UNIVERSITY OF STIRLING

From 'Tree' Based Bayesian Networks To Mutual Information Classifiers: Deriving a Singly Connected Network Classifier Using an Information Theory Based Technique

CLIFFORD S THOMAS

A Thesis Submitted in Partial Fulfilment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Department of Computing Science and Mathematics
University of Stirling, STIRLING, FK9 4LA, Scotland
Telephone +44-1786-467435, Facsimile +44-1786-464551
Email cst@cs.stir.ac.uk

MAY 2005

To my Father William Thomas

ABSTRACT

For reasoning under uncertainty the Bayesian network has become the representation of choice. However, except where models are considered 'simple' the task of construction and inference are provably NP-hard. For modelling larger 'real' world problems this computational complexity has been addressed by methods that approximate the model. The Naive Bayes classifier, which has strong assumptions of independence among features, is a common approach, whilst the class of trees is another less extreme example. In this thesis we propose the use of an information theory based technique as a mechanism for inference in Singly Connected Networks. We call this a Mutual Information Measure classifier, as it corresponds to the restricted class of trees built from mutual information. We show that the new approach provides for both an efficient and localised method of classification, with performance accuracies comparable with the less restricted general Bayesian networks. To improve the performance of the classifier, we additionally investigate the possibility of expanding the class Markov blanket by use of a Wrapper approach and further show that the performance can be improved by focusing on the class Markov blanket and that the improvement is not at the expense of increased complexity.

Finally, the two methods are applied to the task of diagnosing the 'real' world medical domain, Acute Abdominal Pain. Known to be both a different and challenging domain to classify, the objective was to investigate the optimality claims, in respect of the Naive Bayes classifier, that some researchers have argued, for classifying in this domain. Despite some loss of representation capabilities we show that the Mutual Information Measure classifier can be effectively applied to the domain and also provides a recognisable qualitative structure without violating 'real' world assertions. In respect of its 'selective' variant we further show that the improvement achieves a comparable predictive accuracy to the Naive Bayes classifier and that the Naive Bayes classifier's 'overall' performance is largely due the contribution of the majority group Non-Specific Abdominal Pain, a group of exclusion.

Acknowledgements

This thesis has been possible due to the support of many people both within the University of Stirling and other institutions. I wish to thank the staff of St John's Hospital, Livingston for their permission to use the CADA (**C**omputer **A**ssisted **D**iagnosis & **A**udit) database and in particular Mrs Julie Mclaren for her invaluable help in understanding its contents. In addition, my gratitude goes to her for the diligence in getting extremely busy A&E doctors to complete my questionnaire.

In conjunction with the CADA database, I would also like to thank the staff of the General Infirmary, Leeds for the use of their Acute Abdominal Pain database and in particular the late Professor Tim de Dombal for his professional and expert guidance in evaluating this difficult domain, also Dr Susan Cramp for her support and for providing the data sets.

As a part-time student this research has taken several years to complete and I wish to thank my supervisors Professor Leslie Smith and Kate Howie for their continued support during this period. I also wish to extend this to their guidance and suggestions in publications and for taking the time to read this thesis along with Dr Amir Hussain. With respect to the publication in the Intelligent Data Analysis Journal, I would like to thank the unknown reviewers for their valuable contributions which undoubtedly improved its contents.

Although much of the work was carried out remotely from the University, I would like to thank Sam Nelson for supporting my computer access needs both on and off-site.

On a more personal note, I would like to thank Professor Ken Turner for his assistance and continued support throughout my studies. Thanks also to my wife Janet for her support and putting up with being a virtual widow for many years, and in particular my son Christopher who has unfortunately missed a few years of Father/Son companionship, but has always been understanding and supportive throughout.

I would also extend my gratitude to Dr. Stephen Reiff-Marganiec, Dr. Nhamoinesu Mtetwa and Dr. Mario Kolberg for their valuable advice and assistance.

This work has been financially supported by BAE Systems Ltd and thanks goes to the business group directors for their continued support and encouragement.

DECLARATION

I hereby declare that I have personally composed this thesis and that the work herein was my own except where due acknowledgement is made. The material in this thesis has not been submitted to this or any other University for a degree. However, the early versions of some material presented in this thesis have been published or submitted for publication, which are included in the reference list.

Signed: 

Clifford S Thomas

Dated: 

20th May 2005

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

Bayesian Networks (BN) [Pea88, Nea90] have become the representation of choice for reasoning under uncertainty. Not only do BNs provide a compact graphical way to express complex probabilistic relationships among several random variables, but also tender attractive features not offered in other approaches. One advantage is their ease of comprehensibility to humans as many relationships between domain variables can be easily interpreted directly from the structure.

Extracting knowledge from experts in complex domains to solve 'real' world problems is however, arising as a major obstacle in constructing BNs [DG00]. The alternative is to learn directly from data, examples adopting this strategy can be found in [CMN00, CGK02, SV95, KS00, and Lar02], but except where models are considered 'simple' the task of construction and inference is in general provably NP-hard [Coo90, CLR90]. Chickering [Chi96] also found this to be the case when learning the structure of a general directed probabilistic network, even if each node is constrained to have at most two parents. For modelling larger 'real' world problems this computational complexity has been addressed by methods that approximate the model. The Naive Bayes classifier (NB) which has strong assumptions of independence among features is a common approach whilst the class of trees another less extreme example.

This thesis explores a new direction of applying information theory based methods for inducing a Classifier from a BN in the form of a 'tree' structure. The study examines the possibility of extending the use of 'mutual information' to the task of classification outside, but complementary to, the traditional roles of structure learning and the identification of relevant features.

Its purpose is to avoid the dependence upon prior node ordering and the subsequent inference complexity when the network topology leads to large Conditional Probability Tables (CPT).

It is hoped that the results presented in this thesis will encourage the use of this new classifier to assist in classifying in 'real' world applications particularly, where prior node ordering is not

generally available and in domains which can extend to hundreds, sometimes thousands of random variables.

1.1 Thesis Statement

As indicated previously, learning BNs from data is a rapidly growing field of research, with the specialist task of classification considered a very important undertaking in many data analysis activities [FGG97, DH01]. Typical areas of application are speech recognition, image understanding, spam filters and medical diagnosis.

Characteristic of a BN and key to defining its representation, in respect of the domain it models, is the determination of edge directionality of the graph. Where possible a domain expert can specify the node/vertices[1] ordering, that is, the domain knowledge used to specify a causal order of nodes or variables of the domain. However, where expertise is scarce, finding a node ordering by alternative means that will represent a useful BN can be a difficult task.

Whilst it is possible to find a BN for any given ordering (as the Joint Probability Distribution can be written by successive applications of the chain rule) it is clearly not practical to search among all possible orderings of nodes. Moreover, if we choose a poor order we may get a more complicated network. As the topology changes more tenuous relationships can occur, which may in turn require unnatural and problematic probability judgements.

The dependence BNs have on node ordering has led to researchers actively developing algorithms to efficiently determine edge directionality. One approach uses a search and scoring method to find the correct directions of the edges [LB94, FG96] but this was found to be slow as the search space can be large if prior node ordering is not supplied. Another more common approach uses Conditional Independence (CI) tests and has been used in many edge orientation algorithms [RP87, SGS90, VP92 and SGS91]. These methods are generally exponential in complexity. Singh [SV93] proposed a variant on the CI tests generating a "good" node ordering from data. Although offering an improvement in complexity, it was noted that the quality of the recovered network structure was very sensitive to the node ordering determined by their algorithm. Further strategies can be found in [Pan02, CG01, ACH01, LK+96 and GP+02].

---

[1] These terms are used interchangeably throughout the thesis.

Singly Connected Networks (SCN or 'polytree') are a restricted class of networks that can efficiently be solved in time linear in the number of nodes. However, despite this reduction in complexity the task of finding edge directionality on a skeleton tree structure, thus completing the 'polytree', is still as complicated [Cam96] to resolve. 'Polytree' recovery techniques have been proposed [RP87, Cam96 and Das99] based on CI tests, but in some situations full recovery was not always possible, leaving some edges undirected. When directionality was fully recovered it was found that even with a small number of parents, a node's CPT still required an unrealistic number of values to complete its description. Since each variable state must be specified in a CPT, there will be an exponential growth corresponding to the number of parents associated with the child node.

One of the main objectives of the research is to investigate the use of an information theory based technique as a mechanism for efficient inference in BNs. The aim is to avoid the issues concerning large CPTs and the dependence upon prior node ordering. This is tackled by taking advantage of the existing tree structuring algorithms. The concept proposed builds on the efficient Singly Connected Network (SCN) or 'polytree' as described by Pearl [Pea88], using the orientation of the tree edges, with respect to the class node, as a heuristic for assigning edge directionality. We call this SCN variant a Mutual Information Measure (MIM) Classifier as it corresponds to the restricted class of trees built from mutual information.

To demonstrate the validity and effectiveness of the classifier it has been applied to several benchmark problems taken from the UCI repository[2] [MA95, BM00]. The MIM classifiers learned are shown to perform significantly better than the NB classifier, one of the most widely studied methods, as well as displaying accuracy comparable to a general Bayesian network (GBN) and SCN learned classifiers. In addition, properties of the kinds of problems are identified where the MIM tree based classifiers will be most useful as opposed to the alternative representations.

Network topology and the number of variables govern the complexity of probabilistic inference in practice. As a consequence the application of BNs is often dismissed as unfit for 'real' world domains. A common approach to dealing with this is to learn selective networks using only 'relevant' attributes. The idea is to use only a subset of the available attributes to model the domain and thus make them computationally simpler to evaluate.

---

[2] UCI – University of California, Irvine's repository of machine learning databases.

Our second aim is to investigate the possibility of expanding the class Markov blanket (MB) and thus optimise the MIM classifier's predictive performance. In this thesis we consider the initial tree structure learned to represent a '*lower bound*' or implied feature sub-section focusing attention on the class node.

Experiments carried out using the UCI repository database show that in general the 'overall' performance of the MIM classifier can be improved by expanding the class Markov blanket, with predictive accuracies found to be significantly better than those obtained for the NB classifier. In addition, when compared to the results obtained using a selective NB (SNB) classifier [LS94], the number of relevant (with respect to the class node) features required was, in many of the data sets used, found to be similar in size as those selected and utilised by the SNB models.

Finally, the algorithms for inducing the MIM classifier and its optimised variant from data, are applied to the task of diagnosing a medical database concerning Acute Abdominal Pain (AAP). This domain is well known to be difficult with diagnosis complicated by other diagnoses, which often present similar signs and symptoms. In fact despite being a high dimensional problem with several variables, the NB classifier is considered optimal for classifying this domain [TS94]. Our experiments show that the MIM classifier achieves a comparable predictive performance to that of the NB classifier when evaluated with 'external' data of the domain and in general is further improved in performance by expanding the class MB. We further show that the approach provides a recognisable (by the doctors) qualitative structure without violating 'real' world assertions.

1.2 Summary of Contributions

The following sections list the main contributions of the thesis.

Mutual Information Measure (MIM) classifier. Node ordering choice and subsequent CPT dimensionality are known to impact on a BN's ability to perform well as a classifier. A bad choice may not only result in a topology which leads to an intractable solution but also to the possibility that CPTs will require an unrealistic number of probabilities to both estimate and subsequently update in respect of new evidence being presented. Moreover, in situations where domain feature dimensionality is high and data sets sparse, these probabilities may even be unreliable.

To address these issues we have derived a new classifier based on information theory techniques. Initially by application of the Chow & Liu [CL68] (CL) algorithm an efficient 'tree' based BN classifier is constructed then by using the notion of branch 'weights' we derive a method to discriminate between class-states (i.e. classify new evidence of the domain).

Specifically:

- Empirical studies show that neither the restriction to 'tree' topology nor CPT dimensionality affected the performance accuracy when compared to the less restricted GBN models.
- We show the technique adopted is independent of node ordering choices by employing a heuristic which identifies the class MB consistently.
- For large sample sizes and medium dimensionality of features (14+) we show the MIM classifier performs better than the NB.[3]
- For high feature dimensional domains with 'multi' parented class nodes, the MIM classifier is shown to perform better than the GBN.[3]

Selective MIM classifier (SMIM). Feature selection is an approach used to overcome the complexity of BNs by attempting to identify and remove irrelevant features of a domain to be modelled and thus improve the 'overall' performance.

To improve the performance of the MIM classifier we derive a 'Wrapper' type selective variant of the MIM classifier and show that the same efficient 'tree' based classifier (MIM) can also be used as the Wrapper evaluation function.

Specifically:

- Empirical studies show the class MB derived via the CL algorithm represents a satisfactory 'initial' class MB for large data sets (i.e. those that contain sufficient data to fully characterise the individual class-states).
- For small data sets, the performance of the corresponding MIM classifier can be improved by a focused expansion of the 'initial' class MB.

---

[3] In respect of the UCI data sets studied in this thesis.

- Empirical studies confirm the classification technique is not restricted to 'tree' based structures (with respect to the class MB) and is thus independent of the underlying topology of the domain being modelled.
- In general only a marginal number of features are required to be added to the 'initial' class MB in order to improve performance.

Case Study –AAP. The domain of AAP is known to be difficult and challenging to classify. Many researchers [ED84, Dom91 and TS94] have argued that the NB classifier is 'optimal' for this domain despite the fact it violates 'real' world assertions. We investigate this claim by carrying out detailed experiments comparing the performance accuracy of the MIM and SMIM classifiers with that of the NB classifier.

Specifically:

- We show the MIM classifier and its selective variant (SMIM) can be efficiently applied to the domain without making the assumption of extreme CI given the class. In the case of the MIM classifier we additionally show the qualitative structure constructed is recognisable by domain experts in contrast to the trivial structure of the NB classifier.
- Empirical studies confirm that the NB classifier is probably 'optimal' for the domain, however it is only in respect of its *overall* performance accuracy.
- The *overall* performance of the NB classifier is due to the contribution of the majority group Non-Specific Abdominal Pain (NSAP) (a group of exclusion). When this influence is removed we further show that the SMIM's *overall* performance matches that achieved by NB.
- Empirical studies show that the MIM classifier identifies with greater predictive accuracy, more individual disease groups than NB. This is important as each group has a different level of significance. For example the group appendicitis can be fatal if not identified quickly.

1.3 Thesis Organisation

- **Chapter 2** - provides an overview of the methods that will be used to make comparisons against the MIM classifier (described in Chapter 3), in particular, the representation, inference and induction of the General Bayesian Network, 'polytree' and NB classifiers.
- **Chapter 3** - describes a new technique for inducing a 'tree' based classifier from a BN [THS05a]. The approach is based upon the well known Chow and Liu algorithm [CL68] to both construct a tree structure and subsequently classify unseen observations of the problem domain.
- **Chapter 4** - demonstrates the validity and effectiveness of the MIM classifier by carrying out detailed experimental studies on a number of benchmark databases selected from the UCI repository of Machine Learning databases [MA95, BM00]. The results and subsequent comparisons against the methods described in Chapter 2 are discussed.
- **Chapter 5** – in this chapter the MIM classifier is considered as representing an implied feature selector and therefore the use of the Chow and Liu algorithm can be considered as a mechanism for deriving an 'initial' Markov blanket. This chapter thus discusses the possibility of improving the performance of the MIM classifier by expanding the 'initial' class Markov blanket, whilst maintaining the MIM classifier classification efficiency as described in Chapter 3.
- **Chapter 6** - provides the evaluation of the performance of the 'selective' variant of the MIM classifier. As in Chapter 4, the same UCI databases are again utilised for carrying out a series of experiments on these methods. The resulting performance is compared to the previous 'non-selective' variants (evaluated in Chapter 4), together with a 'selective' variant of the NB classifier [LS94].
- **Chapter 7** - describes a case study where the two techniques, proposed in Chapter 3 and Chapter 5, are applied to a medical domain. The domain of Acute Abdominal Pain (AAP) represents a 'real' world problem and is known for its difficulty in respect of the task of classification. Two data sets of the domain are utilised and the results obtained from all methods studied in this thesis compared, along with those of the experts' own diagnosis.

- **Chapter 8** - represents the final chapter and reviews the main contributions of the thesis and provides some suggestions for further work.
- **Appendix**

  A – CADA database contents used in the experimental studies.

  B – Responses from the Doctor's questionnaires.

  C – Questionnaire template.

  D – Patient Record Sheet used by St. John's Hospital A&E Department.

  E – Acronyms and notation used in the thesis.

  F – Publications.

Chapter 2

Bayesian Networks as Classifiers

Probabilistic graphical models or Bayesian networks offer a unified qualitative and quantitative framework for representing and reasoning with probabilities and independencies. In a BN vertices represent propositional variables in a domain, and edges between vertices represent the dependency relationships among the variables. By taking advantage of the independencies existing between subsets of variables in the domain, they model the joint densities that limit the problems of dimensionality, namely parameter space. Of particular interest is the use of BNs to characterise the specialist application as a classifier. Essentially, the BN represents a function that assigns a class label to an instance described by a set of features. Despite the obvious disadvantage in restricting a BN to only answering specific queries, the attractive features BNs offer have lead to several examples of 'real' world applications. The diversity ranges from medical [MC+01, BL+01], Web intelligence [JL+04] to waste water treatment [SB00]. Further examples can be found in [FF95, SP+02, CS+03 and EN95].

Introductory Bayesian network theory can be found in [Jen96, Ste00, and Jen01], whilst a review of the algorithms and literature on learning BNs in [Hec98, Bun96, CG01, HGC95, Pan02, and FG96].

The purpose of this chapter is to provide an overview of the methods that will be used to make comparisons against the MIM classifier (Chapter 3). In section 2.1, the representations and corresponding techniques for inference concerning a General Bayesian Network (GBN), a Naive Bayes network and a Singly Connected Network or 'Polytree' are reviewed. In section 2.2, the methods for inducing Bayesian Networks from data are briefly discussed, with secion 2.3 providing a summay of the chapter.

2.1 Representation / Inference

2.1.1 General Bayesian Networks (GBN)

Modelling a Bayesian Network consists of determining the qualitative graph structure G and the quantitative parameter θ. The qualitative network structure $G(N, A)$ is a directed acyclic graph (DAG). Each of the vertices $n \in N$ represents a domain variable, and each edge $a \in A$ between vertices represents a probabilistic dependency [Pea88].

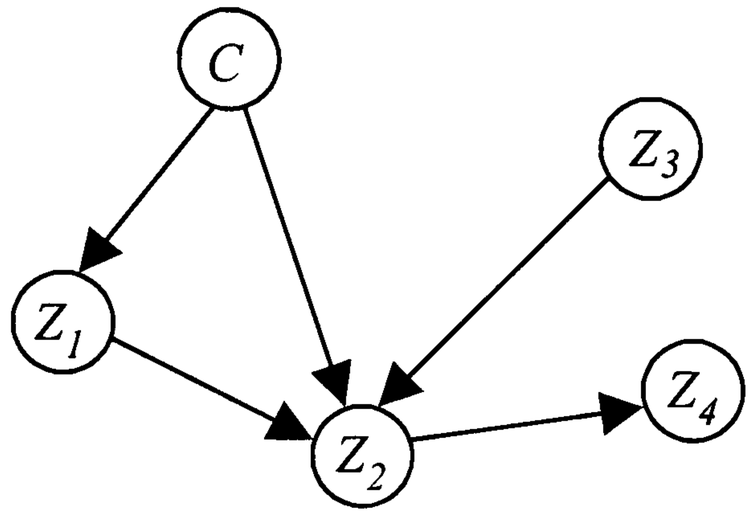


Figure 2.1. A General Bayesian Belief Network Example.

Edges in the Bayesian Network (Figure 2.1) represent the dependencies among the variables[4] $Z = \{Z_1, \ldots, Z_n\}$ with the parents of Z_i, $pa(Z_i)$ the direct predecessors of Z_i in G. An absence of edges indicates that there is conditional independence. The quantitative parameter θ consists of the joint probability distribution (JPD) $P(Z_1, \ldots, Z_n)$.

This is the general product and can be written:

$P(Z_1, \ldots, Z_n) = \prod_{i=1}^{n} P(Z_i \mid pa(Z_i))$ where $pa(Z_i)$ is designated as the parent of Z_i.

The resulting DAG encodes a group of conditional independence relationships among the vertices, according to the concept of *d-separation* [Pea88].

Definition 1: Two sets of variables X and Y in a network are *d-separated* by a third set of variables Z if and only if all paths that connect any vertex in X and any other vertex in Y have the following property:

There is a vertex v in the path such that either:

---

[4] The class varaiable C, shown in Figure 2.1, is included within the feature set depicted by Z.

$v \in Z$ and the arrows along the path do not converge at v.

$v \notin Z$, any descendent of v is not in Z, and the arrows along the path converge at v.

By using this definition conditional independence can be identified. That is, a group of variables X is conditionally independent of another group of variables Y given a third group of variables Z if the set Z *d-separates* X from Y [Jen96].

If the network is built in collaboration with domain experts, the determination of the structure is often a relatively easy task, since this task usually fits well with knowledge that for example, medical experts often have about causal relationships between variables. In an automated approach a data set can be utilised but this task in general is considered to be difficult [Coo90, CLR90] as reviewed in section 2.2. For the quantitative part (that is quantifying the conditional probability tables in the network) this aspect is often considered by medical experts as a much harder or even impossible task. The reason is that medical domain experts themselves often have no idea about these probabilities. When available a domain data set can provide estimates of the probabilities more readily than the experts.

The classification process involves a class variable C that can take on values $C_1,\ldots,C_m$, and a feature vector Z of n features that can take on a tuple of values denoted by $\{Z_1,\ldots,Z_n\}$. Given a case Z represented by an instantiation $\{Z_1,\ldots,Z_n\}$ of feature values, the classification task is to determine the class value C_i to which Z belongs[5]. Reviews of commonly used techniques concerning GBN inference can be found in [Pea88, Nea90]. When using a GBN classifier on complete data, the Markov blanket (MB) of the classification node forms a natural feature selection, as all features outside the MB can be safely ignored. Thus prediction using a GBN classifier examines only the relevant features [Pea88] defined by the MB of the class variable. Figure 2.2 illustrates a possible class MB.

Definition 2: For a node n in a GBN the Markov blanket is the union of $n's$ parents, $n's$ children, and the parents of $n's$ children.

[5] In this thesis we restrict our discussions to domains with only discrete variables.

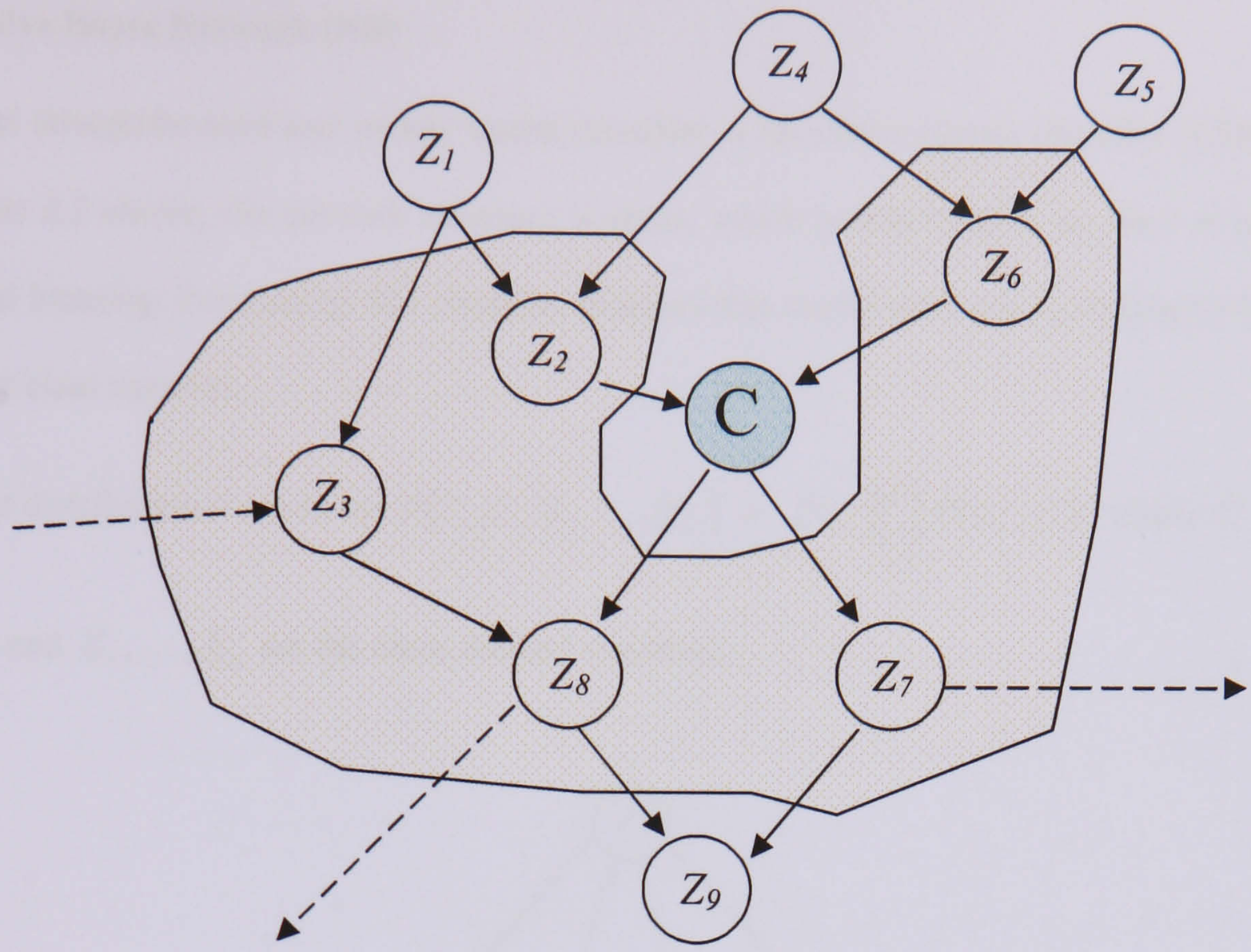


Figure 2.2. The Markov boundary of a node n is a BN, where n's Markov boundary is a subset of nodes that 'shields' n from being affected by any node outside the boundary. One of n's Markov boundaries is its Markov blanket. In this figure the Markov blanket of the Class variable C is defined.

For BNs there are essentially two types of inference, exact and approximate. Exact procedures are useful when networks are not too complex (in the general case inference is NP-hard). The most popular method is the junction tree algorithm [Jen96], this can however be exponential in the size of the cliques, thus restricting its use to low complexity applications. Other methods are arc reversals [Sha88], trees of cliques [LS88, JL90], symbolic manipulations of sums and products [DAm91] and Pearl's message passing algorithm [Pea86].

However, as shown by Cooper [Coo90] exact probabilistic inference is NP-hard. As the network size increases in magnitude, exact inference times grow and relatively small changes in the network topology can inevitably transform a simple problem into an intractable one. Approximate inference procedures are used when clique state space size is too large. Of the various approaches the more recent are Monte Carlo sampling methods [Mac98] and mean field and variational techniques [SJJ96, JJ98]. Other methods of approximate inference are discussed in [Dra95] concerning localised partial evaluation, [Kja94] weak arc removal, [Hen88, DAA94] using logic sampling, and [SP89, CC90 and Pea87] via stochastic simulation techniques. As is the case for exact inference, approximate inference has also been shown to be NP-hard [DL93].

2.1.2 Naive Bayes Network (NB)

The most straightforward and widely tested classifier is the Naive Bayes classifier [DH01, LIT92]. As Figure 2.3 shows, the network structure is static, which means there is no need to perform any structural learning. Essentially, this classifier assumes that the features are conditionally independent given the class variable.

The joint distribution is given by: $P(C, Z_1, Z_2, \ldots, Z_n) = P(C)\prod_{i=1}^{n} P(Z_i \mid C)$ where C is the class variable and $Z_1, \ldots, Z_n$ are the other domain variables.

