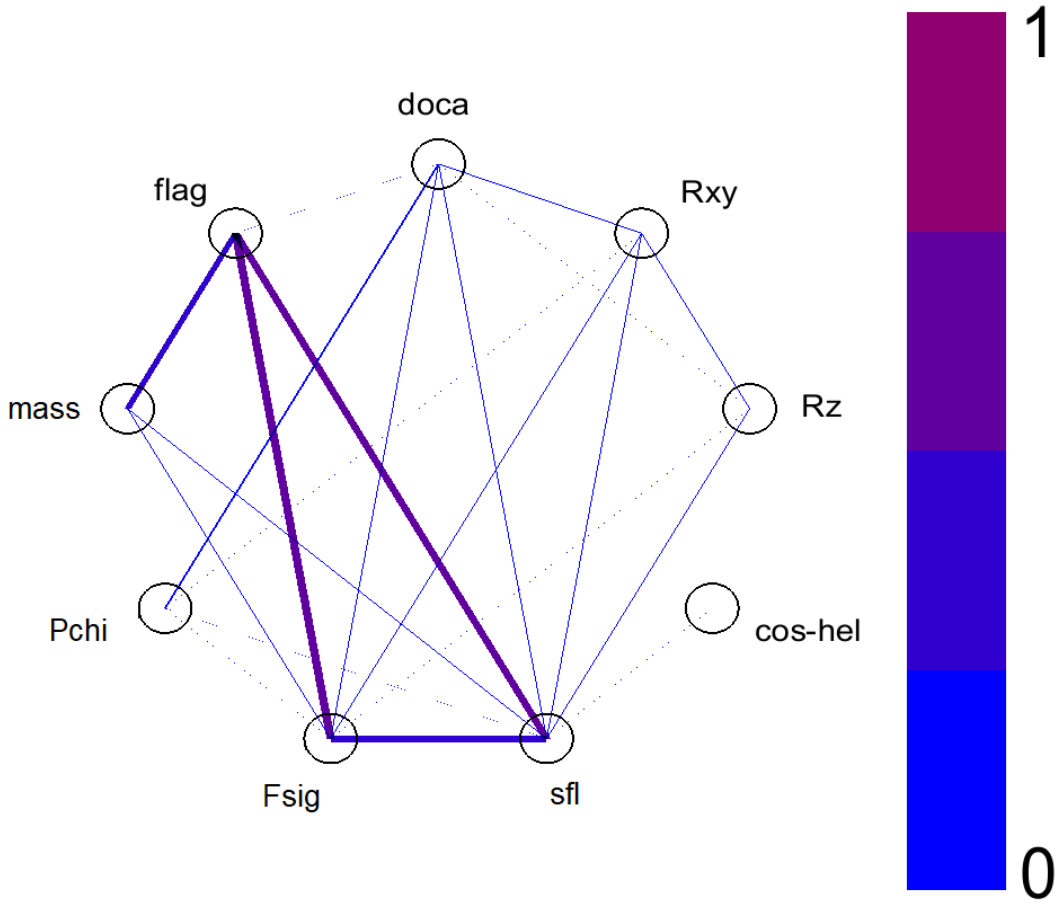

Figure 6.19: The VID for all variable SI values between 0 and 1, including with the class variable for the equiquantised Particle data set.

although *Pchi* is not directly related to the class, when in combination with *sfl* there is a three-way correlation that marginally increases the accuracy obtained when compared with model for *sfl* alone.

The VID shows a significant SI value between the variables *Fsig* and *sfl*, revealing that there is a correlation that may be of interest. The scatter plot, in figure 6.18, confirms this correlation. However, for the classification problem, this suggests that including both variables may not be optimal for machine learning.

We can quantitatively assess which combinations would be suitable for a machine learning model using the CDR values. In two dimensions *Rxy* and *Fsig* has the largest CDR value, indicating a high predictive ability. As with $\{sfl, Pchi\}$, this is also an interesting relationship, as *Rxy* does not show a direct correlation with the class. However, its ability to distinguish between the classes is apparent from the pair's joint distribution. [29], similarly found that simultaneously cutting on the variables *Fsig* and *Rxy* was the most successful, achieving an accuracy of 94.0%. The classification model for this subset similarly supports this conclusion, achieving a $\kappa = 0.871$ and 95.08% accuracy. This two-dimensional subset is the best one can achieve without extending to further dimensions, and is only $\approx 1\%$ less accurate than the model based on the full 8 dimensions.

Variables	SI (bits)	CDR (bits)	Accuracy estimate (%)	*kappa*
Fsig	0.44	1.708	90.40	0.763
sfl	0.41	1.553	89.54	0.735
mass	0.25	0.897	82.72	0.557
doca	0.01	0.133	74.72	0.00
Rxy, Fsig	-	2.405	95.08	0.871
Fsig, mass	-	2.400	94.22	0.845
sfl, mass	-	2.343	93.24	0.820
doca, Fsig	-	2.194	93.32	0.828
Rxy, Fsig, mass	-	3.058	95.96	0.892
sfl, Fsig, mass	-	2.945	94.72	0.859
Rxy, sfl, mass	-	2.885	94.70	0.858
doca, Fsig, mass	-	2.883	94.76	0.860
Rxy, sfl, Fsig, mass	-	3.568	95.52	0.881
doca, Rxy, Fsig, mass	-	3.333	95.92	0.891
doca, sfl, Fsig, mass	-	3.276	94.58	0.856
Rxy, cos-hel, Fsig, mass	-	3.261	96.00	0.893
Rxy, cos-hel, Fsig, Pchi, mass	-	3.215	95.98	0.892
Rxy, Rz, cos-hel, Fsig, Pchi, mass	-	3.215	96.14	0.896
doca, Rxy, cos-hel, Fsig, Pchi, mass	-	3.573	96.02	0.893
All	-	3.961	96.18	0.898

Table 6.2: The SI and CDR values for variable subsets in the Particle data set, as well as the accuracy estimate and *kappa* statistic obtained from a PART decision tree model. The SI values given are for the individual variable with the class, the SI for all other individual variables were zero and '-' indicates that it is not-defined for beyond two dimensions.

From table 6.2 we can see that there are only marginal improvements extending beyond two dimensions. The three-dimensional subset with the largest CDR consists of *Rxy*, *Fsig* and *mass*, obtaining an accuracy only 0.22% less than the full data and 0.88% more than the two-dimensional data set. This is about as well as the Particle data set can do as the CDR value plateaus for increasing dimensions.

6.3.5 Coronary Heart Disease Data

The data set is presented as a parallel coordinate plot in figure 6.20. The purpose of this data collection was to identify the risk factors of heart disease in a high prevalence region. When it comes to risk-factors the relationship between the variable and the outcome can be weak and only increase the probability of developing the condition. Simultaneously, it can also be the case that any variable on its own is not enough to give an individual an increased chance of developing CHD and only when considered in conjunction with other factors can we identify those who are at risk. Resulting in small mutual information values. These types of data sets are common in the medical field. These features can be seen in the parallel coordinates plot as, unlike in previous examples, there are few regions which clearly differentiate the two classes. This is reflected when classification algorithms are applied to the data set. The logistic-regression model performed the best, achieving an accuracy estimate of 71.212% and $\kappa = 0.336$ on the full data set, considerably less than previous examples, however more than expected from the CDR value of 0.163 bits.

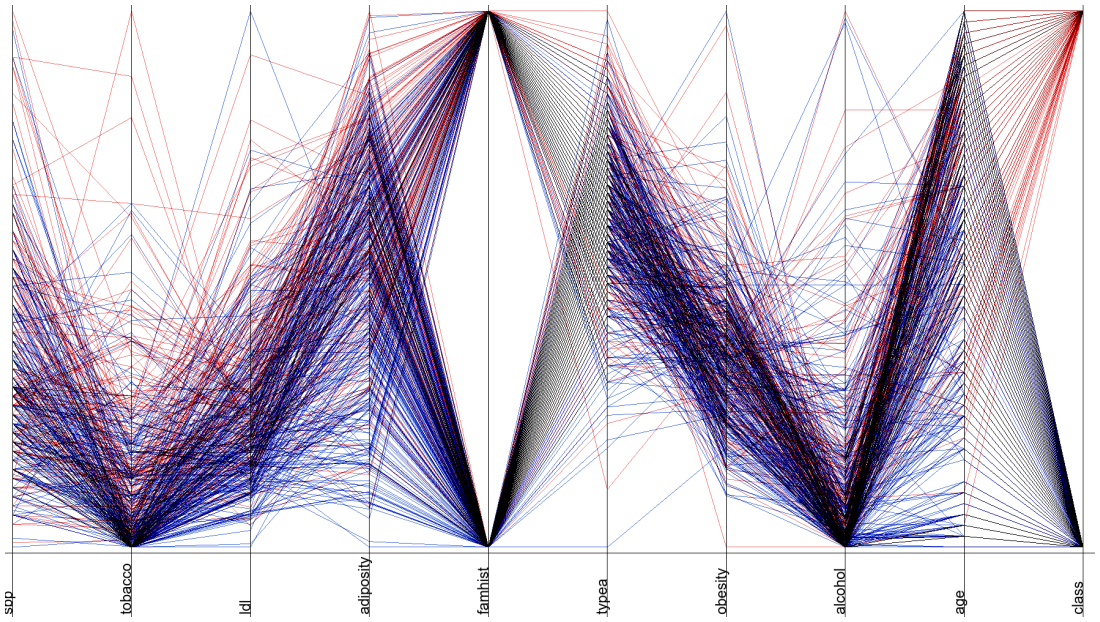

Figure 6.20: Parallel coordinates plot for the coronary heart disease data set coloured such that the control group is blue and the coronary heart disease patients are red.

So how does one interpret such a data set? Here, we first consider the relevancy of the individual variables. The SI value of each of the variables with the class are visualised in the supervised VID in figure 6.21. The SI values for this data are significantly less than what we have seen before. For perverse data sets, such as this one, an increased number of iterations is recommended in order to stabilise the values and reduce the errors. Here, we used 250 iterations. The variable with the largest SI with the class being *age* with an $SI = 0.06 \pm 0.01$. This is the optimistic error estimate and it could in fact fall between 0.01 and 0.03, as discussed in section 4.6. For data sets with weak correlations of the same order as the error we can consider the SI alongside the CDR to confirm these correlations. In the case of *age* the CDR supports the conclusion that this is the most relevant variable, followed by *family history*, *tobacco* and *ldl* in descending order of relevance. We could create a model based off of these 4 variables yielding an accuracy estimate of 72.078% and $\kappa = 0.358$. From this simple visualisation we have already improved on the classification model, however, the *kappa* still indicates a poor classification.

We can also evaluate the predicative abilities of the individual variables by considering the Kullback-Leibler divergence. The CDR values, shown in table 6.3, suggest similar conclusions to the SI with *age* once again ranked as the most predictive of the class. Alongside the CDR and SI values are the accuracy estimate and Cohen's *kappa* values for the individual variables evaluated using the logistic-regression algorithm. Interestingly, *family history*, which was identified as relevant by both the CDR and SI, performs worse than just classifying all of the instances as the control group. In addition the CDR values indicate that *adiposity* and *sbp* may also have some predictive abilities of the class, this is supported by the marginal improvements to the accuracy estimates for these variables, although the *kappa* values are very small.

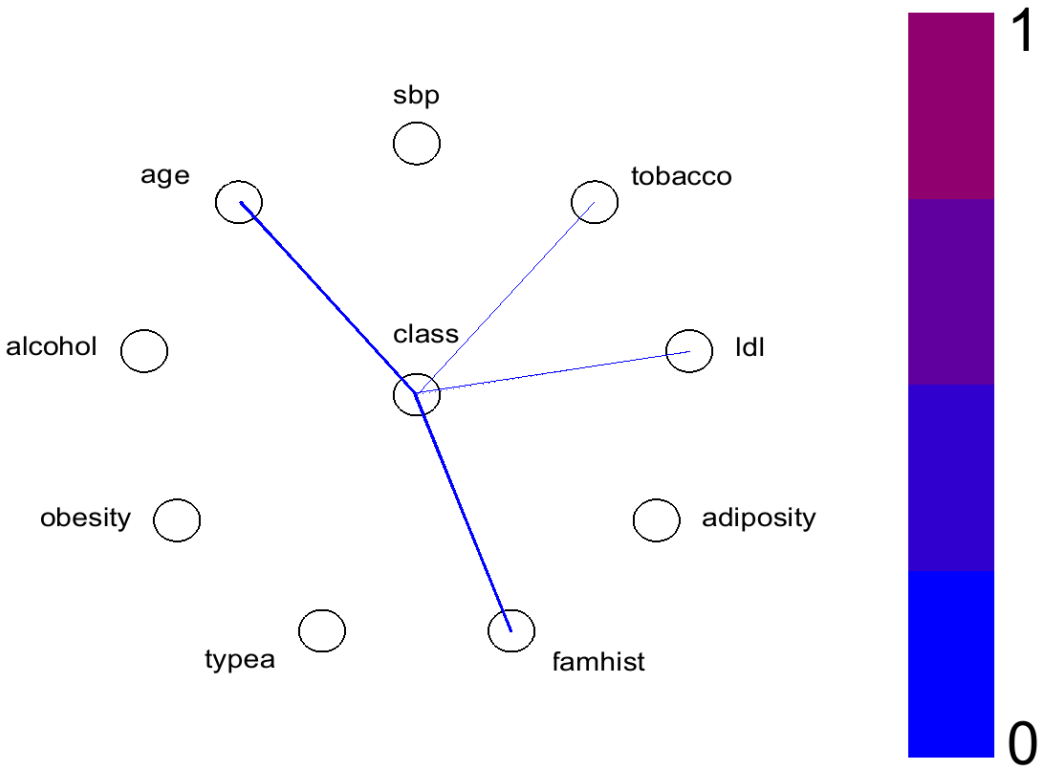

Figure 6.21: The supervised VID of the CHD data set for SI values between 0 and 1.

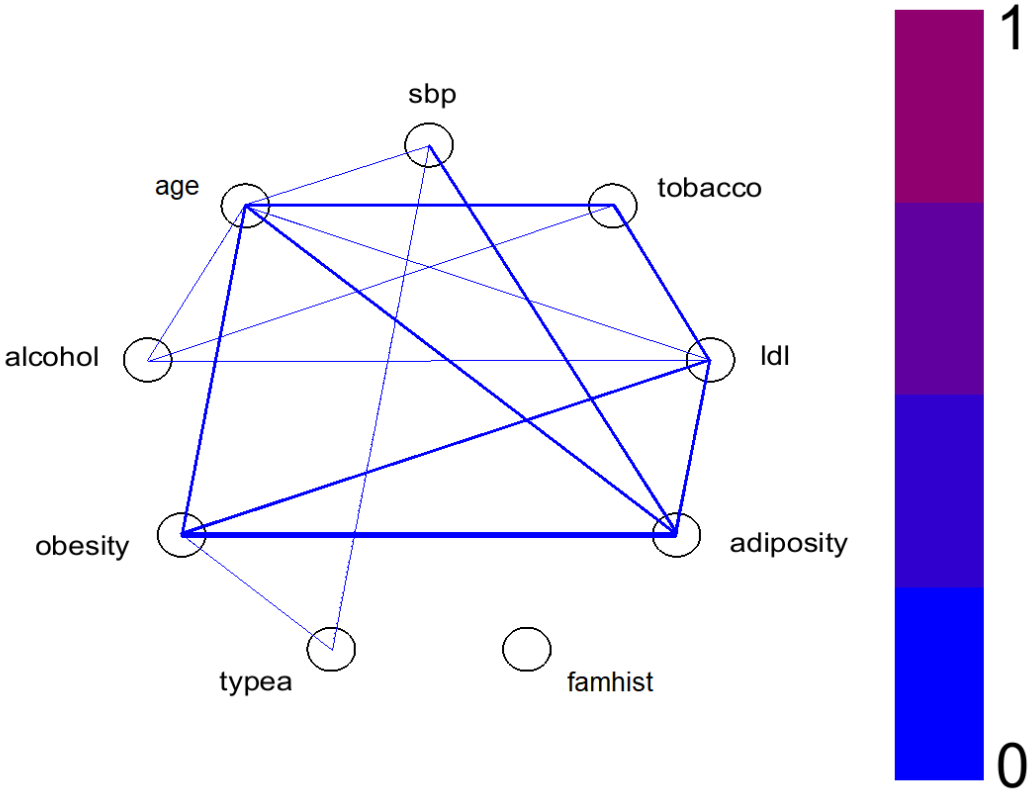

Figure 6.22: The unsupervised VID for all the variable in the CHD data set, excluding the class variable for SI values between 0 and 1.

Variable	SI (bits)	CDR (bits)	Accuracy estimate	*kappa* (%)
age	0.06	0.270	67.316	0.232
tobacco	0.01	0.184	69.913	0.220
family history	0.03	0.142	62.338	0.070
ldl	0.01	0.109	66.883	0.138
adiposity	0.00	0.074	67.100	0.139
sbp	0.00	0.059	66.667	0.093
alcohol	0.00	0.006	65.152	−0.004
type A	0.00	0.000	65.368	0.000
obesity	0.00	0.000	64.935	−0.005
ldl, family history, age	-	0.333	73.16	0.374
tobacco, ldl, family history, age	-	0.120	72.078	0.358
tob., ldl, adip., fam. hist., type A, age	-	0.404	73.810	0.398

Table 6.3: CDR and similarity index values for individual variables with the class variable of the Coronary Heart Disease data set, ordered from the highest to the lowest CDR value. All variables combinations are in relation to the class. Note that mutual information, and consequently SI, is not defined for more than two-dimensions, indicated by the '-'.

At this stage it will be valuable to consider the inter-correlations between the variables, these are visualised in figure 6.22. We observe that *obesity* and *adiposity* are the most correlated variables in the data set, this is not unexpected as both are measures of how overweight an individual is. Including two closely related variables, such as these, can be confusing for the classification algorithm. This is clearly demonstrated in this example as by removing the variable *obesity*, which was evaluated to be less relevant than *adiposity*, we improve the accuracy of the model to 73.160% with $\kappa = 0.380$, this is a $+1.95\%$ improvement on the model fitted to the full data set of variables.

In the unsupervised interaction diagram we observe that *family history*, a variable identified to be relevant by the proposed method, does not exhibit any correlations with the other variables. This indicates that this variable contains information about the class that no other variable does and will therefore be invaluable in a variable subset. The remainder of the variables identified to be relevant to the class, on the other hand, demonstrate relationships between all possible combinations with the weakest relationship being between *age* and *ldl*. This potentially suggests that *tobacco* may be superfluous in the variable subset. To evaluate this we apply the logistic-regression algorithm to the relevant variables minus *tobacco*. The resultant model gives an accuracy estimate of 73.160% and $\kappa = 0.374$. This achieves the same accuracy estimate and only a marginal reduction in *kappa* compared to the 8 variable subset we previously looked at. This drastically simplifies the classification model while maintaining performance.

Using a wrapper subset evaluator in WEKA we implemented an exhaustive search on the full set of variables to find the optimal variable subset using the logistic-regression algorithm. The most predictive subset was found to consist of 6 variables: *tobacco*, *ldl*, *adiposity*, *family history*, *type A* and *age*, yielding an accuracy estimate of 73.810% and $\kappa = 0.398$. This is the most accurate model that this algorithm can build from this data set. Interestingly, *type A*, which was included in the 6 variable subset, was evaluated to be irrelevant by the CDR as well as the SI, achieving

a $\kappa = 0.000$ when considered as a solitary variable. Nonetheless, it was found to improve the model, however, only when considered in conjunction with *tobacco* and *adiposity*.

Repeating the exhaustive search with *obesity* and *type A* removed, two variables which we identified to be irrelevant, comes to the conclusion that *ldl*, *family history* and *age* achieves the most accurate model when *type A* is not available. This is a marginal deterioration of -0.649%. Nonetheless, both of these subsets achieve more accurate models than using the full data set of variables. For such perverse data sets, reducing the dimensional complexity can improve on the accuracy of any subsequent model. Interestingly, *family history*, the variable which performed badly when used to classify instances on its own, was found necessary to the subset. Demonstrating the ineffectiveness of attribute ranking alone.

6.3.6 Prostate Cancer Data

This Prostate data set is different from the others, in that it has a continuous class variable, *lpsa*, with 85 distinct values out of the 97 sample size. Therefore, the class has been brushed so that a colour gradient corresponds to the class value. From the parallel coordinates plot in figure 6.23, we can see that the variables *svi* and *gleason* are discrete variables, and that *pgg45* can only take on set values, potentially due to limited precision. Correlations between discrete variables can sometimes be difficult to visually identify. Nonetheless, a positive correlation between *gleason* and *pgg45* is observed. After some playing with the axes ordering a number of other variables correlations become apparent, indicated by the characteristic parallel lines. Some correlations with the class are also evident from distinguished areas of colour, for example the variables *lcavol* and *gleason*, which show a clear separation. In figure 6.24 we show the scatter plots for *lcavol* and the class variable, *lpsa* on the left, and *lcp* and *lcavol* on the right. From the parallel coordinates we can see that the variable *lcp* has the majority of instances take on a single value, while the remaining instances show correlation with *lcavol*. This is reflected in the scatter plot.