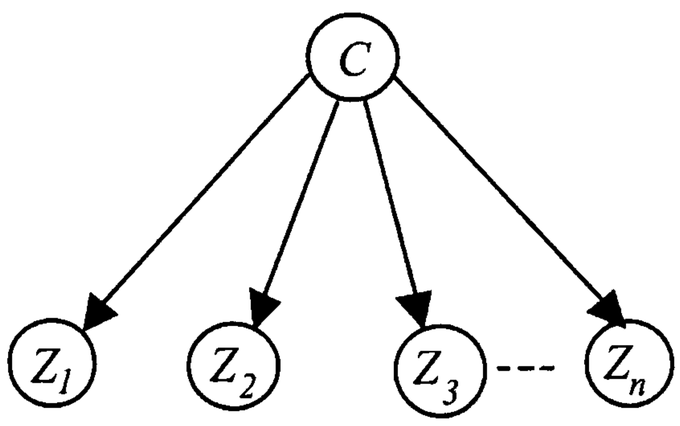


Figure 2.3. A Naive Bayesian Network Example

In this case inference is straightforward. To perform this task, we assume that we have the prior probabilities, $P(c_i)$, for each value c_i of the class variable. Further, we assume that we have the conditional probability distribution for each feature value z_j given the class value c_i, $P(z_j \mid c_i)$.

Using Bayes' rule, a new case, $Z = \Lambda_j z_j$ (Λ denotes conjunction), can then be classified as:

$$P(c_i \mid Z) = \frac{P(c_i)P(Z \mid c_i)}{P(Z)} = \frac{P(c_i)P(\Lambda_j z_j \mid c_i)}{\sum_k P(\Lambda_j z_j \mid c_k)P(c_k)}$$

Despite the controversial assumption of independence, this classifier has outperformed many state-of-the-art classifiers [LIT92, DP97 and HJR00]. Further analysis of the Naive Bayes classifier can be found in [EN95, LIT92, Ped98, Ris01a, CP+03 and Ris01b].

2.1.3 Singly Connected Networks or 'Polytrees'

A Bayesian network where a vertex may have multiple parents, and which is singly connected (that is, no more than one undirected path exists between any two parents vertices), is called a causal 'polytree' or Singly Connected Network, Figure 2.4. In using SCNs we gain in efficiency in procedures for learning the networks [AC$^+$91, HC93 and Cam96] and in performing exact evaluation or propagation [Pea88].

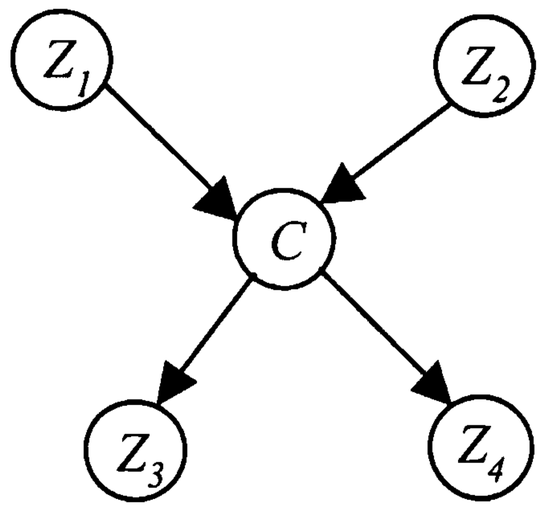


Figure 2.4. 'Polytree' Example

In a tree $P'(Z) = \prod_{i=1}^{n} P(Z_i \mid Z_{j(i)})$ where $Z_{j(i)}$ represents the parent of Z_i, the root vertex (selected arbitrarily) has no parents. In the case of the 'polytree' $P(Z) = \prod_{i=1}^{n} P(Z_i \mid Z_{j(i)}, \ldots, Z_{j_n(i)})$ where $Z_{j(i)}, \ldots, Z_{j_n(i)}$ represents the parents of Z_i. The parents in a 'polytree' are identified by determining, via some algorithm, directionality of the tree edges or alternatively supplied by a domain expert.

'Polytrees' represent much richer dependency models than trees, as they support products of higher-order distributions. Moreover, they admit tractable inference and can be identified by a Maximum Weight Spanning Tree (MWST) algorithm [DL93, Pea88] to find the structure and thus only require second-order statistics to establish the branch weights. Further details for constructing a 'polytree' will be covered in Chapter 4, in particular the specific implementation that will be used to evaluate the MIM classifier.

A brief overview of propagation in a 'polytrees' follows with [Pea88] providing a more detailed description. Essentially, when diagnostic evidence (that is, evidence from the children vertices) arrives at a vertex the prior belief undergoes a revision. This is achieved by using 'local' calculations

to send messages to neighbouring vertices which effectively then propagate around the whole tree structure. Consider the representation depicted by Figure 2.5, the vertex V_j blocks the path between the parent vertices and child vertices. That is, V_j 's parent vertices are conditionally independent of V_j 's child vertices given a known V_j .

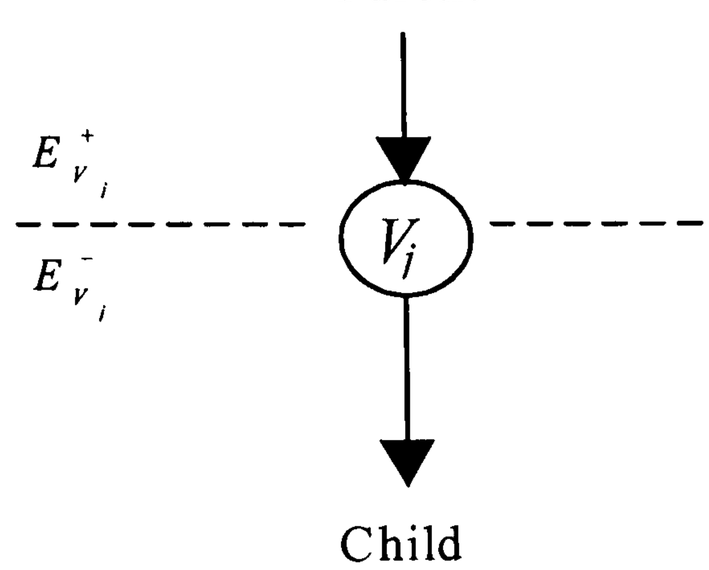


Figure 2.5. 'Polytree' (subset)

This property is useful for calculating posterior probabilities. For a variable V_j, the posterior probability of V_j given some evidence E can be represented by $P(V_j \mid E)$. The evidence E is a set of vertices instantiated so that E can be divided into two parts: $E^+_{V_j}$ and $E^-_{V_j}$ which represent the subset of evidence E .

From Bayes rule:

$$P(V_j \mid E) = P(V_j \mid E^+_{V_j}, E^-_{V_j})$$

$$= \frac{P(V_j, E^+_{V_j}, E^-_{V_j})}{P(E^+_{V_j}, E^-_{V_j})}$$

$$= \frac{P(E^+_{V_j}) P(V_j \mid E^+_{V_j}) P(E^-_{V_j} \mid V_j, E^+_{V_j})}{P(E^+_{V_j}, E^-_{V_j})} \qquad \text{.....(1)}$$

Due to conditional independence in the structure, Figure 2.5, $P(E^-_{V_j} \mid V_j, E^+_{V_j}) = P(E^-_{V_j} \mid V_j)$ so by substitution in (1) we get:

$$P(V_j \mid E) = \frac{P(E^+_{V_j})}{P(E^+_{V_j}, E^-_{V_j})} P(V_j \mid E^+_{V_j}) P(E^-_{V_j} \mid V_j)$$

where $\frac{P(E^+_{V_j})}{P(E^+_{V_j}, E^-_{V_j})}$ is a normalising constant called α .

By defining two messages for V_j: $\pi(V_j) = P(V_j \mid E^+)$ and $\lambda(V_j) = P(E^- \mid V_j)$ we get the expression: $P(V_j \mid E) = \alpha\pi(V_j)\lambda(V_j)$

With this expression we can thus calculate the posterior probability of V_j given evidence E by only using two messages π and λ, where $\pi(V_j)$ represents the causal support from the parents and $\lambda(V_j)$ represents the diagnostic support from V_j's children. Since only the parent and child vertices are involved the compilation of the posterior probability is local. Based on these observations Pearls' [Pea88] algorithm enables propagation in a 'polytree'.

2.2 Induction of Bayesian Networks

Learning methods and performance of BNs as classifiers are studied in Friedman [FGG97] and Cheng [CG99]. Further details of learning Bayesian networks can be found in [KS04, KaS04, PD03, GD04, GM04, Nea03, ZL02, Ste00, and Hec99].

The process of learning a Bayesian Network is defined by two activities: learning the graphical structure, and then learning the parameters for the structure [Pea88].

For learning the parameters the empirical conditional frequencies from the data can be used [CH92].

In the case of the structure, there are in essence two popular approaches used to learn a BN topology. The first is a scoring-based learning algorithm that searches for a structure that maximises a scoring function. Typically, this function can be a Bayesian method [CH92, HGC94, Yor92, and MR94], a minimum description length (MDL) score [Suz96, LB94, KS00] or an entropy based measure [Her91]. Score based approaches exhibit less time complexity for densely connected DAGs, however they may not find the best solution due to their heuristic nature.

The second approach is a constraint-based or CI-based algorithm. Learning the structure in this approach requires the identification of the conditional independence relationships among the vertices. Conditional independence tests are carried out to assess the dependency relationships among vertices and then used to constrain the building process. Various CI algorithms are described

in [SGS96, SRA90, CBL97, SGS00, and Shi00]. This approach generally needs large quantities of data and is computationally more expensive than the score based approaches.

2.3 Summary

In this chapter, a variety of techniques for learning a BN from data and subsequent inference has been reviewed. We began with the unrestricted GBN which represents a compact encoding of the JPD by means of an underlying graphical structure. This is an attractive feature as it enhances the understanding of the model, since relationships between the domain attributes can be simply read off the structure. Unfortunately generating a network from data is considered a difficult task, as is subsequent inference both being in general NP-hard [Coo90, CLR90]. 'Polytrees' or Singly Connected Networks represent a much richer dependency model than trees and as a restricted representation can be derived efficiently, however even though it offers tractable inference, full recovery of the 'domain' is not always achievable [Pea88]. NB on the other hand offers a straightforward approach and has been demonstrated as a very efficient classifier [LIT92, DP97 and HJR00]. Despite this achievement it provides no qualitative aspect and violates 'real' world assertions by making strong assumptions of conditional independence given the class.

In the chapter that follows we introduce a new classifier derived from an information theory based technique [THS05]. The approach is based upon the class of 'trees' [CL68, Gei92 and Pea88] which have been shown to be efficiently learnt, with particular reference to the MWST as derived via the CL algorithm [CL68]. By using tree based dependency approaches, we avoid the assumptions underlying the NB classifier whilst enabling a qualitative representation to be retained. In contrast to the BN's approach to inference, the new classifier further utilises and extends the concept of Mutual Information (MI), providing for efficient inference based upon pair-wise marginals rather than CPT probability estimates.

Chapter 3

Tree Structures and Classification

A Tree, like any graphical model, has the ability to express the dependencies between variables separately from the detailed forms of these dependencies, contained in the parameters. In doing so it provides a property that offers excellent support for human intuition and allows for the design of inference and learning algorithms.

Trees are simple models: this is especially evident when examining the algorithms that fit a tree to a given distribution. All the information about the target distribution that a tree can capture is contained within a small number, at most $(n-1)$, pair-wise marginals. Simplicity leads to computational efficiency, where efficient inference is a direct consequence of the fact that trees are decomposable models with a small clique size. Trees have a probability distribution that can be mapped perfectly as both a Bayes net and Markov net, where a Markov net is defined by a structure, which is an undirected graph with an arbitrary topology.

Tree structures require that exactly one variable be considered as a cause of another given variable. Although this restriction simplifies computations, its representational power as a consequence is reduced, since it forces a single vertex from all causes sharing a common consequence. For example, when a doctor discovers evidence in favour of one disease, it reduces the likelihood of other diseases that could explain the patient's symptoms. However, despite this representational loss we will show in Chapter 4 that 'tree' based BN structures can be just as effective as less restricted models. In the following section we describe a technique for inducing 'tree' structures with section 3.2 introducing the MIM classifier. In section 3.3, we review some related work concerning the use of the Chow and Liu algorithm. Section 3.4 describes how we use the 'tree' structure and corresponding branch weights to classify new evidence, with section 3.5 reviewing related work in respect of other 'arc weight' concepts. Finally, in section 3.6 we summarise the chapter.

3.1 Induction of Tree Structures

Learning algorithms essentially measure the volume of the information flow between two nodes and this in turn is used to guide the construction of the Bayesian network from a given data set. A common technique for measuring this flow of information is by use of Mutual Information (MI). In information theory, mutual information is used to represent the expected information gained on sending a symbol and receiving another. In Bayesian networks, if two nodes are dependent, knowing the value of one node will give us some information about the value of the other node. This information gain can be measured using mutual information. Therefore, the mutual information I between two nodes can tell us if two nodes are dependent and how close their relationship is.

An algorithm that has the characteristics of graphical learning approaches is that proposed by Chow and Liu [CL68]. This algorithm adopts a *search* and *scoring* based method and views the learning problem as a search for the structure that can best fit the data. The algorithm starts with a graph without any edges and uses a *search* method to add on edges to the graph. Once found a *scoring* method is used to see if the new structure is better than the old one. If it is, the newly added edge is retained and the algorithm continues by trying to add another one. This is essentially repeated until no further new structure is better than the previous one. In the case of the Chow and Liu algorithm, the Kullback-Liebler [Kul68, KL51] (K-L) cross entropy is used as the measure of best *score*.

3.1.1 The Chow and Liu Algorithm

The Chow and Liu algorithm takes a probability distribution P as its input and constructs a Bayesian network in the form of a Tree as its output. This is achieved in only $O(N^2)$ pair-wise dependency calculations with each calculation using only second-order statistics, where N is the number of nodes.

The concept uses a notation of tree dependence to approximate the underlying probability distribution data. In particular, the algorithm allows us to find the best approximation of an n-order distribution by a product of $(n-1)$ second order distributions. The main result can be formulated as follows.

A probability distribution is called a distribution of the tree dependence if it has the following form:

$$P'(Z_1, Z_2, \ldots, Z_n) = \prod_{i=1}^{n} P(Z_i \mid Z_j) \quad j \in (0,1,\ldots,n) \; and \; i \neq j$$

Where $P(Z_i \mid Z_0)$ is by definition equal to $P(Z_i)$. A probability distribution of a tree dependence $P^t(Z_1, Z_2, \ldots, Z_n)$ is an optimum approximation to the "real" distribution $P(Z_1, Z_2, \ldots, Z_n)$ if and only if its dependence tree t has maximum weight, where the weight is determined by the mutual information $I(Z_i, Z_j)$ between two variables Z_i and Z_j :

$$I(Z_i, Z_j) = \sum_{Z_i, Z_j} P(Z_i, Z_j) \log\left(\frac{P(Z_i, Z_j)}{P(Z_i)P(Z_j)}\right) \geq 0$$

It has been proved by Chow and Liu that, maximising the total branch weight is equivalent to minimising the Kullback-Liebler measure:

$$D(P, P^t) = \sum_{Z} P(Z_1, Z_2, \ldots, Z_n) \log\left(\frac{P(Z_1, Z_2, \ldots, Z_n)}{P^t(Z_1, Z_2, \ldots, Z_n)}\right)$$

This measure can be interpreted as the difference between two distributions. It is always positive when the distributions are different and is zero when they are identical.

Procedure 1 (Restricted network) [CL68]

The procedure of Chow and Liu can be summarised as follows.

1. Compute the Mutual Information

$$I(Z_i, Z_j) = \sum_{Z_i, Z_j} \hat{P}_D(Z_i, Z_j) \log\left(\frac{\hat{P}_D(Z_i, Z_j)}{\hat{P}_D(Z_i)\hat{P}_D(Z_j)}\right) \quad \text{between each pair of variables } i \neq j \ \ldots(1)$$

2. Build a complete undirected graph in which the vertices are the variables in Z .

   Annotate the weight of an edge connecting Z_i to Z_j by $I(Z_i, Z_j)$.

3. Build a maximum weighted spanning tree of the graph [CLR90, Pea88].

Here $Z = \{Z_1, \ldots Z_n\}$ is the feature set of discrete variables and $\hat{P}_D$ is the measure defined by the frequencies of events in the data D .

Figure 3.1. Chow Liu Tree Algorithm Procedure.

3.2 The Mutual Information Measure Classifier

The task of classification is an important activity as it represents the foundations on which we make future decisions. It seems logical therefore to want to demonstrate the feasibility and usefulness of a new classifying algorithm on problems taken from the 'real' world in order to facilitate its validation. The approach most commonly adopted by other researchers for assessing these types of algorithms utilises the UCI repository[6] to test the new algorithm and subsequently makes comparisons with other competing methods, typically the NB classifier. Examples illustrating this approach can be found in [KS00, Paz96, CG99, CG01, Sin98, FGL98, FGG97 and LKM01]. In this thesis, we also adopt this methodology, however for our experimental work the use of the UCI repository is complimented by a 'real' data set describing the medical domain Acute Abdominal Pain (Refer to case study in Chapter 7 for more details).

In our investigation the objective is to produce a new classifier that takes advantage of the topology of a 'polytree' or Singly Connected Network (SCN), that is, modelling a restricted qualitative structure induced by a BN. We consider this particular aspect significant as it provides a valuable insight into the domain under study, which in turn leads to a mechanism for knowledge discovery. The approach does however have a drawback. In order to apply the proposed learning algorithm we have to assume that all the domains can be approximated by an underlying 'tree' based distribution. Simplified models, such as Singly Connected Networks, have been shown to represent good approaches to automatic classifier construction [Cam96] alleviating the time consuming processes of learning and inference compared to that required for GBNs. Despite their loss of representation capabilities, SCNs gain in efficiency and simplicity as they can be built from data using only pair-wise marginals. That is, simplification is achieved by selecting a topology that allows efficient propagation, for example a SCN or 'polytree' [Das99, Cam96, RP87]. In taking this approach, in respect of the development of the technique, we are adopting a theoretical method, however in the same way as those cited previously, the actual assessment/evaluation of the new technique is experimental. The precise details of the statistical tests employed for analysing the generated

---

[6] A collection of benchmark data sets

quantitative data are in Chapters 4 and 6, in respect of the UCI data, and Chapter 7 for the AAP data set[7].

In the sections that follow, we propose a new inference technique based upon the well-known mutual information between two random variables. We call this a Mutual Information Measure (MIM) classifier as it corresponds to the restricted class of trees built from mutual information. This is an extension to Pearl's 'polytree' construction utilising the Chow and Liu (CL) tree building algorithm. Moreover, rather than use the inference method traditionally employed by BNs, we consider the mutual information or 'branch weight' as a measure of strength for an edge linking multi-state vertices and further demonstrate that this branch weight representation can be used to classify locally, new evidence presented to a SCN. The advantage this approach offers is that the qualitative aspect can be satisfied by employing the efficient CL algorithm, whilst the complimentary use of pair-wise marginals for branch weights, ensures the new method is not influenced by unreliable CPT probability estimates. The concept of information 'weight' has been researched and used in many other approaches [Boe95, Dra95, JN97] together with applications that have utilised the mutual information measure [YZ01, NJ98, Bat94, KJ96, LTD01 and CGK 02].

Prior to inference we first construct a SCN based upon an information theory based technique. We achieve this by building the Mutual Information Measure (MIM) classifier structure in two stages. In the first, we use the Chow and Liu tree building algorithm to build the skeleton structure as described in Procedure 1, Figure 3.1. Once constructed, stage two transforms the structure to a singly connected network. In stage two we determine the node ordering from the orientation of the tree edges with respect to the class node.

The formal algorithm for constructing the Maximum Weight Spanning Tree (MWST) can be described by the pseudo code shown in the Figure 3.2.

The algorithm results in $n(n-1)/2$ pairs of $I(Z_i, Z_j)$ being generated with the algorithm terminating when $(n-1)$ branches have been selected, at which point the dependency tree has been constructed. Essentially, by looking at the association of all variables in terms of couples, an $(n-1)$ undirected branched tree can be constructed, where n is the number of variables.

---

[7] Under the section heading '*Experimental Design*'

FOR i=1 to N-1 **DO**
BEGIN
FOR j=i+1 to N **DO**
BEGIN
Find all second-order probability distributions $P(Z_i, Z_j)$

- (A) From the given (observed) distribution $P(Z)$, compute the joint distribution $P(Z_i, Z_j)$ for all variable pairs.

Calculate mutual information measures $I(Z_i, Z_j)$

- (B) Using the pair-wise distributions (A), compute all $n(n-1)/2$ branch weights and order them by magnitude.

END
END
Branches No=0
WHILE (Branches No < (N-1))

- (comment) Repeat (C) until $(n-1)$ branches have been selected.

BEGIN
Select two variables Z_i, Z_j that have largest $I(Z_i, Z_j)$

- (C) Assign the largest two branches to the tree to be constructed.

Add the branch (Z_i, Z_j) to the tree
IF (there is a loop in the tree)
Delete the branch (Z_i, Z_j)
ELSE
Branches No = Branches No + 1
END IF

- (comment) now Examine the next-largest branch, and add it to this tree.

END

Note: for those branches having equal weight, the first largest branch found will be selected to define the structure of the MWST.

Figure 3.2. Maximum Weight Spanning Tree Algorithm.

Figure 3.3 (a-c) illustrates the stepd taken in constructing the MIM classifier for the domain 'Flare' using the MWST algorithm of Figure 3.2 and the procedure detailed in Figure 3.4. The 'initial' process uses the CL algorithm as defined by equation (1) (Procedure 1, Figure 3.1) to calculate the $n(n-1)/2$ mutual information measure values for every feature pair over the 'n' features of the domain. Figure 3.3(a) shows a subset of the sorted values arranged in descending order by MI size along with their corresponding edges.

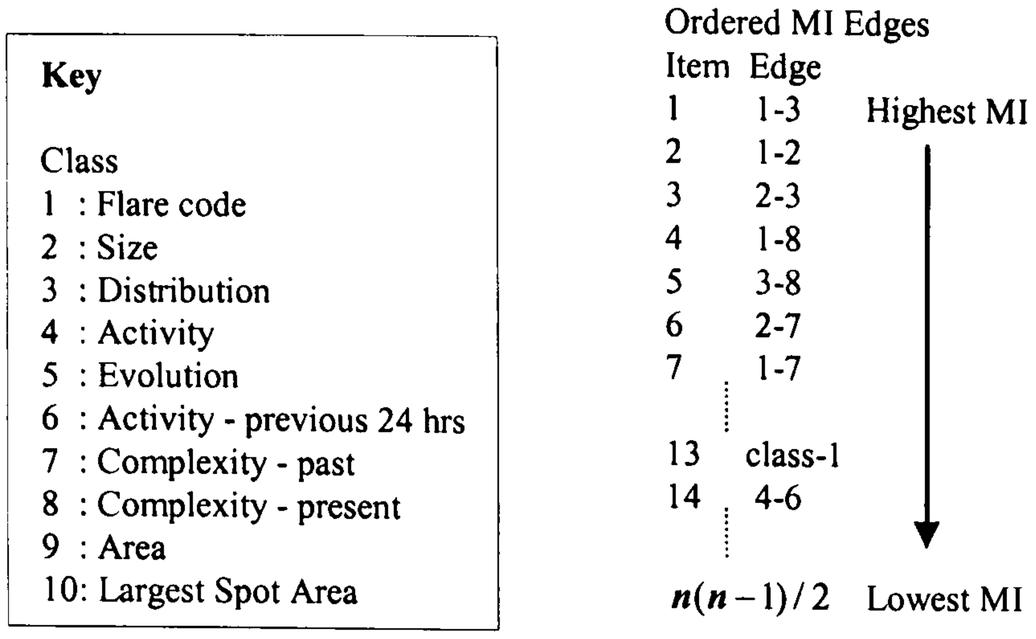


Figure 3.3(a). MI values – 'Flare' Data Set

The first phase (step 1) for constructing the classifier is depicted by Figure 3.3(b). Starting with the highest MI valued edge, here item 1, edge 1-3[8], a tree structure is built by selecting additional edges by the MI order from the sorted list. In the event a loop occurs, the edge is discarded and the next valid one considered. The process is repeated until a $(n-1)$ branched tree is constructed which represents the completed MWST.

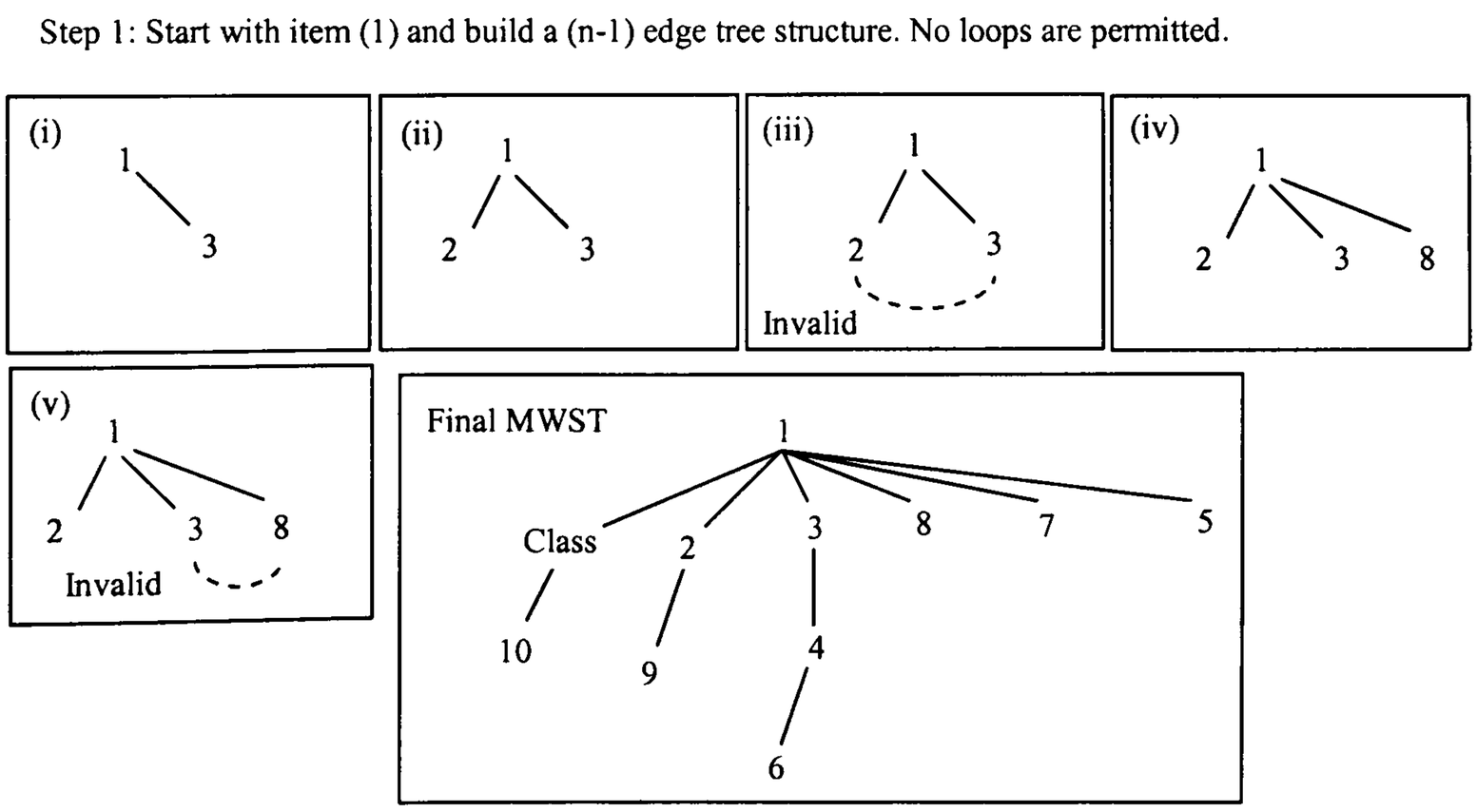
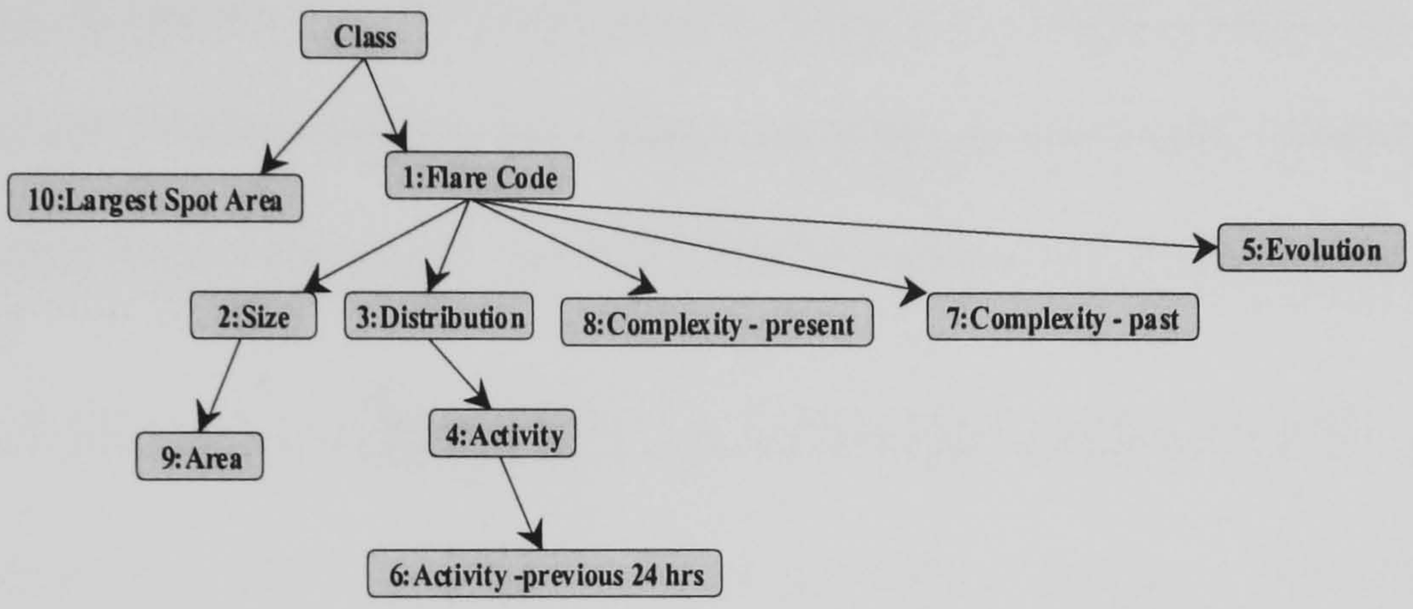


Figure 3.3(b). MWST Learning Procedure.