To begin to construct a idea of how these variable interact we consider the VIDs for supervised and unsupervised problems, shown in figures 6.25 and 6.26 respectively.

The supervised VID illustrates a number of significant correlations with the class variable. The fact that the class is continuous does not affect this feature. As observed from the parallel coordinates plot *lcavol* and *gleason* demonstrate correlations with the class, along with *lcp*, *svi* and *lweight*. This is confirmed by the correlation coefficients of the individual variables. We see that there are a number of variables - age, lbph and pgg45, not directly linked to the class. These could confuse a classification problem or at the very least add very little to the model.

Previously, we used linear regression to evaluate the Prostate data set due to it superior performance as well as its ability to deal with continuous class variables. When applied to the full data set the linear regression model contained 7 out of the 8 variables (removing *gleason*) achieving

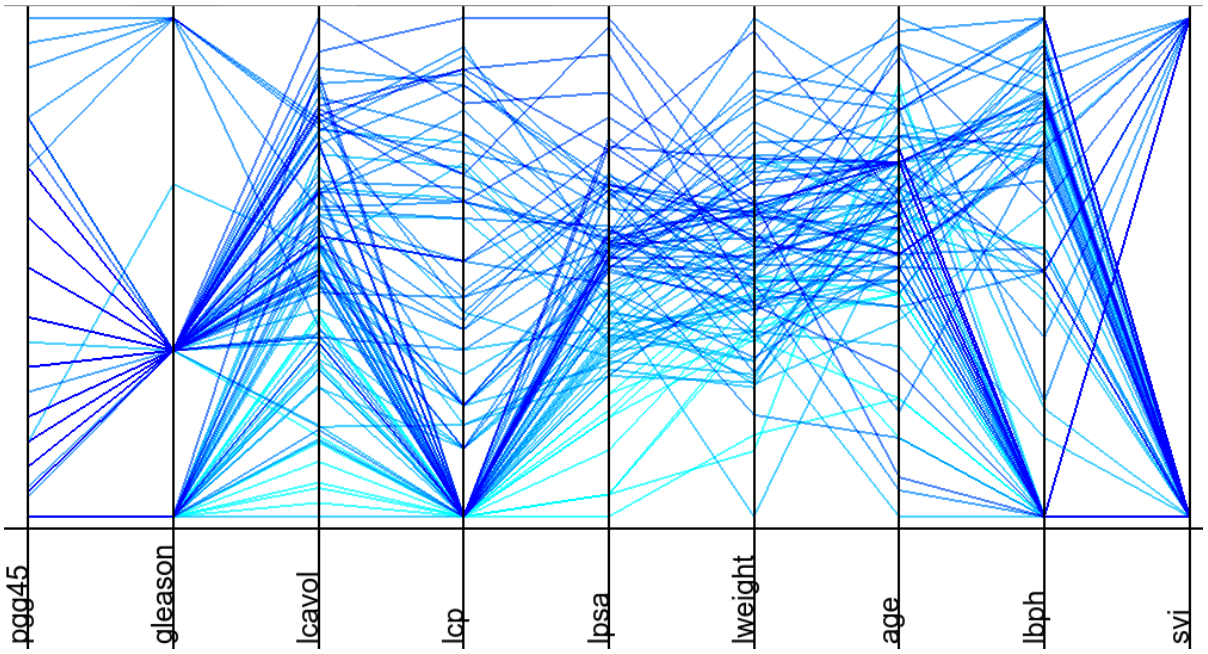

Figure 6.23: Parallel coordinates plot for the Prostate data set.

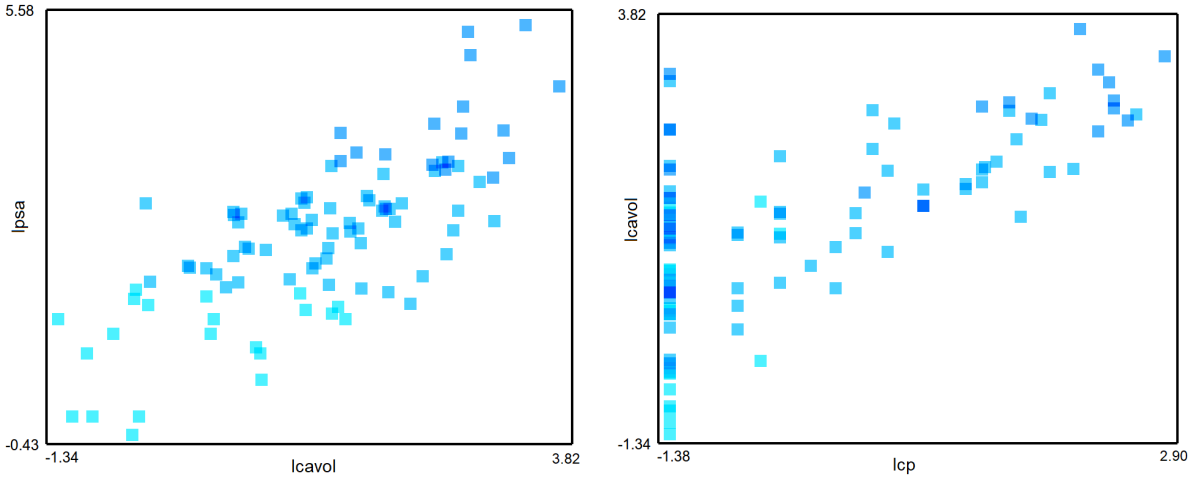

Figure 6.24: Scatter plots for the Prostate data set. On the left is *lcavol* vs the class variable *lpsa* and on the right *lcp* vs *lcavol*. The data has been brushed with the same colour gradient as in the parallel coordinates plot.

a correlation coefficient of 0.7626 with the true class variable. However, by removing the variable not directly linked to the class the resultant linear regression model achieved a correlation coefficient of 0.776, a marginal improvement compared to the full data set. Interestingly, this model only consisted of *lcavol*, *lweight* and *svi*. This is because when *lcavol* is in the model, the variables *lcp* and *gleason*, which were previously identified as relevant, are no longer significant. Here we can see that from this one diagram we have quickly reduced the dimensionality of the model while simultaneously increasing the accuracy. The exhaustive search confirms that this three-dimensional subset is the best that one can achieve for this data set. From the unsupervised VID, it is apparent that many of these variables share information, thus, there is no benefit to including them all in the model. The variable *lweight*, on the other hand, is only weakly correlated

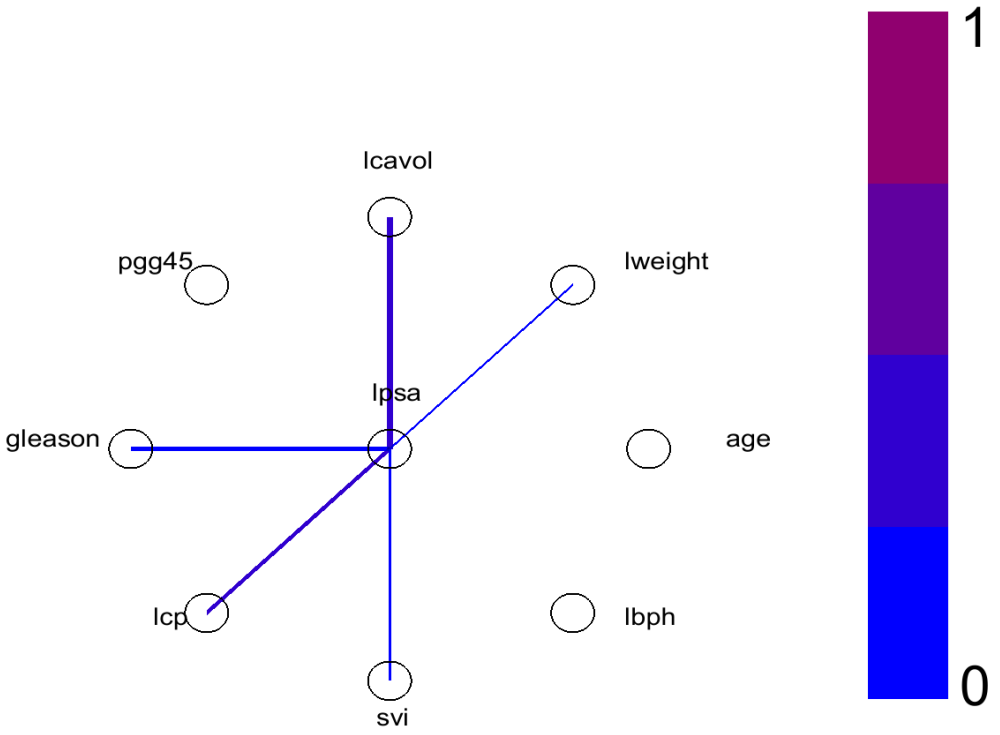

Figure 6.25: Supervised VID for the Prostate data set for SI values between 0 and 1.

to the other relevant variables making it an ideal choice in an optimised subset.

6.4 Discussion

Many factors affect the success of a data analysis task. By taking an exploratory approach, one can understand and influence the outcome through better knowledge of the data. Exploratory analysis is approachable from two directions. Visual analysis focuses on visualisations to help a user understand and interpret. One can also explore the data through statistical tests and queries. By tackling a data analysis problem two-fold, the data structures quickly become apparent.

This chapter presented several case studies and empirically analysed the suitability of algorithm W on real-world data. First, algorithm W calculated the SI and CDR values, which successfully detected variable-class and variable-variable interactions. Next, the data and SI values were presented visually in a comprehensible form for human interpretation. We then demonstrated their use in untangling complex data structures and understanding variable combination choices for feature selection and explaining machine learning models.

Feature selection and explainable machine learning come hand-in-hand. A suitable feature selection reduces the dimensionality, which in turn simplifies the model, aiding understanding. A good variable subset includes variables that are correlated with the class and uncorrelated with each other. However, the relevancy of a variable beyond pair-wise interactions with the class

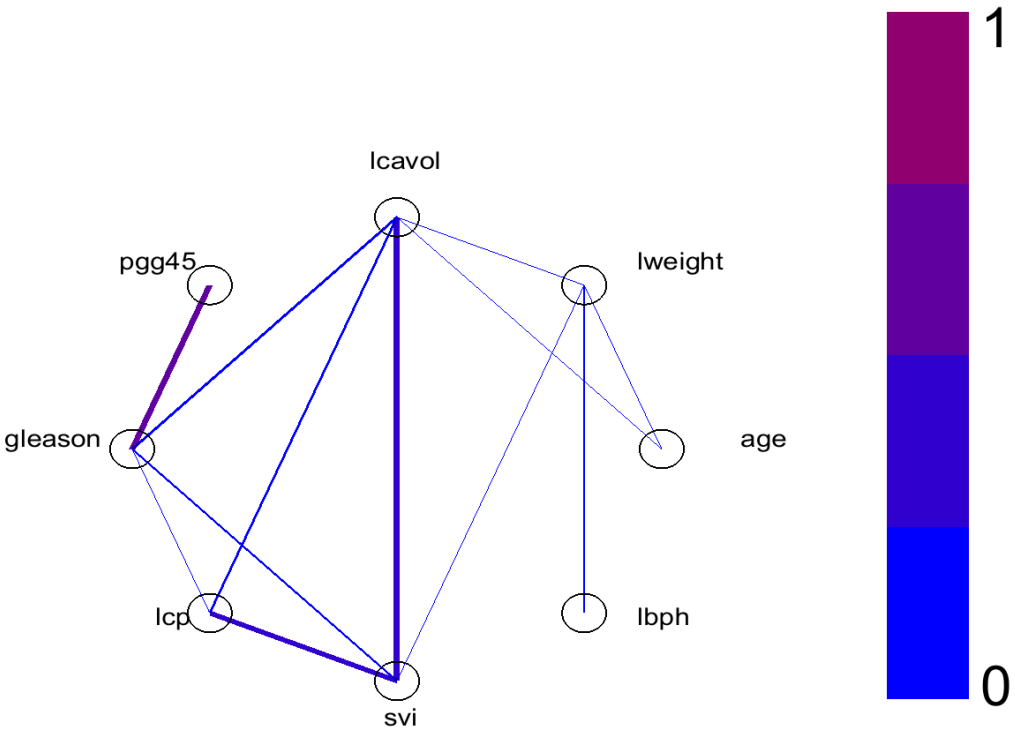

Figure 6.26: The unsupervised VID for the Prostate data set for SI values between 0 and 1.

must be considered. The SI and CDR values, where applicable, were agreeable in their analysis. The SI values successfully ranked pair-wise interactions, which we confirmed via visualisations and machine learning tasks — as a result, guiding the user to interesting correlations while bypassing the need for an exhaustive search. Similarly, the CDR values successfully identified key variable subsets that performed well to train a classification model. Thus, avoiding the pitfall of only considering relevance.

All these case studies were carried out using the exploratory data analysis application, DataViewer. While the need for a quantitative method for identifying interactions is paramount, human comprehension thrives on visualisations. The use of standard one- or two-dimensional visualisations, such as histograms and scatter plots, still prove valuable when used in conjunction with the idea of co-plots or linked brushing. However, in the era of big data, the ability to consider multiple dimensions and interactions concurrently is more necessary than ever. Parallel coordinates did some way in providing this, and with brushing, enabled detection of probability distributions and variable correlations. However, only adjacent axes are comparable, and the visualisation is vulnerable to subjective interpretations.

To this aim, we presented a VID as a visualisation method for unsupervised and supervised problems. The VID is a compact summary of the variable interactions and can identify relevancy and redundancy in clusters of variables and connect pairs of interacting variables. We show numerous examples of the use of the VID to summarise interactions. In future, when there are many variables, it may be advantageous to add shading to the VID, so it is easier to identify

which variable label corresponds to which node. It would also be desirable to add interactive abilities into the VID, such as clicking on a variable node would reveal the "supervised" VID for the interactions with that variable. Similarly, clicking on connections could highlight that connection and subsequent connections to track dependencies through the data.

We presented an equiquantisation technique based on efficient coding ideas in information theory. The equiquantised-visualisations illustrate the same information as the original data, but the data structures are amplified by applying a linear transform. This amplification occurs due to dense information regions spreading out to use the marginal space more efficiently. Thus, by sacrificing the preservation of the distribution shape, the information estimates for variables with extremes of high and low-density regions are improved, and dependencies otherwise hidden are revealed. Through the examples given, equiprobable histograms and scatter plots prove an unmatched commodity for visualising correlations otherwise undetectable to a viewer.

References

[1] S. García, J. Luengo, and F. Herrera, "Feature selection," *Intelligent Systems Reference Library*, vol. 72, no. 6, pp. 163–193, 2015, ISSN: 18684408. DOI: 10.1007/978-3-319-10247-4_7.

[2] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003, ISSN: 00032670. DOI: 10.1016/j.aca.2011.07.027.

[3] G. Doquire and M. Verleysen, "A comparison of multivariate mutual information estimators for feature selection," *ICPRAM 2012 - Proceedings of the 1st International Conference on Pattern Recognition Applications and Methods*, vol. 1, pp. 176–185, 2012. DOI: 10.5220/0003726101760185.

[4] A. Jović, K. Brkić, and N. Bogunović, "A review of feature selection methods with applications," *38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 1200–1205, 2015, ISSN: 18790534. DOI: 10.1016/j.compbiomed.2019.103375.

[5] A. Jung and P. H. Nardelli, "An Information-Theoretic Approach to Personalized Explainable Machine Learning," *IEEE Signal Processing Letters*, vol. 27, pp. 825–829, 2020, ISSN: 15582361. DOI: 10.1109/LSP.2020.2993176. arXiv: 2003.00484.

[6] S. Morgenthaler, "Exploratory data analysis," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 1, no. 1, pp. 33–44, 2009, ISSN: 19395108. DOI: 10.1002/wics.2.

[7] D. B. Carr, R. J. Littlefield, W. L. Nicholson, and J. S. Littlefield, "Scatterplot matrix techniques for large n," *Journal of the American Statistical Association*, vol. 82, no. 398, pp. 424–436, 1987. DOI: 10.1080/01621459.1987.10478445. eprint: https://doi.org/10.1080/01621459.1987.10478445. [Online]. Available: https://doi.org/10.1080/01621459.1987.10478445.

[8] B. Alpern and L. Carter, "The hyperbox," *Proceeding Visualization '91*, pp. 133–139, 1991.

[9] E. Kandogan, "Star coordinates: A multi-dimensional visualization technique with uniform treatment of dimensions," *In Proceedings of the IEEE Information Visualization Symposium, Late Breaking Hot Topics*, vol. 650, pp. 9–12, 2000. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.4.8909&rep=rep1&type=pdf.

[10] J. Sansen, G. Richer, T. Jourde, F. Lalanne, D. Auber, and R. Bourqui, "Visual Exploration of Large Multidimensional Data Using Parallel Coordinates on Big Data Infrastructure," *Informatics*, vol. 4, no. 3, p. 21, 2017, ISSN: 2227-9709. DOI: 10.3390/informatics4030021.

[11] A. Inselberg, "The plane with parallel coordinates," *The Visual Computer*, vol. 1, no. 4, pp. 69–91, 1985, ISSN: 01782789. DOI: 10.1007/BF01898350.

[12] J. Yang, W. Peng, M. O. Ward, and E. A. Rundensteiner, "Interactive hierarchical dimension ordering, spacing and filtering for exploration of high dimensional datasets," *Proceedings - IEEE Symposium on Information Visualization, INFO VIS*, pp. 105–112, 2003, ISSN: 1522404X. DOI: 10.1109/INFVIS.2003.1249015.

[13] E. J. Wegman, "Hyperdimensional data analysis using parallel coordinates," *Journal of the American Statistical Association*, vol. 85, no. 411, pp. 664–675, 1990, ISSN: 1537274X. DOI: 10.1080/01621459.1990.10474926.

[14] L. F. Lu, M. L. Huang, and J. Zhang, "Two axes re-ordering methods in parallel coordinates plots," *Journal of Visual Languages and Computing*, vol. 33, pp. 3–12, 2016, ISSN: 1045926X. DOI: 10.1016/j.jvlc.2015.12.001. [Online]. Available: http://dx.doi.org/10.1016/j.jvlc.2015.12.001.

[15] L. Lu, W. Wang, and Z. Tan, "Double-Arc Parallel Coordinates and its Axes re-Ordering Methods," *Mobile Networks and Applications*, 2020, ISSN: 15728153. DOI: 10.1007/s11036-019-01455-9.

[16] J. Alsakran, N. Alhindawi, and L. Alnemer, "Parallel coordinates metrics for classification visualization," *2016 7th International Conference on Information and Communication Systems, ICICS 2016*, pp. 7–12, 2016. DOI: 10.1109/IACS.2016.7476078.

[17] E. J. Wegman and Q. Luo, "High Dimensional Clustering Using Parallel Coordinates and the Grand Tour," pp. 93–101, 1997. DOI: 10.1007/978-3-642-59051-1_10.

[18] A. Jakulin and I. Bratko, "Quantifying and Visualizing Attribute Interactions," 2003. arXiv: 0308002 [cs]. [Online]. Available: http://arxiv.org/abs/cs/0308002.

[19] M. Muja and D. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration.," vol. 1, Jan. 2009, pp. 331–340.

[20] M. Matsumoto and T. Nishimura, "Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 8, no. 1, pp. 3–30, 1998.

[21] G. Marsaglia, "The marsaglia random number cdrom including the diehard battery of tests of randomness," *http://www. stat. fsu. edu/pub/diehard/*, 1996.

[22] R. Rosipal, "Kernel-Based Regression and Objective Nonlinear Measures to Assess Brain Functioning," Ph.D. dissertation, 2001, p. 106.

[23] R. E. Valdes-Perez and R. C. Conant, "Information Loss Due to Data Quantization in Reconstructability Analysis," *International Journal of General Systems*, vol. 9, no. 4, pp. 235–247, 1983, ISSN: 15635104. DOI: 10.1080/03081078308960824.

[24] K. Hlaváčková-Schindler, M. Paluš, M. Vejmelka, and J. Bhattacharya, "Causality detection based on information-theoretic approaches in time series analysis," *Physics Reports*, vol. 441, no. 1, pp. 1–46, 2007, ISSN: 03701573. DOI: 10.1016/j.physrep.2006.12.004.

[25] W. N. Street, W. H. Wolberg, and O. Mangasarian, "Nuclear Feature Extraction For Breast Tumor Diagnosis," *International Symposium on Electronic Imaging: Science and Technology*, vol. 1905, pp. 861–870, 1993.

[26] O. L. Mangasarian, W. N. Street, and W. H. Wolberg, "Breast Cancer Diagnosis and Prognosis via Linear Programming," *INFORMS*, vol. 43, no. 4, pp. 570–577, 1995.

[27] M. Ankerst, S. Berchtold, and D. A. Keim, "Similarity clustering of dimensions for an enhanced visualization of multidimensional data," in *Proceedings IEEE symposium on information visualization (Cat. No. 98TB100258)*, IEEE, 1998, pp. 52–60.