[8] For simplicity feature names are coded as shown in the 'key' shown in Figure 3.3(a).

Figure 3.3(c) illustrates the final phase (step 2) by identifying the class variable and orientating the tree structure such that the class becomes the root of the tree[9]. Directionality for the edges is then taken outwards from the class, completing the qualitative representation of the classifier. In respect of the quantitative aspect, MI values are assigned to the corresponding edges as calculated by equation (1) (procedure 1, Figure 3.1), essentially those shown in the list of Figure 3.3(a).



Step 2: Set Class variable as root of the tree and assign directionality outwards from the root.

Figure 3.3(c). MIM Classifier – 'Flare'.

The procedure for learning the MIM Classifier can be further summarised as detailed in Figure 3.4.

1. Input Training Data of the domain $\{C, Z_1, \ldots, Z_n\}$.
2. Build the undirected tree structure using the Chow and Liu algorithm.
3. Select the domain 'Class' variable as the root of the undirected graph.
4. Transform the graph from (3) into a directed (SCN) by setting the direction of all edges to be outwards from the class vertex.
5. Output SCN (MIM Classifier structure) $G(N, A)$

Figure 3.4. MIM Classifier Learning Procedure.

[9] Feature names are included for completion.

Consider the structure in Figure 3.5, which represents a subset of a MIM classifier structure as generated from the procedure detailed in Figure 3.2.

In this example the class vertex C has $C_1,\ldots,C_m$ multi-state values and the attribute Z_1 $\{Z_{1-1}, Z_{1-2}\}$ multi-state values. The mutual information measure $I(C,Z_1)$ as defined by equation (1) (Procedure 1, Figure 3.1) is a measure of the dependence between the two variables C and Z_1.

The value $I(C,Z_1)$ represents the summation of the 'individual' mutual information that is associated with each pair of class-attribute state values, that make up the overall 'branch weight' $I(C,Z_1)$. We call these 'individual' values mutual information elements and denote them by $I'(\ )$.

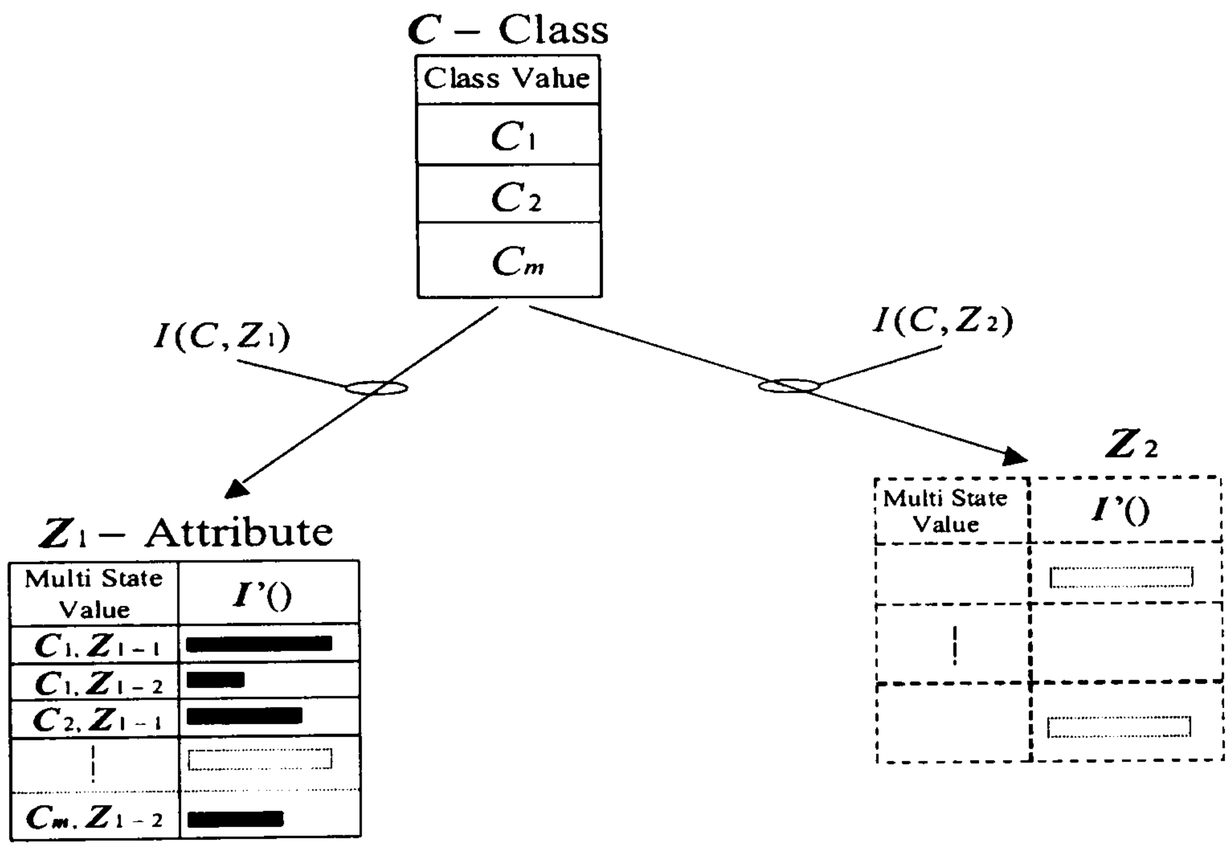


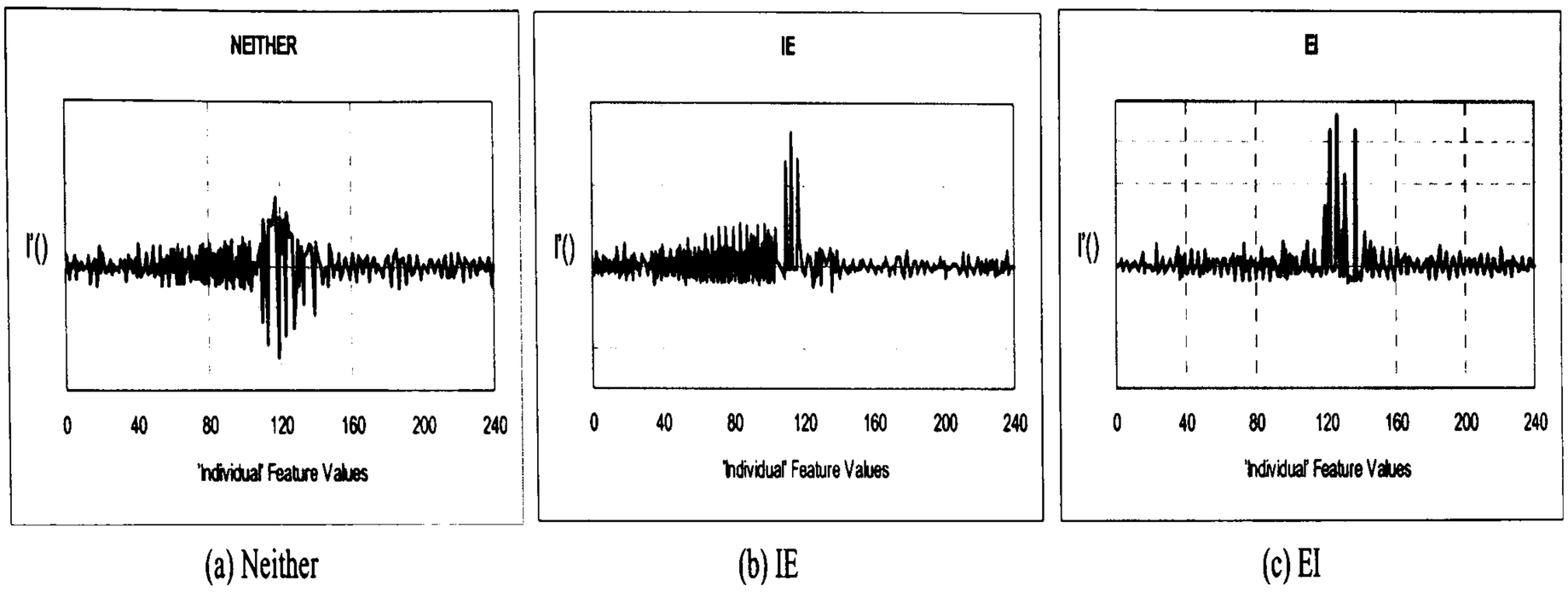
Figure 3.5. MIM Classifier Tree Structure (subset).

The plots in Figure 3.6, using the UCI 'DNA' database, illustrate the distribution of calculated $I'(\ )$ values with respect to the three class labels describing the primate splice-junction gene sequences (DNA).

For each class $C=\{C_1,\ldots,C_3\}$ the distribution of $I'(\ )$ values reveals that there is a characterisation 'profile', which is distinctly different for each class label.

During the process of classification we 'introduce' evidence in the form of a feature vector $\{Z_1,\ldots,Z_n\}$ for $n=60$ attribute instantiations. To propagate this information or evidence in our SCN

we update the 'branch weight' elements $I'(\ )$ in respect of each class $C = \{C_1, \ldots, C_m\}$ where $m=3$, representing the class-states Neither, IE and EI.



(a) Neither (b) IE (c) EI

Figure 3.6. DNA Class Distributions of $I'(\ )$ values for each of the three Class-attribute measures of dependence. Where (a) represents the class 'Neither', (b) the class 'Intron/Exon' boundary or donors and (c) the class 'Exon/Intron boundary or acceptors. Each of the 60 attributes has four states (A,C,G,T) resulting in a total of 240 'individual' feature values. We omit values from the $I'(\ )$ axis as only the 'profile' characterising each class is relevant.

3.3 Related Work

The Chow/Liu (CL) tree structuring algorithm has been the corner stone for many approaches since the introduction of several probabilistic proposals that have extended the spanning tree model.

One prominent example is TAN or Tree Augmented Naive Bayes [Gei92], which maintains the computational simplicity of NB but allows additional edges between features. It is well know that NB performs well even against state of the art classifiers such as C4.5 [Qui93]. What TAN attempts to do is capture correlations among the features by adopting less restrictive assumptions than NB, but without loss to its prediction capability. In the augmented structure, Figure 3.7, an edge from Z_1 to Z_2 implies that the two attributes are no longer independent given the class variable C.

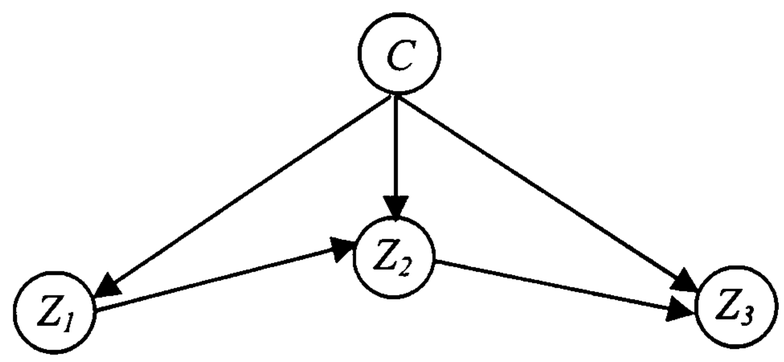


Figure 3.7. A simplified TAN structure example.

From Figure 3.7, a TAN is defined as a NB with augmenting edges between features with the class variable C having no parents. Attributes have as parents the class and at most one other attribute. The algorithm for learning TAN classifiers [FGG97] learns a tree structure over $Z \setminus \{C\}$, using mutual information tests conditioned on C. Where $Z = \{Z_1, \ldots, Z_n, C\}$ are the features of the data with C representing the class variable. Procedure 2, Figure 3.8, describes the TAN learning algorithm.

Procedure 2:

1. Input learn partition $Z \setminus \{C\}$
2. Carry out conditional MI test $I(Z_i, Z_j \mid \{C\})$ between each pair of attributes $i \neq j$
3. Assign weights to edges $I(Z_i, Z_j \mid \{C\})$
4. Build MWST
5. Select a root variable and set directions of all edges to be outward from it.
6. Attach C as a parent of every Z_i, where $1 \leq i \leq n$
7. Learn parameters of TAN
8. Output TAN

Figure 3.8. TAN Learning Algorithm Procedure.

The TAN learning algorithm extends the CL algorithm requiring $O(N^2)$ conditional MI tests and assigning 'weights' $I(Z_i, Z_j \mid \{C\})$ to only the Z-Z[10] edges. In the MIM classifier the edges are unconditional MI and assigned to all pairs which includes the C-Z edges. Unlike the MIM classifier, TAN builds a MWST, which excludes the class variable adding it once the structure (tree) is complete and then attaches edges from the class variable to all features $\{Z_1, \ldots, Z_n\}$. In contrast, the MIM classifier tree structure is derived using all the data features including the class variable. In the

[10] Z-Z denotes feature-feature association, C-Z class-feature association.

first stage of construction the class variable is treated as an ordinary feature and only selected as the root, after the MWST is completed, which includes the class variable.

By the use of the CL algorithm, TAN imposes a tree-based topology that restricts a node to have at most two parents. Whilst this does reduce the search space and enable the CPT size to be more manageable, the approach however, does not avoid the possibility of intractability.

The BAN, BN Augmented Naive-Bayes [CG99] further extends TAN by allowing the features to form an arbitrary graph, rather than just a tree, Figure 3.9.

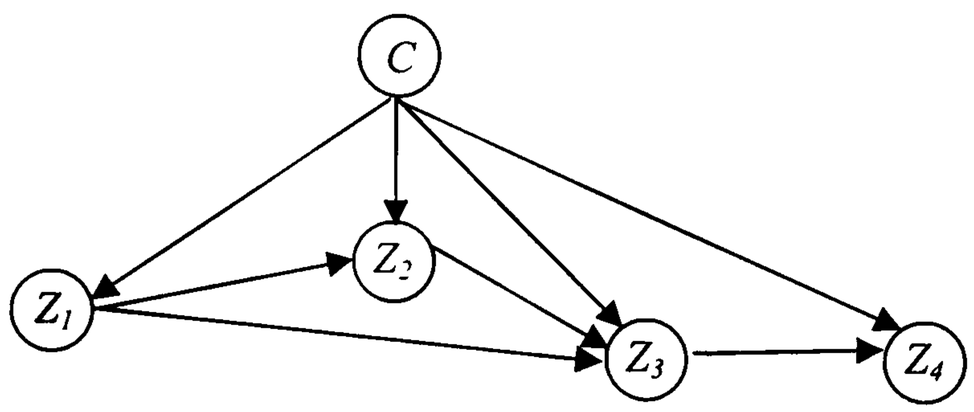


Figure 3.9. Simplified BAN structure example

The algorithm for learning BAN also extends the CL algorithm and is defined by a three phase BN learning algorithm.

Phase I, 'drafting', computes mutual information of each pair of attributes as a measure of closeness, and creates a draft based on this information (a SCN). In phase II, 'thickening', edges are added when pairs of attributes cannot be *d-separated*, with phase III, 'thinning', verifying the necessity of each edge. In this last phase, each edge is examined using CI tests and removed if the two attributes of the edge can be *d-separated*. The CI tests utilise conditional mutual information tests, as used for TAN construction. Cheng's algorithm [CG99] also adds an edge orientation procedure to phase III to determine directionality on the structure. When prior node ordering is supplied the complexity is $O(N^2)$ otherwise it rises to $O(N^4)$.

As was the case for learning TAN, the class node is not included as part of the initial construction but added after completion of the structure (in this model a network), with edges from the class variable being attached to all features. Since BAN still represents an unrestricted BN learning algorithm, the issues of inference complexity will nevertheless be present. As Chickering [Chi96] pointed out even restricting the structure to have only two parents, does not prevent the inference from becoming NP-

hard. Where poor prior node ordering is supplied, as for any GBN, the constructed BANs can still potentially represent an intractable solution.

A common feature of TAN, BAN and the MIM classifier is the assignment of the class variable as the root of the derived structure. This is in contrast to the GBN which treats the class variable as an ordinary node. Although this does permit an even more flexible structure, some researchers [CG99, FGG97] have argued that the former approach can improve classifier performance. Whilst, by design, this is the same structurally for the NB classifier, TAN, BAN and the MIM classifier offer a more expressive representation than NB.

Cheng [CG99] concluded from his experiments that methods based on CI tests, such a mutual information tests, are more suitable for BN classifier learning than the more standard scoring methods. In the case of the MIM classifier, the use of Chow/Liu's algorithm means it characterises both a constraint (CI) and score based learning approach. The CL algorithm finds a structure with the best score (cross-entropy K-L) but does so by analysing the pair-wise dependencies. Heckerman [HMC97] compared score and CI learning approaches and found score based were better than CI for modelling a distribution, whilst Friedman [FGG97] on the other hand found that using general scoring methods may result in poor classifiers, a conclusion also arrived at by Greiner [GGS97].

As we shall show in Chapter 5, whilst we initially define our MIM classifier model via a search and score approach, we can in some cases improve classification performance using a form of 'constraint' edge addition.

In Cheng's approach, when the underlying structure is a 'tree' the drafting phase essentially defaults to that of the MIM classifier's structure. The difference lies only in the choice of node ordering, although as a classifier, the class variable will in both approaches, be assigned as the root. Phase II and III in this situation will not make any additional changes to the model.

Pearl [Pea88] and Rebane [RP87] further extended the CL algorithm to recover an undirected Markov tree from a given discrete JPD using a MWST algorithm. They show the same algorithm also recovers the undirected skeleton (topology) of a 'polytree'. Structurally, this is the exact topology of the MIM classifier. Pearl and Rebane further developed an algorithm to recover the causal directionality of the edges, but as reported by Pearl, full recovery was not always possible requiring completion by an expert of the domain. In the MIM classifier node ordering is orientated in

conjunction with the class variable assigned as the root. As we shall demonstrate in Chapter 4, a poor choice of node ordering does have an impact on the 'polytree's' predictive performance as a classifier. As this choice defines the edge directionality, the dimensions of the corresponding CPTs may be large resulting in a model that is intractable. This may be more apparent especially in cases where an attribute has 'multi' parents. Since the MIM classifier in contrast uses branch 'weights', which are already available from the MWST branch list, directionality and CPT size do not impact on its inference complexity in respect CPTs.

A novel application of the CL algorithm is the CL multinet classifier [FGG97, HKL03]. This classifier allows different attribute dependencies for different class-states. In this approach a different network is learned per class-state, with each network, a tree, constructed using the CL algorithm. By constructing trees the complexity of the network is constrained [FGG97]. In contrast, a TAN forces relations among features to be the same for all the different instances of the class variable. This is essentially the case, structurally, for the MIM classifier. However, when used as a classifier there is an implied 'multinet' by use of the class 'profiles' which define each class-state characterisation. Unlike the MIM classifier, CL multinets are expensive to learn [FGG97]. Learning a CL multinet involves the application of the CL algorithm, but in this approach it is in respect of each class-state data partition, that is, its' equivalent JPD. For those edges relating to C-Z there will be a correspondence to those of the MIM classifier (although as a single tree structure not as separate trees for each class-state). By partitioning the data by class-state the multinet takes care of the C-Z associations. In the case of the Z-Z relationships, the CL multinet will use subsets, corresponding to a class-state, of the 'learn' data partition, whilst the MIM classifier uses the entire 'learn' data partition. This means the branch weights for Z-Z edges reflect strengths from the entire data set, whilst for the CL multinet it is only with respect to the corresponding class variable state.

Another network structure derived from the underlying tree structure of the CL algorithm is proposed by Sucar [SP$^+$97]. This approach begins by constructing a skeleton tree structure using the CL algorithm, then determines a node ordering on the undirected graph in accordance with the size of the MI edge 'weights'. New edges are then added to the tree, again by the same edge 'weight' size, and are controlled by the assigned node ordering to avoid producing cycles. As the objective is to predict a specific hypothesis node (set as the root) termination of the algorithm is determined when

the desired performance level is reached. Classification using the completed structure, further requires the parameters to be estimated (CPTs) from the available data. This approach however, can lead to dense and possibly intractable solutions. Since edge additions are governed by the size of MI, this can result in having to add several irrelevant (with respect to the chosen hypothesis node) edges to the structure to obtain the desire level of performance accuracy. In Chapter 5, we will show that the MIM classifier can be enhanced by the addition of edges to the initial CL tree structure just as Sucar proposed. By focusing on the class variable (as root) performance can, in some domains, be improved, but in the case of the MIM classifier it is not at the expense of inference complexity. Further applications that use the CL tree structuring algorithm as their underlying structure are described in [Mei99], who proposes a mixture of trees together with two accelerated CL algorithms and in [FZ00], which extends TAN to a credal classifier (TANC). A credal classifier is a function that maps instances of the features to a set of class-states. That is, when the available knowledge is insufficient to isolate a single class-state, a set of alternatives is presented. Despite being modelled on the efficient tree-based classifiers, TANC is reported to suffer from overfitting [ZF03] with feature selection and model pruning proposed in order to overcome this problem, characteristic to networks. The use of the CL algorithm for constructing a tree-based model has been shown to offer a competitive approach in many classification problems. However, in doing so, these applications make the assumption that the underlying structure of the training dataset is a 'tree'. Work carried out by Huang [HKL02] proposed a procedure to avoid violating this assumption. By considering nodes as a 'large node', networks can be transformed to represent a tree whilst maintaining their original conditional independence relationships. This is achieved by firstly constructing a tree in which the nodes are a subset of the features of the data set. First they create a CL tree as a draft approximation over the data set, and then they refine the tree into a 'large node' structure. The construction is guided by the frequent 'itemsets' which are essentially subsets of features which come out together with each other frequently.

Since the MIM classifier is based upon the use of the CL algorithm for defining its class MB, it too makes the same assumption, that is, the underlying structure is tree-based. Despite this, as shown in Chapter 4, the predictive accuracy of the MIM classifier remains competitive with the unrestricted GBN for many of the datasets studied.

Common to the approaches reviewed is the underlying assumption that the domain can be represented by a tree and that the structures can be derived by an application of the CL algorithm. However, despite the use of this efficient algorithm, node ordering and subsequent CPT dimensionality still present a problem. The approach taken by the MIM classifier is to adopt a heuristic for node ordering which orientates the structure such that the class node is configured as the root of the tree. This placement enables the classifier to be independent of any node ordering choice whilst defining a class MB consistently. Both BAN and 'polytree' are influenced by the final topology and may perform as a classifier badly as a result of poor node ordering choices. For BAN where no prior node ordering is provided the complexity is $O(N^4)$, whereas for the 'polytree' full recovery is not always even possible without expert guidance. In the case of TAN this problem is contained but at the cost of representational constraint by restricting nodes to have at most only two parents.

With regards to the task of classification the use of branch weights by the MIM classifier enables new evidence to be classified without the concerns associated with CPTs. This approach effectively avoids the issues relating to large numbers of probabilities to be estimated, particularly in circumstances when nodes have 'multi' parents. Moreover, when the dimensionality of the domain is high and the data set sparse, populating the CPTs may result in estimating unreliable probabilities. In contrast, the MIM classifier's approach calculates branch weights using the pair-wise marginals and only these (focused on the class MB) are required to be updated when presented with new evidence to be classified. This technique avoids the problems of having to update a potentially unrealistic number of probabilities associated with the CPTs that the other approaches need to deal with.

Multinet construction uses a subset of the domain data as defined by the class-states. As a consequence when there are class-state imbalances, typical in 'real' world domains, the selection of Z-Z feature associations may not adequately model the domain and subsequently be reflected in the reliability of the corresponding CPTs. The approach adopted by the MIM classifier however uses the entire data set to calculate Z-Z feature associations and thus its edges will have a 'stronger' measure of association between features than those of the multinet. The resulting MIM tree structure, directed by the magnitude or strength of the Z-Z and C-Z edges, will better model the domain and in turn define a more representative topology for the class MB. For the MIM classifier, there is only one tree

structure which is common for all class-states, and thus it is not as computationally expensive to construct as the multinet, which requires one structure (tree) for each class-state. In addition, as we will demonstrate in Chapter 5, the MIM classifier is not constrained by topology in order to manage model complexity or the task of classification.

3.4 Classification - MIM Classifier

Let T_m represent a MWST for a tree dependent probability distribution P_t, where P_t is a Markov field relative to a tree [Pearl 88]. If a feature vector $Z = \{Z_1, \ldots, Z_n\}$ describes a new observation of the domain then the probability distribution P_t will be updated to $P_t^{'}$. For Z belonging to a particular class value C_i, where $i = (1, 2, \ldots, m)$, the new MWST for $P_t^{'}$ can be represented by $T_{m_{Ci}}$. If we repeat this for each possible value of C_i then m MWSTs will be constructed. In order to assign a feature vector Z to a classification value of class C_i, we need only find the maximum $T_{m_{Ci}}$ from the m MWSTs. As was shown by Chow and Liu [CL68] this is equivalent to finding the maximum total branch weight for T_m, thus minimising the K-L measure. In other words we are calculating the relative difference between each $P_t^{'}$ in respect of the m probability distributions represented by $T_{m_{Ci}}$ for $i = (1, 2, \ldots, m)$. Identifying the specific class value i to which the new observation Z belongs, requires finding the MWST ($T_{m_{Ci}}$) that has the greatest total branch weight. The winning class value $i = (1, 2, \ldots, m)$ will thus identify one of the mutually exclusive classes C_i that corresponds to the maximum $T_{m_{Ci}}$.

For the representation of Figure 3.5, considering the subset only, the joint probability distribution $P(C, Z_1, Z_2)$ can be written as $P(C)P(Z_1 \mid C)P(Z_2 \mid C)$ by the chain rule. If we ignore the edges $C - Z_1$ and $C - Z_2$ then the three vertices can be considered independent giving:

$$P^{'}(C, Z_1, Z_2) = P(C)P(Z_1)P(Z_2)$$

and from equation (1) (Procedure 1, Figure 3.1) .

$$I(C,Z_1,Z_2) = \sum_{C,Z_1,Z_2} P(C,Z_1,Z_2)\log\frac{P(C,Z_1,Z_2)}{P'(C,Z_1,Z_2)}$$

$$= I(C,Z_1) + I(C,Z_2)$$

This is the sum of the individual branch mutual information measure values between pairs of neighbours. More generally in terms of mutual information measure 'branch weights' $W_t(Z_0,Z_1,Z_2) = W_t(Z_0,Z_1) + W_t(Z_1,Z_2)$. So if we sum over all these branch weights $W_t(\ )$ we will have a combined measure of their effect. Since mutual information is symmetric then $C \rightarrow Z_1$ is the same as $C \leftarrow Z_1$, so that directionality in our representation will not alter the value of the branch weight $I(C,Z_1)$.

If we now consider a representation of the domain data set as depicted in Figure 3.10 we can see that an instantiation of an evidence vector $Z = \{Z_1,\ldots,Z_n\}$ can be classified as belonging to one of the mutually exclusive class labels $C = \{C_1,\ldots,C_m\}$ by local computation.

In Figure 3.10, the training sample of the domain can be viewed as a series of class partitions characterising samples belonging to a particular class.

Each partition is described by a vector of class features $Z = \{Z_1,\ldots,Z_n\}$ and this will be the case for each class label where $C = \{C_1,\ldots,C_m\}$. The actual dimensions of the partitions may or may not be the same for each class and will correspond to the specifics of the domain data set.

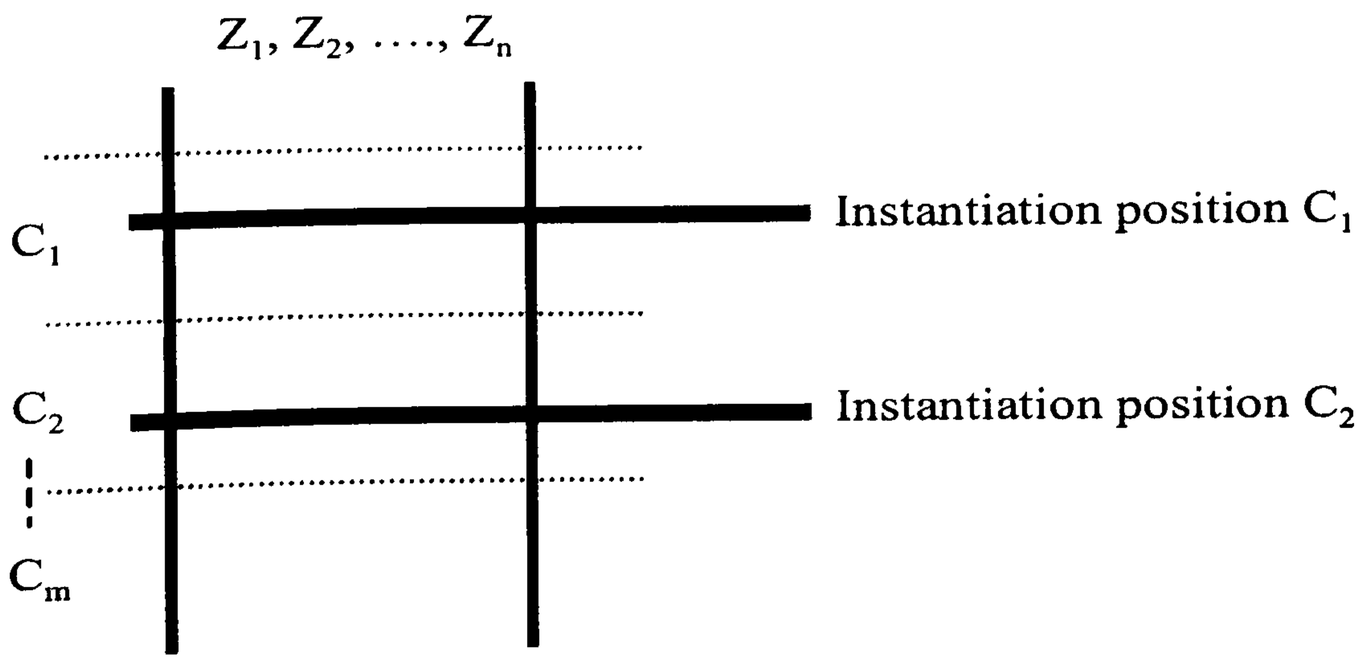


Figure 3.10. Domain Data set Representation example.

An instantiation of an evidence vector $Z\{\ \}$ in position C_1 will increase the marginal $P(C_1)$ and update the joint probabilities $P(C_1, Z)$ in respect of the evidence vector $Z = \{Z_1, \ldots, Z_n\}$ and their values. Similarly, for an instantiation in positions $C_2, \ldots, C_m$. Since the evidence vector $Z\{\ \}$ will be common for all $C\{\ \}$ instantiation positions, the marginal probabilities $P(Z)$ for each value of Z, due to the evidence, will be updated, but remain at a constant value for each instantiation position.

In terms of our MIM structure this implies that any changes of information, branch weights, due to observing an evidence vector $Z\{\ \}$, will only be measurable on edges that are directly associated with the class vertex. The corresponding information on edges not associated with the class will remain at a constant value, for each instantiation position $\{C_1, \ldots, C_m\}$. Thus, the classification can be achieved locally using a subset of the domain features as defined by the SCN. In situations where the structure corresponds to that of Naive Bayes, the feature size will be n. However, unlike Naive Bayes the same extreme assumption of conditional independence for all features given the class is not made.

3.5 Related Work

The idea of branch weights, sometimes referred to as arc weights, in BNs is a versatile concept. The notion of connection strength was first proposed by Boerlag [Boe95], who used arc weights to draw contour maps of a BN as well as evaluation. This particular application however was limited to only binary nodes. Based on the idea of connection strength Jitnah [JN97] proposed a framework for implementing incremental evaluation of BNs called treeNet. The procedure, which is limited to 'polytree structures', first transforms a BN into a tree structure and then applies anytime inference. Anytime algorithms are concerned with the improvement of the quality of a result with computation time. This is particularly useful for real-time decision systems such as medical diagnosis. The transformation is guided by setting the query node (as selected) to be the root of the treeNet. A similar approach is taken to define the topology of the MIM classifier which is also guided by the placement of the class node as its root. Unlike treeNet however, the class node is identified after construction of the tree and edge directionality assigned as described previously in Figure 3.4. When

new evidence is presented to treeNet the posterior probability of the query node is re-calculated using a modification of the 'polytree' message-passing algorithm [Pea88]. This modification essentially restricts the messages to be passed in one direction only, that is, towards the query node.

Arc weights are used as a measure of strength to provide an estimate of how much a node can be affected by each neighbour. The neighbour at the arc with the heaviest weight will have the greatest influence on the node of interest. In contrast, the branch weights of the MIM classifier collectively provide a measure of change, in respect of the child nodes, that affects the class or parent node. Unlike treeNet it is the relative difference in MI weights summed over the class MB that identify a target class-state, rather than combining incoming messages from child nodes to compute the query node belief. In the case of the MIM classifier, the branch weights themselves are updated in support of new evidence and in the worse case will be, for m class states, $m(n-1)$ where the topology configures as that depicted by NB. In treeNet arc weights are used to traverse the structure and are computed locally at each arc using only the values stored in the associated CPT. However, with 'polytrees', just as for GBNs, the possibility of 'multi' parented nodes and the corresponding exponential rise in size of the CPTs means estimating reliable weights may be difficult in some domains. Even with a modest complexity, updates to node beliefs may be intractable. The MIM classifier on the other hand avoids this by calculating branch weights based upon pair-wise marginals and is not affected by the CPT dimensionality and to some extent, the scarceness of the domain data set.

Although arc weights provide a flexible way of evaluating the treeNet the limitation to 'polytree structure' application discounts its use for some 'real' world problems[11]. Despite being derived as a 'tree' based structure, this limitation is not true for the MIM classifier. As section 3.3 pointed out, the task of classification is focused on the class MB and is therefore independent of the 'overall' domain representation whether portrayed as a tree or a network.

Draper [Dra95] proposed a propagation algorithm called Localized Partial Evaluation (LPE) which computes interval bounds on the marginal probability of a specific query node. Arc weights are used to search for the best vertices to include in an 'active set', a subset of the entire network, and then a standard message-propagation [Pea88] is used over this set. In Drapers' approach interval-valued

---

[11] Extensions to treeNet are proposed as 'further work' in order to deal with networks [JN97].

messages are used instead of the normal point-valued messages and as for treeNet message passing is in the one direction towards the query node. Unlike treeNet, LPE is not limited to 'polytree structures' and can be applied to multiply connected networks. This is achieved by converting the network into a tree structure using typically the technique proposed by Lauritzen and Spiegelhalter [LS88].

In the MIM classifier the CL algorithm acts as an implied feature subset selector and so the equivalent 'active set' is defined by the class MB. Since classification focuses on the MB, as Chapter 5 will show, the technique is not limited to tree structures but can potentially be applied to networks. Unlike Drapers' approach the 'active set' in the MIM classifier may not be the best set of vertices. This is due to the use of the CL algorithm which is prone to either missing out relevant features or adding irrelevant features.

A further use of arc weights was proposed by Thomas [Tho96]. MI arc weights defined by the branches of a MWST structure were considered as 'weights' of a feed forward Neural Network Expert System, a model similar to Gallents' MACIE [Gal94]. The substitution of MI 'weight values' into Gallents' linear discriminant function implied an overall strength or ordering with respect to each class state instantiation. Classification of new evidence presented to the neural network is easily determined by selecting the highest resulting function value. Unfortunately, the approach is limited to topologies restricted to models that assumed extreme CI.

Whilst MI can be considered 'local', that is between neighbours, Nicholson [NJ98] considered the information content of a set of vertices, forming a connected region. The approach taken is similar to that adopted by the MIM classifier. MI is used to measure branch relevance during the construction of the MWST and then the 'connected region' identified by the class and its associated child vertices. Nicholson also proposed the idea of a path weight in relation to a query vertex. The approach assesses the relative impact of selected vertices on the posterior belief of a query vertex even when the vertices are not necessarily neighbours. For the MIM classifier however, the path weight is trivial as the region of the structure with respect to the class vertex, has a similar topology as that of NB. A problem with the approach of Nicholson is the focus on the most informative set of vertices with respect to the query (class). The use of MI does not guarantee the path or region selected will exclude irrelevant edges/vertices to provide for efficient classification. Strong MI associations between nodes

can be due to a 'commonality' and not true relevance. Although selected for the path they may not be a useful contributing factor in respect of the query vertex. The MIM classifier on the other hand, defines the connected region within the MWST with respect to the class vertex. Despite the fact that all associations will correspond directly to the class, the set is still not guaranteed to be optimal. In Chapter 5 we show that in fact they may not be optimal and that the set can in some cases be expanded, potentially improving performance.

[JN99] further extended the basic framework of BN evaluation by introducing a temporal aspect into the graphical representation. The Dynamic BN assigns a series of instances of variable state changes to specific time-slices. Edges connecting vertices across time-slices represent the dynamic behaviour of the domain. The principle idea is to selectively remove edges by use of an arc weight measure based on MI. Essentially at each time-slice edges are deleted from past slices if their weight is less than a pre-selected threshold parameter. Since edges of smaller weights are deleted first, information most relevant to the current belief state is retained. The threshold is generally set by the desired level of accuracy enabling a variation in information amount.

In general arc weights are used to either guide belief update or focus on an area of relevance. Common to all these approaches is the heavy dependency of a pre-defined graph, fully structured and with manageable CPT sizes. For the MIM classifier, arc weights are derived as part of the build process and via the MWST together with the class variable, a specific region of relevance is defined, that is, the class MB. Unlike the reviewed approaches however, the MIM classifier additionally expands the concept of arc weights to actually classify new evidence and has the benefit of avoiding the issues associated with CPT dimensionality and the corresponding reliability of its populated probabilities.

3.6 Summary

In this chapter, we introduced the MIM classifier and described a method for inducing a 'tree' and subsequently, a new technique to classify unseen evidence of the domain using mutual information. By use of the CL algorithm an efficient approach for learning and classifying new observations of the domain has been shown. However, the same algorithm is prone to generating structures that have either missing relevant features or adding irrelevant features. Moreover, it assumes that the

underlying structure is a 'tree' which for some domains may be violated. As a consequence, the MB utilised by the MIM classifier may not be optimal and impact on its predictive accuracy.

In the following chapter we evaluate the MIM classifier on several different data sets and compare 'tree' based methods to networks (GBN). In addition, the performance of the MIM classifier is compared to that of the NB which represents a 'tree' model that assumes extreme CI given the class variable.

Chapter 4

Evaluation – MIM Classifier

In Chapter 3, we introduced the MIM classifier, described by a 'tree' structure representation of a BN derived from an information theory based approach. In this Chapter, we demonstrate the validity and effectiveness of the classifier by carrying out detailed experimental studies on a number of benchmark databases selected from the University of California, Irvine (UCI), repository of machine learning databases. In the section that follows we begin by describing our main objectives of the experiments, with section 4.2 detailing the various data sets used in the evaluation. The experimental methodology and design are discussed in section 4.3 and section 4.4 respectively, whilst the results of comparing the MIM classifier to a NB, GBN and a 'polytree' are described in section 4.5. In section 4.6, we consider the implementations of the results, identifying some drawbacks in section 4.7 and consider learn rates in section 4.8. Section 4.9 provides a summary of the contributions and finally, in section 4.10 we summarise the chapter.

4.1 Objectives

Learning and inference in representations depicted by 'tree' structures have been shown to be computationally feasible [Pea88] in comparison to those of GBNs. However, the restrictions imposed by 'tree' structures results in some loss of representation capabilities, as pointed out in Chapter 3. One of the main objectives of the experiments was to measure the predictive performance to ensure the restriction in topology was not at the expense of predictive accuracy. Comparisons were made specifically against those of the GBN and the MIM classifier.

Another aspect of learning and inference in BNs is the dependency on node ordering. Whether given prior to construction or derived via some algorithm, this aspect impacts heavily on the model complexity. Although comparisons with the GBN would provide some measure of this issue, we further selected a model that has the same representational topology. To evaluate the effect of node ordering dependence and predictive performance, we constructed a 'polytree' using the implementation proposed by Rebane and Pearl [RP87]. Since the MIM classifier is structurally

(skeleton) the same, as both are derived from the CL algorithm, the only difference lies in the edge directionality of the 'polytree'. As such we compared the MIM classifier to the 'polytree' to evaluate the effect of the node ordering (derived by Rebane's algorithm) on its predictive accuracy.

As reviewed in Chapter 2, an option to reduce inference complexity of BNs is to restrict the network topology, for example to tree structures. Taking this approach to its extreme, making strong independence assumptions about the domain features, is the NB classifier. Experimental studies [LS94] showed that this simple classifier could outperform state of the art classifiers, such as C4.5 [Qui93] in many 'real' world domains. Another objective of these experiments is to compare the performance of the MIM classifier with that achieved by NB, but in the case of the MIM classifier, without violating 'real' world assertions.

Since 'tree' based classifiers must estimate on a much smaller set of parameters than the corresponding GBN, they should learn at a faster rate and asymptote faster. In addition, as learning requirements increase exponentially with the number of features, it may be difficult to estimate adequate conditional probabilities for large networks, especially when data is sparse. With this in mind, we are interested in comparing the performance of the 'tree' based classifiers (in particular MIM) against networks (GBN) as a function of the number of cases used for learning the model.

Finally, if we consider the MIM classifier as a middle ground between the GBN and NB classifiers, and given that each representation will have its advantages and disadvantages, our last objective is to categorise the types of data sets most suitable for each approach, especially those where most benefit is obtained by using the MIM approach.

4.2 Description of Data Sets

For our experiments, we used twenty benchmark problems taken from the UCI repository [MA95, BM00]. The data sets selected[12] are summarised in Table 4.1. The choice of data sets listed in Table 4.1 was motivated by two factors. In the first instance, we want to compare the performance of the algorithm for constructing the MIM classifier with other BN approaches. However, we also want to try to categorise the type of problems where the MIM classifier may offer the greatest benefit to the alternative approaches.

---

[12] Further information regarding these data sets can be obtained from the UCI repository [MA95, BM00].

Table 4.1: Data sets used in the experiments

Database Name	Attribute size	Class size	Sample size	Train size	Test size
Vehicle #	18	4	846	Cv5	-
DNA	60	3	3186	2000	1186
Car_Evaluation	6	4	1728	Cv5	-
Flare	10	2	1066	Cv5	-
Chess	36	2	3196	2130	1066
Vote*	16	2	435	Cv5	-
Mushroom *	22	2	8124	5416	2708
Letter	16	26	20000	14000	6000
Hepatitis *#	19	2	155	Cv5	-
Nursery	8	5	12960	8640	4320
CRX *#	15	2	690	Cv5	-
Soybean_Large	35	19	683	Cv5	-
Segment #	19	7	2310	1540	770
Vote1	15	2	435	Cv5	-
Cars	8	3	392	Cv5	-
Austria #	14	2	690	Cv5	-
Heart #	13	2	270	Cv5	-
Promoter	57	2	106	Cv5	-
Glass #	9	7	214	Cv5	-
Ann-Thyroid	21	3	7200	3772	3428

Key: * Indicates data sets with 'missing' attribute value. # Indicates continuous valued attributes.

Cv5 indicates 5 fold Cross Validation.

A diversity of data sets has been selected with varying numbers of features, classes and sample sizes. For example, Promoter represents a high dimensionality, small data set, whilst 'Nursery' and 'Car_Evaluation' characterise a low dimensionality, large data set.

4.3 Experimental Methodology

The MIM classifiers were constructed using the process introduced in Chapter 3 and additionally described in Thomas [THS05a]. The NB classifier was implemented using the method suggested in Gu [GG90]. For the purposes of this investigation, we used PowerConstructor [Che98] to both learn and test the GBN classifier. In the case of the 'polytree' classifier we implemented a version based on the Rebane & Pearl [RP87] model as described by the pseudo code detailed in Figure 3.2, Chapter 3[13].

Once the skeleton tree structure has been constructed, subsequent directionality discovery can be established. The initial process first identifies an internal vertex, that is, a vertex that has more than

[13] This procedure only constructs an undirected tree structure.

one neighbour, and then applies a dependency test. Essentially, if, say, Z_3 is an internal vertex and has at least two neighbours Z_1 and Z_2, then the objective is to try to establish whether Z_1 and Z_2 are marginally independent. If they are, then we assign directions from Z_1 to Z_3 and from Z_2 to Z_3, as shown in Figure 4.1.

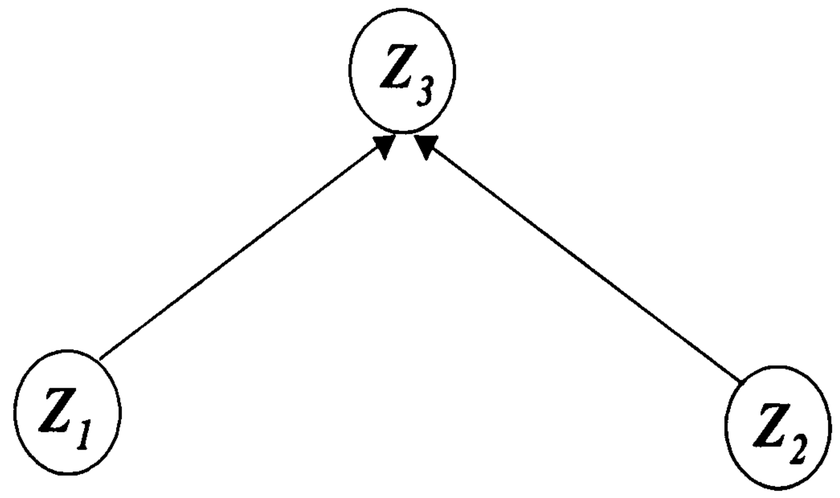


Figure 4.1. Directionality: Case when found marginally independent

If, however, Z_1 and Z_2 are not marginally independent then we assign the opposite directions: from Z_3 to Z_1 and Z_2, Figure 4.2.

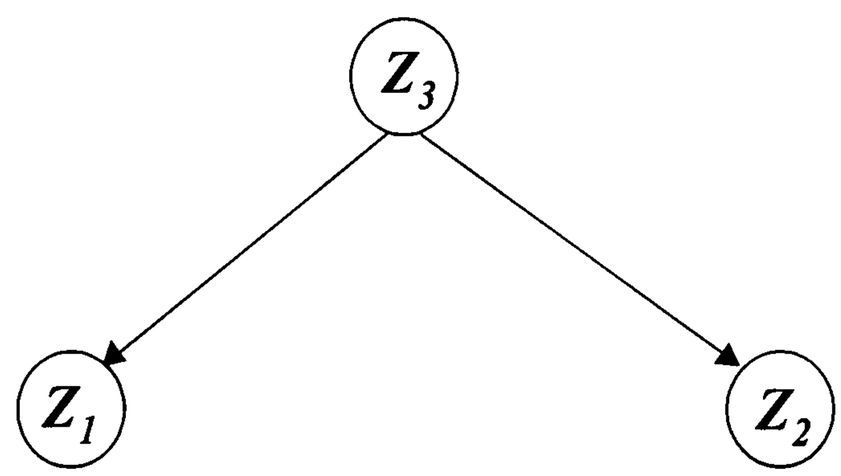


Figure 4.2. Directionality: Case when not found marginally independent

This is effectively repeated for every combination of pairs in respect of each internal vertex, and directions assigned as appropriate. Some assignments, however, may be inconsistent with the arrows associated with an edge going in both directions. This occurrence is not uncommon as [Pea88] noted that only one type of dependency can be uniquely identified, and therefore only "partial" recovery may be possible.

For the independency test, we used the mutual information measure. This allows us to exploit the fact that, if Z_1 and Z_2 are independent then the measure is asymptotically distributed as non-central χ^2 with $(r-1)(c-1)$ degrees of freedom, where r and c are the number of values of variables Z_1 and Z_2 respectively [Kul68]. Refer to Section 4.10 for more detail on this aspect. The implementation used in our experiments is described by the pseudo code detailed in Figure 4.3.

FOR i = 1 to N **DO**
  BEGIN
    IF Z_i has more than one neighbour
    THEN put Z_i in Multiple_Set
END
FOR each Z_i in Multiple_Set
BEGIN
 FOR any pair of neighbours Z_j, Z_k of Z_i **DO**
 BEGIN
 IF (Z_j and Z_k are independent)
 THEN $2I$ is distributed as χ^2 with (r-1)(c-1) degrees of freedom
    (Where I is the mutual information measure)
    $Z_j \rightarrow Z_i, Z_k \rightarrow Z_i$
 ELSE
    $Z_i \rightarrow Z_j, Z_i \rightarrow Z_k$
 END
END

Figure 4.3. 'Polytree' Construction Algorithm for Directionality Discovery.

In the event that edges remained undirected after applying this algorithm we applied two 'rules' in order to allow the conditional probability tables to be calculated. The first rule was taken from Verma & Pearl [VP92], whilst the second, a heuristic derived from the partially completed 'polytrees'. Directionality was essentially assigned to the undirected edges in conjunction with those edges that had already been successfully recovered.

The two rules applied were as follows:

Rule 1: If $a \rightarrow b$ and a is not adjacent to c then direct $b \rightarrow c$

Rule 2: If $a \rightarrow b \rightarrow c$ and $d \rightarrow b$ then direct $d \rightarrow e$ that is as shown in Figure 4.4.

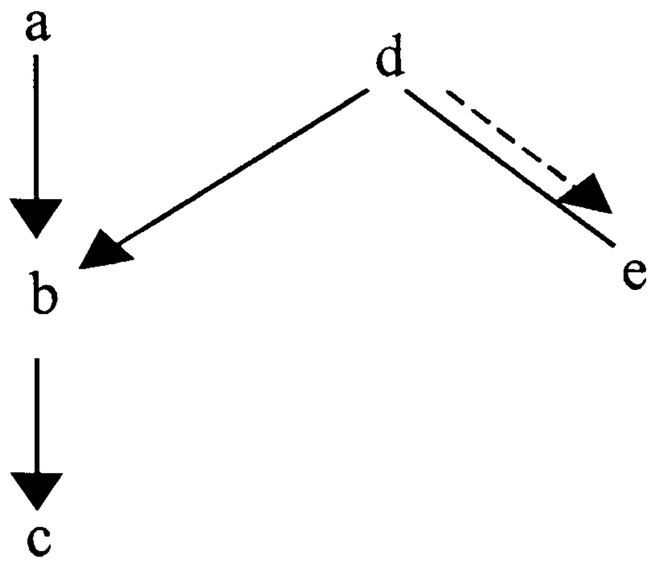


Figure 4.4. Rule 2 - Directionality.

As we are using the Chow and Liu algorithm, we have restricted our investigations to discrete data sets. All continuous features were therefore made discrete prior to application. This was achieved by use of the utility provided by MLC++ [KJ+94] on the default setting.

The simplest way to handle missing data is to merely drop cases for which the values of all variables were not observed. Whilst this approach may be fairly straightforward to implement for small amounts of missing data, the loss in sample size can be considerable in domains where the number of variables is large. According to Friedman [Fri98] the learning procedure should use 'missing' values in order to maximise the probability of the 'actual observations'. As such, all data sets used in our experimental studies that contained 'missing' feature values were dealt with by treating them as an additional element of that feature. Although this approach has the disadvantage of adding an additional parameter to each attribute and thus more to estimate, we consider the increase advantageous in meeting our experimental objectives, particularly since one of our aims is concerned with the complexity associated with 'multi' state attributes and the actual number of states that defines an attribute and its corresponding CPT.

4.4 Experimental Design

Residual evaluations do not provide an indication of how well the classifier will perform when required to make a prediction for data it has not already observed. This problem can be avoided by not using the entire data set when constructing each classifier structure. For the larger data sets[14], we randomly partitioned each data set into two parts. The first part comprised 2/3 of the entire sample and was used for training/construction of the four classifier structures. The second part, the remaining 1/3, was subsequently used for evaluating the predictive accuracy of each of the classifiers constructed (this represents the simplest kind of cross validation or hold-out technique). Unfortunately, with the hold-out technique there is a possibility of obtaining an over-represented class in one partition and an under-represented one in another. To overcome this, we applied a stratified distribution for each of the large data sets, which was maintained in respect of the two partitions. For those data sets that were small, the hold-out technique was not applied but a 5-fold cross validation as indicated in Table 4.1. Our choice of folds (k = 5) is based on the recommendations of Kohavi [Koh95]. As for the hold-out technique, stratification was also employed for the small data sets, with the folds containing approximately the same proportions of class labels as the original data set. According to Kohavi [Koh95] stratification is generally a better scheme, both in terms of bias and variance, when compared to regular cross-validation.