[28] M. J. Kim and I. Han, "The discovery of experts' decision rules from qualitative bankruptcy data using genetic algorithms," *Expert Systems with Applications*, vol. 25, no. 4, pp. 637–646, 2003, ISSN: 09574174. DOI: 10.1016/S0957-4174(03)00102-7.

[29] L. Teodorescu, "Gene expression programming approach to event selection in high energy physics," *IEEE Transactions on Nuclear Science*, vol. 53, no. 4, pp. 2221–2227, 2006, ISSN: 00189499. DOI: 10.1109/TNS.2006.878571.

[30] M. Paluš, "Testing for nonlinearity using redundancies: quantitative and qualitative aspects," *Physica D: Nonlinear Phenomena*, vol. 80, no. 1-2, pp. 186–205, 1995, ISSN: 01672789. DOI: 10.1016/0167-2789(95)90079-9.

[31] M. Vejmelka, "Quantifying interactions between complex oscillatory systems: a topic in time series analysis," Ph.D. dissertation, 2008.

Chapter 7

Conclusions and Future Work

In this thesis, we explored various aspects of the estimation and applications of information-theoretic statistics. Estimating entropy and mutual information is an essential task with many data analysis and machine learning applications. In particular, exploratory data analysis or features selection methods use mutual information criteria and would greatly benefit from a reliable way to estimate information measures for a wide range of applications. Unfortunately, most information estimators apply to either purely continuous or discrete samples, making them impractical in real-world cases. Numerous attempts have been made to quantise or add noise to samples to rectify this, to no avail. New algorithms in chapter 4 have solved this problem.

The main aim of this thesis was to develop a robust method for detecting nonlinear relationships that could be used for feature selection and classification tasks. Through our research, we propose a novel algorithm that uses a noisy resampling technique to identify pairwise correlations for finite samples. We implement these ideas into a new visualisation software package to improve understanding of data structures and their impact on machine learning. Our work suggests that information theory provides a quantitative understanding of data limitations and has many applications in data analysis when estimated reliably. Thus, a reliable approach to estimating entropy and other information measures will have knock-on effects in many data-centric disciplines.

7.1 Main contributions

Our main contributions will be summarised here. Note that the figures shown here are repeated from previous sections to provide a compact summary of the key findings from this work.

- **Density estimation:** We have analysed the behaviour of the Shannon entropy of continuous finite samples for different quantisations. Our work indicates that the Shannon entropy can be fixed at $H = \log_2(N)/2$, independent of the underlying probability distribution, see figures 7.1 and 7.2. This observed independence contrasts with traditional quantisation methods that aim to minimise measures, such as the MISE of the density estimate, requiring prior knowledge of the underlying pdf. In addition, quantisation methods often require

subjective human input for smoothing, resulting in unpredictable biases in the entropy estimate. Thus, it is advantageous to determine the optimal bin width by setting the Shannon entropy to a constant for all variables, removing these subjective biases. Furthermore, this method is non-parametric and does not make any assumptions about the underlying pdf.

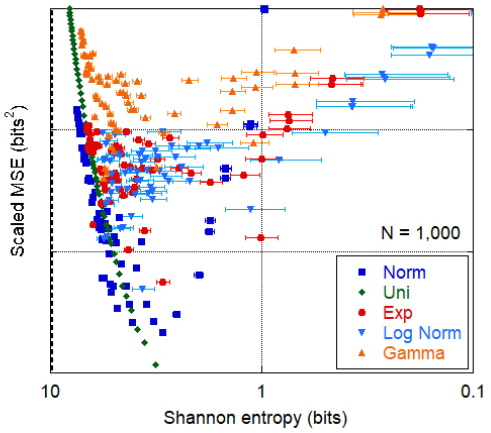

Figure 7.1: A normalised graph demonstrating the distribution-independent minimum for MSE of the entropy estimate for quantised one-dimensional continuous distributions.

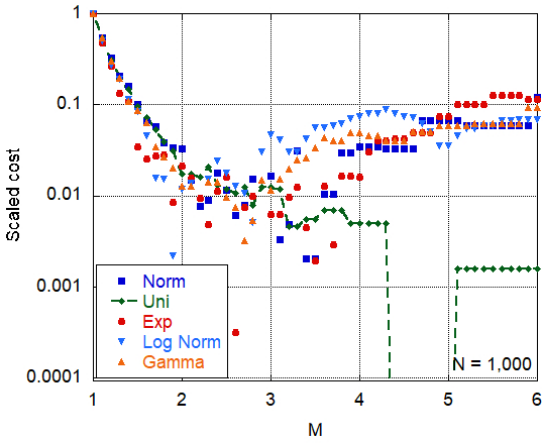

Figure 7.2: A graph demonstrating that statistical fluctuations in a histogram, measured using a normalised cost function, have dissipated for $M \geq 2$.

- **Noise distribution:** We derived a formula for the effect of continuous noise distributions on discrete variables, $h(g) = H(P) + h(u)$, which we found not to preserve the information content of the sample. Instead, the noise distribution adds information to the system dependent on the differential entropy of the distribution used. We verified this behaviour for several distributions through synthetic experiments, where we found that the measured bias from the noisy entropy estimate equated to the entropy of the noise distribution, see figure 7.3. Verifying the derived formula. From these findings, we can now successfully add noise to discrete variables. We recommend a $\mathcal{U}[0, 1)$ noise distribution, where $h(\mathcal{U}[0, 1)) = 0$ bits removes these problems. Thus, improving the applicability of the superior continuous entropy estimator methods.

- **Entropy estimation:** We extend our ideas to propose a novel method for calculating information-theoretic quantities. Thus, we proposed a noisy resampling method, termed algorithm W, for estimating entropy, mutual information and the Kullback-Leibler divergence. We empirically tested the new method on artificial and real-world data sets. The method successfully calculated entropy and mutual information values for known probability distributions, and experiments on artificial data showed that algorithm W is asymptotically unbiased and consistent. The proposed method demonstrated a faster convergence rate and smaller estimation error for several synthetic experiments consisting of mixed, continuous and discrete cases, for example see figure 7.4. Results from real-world data similarly showed that algorithm W effectively detected pairwise correlations and outperformed or was equivalent to

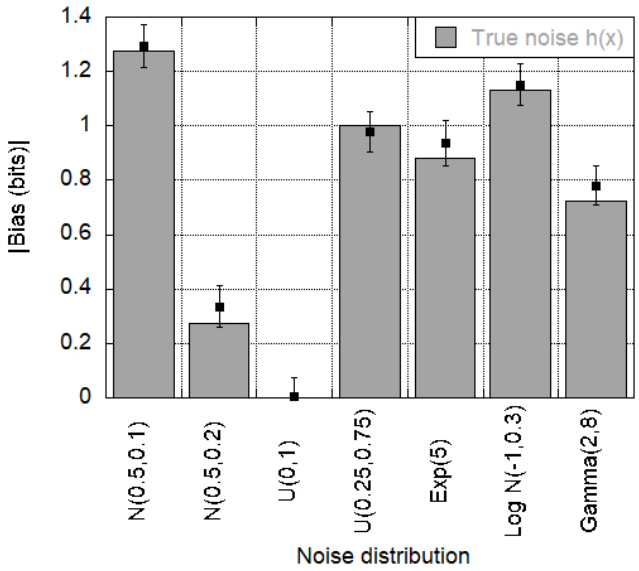

Figure 7.3: Verification that the entropy of a noise distribution is added to a differential entropy estimate resulting in an apparent bias.

the 3H-KL and KSG estimators. Figure 7.5 demonstrates the comparative performance on correlations with a class variable as a function of the accuracy estimate for classification. As seen in chapter 4 a robust estimator is valuable is supervised learning as variables with a small mutual information with the class have proven merit in feature selection subsets.

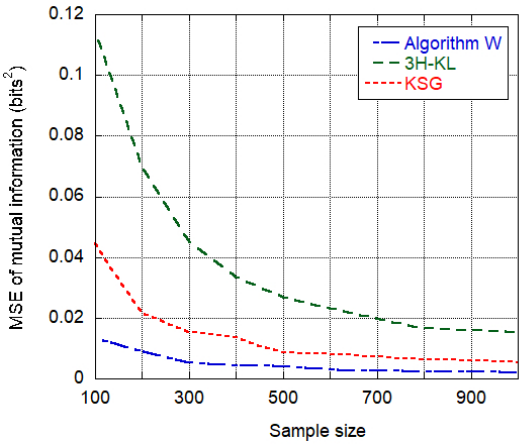

Figure 7.4: Comparison of the MSE for mutual information estimators of simulated independent normal distributions.

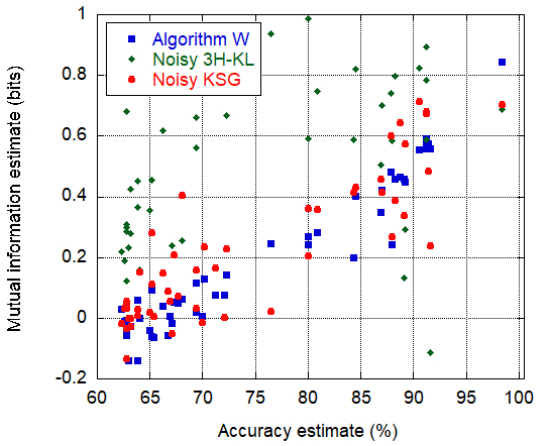

Figure 7.5: Comparison of the mutual information of variable-class pairs given as a function of the classification accuracy estimate for real-world data samples.

- **Kullback-Leibler divergence:** We similarly applied the noisy resampling principles to the Kullback-Leibler divergence k-nearest-neighbour estimator in [1] and empirically demonstrated its superior convergence compared to the raw estimator. For finite data samples, we demonstrate that the Kullback-Leibler estimate saturates, see figure 7.6. As samples become increasingly separated the measure is limited by the information content of the sample. Thus, the estimate saturates, dependent on the sample size $\log_2(N)$, defined by

minimum error limit. Thus, showing that there is a fundamental limit on the discriminating power of finite data samples. We employ a symmetric Kullback-Leibler measure to evaluate the predictive abilities of data sets. Using the Chernoff-Stein lemma, we obtain a relationship between the Kullback-Leibler divergence and the *kappa* statistic, relating information theory to machine learning performance. This relationship is shown in figure 7.7.

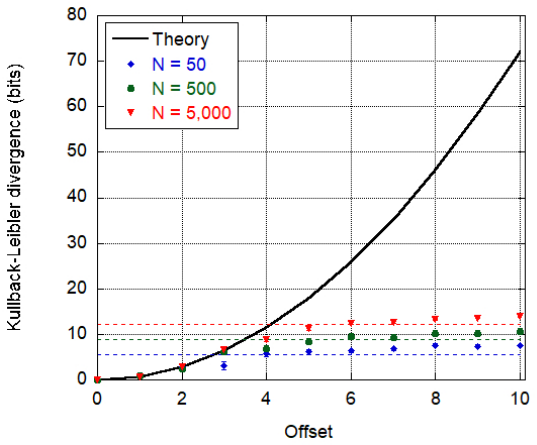

Figure 7.6: A graph depicting the fundamental limit applicable to all estimations of the Kullback-Leibler divergence, where the horizontal lines indicate $\log_2(N)$bits.

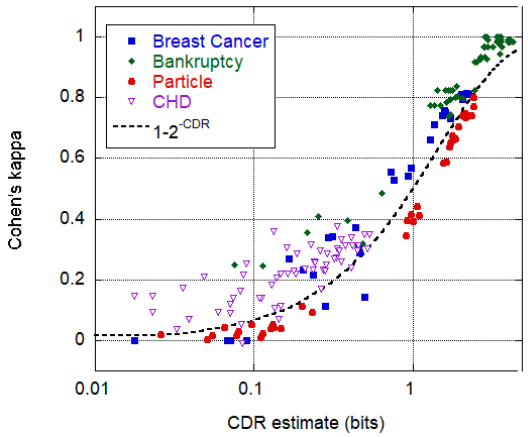

Figure 7.7: A graph demonstrating the relationship between Cohen's *kappa* and the symmetric Kullback-Leibler measure, CDR for real data sets.

- **DataViewer** We developed an innovative visualisation software package and showcased key functionalities of the application via five case studies on machine learning data sets. The software implements the noisy resampling method to estimate the similarity index (normalised mutual information) and symmetric Kullback-Libeler measure, CDR. The similarity index results are illustrated via the novel "variable interaction diagram", which summarises complex network structures in a human interpretative format. An interactive parallel coordinates widget is presented with brushing, pruning, and grouping features. We propose an information-based, equiprobable quantisation technique to improve the information estimates for high-kurtosis distributions. We subsequently present equiprobable visualisations as a valuable exploratory technique for visualising high-kurtosis distributions in one and two dimensions.

7.2 Future work

The concepts presented in this thesis can take several different directions. Here we discuss a few of those ideas.

- **Multiple dimensions:** We believe it is possible to extend the key principles in algorithm W to information-theoretic metrics beyond two dimensions. Common multi-dimensional

measures include co-information [2] and the total correlation [3]. These measures are similarly comprised of entropy terms and can thus be estimated from any entropy estimate. Unfortunately, the bias associated with k-nearest-neighbour estimators increases with dimensions, requiring exponentially more data for the same accuracy. However, as seen for the real-world examples, multi-dimensional interactions play a key role in optimised feature selection.

- **Feature selection:** There are several information-theoretic machine learning algorithms for attribute ranking and feature selection. For attribute ranking, "information gain" and "gain ratio" are the two most commonly used measures for relevancy. These are respectively, the mutual information and similarity index metrics, except that the gain ratio uses the variable's entropy to normalise the mutual information, rather than the smallest entropy. Similarly, various algorithms exist for information-theoretic feature selection [4]–[7]. An interesting direction of future experiments would be to compare the performance of these algorithms using the current methods for entropy estimation and the method proposed in this thesis.

- **DataViewer:** We have incorporated several visualisations into DataViewer. However, other avenues could be beneficial. For example, star coordinates are a valid alternative to visualising high-dimensional data. We discussed various improvements to the GUI throughout section 6.2 for the visualisations currently included in DataViewer. The suggested improvements primarily consist of increasing interactivity, and dynamic visualisations [8]. For the parallel coordinates plot the ordering of the axis is still a crucial, but unanswered question. We hope to address this problem by providing an automated ordering feature in DataViewer that utilises the similarity index in a solution to the travelling salesman problem. We believe these features will be of benefit for exploratory analysis. In future, we hope to test the versatility of DataViewer for user case studies.

References

[1] Q. Wang, S. R. Kulkarni, and S. Verdú, "Divergence estimation for multidimensional densities via κ-nearest-neighbor distances," *IEEE Transactions on Information Theory*, vol. 55, no. 5, pp. 2392–2405, 2009, ISSN: 00189448. DOI: 10.1109/TIT.2009.2016060.

[2] A.J.Bell, "The co-information lattice," *Proceedings of the Fifth International Workshop on Independent Component Analysis and Blind Signal Separation*, pp. 921–926, 2003.

[3] S. Watanabe, "Information Theoretical Analysis of Multivariate Correlation," *IBM Journal of Research and Development*, vol. 4, no. 1, pp. 66–82, 2010, ISSN: 0018-8646. DOI: 10.1147/rd.41.0066.

[4] R. Battiti, "Using Mutual Information for Selecting Features in Supervised Neural Net Learning," *IEEE Transactions on Neural Networks*, vol. 5, no. 4, pp. 537–550, 1994, ISSN: 19410093. DOI: 10.1109/72.298224.

[5] H. Cheng, Z. Qin, C. Feng, Y. Wang, and F. Li, "Conditional mutual information-based feature selection analyzing for synergy and redundancy," *ETRI Journal*, vol. 33, no. 2, pp. 210–218, 2011, ISSN: 12256463. DOI: 10.4218/etrij.11.0110.0237.

[6] M. Bennasar, Y. Hicks, and R. Setchi, "Feature selection using Joint Mutual Information Maximisation," *Expert Systems with Applications*, vol. 42, no. 22, pp. 8520–8532, 2015, ISSN: 09574174. DOI: 10.1016/j.eswa.2015.07.007. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2015.07.007.

[7] X. Wang, B. Guo, Y. Shen, C. Zhou, and X. Duan, "Input Feature Selection Method Based on Feature Set Equivalence and Mutual Information Gain Maximization," *IEEE Access*, vol. 7, pp. 151 525–151 538, 2019, ISSN: 21693536. DOI: 10.1109/ACCESS.2019.2948095.

[8] M. J. Pacione, M. Roper, and M. Wood, "A comparative evaluation of dynamic visualisation tools," in *10th Working Conference on Reverse Engineering*, 2003, pp. 80–89.

Appendices

Appendices A

A.1 Equivalence of Scott's rule and the cost function of Shimazaki and Shinomoto

Scott's formula is a well known bin width optimisation solution, which minimises the IMSE of the density estimate. After some reasonable approximations, this leads to the formula for the bin size

$$\Delta = \left(\frac{6}{\int_{-\infty}^{\infty} p'(x)^2 N dx} \right)^{\frac{1}{3}} \tag{A.1}$$

Where $p'(x)$ is the first derivative of the pdf, $p(x)$. However, equation A.1 requires the first derivative to be non-zero and thus there is no solution for a uniform distribution. Scott's method also requires prior knowledge of the underlying pdf. In [1] Shimazaki *et. al.*, however, develop a non-parametric method for binning neuronal spikes for a time histogram. Using a similar approach to Scott, Shimazaki *et. al.* minimises the mean integrated squared error (MISE) of the underlying spike rate. Two solutions were derived based on properties of the autocorrelation function, $\phi(\tau) = \int_{-\infty}^{\infty} p(x)p(x-\tau)$. When the autocorrelation function is a smooth function of τ [1] derive a formula for the optimal bin width:

$$\Delta = \left(-\frac{6\mu}{\phi''(0)N} \right)^{\frac{1}{3}} \tag{A.2}$$

Here we show that Shimazaki's formula is equivalent to Scott's rule for non-time histograms, where μ, the mean spike rate is equal to 1. We start with a Taylor expansion, $p(x-\tau) = p(x) - \tau p'(x) + \frac{\tau^2}{2} p''(x) + ...$ and substitute this into the autocorrelation function.

$$\begin{aligned} \phi(\tau) &\approx \int_{-\infty}^{\infty} p(x) \left(p(x) - -\tau p'(x) + \frac{\tau^2}{2} p''(x) \right) dx \\ &\approx \int_{-\infty}^{\infty} \left(p(x)^2 - \tau p(x) p'(x) + \frac{\tau^2}{2} p(x) p''(x) \right) dx \end{aligned} \tag{A.3}$$

Writing $\phi'(\tau)$ as the derivative of $\phi(\tau)$ w.r.t. τ, and $\phi''(\tau)$ as the second derivative, gives

$$\begin{aligned}\phi'(\tau) &\approx \int_{-\infty}^{\infty} \left(-p(x)p'(x) + \tau p(x)p''(x)\right) dx \\ \phi''(\tau) &\approx -\int_{-\infty}^{\infty} p(x)p''(x)dx\end{aligned} \tag{A.4}$$

Using integration by parts $u = p(x)$ and $v' = p'(x)$

$$\phi''(\tau) \approx \int_{-\infty}^{\infty} p(x)p''(x)dx = [p(x)p'(x)]_{-\infty}^{\infty} - \int_{-\infty}^{\infty} p'(x)^2 dx \tag{A.5}$$

These terms are integrated over the range of the function, which is by construction smooth. Thus the pdf goes to zero at the end of the range and the first term $[p(x)p'(x)]_{-\infty}^{\infty} = 0$.

$$\phi''(\tau) \approx -\int_{-\infty}^{\infty} p'(x)^2 dx \tag{A.6}$$

When substituted into Shimazaki's solution, in equation A.2, the result is Scott's formula.

$$\Delta = \left(-\frac{6}{N\int_{-\infty}^{\infty} p'(\tau)^2}\right)^{\frac{1}{3}} \tag{A.7}$$

Appendices B

B.1 Noise distribution

By construction in the derivation of $H(p) = h(g) - h(u)$ - where p is the pdf of the random variable, g is the histogram and u is the continuous random number-the consecutive noise distributions did not overlap. Thus, for the above to be true the continuous random number used must not exceed the bins boundaries.

To empirically demonstrate the effects of the noise exceeding the bin boundaries we simulate a discrete uniform distribution $\mathcal{U}[0, 15)$. For samples of size $N = 1,000$ we measure the average bias of 250 i.i.d trials for the KL entropy estimator. We repeat this experiment for normal, uniform and exponential noise distributions over a range of their respective parameters. The results are shown in figures B.1-B.3. Alongside the measured bias, we also plot the theoretical entropy of the noise distribution. When the noise distribution is well confined inside the bin boundaries, the measured bias follows the theoretical entropy curve. However, it is apparent that as the range of the noise distribution increases the measured bias is less than theoretical entropy. For reference, example noise distributions are given in figure B.4 the range of which is compared with a bin width of $\Delta = 1$. These figures show that when the range of the noise distribution increasingly exceeds the bin boundaries the bias is no longer predictable and increases with range.