For each of the four classifiers, the structure was learned/constructed using the training data set and the classifier accuracy determined on the test data set. The classification accuracy was determined as a percentage of the test cases that identified the correct class.

This process was repeated over a series of runs in order to gain a sample average together with the standard deviation for the predictive accuracy using the test partition. Most researchers appear to use about 25-30 trial runs for each data set. However, in our preliminary studies we found that a difference that was significant over 25 trials was often found not to be significant for trials under 10, demonstrating that small sample size can miss small differences. Since the number of trials seems fairly arbitrary, we selected 25 for our experimental work in order to align with our preliminary findings. Within the Machine Learning (ML) community, the paired t-test has been used to

---

[14] We considered 2000 items as a lower limit for application of the hold-out technique based on the recommendations of [MST94]

determine the significance of any differences between algorithms. However, this technique only compares two groups and we wanted to test the statistical significance of the differences among all the classifiers studied. As such the statistical significance of the differences in classification accuracy was measured using an Analysis of Variance (one-way ANOVA). To further determine all pair-wise differences, that is, the magnitude and direction between each pair of methods being compared, we followed ANOVA by Post Hoc Tukey comparisons with an overall confidence level 95%.

Prior to applying ANOVA on each data set we first established the validity of the assumptions, although in the case of the normality assumption ANOVA is quite robust to any violations of normal distribution.

To measure the rate of improvement of the various classifiers, the accuracy was measured for different quantities of the learning cases. More specifically experiments were carried out using 20%, 40%, 60%, 80% and 100% of the cases in the learn partition for constructing the classifiers. Once induced the corresponding predictive accuracies were then measured using the entire test partition, repeating for each of the training sub-partitions.

4.5 Experimental Results

4.5.1 Computational Complexity

Where the GBN model was provided with the prior node ordering the algorithm carried out $O(N^2)$ CI tests to learn an N-node network. This same time complexity applies to the Chow and Liu algorithm (calculating the $N(N-1)/2$ mutual information values) which is utilised for learning both our 'polytree' implementation and the MIM classifier. In contrast the NB classifier's time complexity is $O(N)$, being proportional to the time required to read all of the training data. When the GBN model was not provided with prior node ordering, the time complexity increased to $O(N^4)$. Here the algorithm had the additional overhead of examining N^2 node pairs in order to determine the network edge orientations. In this thesis, both options were explored and the results in Table 4.2 reflect the best predictive values achieved for the GBN.

Of the four algorithms the GBN was noticeably slower to learn the network structure than the tree based algorithms, even when prior node ordering was provided, with relative times corresponding linearly with training sample sizes.

In respect of classification, testing was linear in the representation size of the structures, that is, in the number of features defining the class Markov blanket (MB).

4.5.2 Trees Opposed to Networks

As discussed in section 4.1, there were several objectives in experimentally comparing 'tree' type classifiers to network type classifiers.

First we wanted to gauge the degree to which 'tree' representation affects the inference complexity of the resultant structure by measuring the effect of node ordering and the restriction to 'tree' topology. Moreover, we wanted to compare the classification accuracy of the MIM classifiers and the GBN classifiers.

This was considered important for two reasons, a) we wanted to ensure that adopting a 'tree' structure did not result in a high loss of performance and b) the use of MI 'weighted' edges was not limited by the data set size or dimensionality.

In the following sections, we first describe the results of comparing 'trees' in general with the GBN, followed by the experimental results specifically involving the MIM classifier.

- **Comparison of 'tree' based classifiers with 'network' based classifiers**

In general the Markov blanket for the 'tree' classifiers is smaller than those of the networks (excluding NB whose MB uses all features), this maybe due to fact that most of the features were either irrelevant or redundant, as there was no impact on the overall accuracy. Examples of this are 'DNA' 15(60), 'Promoter' 4(57) and 'Cars' 1(8) where 'trees' actually outperformed the GBN. Although both the MIM classifier and 'polytree' performed better than the GBN for these particular data sets, differences that were found to be statistically significant were only observed for the data set 'DNA' (p-value <0.05 for both models). Thus, parameters for networks will be greater for data sets that have many features. Networks for 'Promoter' and 'Chess' for example, have a greater complexity that impacts on their inference. However, some data sets resulted in learning a 'tree' whose MB comprised of nearly all the domain features. [i.e. Car_Evaluation 5(6), Letter 11(16), Nursery 8(8), Soybean_Large 33(35) and Glass 7(9)].

Table 4.2: Average Predictive Accuracy

DB Name	MIM	NB	GBN	Polytree	Default (overall)
Vehicle	55.66 ± 1.51	58.28 ± 1.79	**61.00 ± 2.02**	56.85 ± 1.77	25.8
DNA	***95.58 ± 0.42***	94.97 ± 0.29	89.90 ± 5.61	**95.62 ± 0.35**	51.9
Car_Evaluation	86.11 ± 0.74	**86.58 ± 1.78**	86.11 ± 1.46	78.81 ± 8.25	70.0
Flare	**82.93 ± 1.26**	80.99 ± 1.28	82.27 ± 1.45	82.66 ± 0.79	79.2
Chess	**96.27 ± 3.56**	87.34 ± 1.02	94.65 ± 0.69	90.14 ± 1.86	52.0
Vote	**95.40 ± 2.41**	89.89 ± 5.29	95.17 ± 1.89	94.94 ± 3.69	54.8
Mushroom	98.56 ± 1.06	95.79 ± 0.39	**99.30 ± 0.16**	98.56 ± 1.06	51.8
Letter	**80.26 ± 0.37**	74.96 ± 1.10	75.02 ± 0.61	79.86 ± 0.80	4.07
Hepatitis	**84.00 ± 7.22**	81.20 ± 3.70	83.22 ± 1.52	82.47 ± 1.44	79.4
Nursery	**95.78 ± 0.30**	94.76 ± 0.45	89.72 ± 0.46	94.85 ± 0.27	33.3
CRX	85.00 ± 0.52	**86.60 ± 0.71**	82.40 ± 1.30	84.70 ± 0.30	55.5
Soybean_Large	**91.29 ± 0.10**	90.78 ± 0.72	89.28 ± 0.42	88.51 ± 0.11	13.5
Segment	**94.49 ± 0.64**	91.95 ± 1.10	94.10 ± 0.83	91.76 ± 0.10	4.80
Vote1	**88.51 ± 1.90**	87.60 ± 2.10	87.10 ± 3.00	88.28 ± 2.70	61.4
Cars	**99.23 ± 0.70**	98.98 ± 0.47	98.97 ± 0.81	98.97 ± 0.81	62.5
Austria	85.07 ± 0.91	**86.38 ± 1.10**	84.35 ± 1.80	84.93 ± 0.66	55.5
Heart	***85.83 ± 2.10***	85.00 ± 1.13	82.78 ± 3.60	***85.83 ± 1.41***	55.6
Promoter	**87.97 ± 1.31**	82.00 ± 2.02	84.25 ± 1.61	87.04 ± 1.97	50.0
Glass	69.37 ± 2.08	68.31 ± 1.98	**70.42 ± 1.40**	69.95 ± 1.60	35.5
Ann-Thyroid	97.17 ± 0.08	**99.11 ± 0.31**	96.28 ± 0.10	97.64 ± 0.57	92.6

Key: MIM – Mutual Information Measure Classifier, **NB** – Naive Bayes Classifier, **GBN** – General Bayesian Network Classifier, **Polytree** – Pearl's SCN Model.

Values in **bold** type indicate the highest model performance achieved by the classifier in respect of each database. ***Bold italic*** values highlight performance levels that are close to the highest level achieved.

The average predictive accuracies, taken over 25 runs, of the classifiers generated for each of the four methods are shown in Table 4.2. Each entry describes the average accuracy along with the sample standard deviation illustrating variations in the predictive accuracy from sample to sample. The default value in Table 4.2 represents the majority classifiers' predictive accuracy. This is equivalent to assigning all new evidence to the class-state that has the greatest frequency of observed samples corresponding to the particular database under investigation. 'Overall' means that the value is measured in respect to the 'entire' dataset and not for the test partition.

Figure 4.5 and Figure 4.6 show the error rates of NB compared to GBN and 'polytree' compared to GBN on the various data sets. In these plots, a point above the diagonal line (representing equality) indicates that the 'network' learned for that particular data set has a higher error rate, and thus was worse, than the corresponding 'tree' based classifier induced from that data set. In Figure 4.5, the NB

has a cluster below the diagonal line otherwise both methods have similar error rates. In the case of Figure 4.6, similar error rates are observed for both methods with respect to the 20 data sets studied.

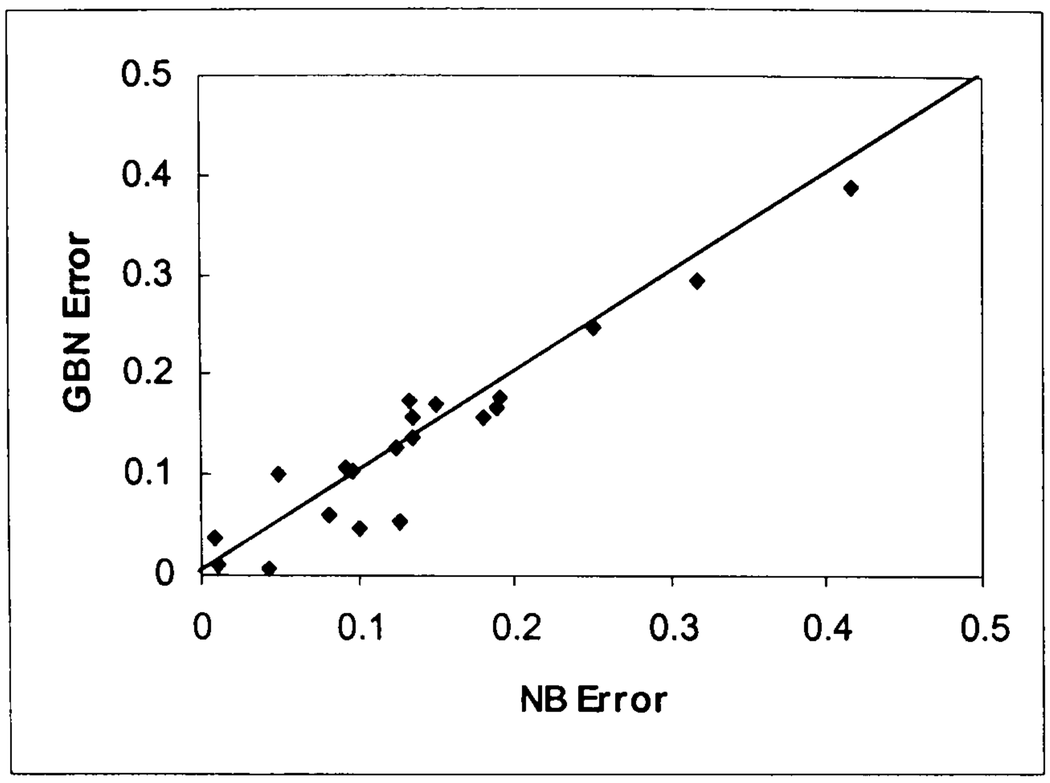


Figure 4.5. Scatter Plots Comparing Error Rates of GBN with NB

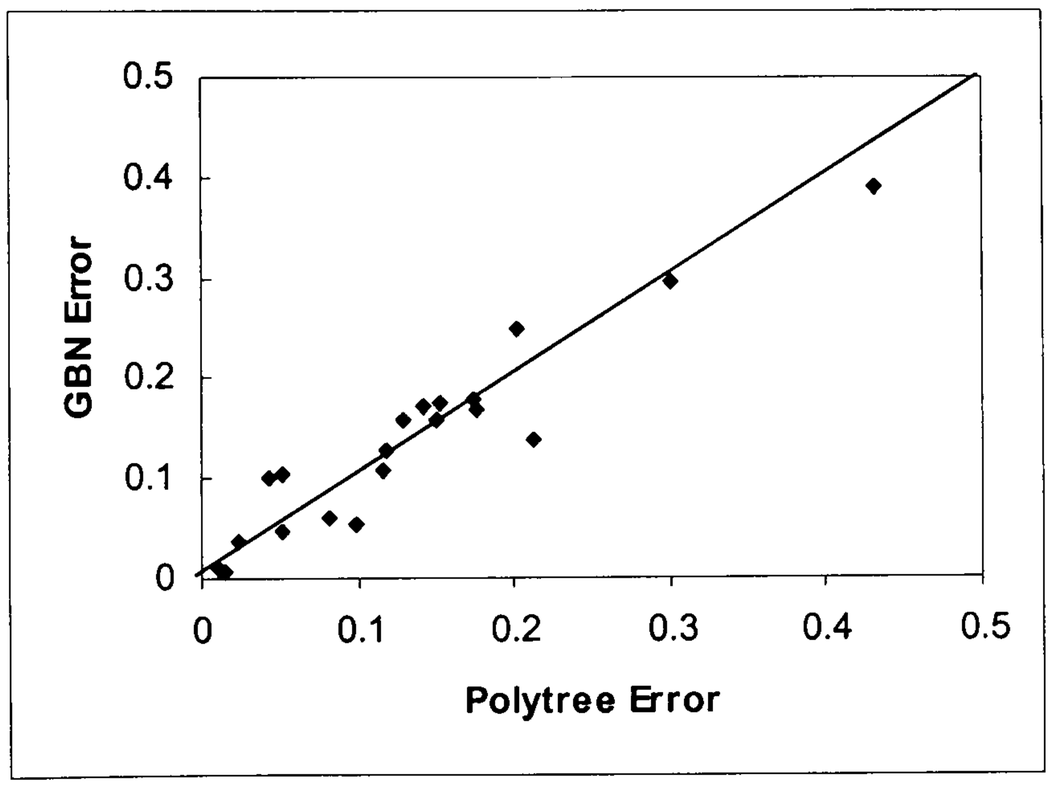


Figure 4.6. Scatter Plots Comparing Error Rates of GBN with Polytree

Figure 4.7 and Figure 4.8 show the difference in accuracies between NB and GBN together with the 'polytree' and GBN respectively for each of the data sets. Each bar shows the average difference in predictive accuracy. A positive value for an algorithm indicates that the NB or 'polytree' performed better on the data set under consideration. The error bars represent the Post Hoc Tukey comparisons with overall 95% confidence for the relative differences. If the confidence interval for a given data set crosses the 'zero' line then the two methods are not statistically different. On the other hand, if

the confidence interval is wholly above, or below, the 'zero' line, then the methods are in fact statistically different.

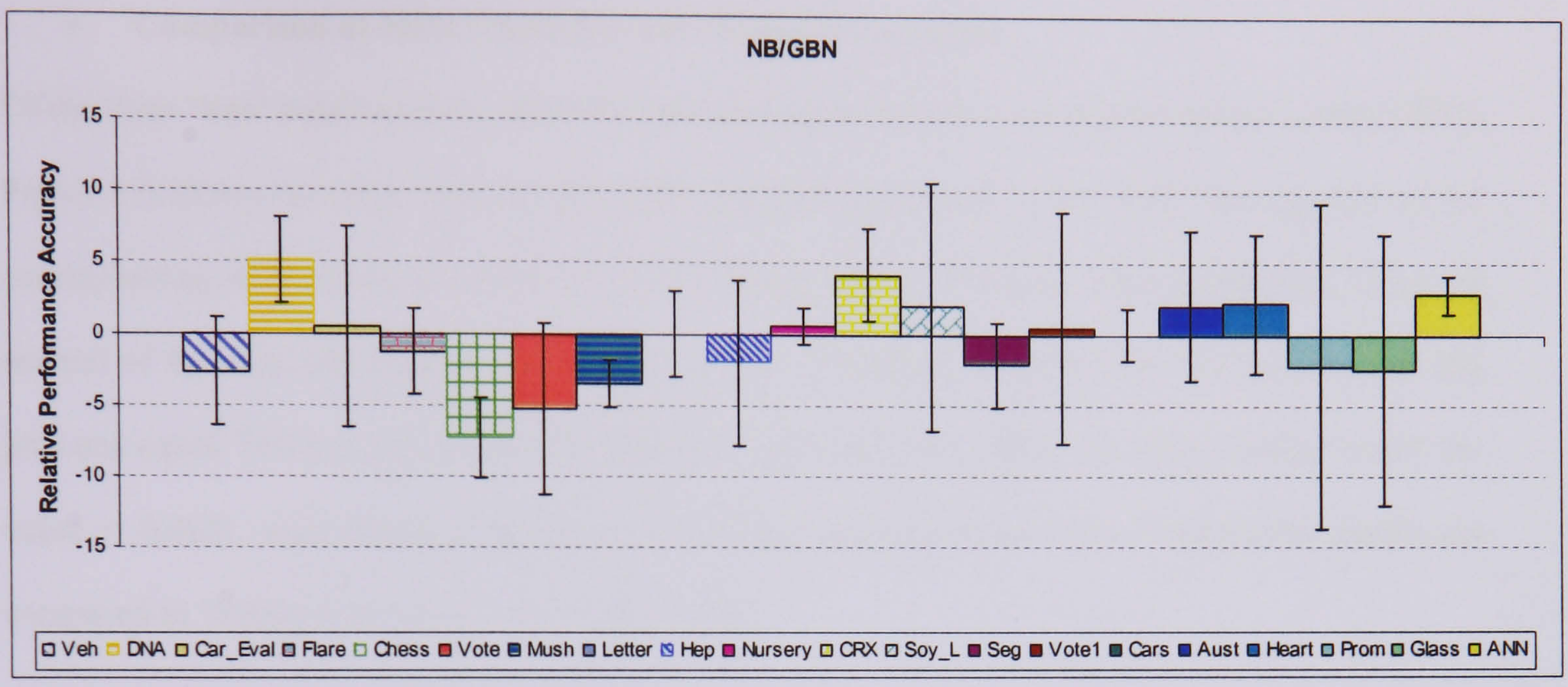


Figure 4.7. Predictive Accuracy relative to NB Classifier.

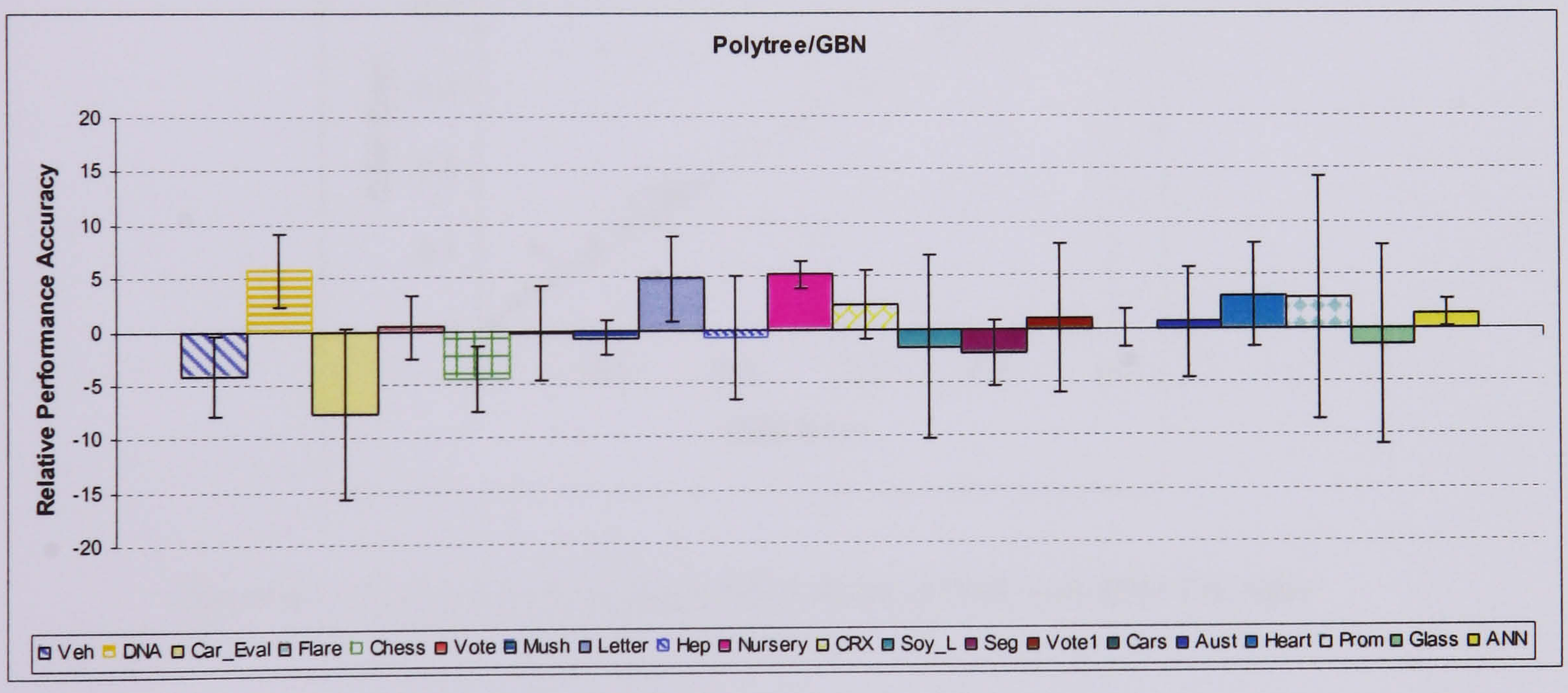


Figure 4.8. Predictive Accuracy relative to Polytree Classifier.

As can be seen from Figure 4.7, NB is better for nine data sets compared to that achieved by the GBN classifier. The NB has three data sets, 'ANN' (p-value = 0.001), 'CRX' (p-value = 0.007) and 'DNA' (p-value = 0.001) for which the differences are statistically significant with GBN having two, 'Chess' (p-value <0.05) and 'Mushroom' (p-value = 0.002). From Figure 4.8, the 'polytree' is better than GBN for ten data sets. In the case of the 'polytree' there are four data sets, 'ANN' (p-value =

0.032), 'DNA' (p-value <0.05), 'Letter' (p-value = 0.001), and 'Nursery' (p-value <0.05) for which the differences are statistically significant compared to those found in respect of the GBN, namely 'Chess' (p-value = 0.022) and 'Vehicle' (p-value = 0.031).

- **Comparison of MIM classifiers with Network classifiers**

Of the three 'tree' based models, the MIM classifier performed the best against the networks (GBN). Figure 4.9 shows the error rates for the MIM classifier compared to the GBN and Figure 4.10 the corresponding differences in accuracies. From Figure 4.9 the GBN has the highest error rates for several of the data sets studied. The MIM classifier was better than the GBN for sixteen data sets plus one equal, Table 4.2. For the MIM classifier, three data sets, 'DNA' (p-value <0.05), 'Letter' (p-value = 0.002), and 'Nursery' (p-value <0.05) had differences that were statistically significant compared to 'Vehicle' (p-value = 0.003) for GBN.

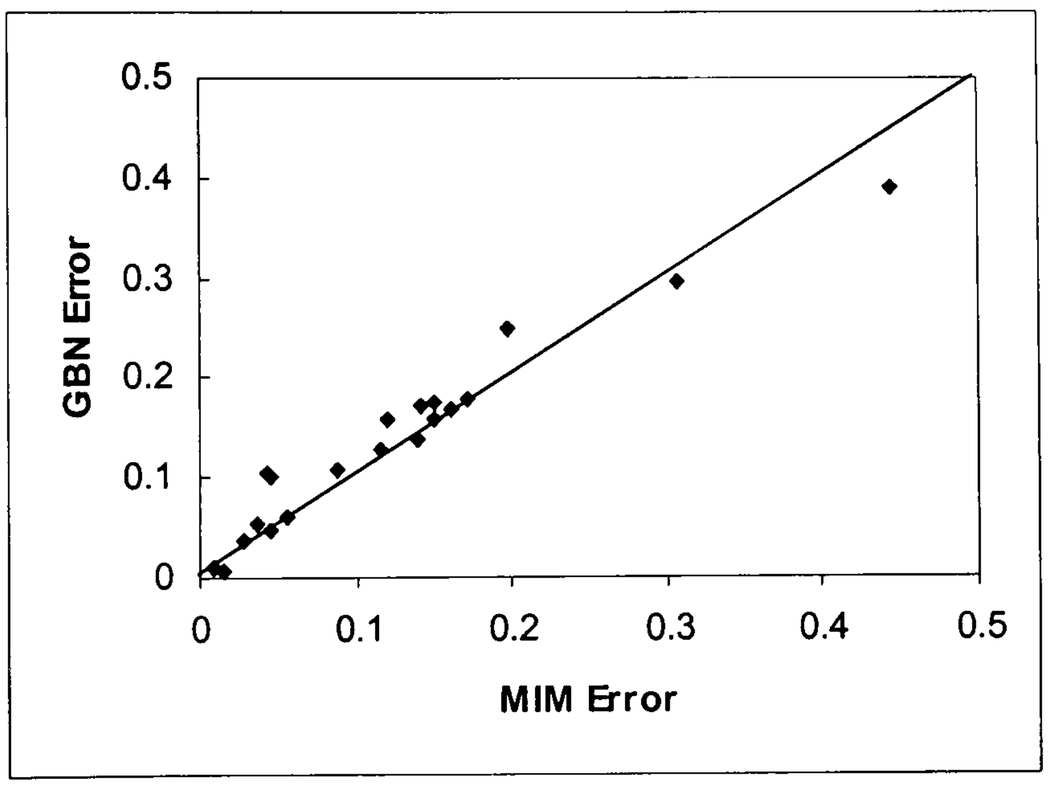


Figure 4.9. Scatter Plots Comparing Error Rates of GBN with MIM Classifier

The GBN performed consistently better for the data sets 'Mushroom', 'Glass' and 'Vehicle' in comparison to all the 'tree' based models. However, when compared to the MIM classifier, only 'Vehicle' (p-value = 0.003) had differences that were found statistically significant. When GBN was compared to NB, it was the data sets 'Chess' (p-value <0.05) and 'Mushroom' (p-value = 0.002) that had differences that were statistically significant, with similar comparisons to the 'polytree' resulting in significance found for the 'Chess' (p-value = 0.022) and 'Vehicle' (p-value = 0.031) data sets.

Langley [LS94] reported NB had a poor performance for the databases 'Mushroom' and 'Chess' as they were highly correlated data sets. In comparisons of the MIM classifier with the GBN this was also observed, however, the data set 'Chess' was actually predicted better by the MIM classifier than by the GBN model. In the case of NB and 'polytree' comparisons with the GBN, the data set 'Chess' was predicted better by GBN as Langley [LS94] observed.