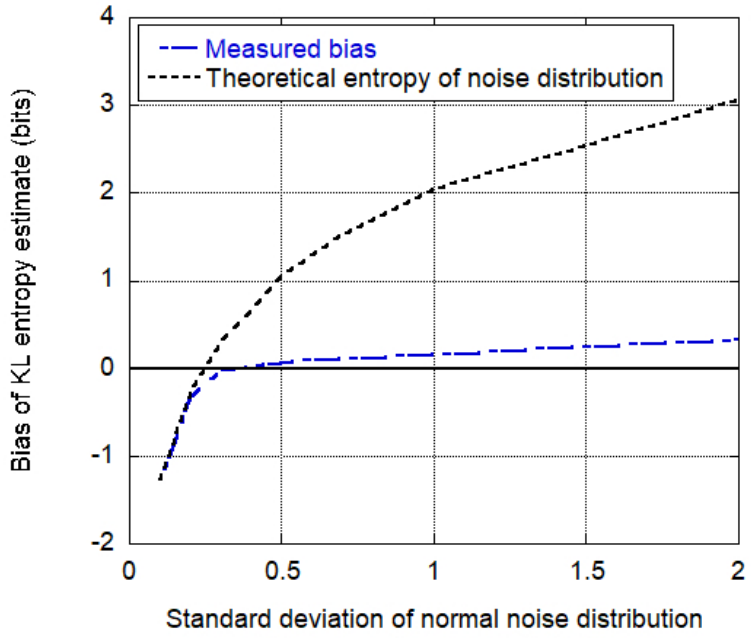

Figure B.1: Bias of the KL entropy estimate as a function of standard deviation of the normal noise distribution.

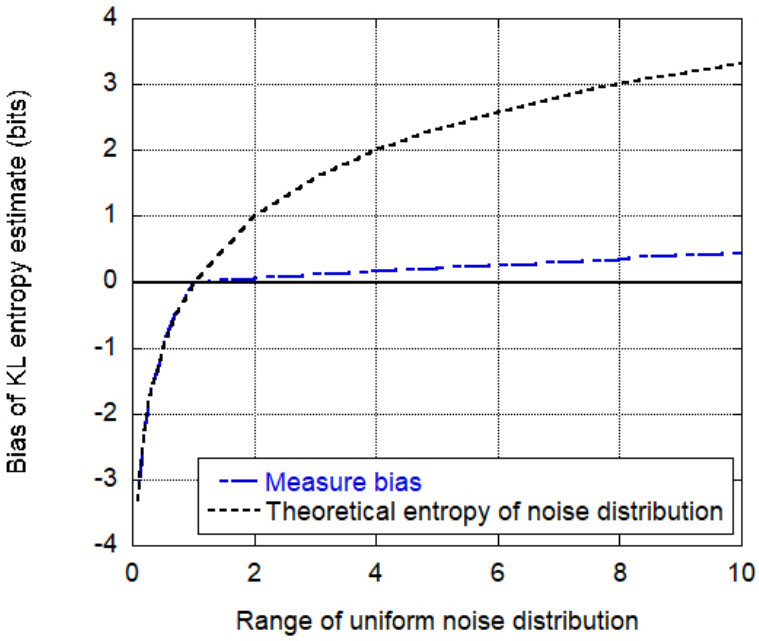

Figure B.2: Bias of the KL entropy estimate as a function of range of the continuous uniform noise distribution.

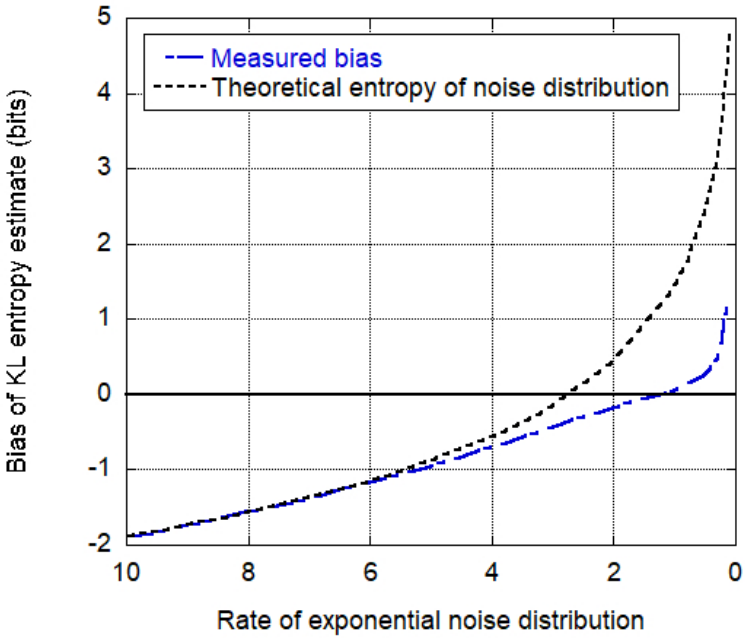

Figure B.3: Bias of the KL entropy estimate as a function of rate of the exponential noise distribution.

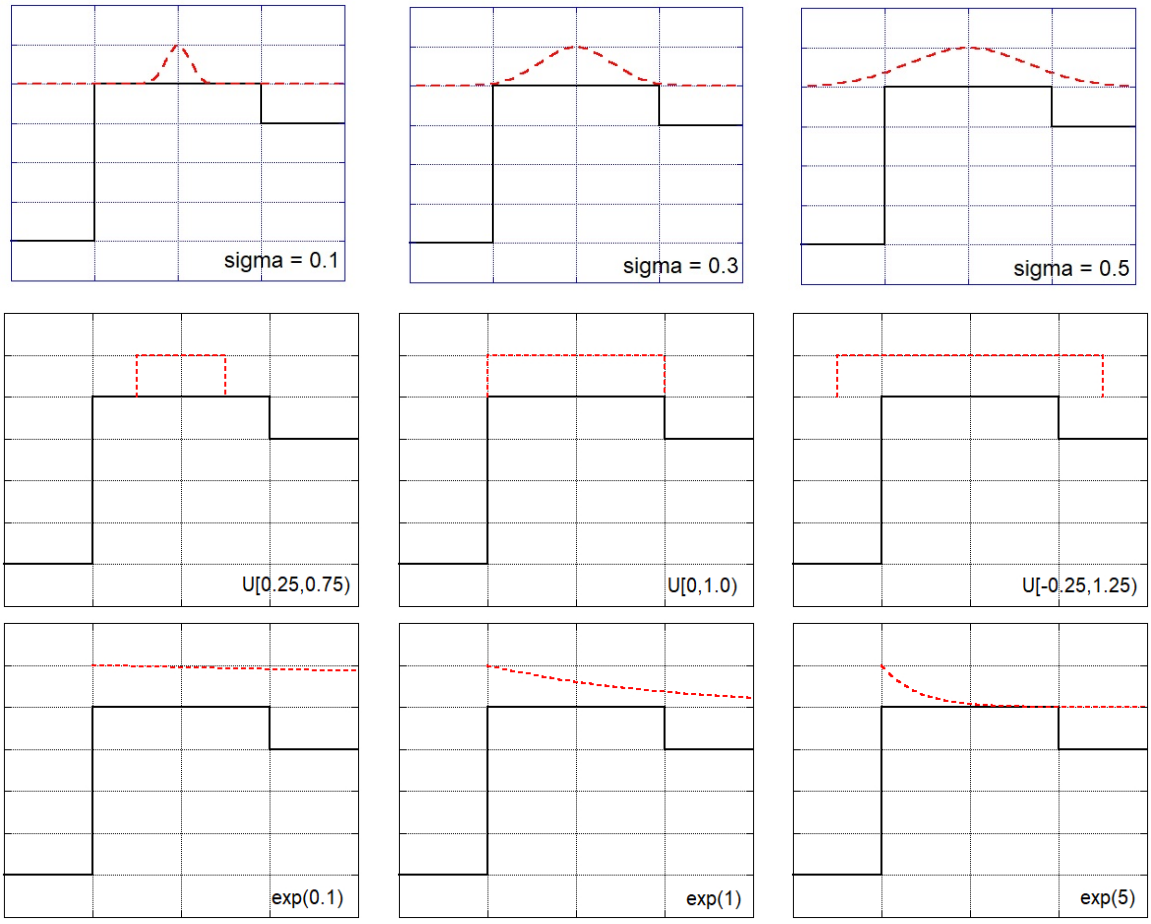

Figure B.4: A schematic of noise distributions for different distributions and parameters in relation to a bin width of 1. The parameters used are shown of each plot and from top to bottom the noise distributions are normal, uniform and exponential.

Appendices C

C.1 Classifier Algorithms

- **PART Decision Tree:** This algorithms uses a "separate-and-conquer" strategy to generate rules iteratively. For each rule a **part**ial C4.5 decision tree is generated for the unclassified instances. The leaf that is able to classify the most instances is made into a rule and the remainder of the tree is discarded. It then removes the instances that have been classified and continues building trees iteratively for the remaining non-classified instances until all the instances have been classified [1].
- **Simple Logistic Regression Model:** This model consists of a standard decision tree structure (parent and child nodes) with logistic regression functions, instead of a class label, at the terminating nodes, "leaves". For each candidate leaf, a logistic regression model is built using the instances at that node. The model at the child node is based on the model at the parent node and higher levels. The idea is that 'global' influences are already encoded in higher-level models. The model at the child node can then be refined by taking into account local influences. This method is implemented using a "LogitBoost" algorithm, which iteratively changes the linear class functions to improve the fit [2].
- **Genetic Algorithm:** In a genetic algorithm the solution to an optimisation problem is evolved via "mutations" and "alterations". Each design solution is evaluated using a chosen fitness function, which quantifies how close the solution is to achieving a set of aims. After each valuation round the worst solutions are removed and new ones are "breed" from the best solutions [3].

Appendices D

D.1 *kappa* and the Kullback-Leibler Divergence

Cohen's *kappa* is defined by the probability of the observed agreement, P_o and the probability of the expected agreement by chance, P_e:

$$P_o = \frac{\sum_{i=1}^{C} m_{Ti}}{N} \tag{D.1}$$

$$P_e = \sum_{i=1}^{C} \left(\frac{n_i}{N} \frac{m_i}{N} \right) \tag{D.2}$$

Where *kappa* is

$$\kappa = \frac{P_o - P_e}{1 - P_e} \tag{D.3}$$

We related *kappa* to the Kullback-Leibler divergence using the Chernoff-Stein lemma [1]. For a two-class system this is

$$m_{F1} = \frac{n_1 n_2}{n_1 + n_2} 2^{-d_{21}} \tag{D.4}$$

$$m_{F2} = \frac{n_1 n_2}{n_1 + n_2} 2^{-d_{12}} \tag{D.5}$$

Where $d_{12} = d(p_1|p_2)$ and $d_{21} = d(p_2|p_1)$ and the remaining terms are as per the confusion matrix table in D.1. First, we need to derive a formula for *kappa* as a function of the above

	class 1	class 2	total
class 1	m_{T1}	m_{F1}	m_1
class 2	m_{F2}	m_{T2}	m_2
total	n_1	n_2	N

Table D.1: A table to visualise the notation for correctly and incorrectly labelled instances for a classifier on a two-class system.

terms. From table D.1 we can easily see the following equations:

$$m_{T1} + m_{F1} = m_1 \tag{D.6a}$$

$$m_{T2} + m_{F2} = m_2 \tag{D.6b}$$

$$m_{T1} + m_{F2} = n_1 \tag{D.6c}$$

$$m_{T2} + m_{F1} = n_2 \tag{D.6d}$$

$$n_1 + n_2 = N \tag{D.6e}$$

$$m_1 + m_2 = N \tag{D.6f}$$

We start by adding D.6c and D.6d and substituting this into D.1

$$P_o = \frac{m_{T1} + m_{T2}}{N} = \frac{n_1 + n_2 - m_{F1} - m_{F2}}{n_1 + n_2} \tag{D.7}$$

Similarly, substituting in equations D.6a, D.6b, D.6c and D.6d into D.2 gives

$$\begin{aligned} P_e &= \frac{n_1 m_1 + n_2 m_2}{N^2} \\ &= \frac{(m_{T1} + m_{F2})(m_{T1} + m_{F1}) + (m_{T2} + m_{F1})(m_{T2} + m_{F2})}{(n_1 + n_2)^2} \\ &= \frac{n_1^2 + n_2^2 - (n_1 - n_2)(m_{F1} - m_{F2})}{(n_1 + n_2)^2} \end{aligned} \tag{D.8}$$

Thus, *kappa* can be written as

$$\begin{aligned} \kappa &= \frac{P_o - P_e}{1 - P_e} \\ &= \frac{\frac{n_1+n_2-m_{F1}-m_{F2}}{n_1+n_2} - \frac{n_1^2+n_2^2-(n_1-n_2)(m_{F1}-m_{F2})}{(n_1+n_2)^2}}{1 - \frac{n_1^2+n_2^2-(n_1-n_2)(m_{F1}-m_{F2})}{(n_1+n_2)^2}} \\ &= \frac{(n_1 + n_2)(n_1 + n_2 - m_{F1} - m_{F2}) - (n_1^2 + n_2^2 - (n_1 - n_2)(m_{F1} - m_{F2}))}{(n_1 + n_2)^2 - (n_1^2 + n_2^2 - (n_1 - n_2)(m_{F1} - m_{F2}))} \\ &= \frac{2n_2 n_1 - 2n_1 m_{F2} - 2n_2 m_{F1}}{n_2 m_{F2} - n_2 m_{F1} - n_1 m_{F2} + n_1 m_{F1} + 2n_1 n_2} \\ &= \frac{2n_2 n_1 (n_1 + n_2) - 2n_1^2 n_2 2^{-d_{12}} - 2n_2^2 n_1 2^{-d_{21}}}{n_2^2 n_1 2^{-d_{12}} - n_2^2 n_1 2^{-d_{21}} - n_1^2 n_2 2^{-d_{12}} + n_1^2 n_2 2^{-d_{21}} + 2n_1 n_2 (n_1 + n_2)} \end{aligned} \tag{D.9}$$

where the final line uses the Chernoff-Stein lemma equations D.4 and D.5. After some algebra, we arrive at:

$$\kappa = \frac{2n_1(1 - 2^{-d_{12}}) + 2n_2(1 - 2^{-d_{21}})}{n_2(2^{-d_{12}} - 2^{-d_{21}} + 2) + n_1(2^{-d_{21}} - 2^{-d_{12}} + 2)} \tag{D.10}$$

Appendices E

E.1 DataViewer Access

A public distribution of the binary is available that runs on Ubuntu. DataViewer - Academic can be supplied to anyone interested if they contact the authors of [1] A download will be provided that includes the binary, installation instructions, a demonstration video and the full licensing details, an overview of which is given in appendix E.2.

E.2 DataViewer Licensing

DataViewer is written in C++ and uses a number of external libraries for specialist tasks. Here we summarise the external libraries and their licenses. Some external libraries have multiple licenses, the primary one, however, is listed first.

Library	License	Link to license
QtAwesome (https://github.com/gamecreature/QtAwesome/blob/master/LICENSE.md)	MIT	https://opensource.org/licenses/MIT
	SIL	http://scripts.sil.org/OFL
	CC	https://creativecommons.org/licenses/by/3.0/
Eigen (http://eigen.tuxfamily.or)	MPL2	https://www.mozilla.org/en-US/MPL/2.0/
	LGPL	https://gitlab.com/libeigen/eigen
GSL (https://gitlab.com/libeigen/eigen)	GPL3	https://gitlab.com/libeigen/eigen
		https://opensource.org/licenses/GPL-3.0
Qt 5.5.1 (https://www.qt.io/downl)	LGPLv2	https://opensource.org/licenses/LGPL-2.1
	LGPLv3	https://opensource.org/licenses/LGPL-3.0
MPIR (http://mpir.org/)	LGPLv3+	https://www.gnu.org/licenses/lgpl-
QCustomPlot (https://www.qcustomplot.com/)	GPL3	https://opensource.org/licenses/GPL-3.0
FLANN (https://github.com/flann-lib/flann)	BSD	https://opensource.org/licenses/BSD-2-Clause
OpenGL (https://www.opengl.org/)	SGI FSL B v1	https://spdx.org/licenses/SGI-B-1.0.html

E.3 DataViewer Code Structure

The algorithms and GUI of DataViewer are both written in object-oriented C++. The code of which is substantial, consisting of 47 files. The full executable file of DataViewer is 37.8 MB and took considerable work. A code structure is provided in figure E.1 for reference. Each visualisation widget has a custom class, which inherits objects from a corresponding algorithm class. These in turn send and receive messages from the software's top level manager class

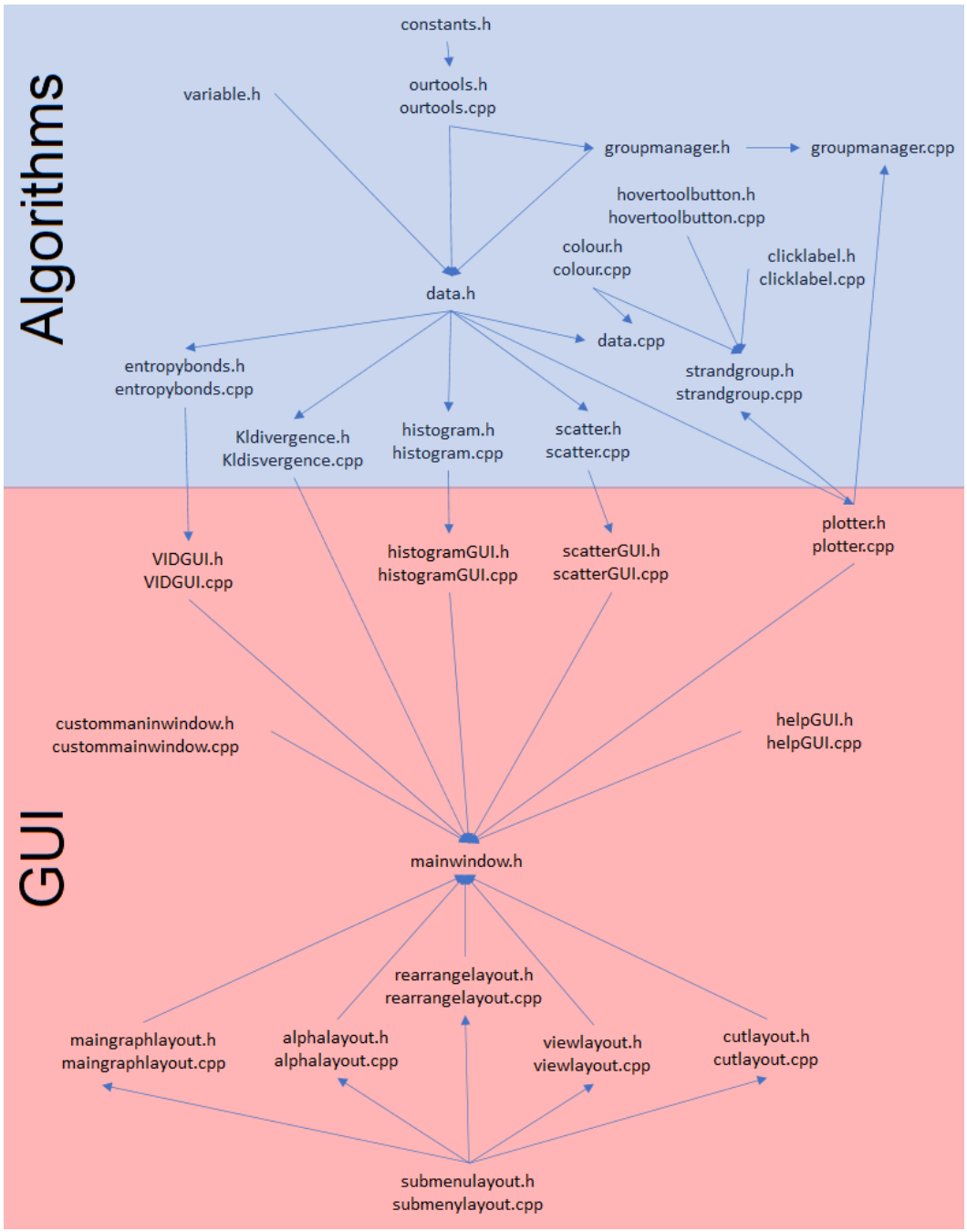

Figure E.1: A diagram showing the code structure for DataViewer. The direction of arrows indicates inheritance.

'data.' Here attributes of the data are updated. Global attributes of variables, such as binning information, is stored in the 'variable' header file, objects of which can be accessed through 'data.' Several default objects are also utilised such as warning and informational messages for the user which are displayed using QMessageBox.

References

[1] S. J. Watts and L. Crow, "Big variates — visualising and identifying key variables in a multivariate world," *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 940, no. December 2018, pp. 441–447, 2019, ISSN: 01689002. DOI: 10.1016/j.nima.2019.06.060. [Online]. Available: https://doi.org/10.1016/j.nima.2019.06.060.