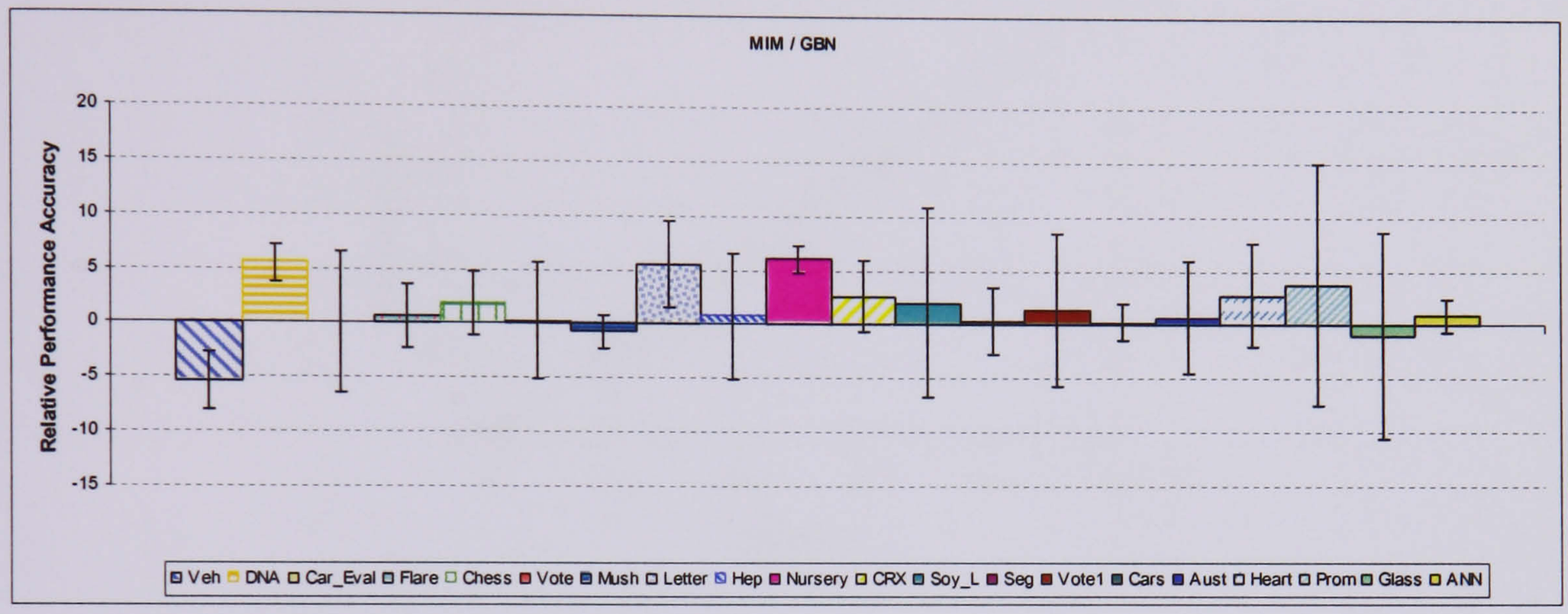


Figure 4.10. Predictive Accuracy relative to MIM Classifier.

Friedman [FGG97] observed that unrestricted networks (GBN) perform worse when the number of relevant features influencing the classification (MB) is small. For example in 'Soybean_Large' the MB in GBN is 5(35) whilst the MIM classifier uses 33(35). From Table 4.2, the MIM classifier topology did not appear to have any major effects on performance of the MIM classifier compared to the unrestricted GBN. The inference approach based on MI 'weighted' edges, as can be seen in Table 4.2, demonstrate that it is an efficient technique in respect of the scope of data sets studied. We observed that when forced to represent a NB type structure, the GBN had a reduction in predictive performance as observed in data sets 'Nursery' and 'Car_Evaluation'. This however was not the case for the MIM classifier, where it predicted an average 6% better than the GBN on the 'Nursery' data set.

4.5.3 Dependence 'tree' models Opposed to Independence models (NB)

Although as shown in the previous section, 'tree' structures have a much lower complexity and efficient inference than networks, whilst maintaining and sometimes improving the accuracy, it is

important to compare dependency tree models (in general trees: 'polytree', MIM) to Naive Bayes classifiers due to their simplicity and polynomial-time inference complexity. In the following sections we first compare NB to 'polytree' induced networks followed specifically by a comparison to the MIM classifiers.

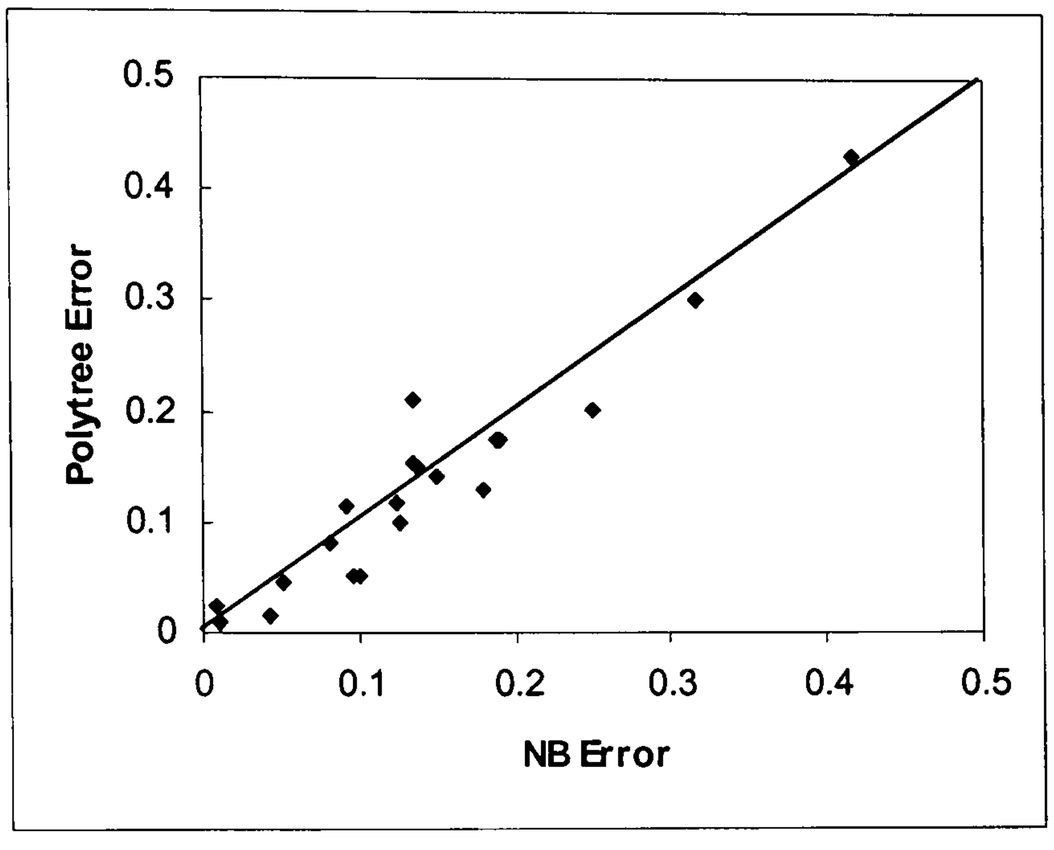


Figure 4.11. Scatter Plots Comparing Error Rates of Polytree with NB Classifier

- **Comparison of 'Polytree' (SCN) classifiers with Naive Bayes classifiers**

The difference in accuracies for the NB and 'polytree' classifiers are shown in Table 4.2. Figure 4.11 shows the error rates for NB compared to the 'polytree'. In Figure 4.11 the NB has the highest error rate for most of the data sets studied. The actual differences between accuracies of the various data sets along with 95% confidence intervals are shown in Figure 4.12 comparing NB with the 'polytree'. The 'polytree' identified twelve data sets better than NB for which three, 'Letter' (p-value = 0.003), 'Mushroom' (p-value = 0.021), and 'Nursery' (p-value <0.05) had differences that were found statistically significant compared to NB which had two, namely 'ANN' (p-value = 0.022) and 'Car_Evaluation' (p-value = 0.014).

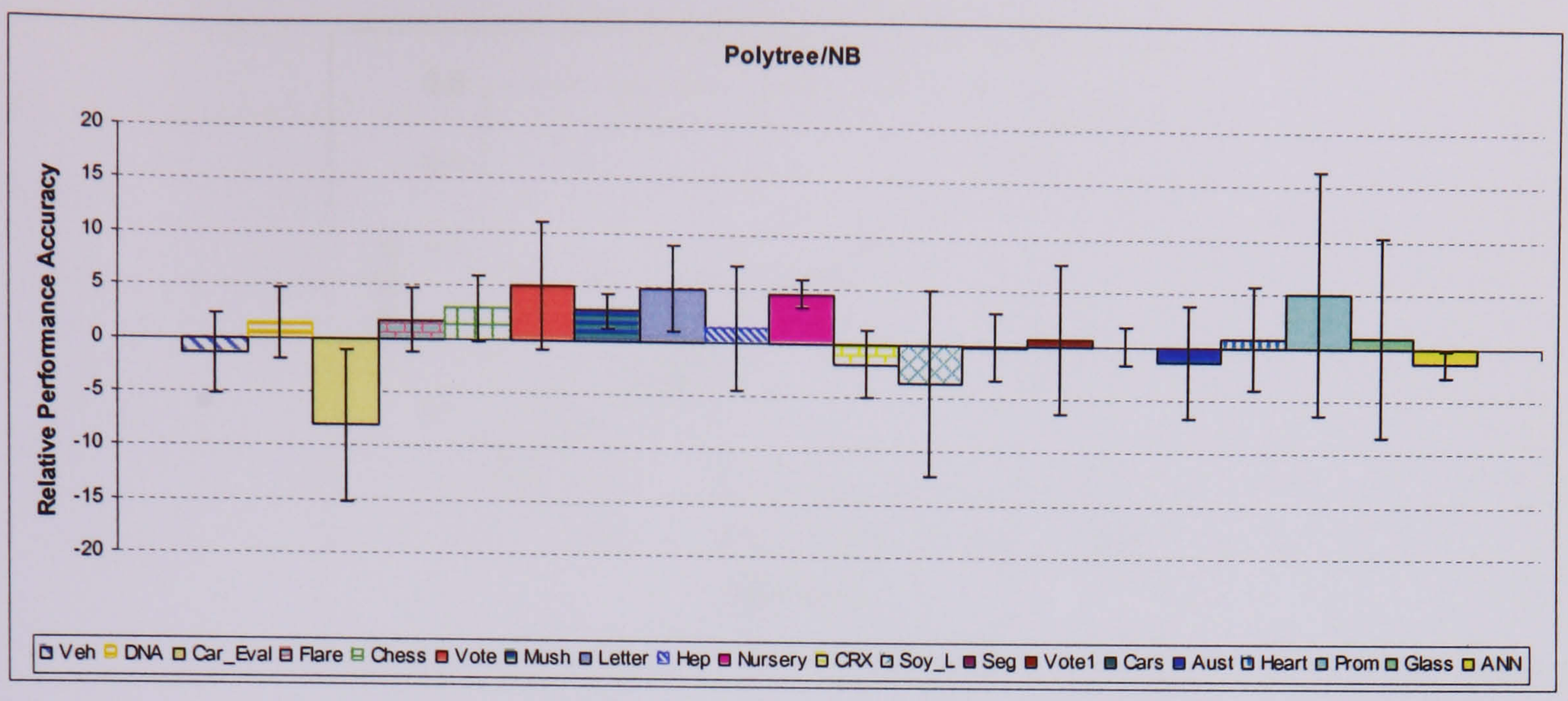


Figure 4.12. Predictive Accuracy relative to 'Polytree' Classifier.

- **Comparison of MIM classifiers with Naive Bayes classifiers**

The results of comparison between the MIM classifier and NB were even better than those of the 'polytree' as shown in Table 4.2. Figure 4.13 compares the error rates between the MIM classifier and NB on the 20 data sets used in the experiments. As can be seen from the results, NB has the highest error rates for the majority of the data sets studied with most above the diagonal line. The difference in accuracies on the various data sets are shown in Figure 4.14. The MIM classifier performance was better on fifteen data sets compared to NB with four having differences that were statistically significant, namely 'Chess' (p-value <0.05), 'Mushroom' (p-value = 0.021), 'Letter' (p-value = 0.002), and 'Nursery' (p-value <0.05), in comparison to NB with only 'ANN' (p-value = 0.006). Although the NB performed well on the data sets, 'ANN', 'Austria', 'Car_Evaluation', 'CRX', and 'Vehicle', the MIM classifier achieved a comparable performance with its MB in general only 50% of the features used by NB. The exception being 'Car_Evaluation', which despite NB performing better was not statistically significant (p-value = 0.986) with NB achieving on average a 0.5% better predictive level than the MIM classifier. In this particular data set, the MB for the MIM classifier was almost the same as that of NB. In fact except for 'Vehicle', NB performed overall better on the remaining four data sets than the other three BN approaches.

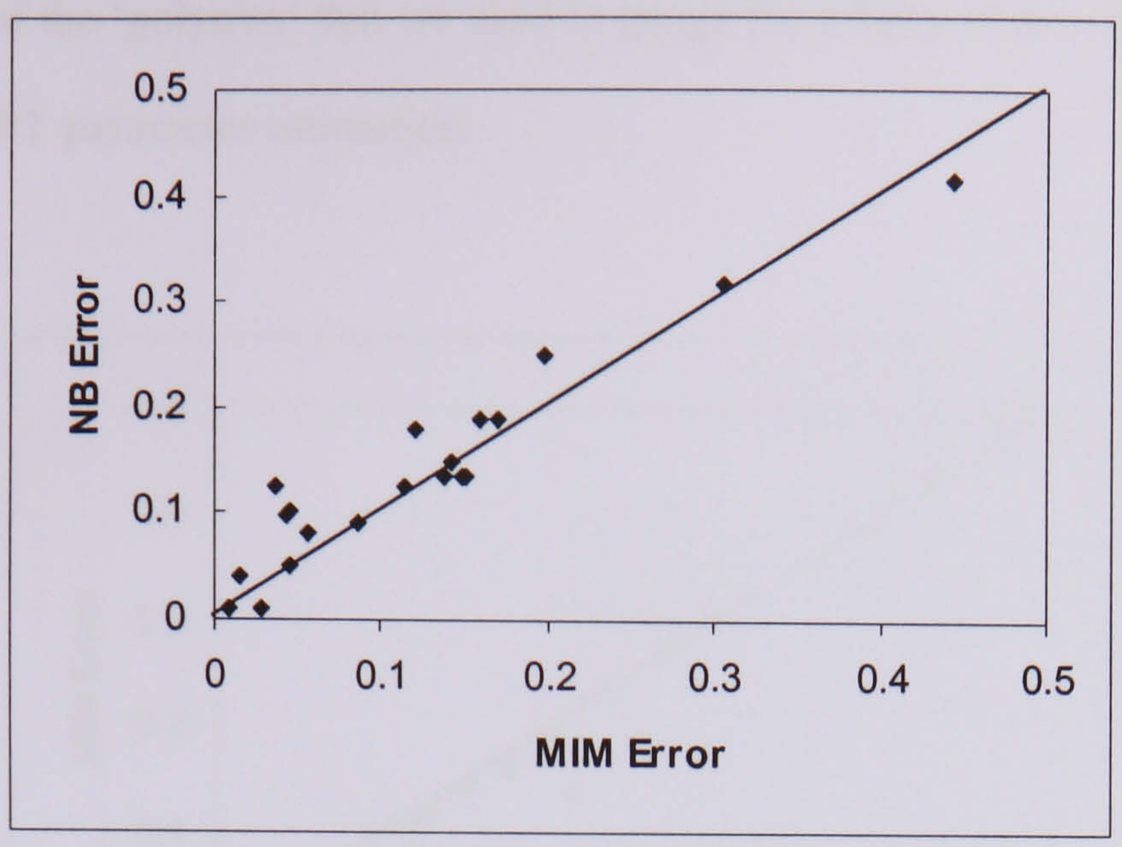


Figure 4.13. Scatter Plots Comparing Error Rates of NB with MIM Classifier

For the data sets 'Nursery' and 'Car_Evaluation', there was an indication that the features of these two databases are almost independent of each other. It was thus no surprise to observe the NB performing well on these two data sets. In the case of the MIM classifier, the predictive performance was comparable to that achieved by the NB.

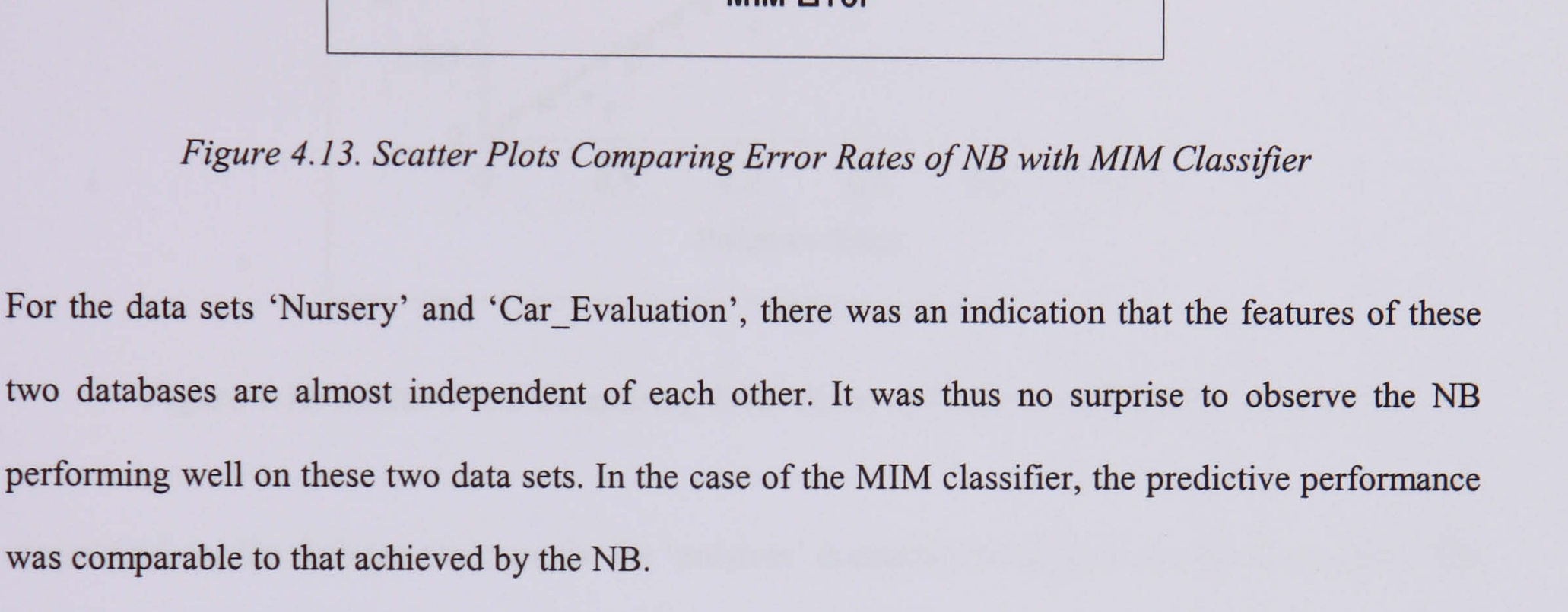


Figure 4.14. Predictive Accuracy relative to MIM Classifier.

4.5.4 MIM classifier Opposed to 'Polytree' classifier (SCN)

The objective specified in section 4.1 justified the rational for this particular experimental study. Our aim was to measure the effect of node ordering on the predictive performance of the resulting networks, learned from the data sets selected from the UCI repository. In section 4.3, we described

our implementation of the 'polytree' that we used to gauge the effects of the choice of node ordering and corresponding CPT parameter estimation.

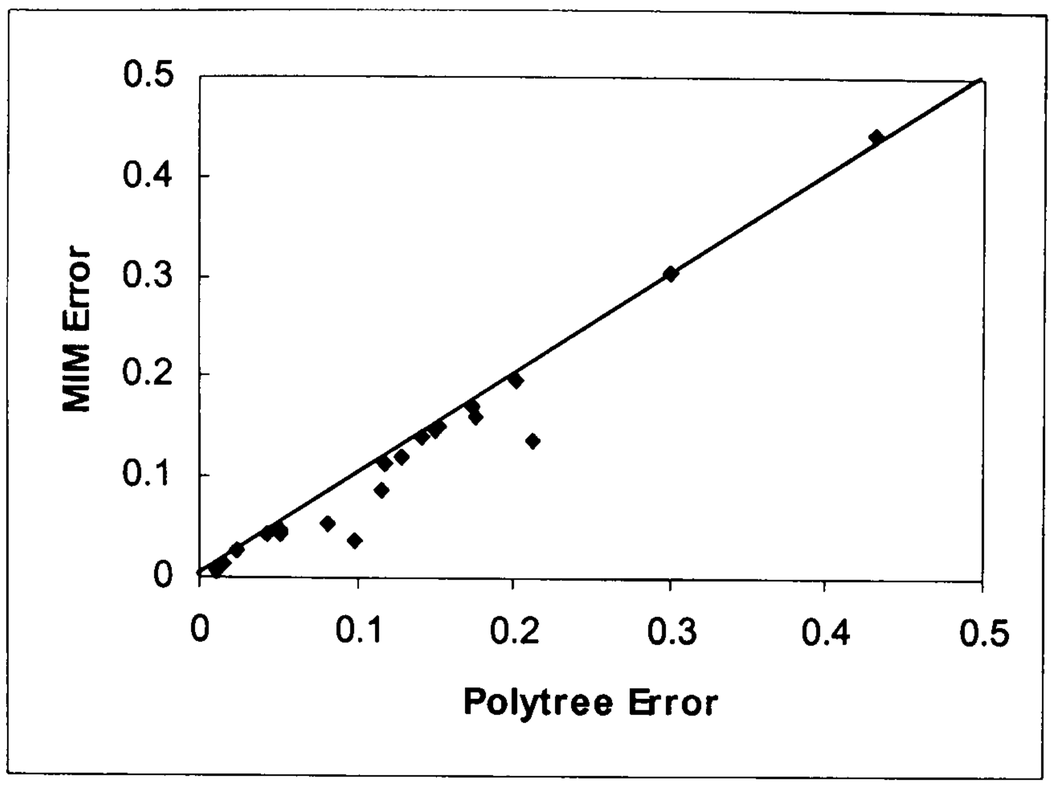


Figure 4.15. Scatter Plots Comparing Error Rates of Polytree with MIM Classifier

As pointed out the skeleton structure for the 'polytree' is exactly the same as the MIM classifier. The difference in the models lies in the selection of directionality for the edges. In the 'polytree' implementation, we use the algorithm of Rebane [RP87] along with some heuristics to complete the model in the absence of domain experts. In contrast, the MIM classifier takes a different approach focused on the class variable to guide the orientation of edge directionality. As such the MB for each will vary depending on the final edge orientation determined, which in turn will reflect the size of the CPT in the case of the 'polytree'. Figure 4.15 shows the error rates for the MIM classifier compared to the 'polytree'. As most values are below the diagonal line, it is clear that the 'polytree' has the highest error rates for the majority of the data sets studied. Figure 4.16 further displays the differences in accuracies of the 'polytree' compared to the MIM classifier. The MIM classifier achieved fourteen data sets better than the 'polytree'. Three of these had differences that were statistically significant, namely 'Chess' (p-value = 0.001), 'Car_Evaluation' (p-value = 0.022), and 'Nursery' (p-value = 0.037), whereas there were none for the 'polytree'. From those predicted by the MIM classifier that were better than the 'polytree' two data sets were very close in predictive levels to the 'polytree' [Flare, Vote1] whilst two of the 'polytree' predictive wins were close to that achieved by the MIM classifier [DNA, Glass].

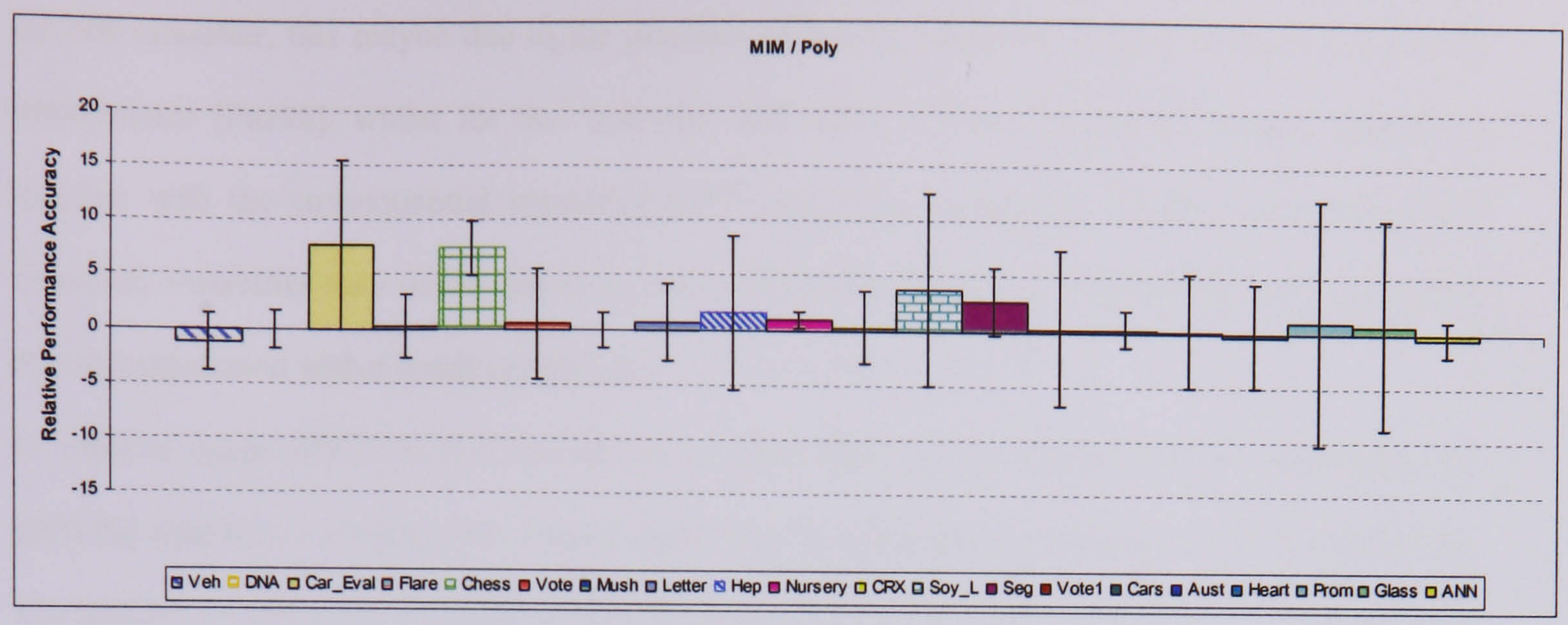


Figure 4.16. Predictive Accuracy relative to MIM Classifier.

The MB for the 'polytree' and thus CPT parameters estimated will be governed by the node ordering. In contrast, the MIM classifier will only require MI 'weights' to be calculated for the MB, which may be the same as that of the 'polytree' for some data sets. In general the predictive accuracies, Table 4.2, show that both the MIM classifier and 'polytree' are comparable throughout, with six data sets either equal or very close in predictive levels [DNA, Flare, Glass, Heart, Mushroom, Vote1]. It would appear however, that the topology has had an effect on the predictive performance of the 'polytree', observed in data sets 'Chess', 'Car_Evaluation' and 'Soybean_Large', however only for 'Chess' (p-value = 0.001) and 'Car_Evaluation' (p-value = 0.002) were the differences found to be statistically significant.

4.6 Discussion

The results shown in Table 4.2 indicate there are some high values of sample standard deviation. This is especially evident for the data sets 'Hepatitis', 'DNA', 'Vote' and 'Car_Evaluation'. In respect of these particular data sets, the values correspond to the MIM classifier, GBN, NB and 'poltytree' respectively. As standard deviation represents a measure of spread, the interpretation of their magnitude translates into structural variations that occurred during the 25 experimental trials. For the dependency models this relates to an inability to establish a 'stable' class MB, whilst for the NB classifier, as its structure is trivial, inconsistency in the estimation of its domain probabilities.

Class MB instability occurs due to the various anomalies characterised by the domain data set. For the NB classifier, this maybe due to the presence of highly correlated features, known to affect its performance [Paz96], whilst for the 'polytree' and GBN, a high dimensional, sparse sample set together with the consequential unreliable CPT probability estimations. In the case of the MIM classifier, instability may occur due to a poorly characterised data set (further discussed in section 4.7) in conjunction with a small sample set.

In order to apply ANOVA, the assumption of equal variance are required to be satisfied and for differing standard deviations this maybe dealt with by carrying out some form of transformation. The penalty however, for methods with high standard deviation values it to affect a reduction in the 'power' of analysis to find a specified effect. Power in this context is defined as the probability of finding a difference (if one exists), and is influenced by the variability in the data.

In the case of the data sets above, none of the methods with high standard deviations were actually found to have differences that were statistically significant. The implication is that for the methods with high standard deviations, the 'differences' would need to be large in order to be measurable (i.e. found to be statistically significant) and so for these particular methods, small 'differences' could potentially go undetected.

In the sections that follow we discusses some implications of the experimental results. In particular, the various properties and assumptions of the different induction methods studied, and identifying characteristics of problems for which each method would be best suited.

4.6.1 Trees Opposed to Networks (GBN)

By examining Figures 4.5 – 4.16 more closely, it is possible to identify the characteristics of data sets on which 'tree' based models may be most beneficial (especially the MIM classifier) over the network approaches.

In general where the database had strong feature independence characteristics the 'tree' based classifiers performed well. However, when the structure was forced to reduce to that of NB type structure, the GBN had a slight degradation in performance. For domain modelling where the class – attribute correspondence was sparse, the MIM classifier appeared to require more samples to adequately learn a model than the GBN. This was evident for the 'Mushroom' database.

The experimental results demonstrate 'tree' based methods (excluding NB) generally induce MBs using only a small fraction of the available domain features. This means, as expected, they will have a lower inference complexity compared to that of the networks. Despite this restriction in topology, performance was not reduced and in some cases found to be better than that achieved by GBN.

The high dimensional problems such as 'DNA' 15(60) and 'Promoter' 4(57) had the greatest reduction in MB size. This implies that data sets with large numbers of features are more likely to have several redundant or irrelevant features, with respect to the class variable, which are not selected by the learning algorithm. In respect of the CL algorithm these features will be low values of MI and thus less likely to be considered during the 'tree' edge selection process.

From the experimental results both the MIM classifier and 'polytree' performed better than GBN on high dimensional domains, especially if the number of cases is small. This was observed for the data sets 'Promoter' and 'DNA'. The reason for this may be due to inadequate modelling. For those domains that resulted in large networks, and were sample sizes were sparse, GBNs may have had difficulties in estimating reliable model parameters. Since the MB for the 'polytree' and MIM classifier are somewhat constrained by having their MB defined by the CL algorithm, the small sample sizes do not appear to have had the same impact as for the GBN. In the case of the 'polytree' the MB will be defined by the recovered directionality and may be larger than that of the MIM classifier.

In general, 'trees' only use a subset of the domain features which leads to better parameter estimates when relatively little data is available. In contrast, overfitting is common for networks which may lead to them picking up spurious dependencies in this situation.

In conclusion, the MIM classifier will generally be more beneficial than GBN for all types of data sets. The benefit will be more so for data sets that are of a high dimensionality with potentially a 'multi' parented class variable, and less for data sets with only a few features, especially if the size of the data sets is large.

4.6.2 Dependence models Opposed to Independence models (NB)

In general, the methods that did not assume extreme conditional independence performed better than the NB classifier as shown in Table 4.2. The experimental results show that this was especially so for the MIM classifier, demonstrating that the extra modelling power (taking dependencies into

consideration) over NB approaches actually makes a difference in practice. Table 4.2, shows that NB was only better for 4/20 data sets over the other dependency models. Specific details concerning dependency 'tree' models and NB were discussed in section 4.5.2 and will not therefore be repeated in this section.

The improved performance is due to the difference in CI assumptions that dependency/independence approaches make. The NB assumes that the features are independent of each other given the class variable. However, this assumption is not normally valid in the 'real' world. Most domains have extensive correlations between the features and the independence assumptions often leads to a degradation in accuracy of the NB classifier. This is supported by Langley [LIT92] and further by Freidman [FGG97], who proposed the NB performance could be improved by relaxing independence assumptions. Our experimental results confirm this and show that more accurate modelling of the dependencies among features does lead to improved classification. Since the 'tree' models are computationally easier to manage than the full unrestricted networks, they may offer a robust middle ground solution between networks and the NB classifiers.

The advantage NB does have however, is its ease of induction and efficient inference. In contrast, dependency models need to be learned and may have intractable inference as a consequence of the final topology recovered. Although 'tree' structures overcome this, 'polytrees' still have issues regarding node ordering and the possibly of large CPTs. The disadvantage of NB is that it violates 'real' world assertions and offers no qualitative information of the domain. From the results shown in Table 4.2, the MIM classifier represents a good middle ground model overcoming the issues that limit the 'polytree' and GBN classifier. Whilst performing better than NB it also offers a qualitative structure and can be both learned and perform subsequent inference efficiently. Like NB it is fairly robust and performs well even for small sample sizes and high dimensional problems.

Despite NBs ease of construction, if the task is to build a classifier, then the performance gained by modelling all features dependencies (even if restricted) has to be weighed against the possible increase in induction and inference costs. For networks without prior node ordering the complexity will be $O(N^4)$ in comparison to 'trees' ('polytree' and MIM classifier) $O(N^2)$. In contrast the complexity for the NB classifier is $O(N)$. However, once constructed, the MIM classifier inference complexity is comparable with NB. Moreover, since inference is reflected by its MB, the MIM

classifier will be potentially less, as only a subset of the features will actually require updating, that is, in respect of the branch weight changes.

To categorise the types of problems where it is worthwhile to use the MIM classifier and when it is more effective to use NB, we consider Figure 4.14. The data sets on which the MIM classifiers are significantly better than NB are those that have large sample sizes. For example data sets, 'Chess', 'Letter', 'Mushroom' and 'Nursery'. This is the case even if the problems have a medium dimensionality such as 'Chess', 'Mushroom' and 'Letter' (around 14 features). On the other hand the MIM classifier was worse generally on those data sets that have small data sets and medium dimensionality (14+ features) such as, 'CRX', 'Austria' and 'Vehicle'. For the latter data sets the NB classifier will be better suited.

4.6.3 MIM classifier Opposed to 'Polytree' classifier (SCN)

For the majority of the databases the MIM classifier was comparable in terms of 'overall' performance with the 'polytree'. This was expected since structurally they are very similar with the only difference being attributed to node ordering defining the final topology. In some cases, the node ordering for the 'polytree' modified the topology sufficiently to reduce its performance, but this was not seen to be significant. Despite the reduction in complexity the 'tree' methods not assuming extreme conditional independence performed comparably with that achieved by the unrestricted GBN on the selected databases.

From our results the 'polytree' performed adequately with high dimensional, low sample problems. However, for domains with multi-parented class variables, especially if the number of attribute states is high, the large CPTs (and thus the considerable number of parameters to be estimated) could lead to an intractable inference problem. For these domains the MIM classifier would be better suited.

4.7 Drawbacks

During the process of experimentation we identified several drawbacks both in design and assumptions in respect of the MIM classifier. The following raises the issues with some initial approaches to dealing with them.

- The use of the CL algorithm, whilst efficient, can have disadvantages, as raised in section 3.6, Chapter 3, as it is prone to generating trees that have missing relevant features or adding irrelevant features to the class MB. In general, our results did not confirm this in practice, except

perhaps for the 'Mushroom' data set, where the MIM classifier's MB was defined by only 1(22) features. This was a highly correlated feature with respect to the class variable, however, the GBN which outperformed the MIM classifier, actually considered more features relevant resulting in a larger and more informative MB. By adopting the approach suggested by Cheng [CG99], as discussed in Chapter 3, the initial CL algorithm used by the MIM classifier could actually be used to construct a network, using only the class MB to perform classification. Essentially, using Cheng's thickening and thinning phases a Wrapper type refinement could be applied to the initial CL algorithm generated MB. In this case the MIM classifier itself would provide the evaluation function in order to determine the optimum classification measure.

- In using the CL algorithm and orienting the tree structure with the class variable as its root, we can consider the algorithm is an implied feature selector. However, due to the issues concerning the CL algorithm discussed previously, the class MB may not represent the optimal for the specific task of classification. Whilst the proposed use of Chengs' approach to discover both an improved qualitative structure and class MB would potentially over come this, it is a computationally expensive solution. This is especially true if no prior node ordering is supplied before hand. In the next chapter, we propose an alternative approach to improving the MIM classifier to potentially enhance its predictive capability and hence reduce the effects of this drawback.
- In order to classify new evidence of the domain the MIM classifier assumes that each class-state is uniquely characterised by a feature 'profile'. For the specific task of classification this will be defined and bounded by the class MB, which is derived via the CL algorithm. If the domain is poorly characterised the class MB will not contain sufficient class-state identifying attributes to enable the MIM classifier to discriminate between some class-states. This will be particularly evident when feature 'profiles' of observed samples within one class-state are very similar to those of other class-states. A good example of a poor characterisation is illustrated in Figure 4.17. The figure represents a random selection of observed samples taken from the 'Vehicle' database. Each individual sample is a plot of the values for the 18 features that make up its description. If we consider sample number 3 and sample number 16 we can see that these particular cases 'profile' very closely to class-state Bus and Van respectively.

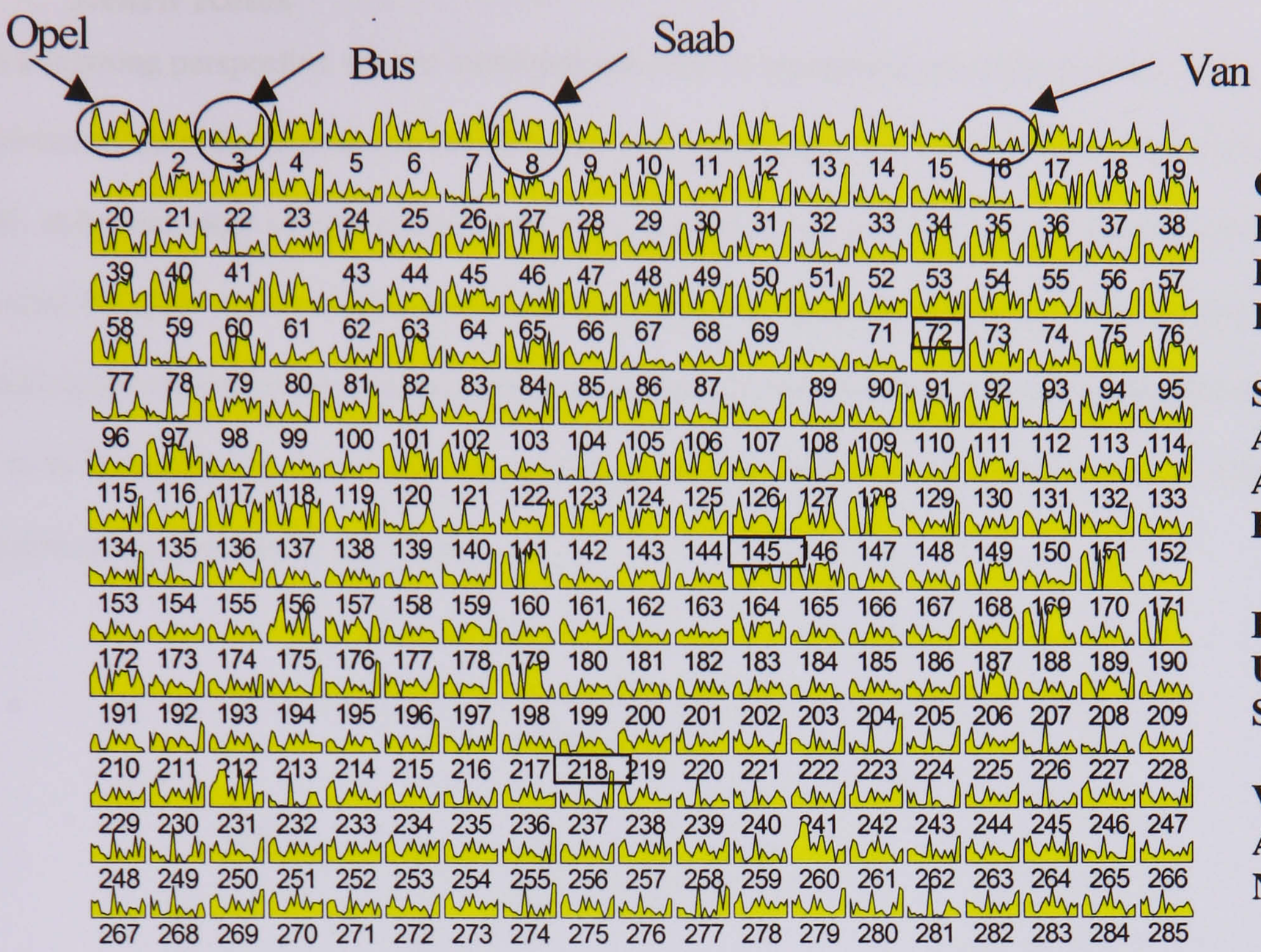


Figure 4.17. 'Vehicle' Database: Sample 'profiles'

However, within the database they have both been assigned as belonging to the class-state Opel. This lack of class-state distinction will impact on the MIM classifiers' predictive capability as the results indicate in Table 4.2. The GBN and NB on the other hand appear to be able to handle this domain a little better, presumably due to having larger and more informative class MBs. In Chapter 6 we will demonstrate that by expanding the class MB the performance of the MIM classifier can actually be improved for this domain. Overall however, all methods studied found this data set challenging and as it may well be a characterisation within the data set itself likely to be difficult to overcome. For the MIM classifier an alternative approach is to learn a model with a view towards maximising the classification measure. The concept of Joint Mutual Information [TF^{+}01] is a possible approach and is reviewed in more detail in future work, Chapter 8.

4.8 Learn Rates

From a learning perspective we are interested not only in asymptotic accuracy but also in the rate of improvement. In order to measure the effect of 'tree' representation (specifically the MIM classifier) on the induction rates of the learned structures compared to networks, learning curves were also generated for some of the databases. From the curves, it is clear that in general the dependency tree methods achieved their asymptote accuracy at a faster rate than the GBN as well as the NB classifier. This is to be expected since 'trees' are generally smaller, and thus need fewer cases to learn their parameter sets compared to the GBN.

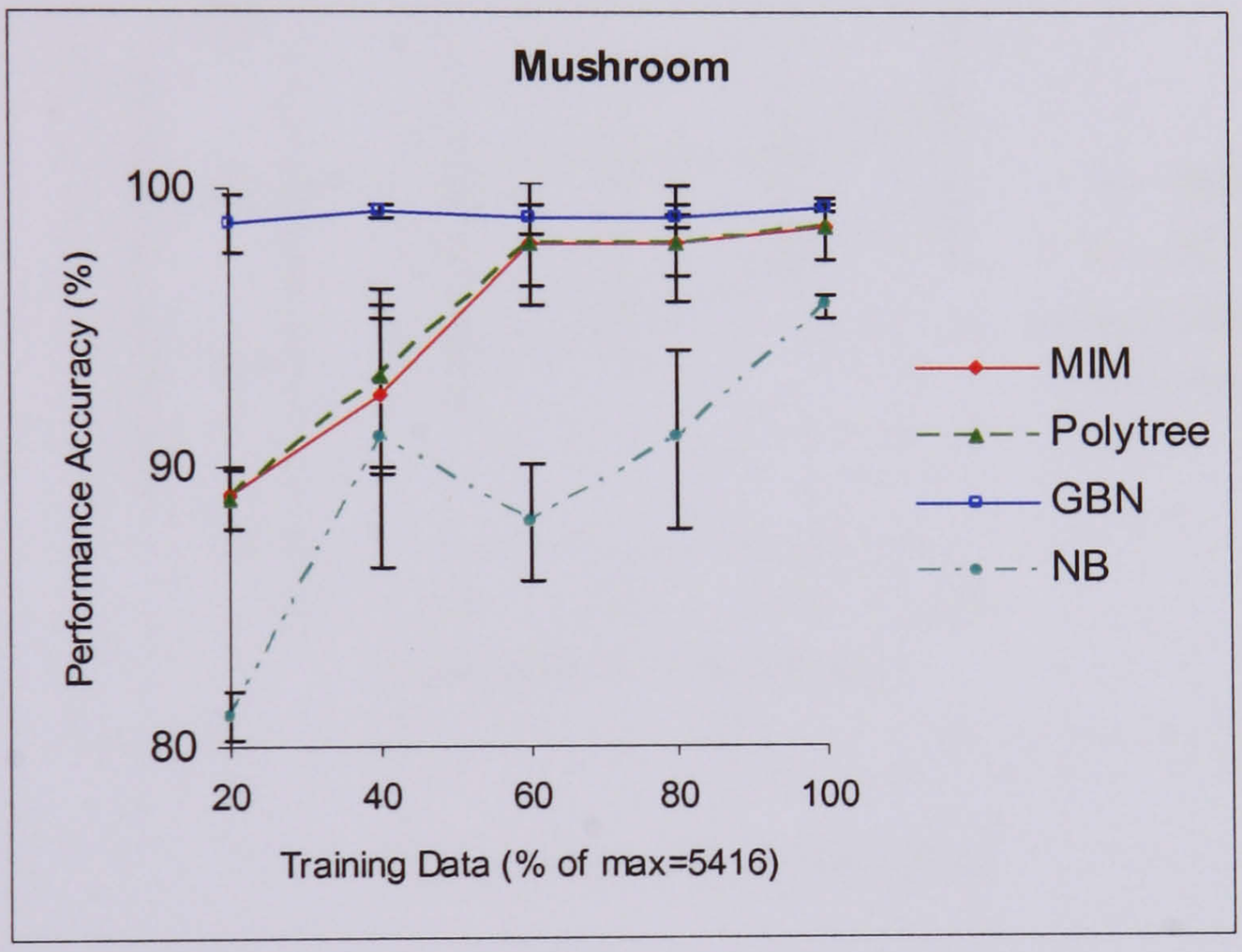


Figure 4.18. Learning Plot: Mushroom

Figure 4.18 - Figure 4.24 illustrate the learning curves for the databases: 'Mushroom', 'Chess', 'DNA', 'Nursery', 'Letter', 'ANN' and 'Segment', based on an average of 25 runs. Only the larger databases, as shown in Table 4.1, were investigated, that is, those using a hold-out approach, as the smaller databases were evaluated by a 5-fold cross-validation approach.

Langley [LIT92] showed that the NB classification performance was poor for databases 'Mushroom' and 'Chess' but good for 'DNA'. In our investigations this was also evident. The MIM classifier and 'polytree' classifiers performed comparably for the 'Mushroom' database and stabilised at 60% of the sample size with both having differences that were statistically significant (p-value = 0.021 for

the MIM classifier compared to NB and p-value = 0.02 for 'polytree' compared to NB). The GBN in contrast was stable throughout the sample sizes performing slightly better and also had differences that were statistically significant (p-value = 0.002 compared to NB). We observed that the 'Mushroom' database did not require many features to classify the majority of the test samples and it was evident that both the MIM classifier and 'polytree' methods did not have sufficient class – attributes in their model representations.

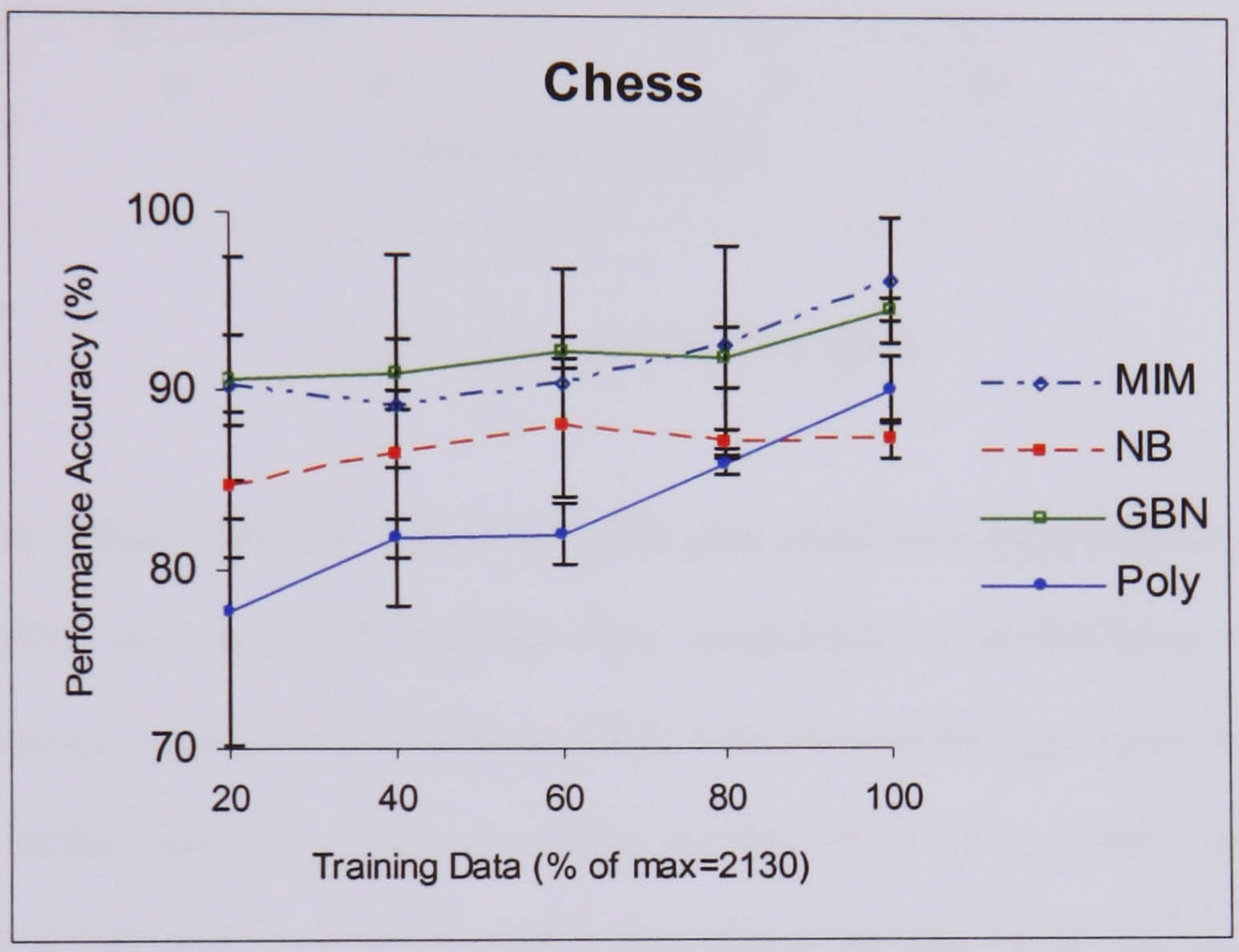


Figure 4.19. Learning Plot: Chess.

For the 'Chess' database, Figure 4.19, both the MIM classifier and the GBN methods improved performance as the sample size increased. However, the 'polytree', although structurally similar to the MIM classifier, performed poorly compared with the MIM classifier which had differences that were found to be statistically significant (p-value = 0.001 compared to 'polytree' and p-value <0.05 compared to NB). This may indicate a poor topology and corresponding bad choice of branch directionality, as determined by the node ordering algorithm, for a more complex structure necessary to model the 'Chess' database.

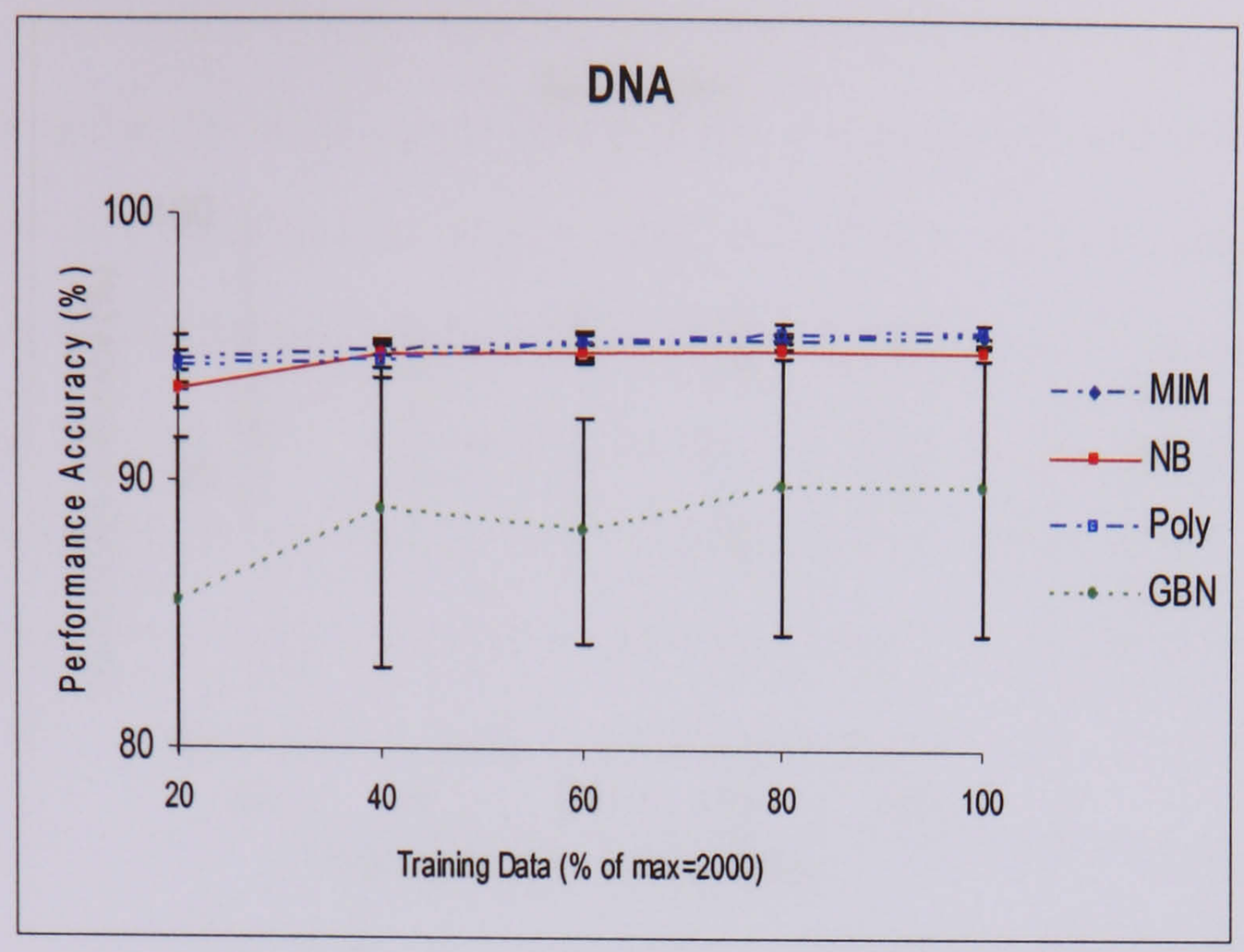


Figure 4.20. Learning Plot: DNA.

In the case of the 'DNA' database, Figure 4.20, NB performed well, outperforming the GBN. Both the MIM classifier and 'polytree' methods were comparable in performance remaining stable throughout all sample sizes and had differences that were statistically significant along with NB (p-value <0.05 for MIM classifier compared to GBN, p-value <0.05 for 'polytree' compared to GBN, and p-value = 0.001 for NB compared to GBN). The 'DNA' database structure has a strong class – attribute association which clearly favours the NB classifier. As the GBN was reduced to this NB representation a degraded performance was observed. This was not the case for the MIM classifier or 'polytree' methods. As the 'Nursery' database, Figure 4.21, also characterised strong feature independence, it was not unexpected that the resulting plot was similar to that of the 'DNA' database. However, unlike for the 'DNA' database, only the MIM classifier and 'polytree' at 60% sample size had differences that were statistically significant (p-value = 0.045 for MIM classifier, and p-value = 0.046 for 'polytree' compared to GBN), which was not the case for NB with a p-value = 0.587 compared to GBN.

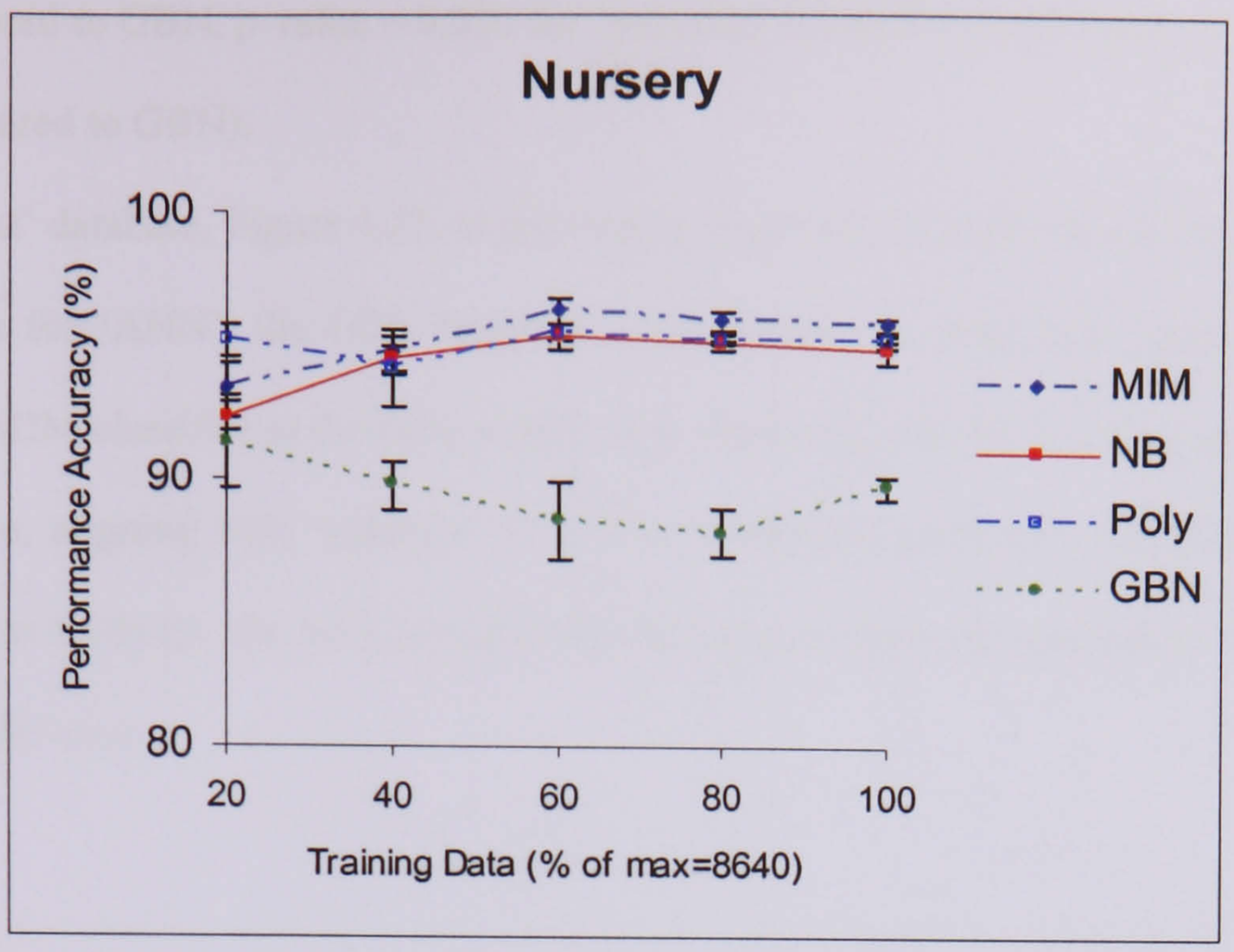


Figure 4.21. Learning Plot: Nursery.

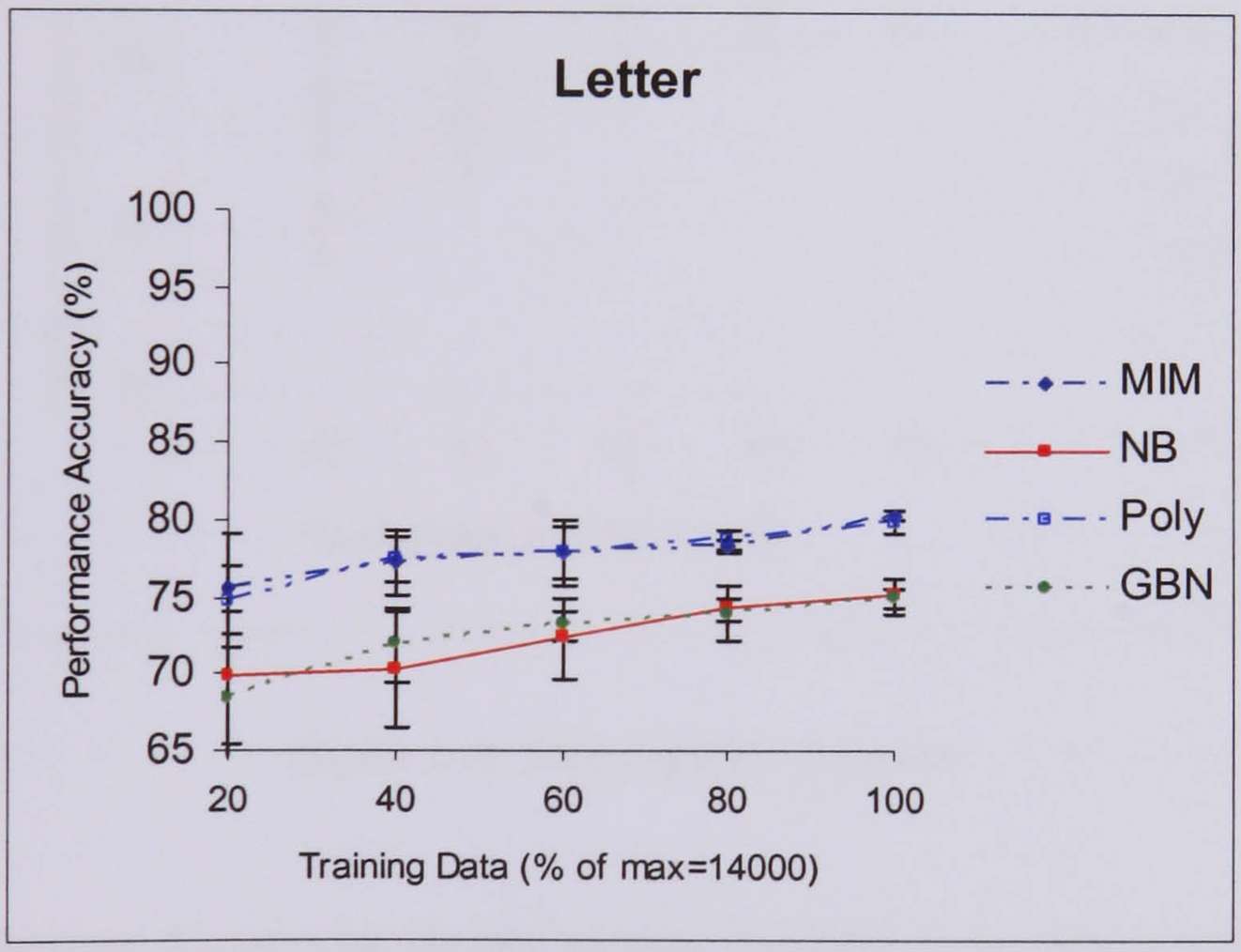


Figure 4.22. Learning Plot: Letter.

In respect of the 'Letter' database, Figure 4.22, all four methods achieved stable performance levels at 60% of the sample size. The GBN and NB classifiers aligned at a lower level of performance accuracy than both the MIM classifier and 'polytree' classifiers, with the latter maintaining similar profiles in terms of performance accuracy and differences that were found to have statistical significance (p-value = 0.002 for MIM classifier compared to NB, p-value = 0.002 for MIM

classifier compared to GBN, p-value = 0.003 for 'polytree' compared to NB, and p-value = 0.004 for 'polytree' compared to GBN).

For the 'Segment' database, Figure 4.23, as previously observed, all methods stabilised at 60% of the sample size. As for 'ANN', the GBN required more samples to learn and matched performance levels with the MIM classifier at the 60% sample size. Similarly, the NB required additional cases to learn the domain, aligning with 'polytree' at a lower predictive level than the MIM classifier and GBN. Of the four methods, the NB classifier had the steepest learning rate before stabilising at the 60% of the sample size.

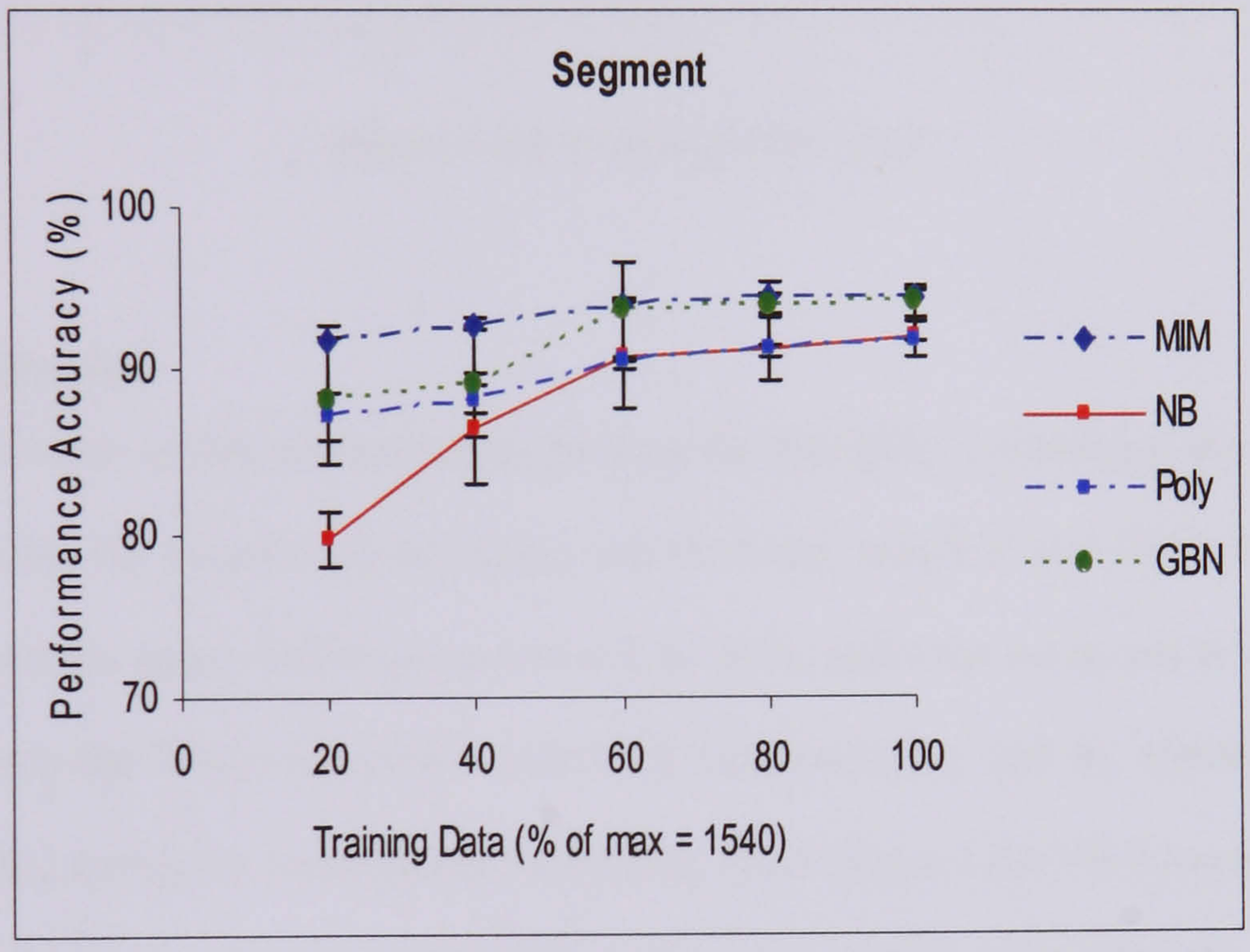


Figure 4.23. Learning Plot: Segment.

In the 'ANN' database, Figure 4.24, all four methods stabilised at the 60% sample size, with NB 'overall' better than the dependency models with differences that were statistically significant for all models (p-value = 0.022 compared to 'polytree', p-value = 0.006 compared to MIM classifier, and p-value = 0.001 compared to GBN). Whilst the MIM classifier and 'polytree' performed virtually the same, the worse was GBN. After an initial steep climb, it too stabilised at 60% of the sample size but achieved a lower predictive level than the 'tree' models. Presumably, there were initially insufficient samples for GBN to learn the domain than required for the 'tree' based approaches.

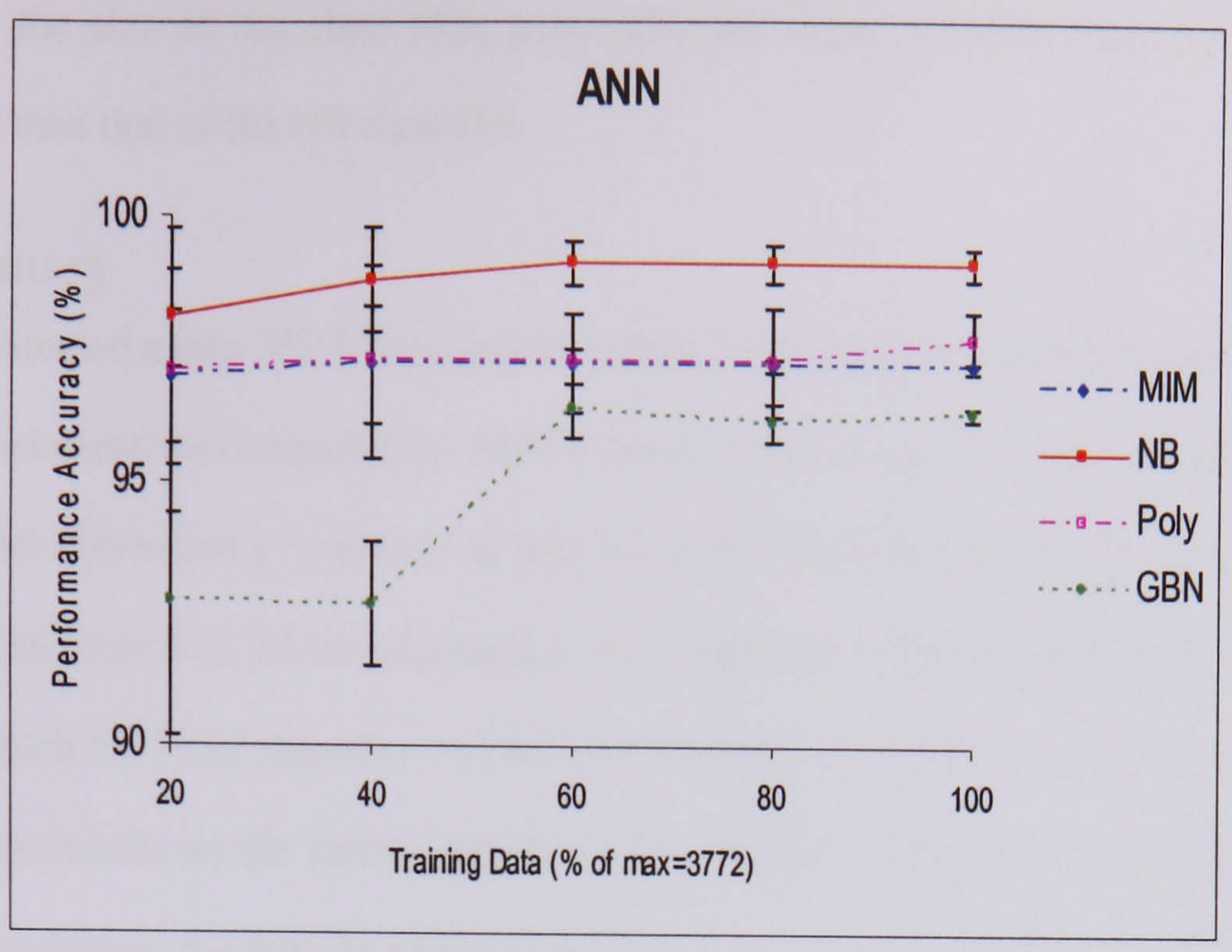


Figure 4.24. Learning Plot: ANN.

4.9 Conclusion

The main contribution of this section lies in showing the feasibility, advantages and effectiveness of the MIM classifier. By overcoming the issues that limit the 'polytree' and GBN models, the MIM classifier represents a good middle ground model. In using pair-wise marginals to calculate branch weights, it avoids the issues associated with CPT dimensionality and by application of a node ordering heuristic, avoids the consequences of making a bad choice. Like NB it has been shown to be fairly robust and demonstrates a comparable performance despite restricting the structure to the topology of a 'tree'. Unlike NB however, the MIM classifier also offers a qualitative structure, which can be efficiently learned by applying the CL algorithm. For small sample sizes and high dimensional problems, particularly where there maybe 'multi' parented nodes, the experimental results show that the MIM classifier performs well compared to unrestricted GBNs. In respect of its learn rate however, more samples are in general required for the MIM classifier to adequately model the domain than the GBN.

The empirical studies show the MIM classifier performed better than NB for many of the larger UCI data sets with a comparable inference complexity. For many of the data sets studied, the class MB for the MIM classifier was defined by a subset of the domain features and thus required only a few branch weights to be updated in order to classify new evidence of the domain. Since the complexity

is reflected by the size of the class MB, potentially the MIM classifier may, for some domains, actually be less than that of the NB classifier.

4.10 Summary

This chapter evaluated a new MIM approach to inference in Singly Connected Networks. In the main part of our experiments we compared the MIM classifier with two other 'tree' modelling approaches, namely the Naive Bayes and a 'polytree' as defined by Pearl & Rebane [RP87], along with a general Bayesian network approach. More importantly, we identified salient characteristics of the types of problems for which the MIM classifier will be most beneficial compared to other representations.

The experimental results on the various data sets show that the MIM classifier generally outperforms GBN, NB and 'polytree' for thirteen of the twenty databases studied. Where the 'trees' demonstrated a low performance, we conclude that the specific data set could not be appropriately approximated by a dependency tree. That is, the assumption that the underlying distribution is tree-based is probably violated.

The 'polytree' and MIM classifier, induced by use of the CL algorithm, were less complex and provided efficient inference for new observations of the domain. Although they had a restricted topology (tree based) they both still demonstrated comparable, sometimes better, predictive accuracy to the unrestricted GBN approach. The MIM classifier outperformed the NB, one of the most widely studied BNs within the ML community, in fifteen of the twenty databases used in the study.

The MIM classifier had differences that were statistically significantly better than NB on four of the twenty databases used. For 'Car_Evaluation', although the NB did perform better the differences were not statistically significant at less than 0.5% performance improvement. In respect of the GBN there were three databases in which it performed better than the MIM classifier, however, in general the overall performance of the MIM classifier matched that achieved by the GBN. The MIM classifier performed better on seventeen of the twenty databases with three of these improvements having differences that were statistically significant. This result was similar in respect of the 'polytree', with the MIM classifier performing better on fourteen of the twenty databases.

The proposed use of mutual information measure 'branch weights' as a mechanism for classifying new unseen evidence has been demonstrated as feasible. The approach taken provides for both an

efficient and localised method of inference in singly connected networks with comparable performance levels of less restricted methods.

By modelling the domain using efficient 'tree' structuring algorithms we have avoided the issues of complexity and overfitting prone to networks. Moreover, the utilisation of Chow and Liu's algorithm allows for tree construction to be achieved using only pair-wise marginals, and although a 'restricted' model, has not required us to make extreme conditional independence assumptions.

Our experimental results on the selected databases have demonstrated that the MIM classifier's performance was not affected by our node ordering approach and did not show any dependence or consequences of making a bad choice as observed in the 'polytree' representation. In addition, for databases that were known to have strong feature independence properties, the reduction of the structure to that of a 'NB' representation appeared not to degrade the performance of the MIM classifier as it did for the GBN. Table 4.3 illustrates a summary of the results in respect of the four methods investigated. Each entry describes a property that is characteristic of a method along with the most appropriate domain for its application.

As pointed out in section 4.7, there are however, a number of ways in which the performance of the MIM classifier can potentially be improved. In the next chapter we consider the possibility of improving the MIM classifier's predictive performance by expanding the class MB.

Table 4.3: Summary of Results

	Method			
	MIM classifier	**NB**	**'Polytree'/SCN**	**GBN**
Structure Format	BN dependency Tree	BN model of independence	BN dependency Tree	BN dependency Network
Strengths	Does not require prior node ordering. Uses pair-wise marginals not CPT for inference. Not influenced by large CPTs. Asymptote rate generally faster than GBN. Generally has smaller class MBs than NB and GBN. Works well with strong feature independence characterised data. Does not make strong assumptions of CI.	Ease of induction (trivial structure - no learning required). Robust model. Efficient inference. Better over GBN for strong feature independence characterised data sets. Not influenced by issues of large CPTs. Does not require prior node ordering. Better than GBN when there is relatively little data available.	Reduced parameterisation – inference complexity reduced. Does not make strong assumptions of CI. Asymptotes at a faster rate than GBN and NB in general. Avoids overfitting. Exact inference (propagation). Requires fewer samples to learn than GBN. In general, has smaller class MB than GBN.	Good for dealing with highly correlated featured data sets (over NB). Does not make strong assumptions of CI. Offers good human interpretation of complex domains. Best representation for dealing with uncertainty. Richer class MB than dependency trees. Compact representation of the JPD.
Weaknesses	Model influenced by the 'characteristics' of the CL algorithm – may miss relevant and/or add irrelevant features. May not represent an optimal class MB. Class-states need to be well characterised within data set.	Violates 'real' world assertions (strong assumptions of CI). Affected by presence of highly correlated features. Asymptote rate lower than dependency trees. Uses all domain features (large class MB).	Issues with large CPTs – unreliable probability estimates, 'multi' parented nodes. Model not always fully recoverable (without 'expert' help). Dependency on prior node ordering.	Prone to overfitting. Issues with large CPTs – unreliable probability estimates, 'multi' parented nodes. Dependency on prior node ordering. In general, asymptotes at a slower rate than the dependency trees. Learn/inference – in general case NP-hard.
Model Build Complexity	$O(n^2)$	$O(n)$	$O(n^2)$	$O(n^2)$ with prior node ordering else $O(n^4)$
Suitable Domains for Approach	High/medium dimensional, small sample size (over GBN). Medium dimensional, large sample size (over NB).	Medium dimensional, small sample size (over MIM). High/medium dimensional, small sample size (over GBN).	High dimensional, small sample size (over GBN). Medium dimensional, large sample size (over NB).	Small/medium dimensional, large sample size (over MIM/polytree). Small/medium dimensional, small sample size - general.
Qualitative Structure	Yes - Restricted	No	Yes - Restricted	Yes - Unrestricted
Inference Method	MI edge 'weights'	$P(C)$ and $P(Z_i \mid C)$	CPT probability estimates	CPT probability estimates

Key: In the context of the data sets studied in the thesis. High dimensional ≡ 35+ features, Medium dimensional ≡ 14+ (max 34), Small dimensional ≡ under 14 features.

4.11 Appendix

Studies and applications utilising contingency tables have long been a part of statistical analysis. Kullback [KL51] in conjunction with the concepts of communication theory proposed that the significance χ^2 [Mor974] could be approximated by the independence component $\hat{I}(H_1 : H_2)$ multiplied by 2. H_2 is the null hypothesis and H_1 the alternative. That is: $H_2 : P_{ij} = P_{i.}P_{.j}$ and H_1 the alternative $H_1 : P_{ij} \neq P_{i.}P_{.j}$.

If we consider a two-way table then we have for $(r-1)(c-1)$ degrees of freedom (df):

$$2\hat{I}(H_1 : H_2) = \sum_{i=1}^{r}\sum_{j=1}^{c} \frac{\left(x_{ij} \frac{x_{i.}x_{.j}}{N}\right)^2}{\frac{x_{i.}x_{.j}}{N}} \sim \chi^2 \qquad \text{.....(1)}$$

Where for N independent observations x_{ij} is the frequency in the i[th] row and j[th] column and:

$$x_{i.} = \sum_{j=1}^{c} x_{ij} \quad x_{.j} = \sum_{i=1}^{r} x_{ij} \text{ and } N = \sum_{i=1}^{r}\sum_{j=1}^{c} x_{ij}$$

Denoting probability by P with the corresponding subscripts that is:

$$P_{ij} = \frac{x_{ij}}{N} \quad P_{i.} = \frac{x_{i.}}{N} \text{ and } P_{.j} = \frac{x_{.j}}{N}$$

The equation (1) can be written as:

$$2\hat{I}(H_1 : H_2) = 2\sum_{i=1}^{r}\sum_{j=1}^{c} NP_{ij} \log \frac{P_{ij}}{P_{i.}P_{.j}} \qquad \text{.....(2)}$$

which is :

$$2\hat{I}(H_1 : H_2) = 2\sum_{i=1}^{r}\sum_{j=1}^{c} x_{ij} \log \frac{Nx_{ij}}{x_{i.}x_{.j}} \approx \chi^2 \qquad \text{.....(3)}$$

Using (3) and equation from equation 1 (Procedure 1, Figure 3.1) this can be rewritten as:

$$I(X_i, X_j) = \sum_{x_i x_j} P(X_i, X_j) \log \frac{P(X_i, X_j)}{P(X_i)P(X_j)}$$

and so: $2IN \approx \chi^2$ for df $(r-1)(c-1)$. .....(4)

Chapter 5

Selective MIM Classifier

In Chapter 4, we demonstrated that the performance of the MIM classifier was competitive compared to both NB and the more complex representations of GBNs. However, in section 4.7, we identified a drawback concerning the class MB defined by the CL algorithm. Whilst the 'tree' based structure constructed may offer a satisfactory qualitative representation of the domain, it may not, in respect of the task of classification, offer the best solution. This aspect was previously discussed in Chapter 3, section 3.3. Since the relevant area of features for classification is identified by the MB [Pea88], the quality of the subset, defining the MIM classifier class MB, will accordingly influence its performance. In this chapter, we propose a method for improvement of performance by focusing on the MIM classifier's class MB. Targeting the MB has been studied by other researches such as Tsamardinos [TA03] who showed that the MB corresponds to the strongly relevant features as defined by Kohavi [KJ97] and further by Margaritis [MT00] whose efficient algorithm identified the MBs of domain nodes for subsequent learning of BNs. Madden [Mad02] on the other hand demonstrated that by focusing on the MB alone, an efficient classifier could be constructed, whilst Cheng [CH$^+$02] similarly identified the MB during network construction and showed that in respect of the task of classification, all features outside the MB could be safely deleted. More recent approaches concerning the MB can be found in [BCS04] for Bayesian networks and [FF03] which identifies the MB with decision tree induction. In the case of the MIM classifier, if we consider the class MB, defined by the CL algorithm, as representing an 'initial MB', then the task of improving performance can be considered as a 'feature selection' problem. Essentially, the CL algorithm, in respect of the class variable, acts as an implied feature selector which in general selects a subset of the domain features.

In the following section we review the main approaches to feature selection with sections 5.2 and 5.3 describing a technique for inducing and classifying with, a selective variant of the MIM classifier

respectively. In section 5.4, we discuss some related work concerning the MIM proposed technique, and in section 5.5, we summarise the chapter.

5.1 Introduction to Feature Selection Approaches

Feature selection (FS) is fundamental to a number of different tasks such as data mining, image processing and classification. The main usage is in dimensionality reduction where algorithms are required to identify and select the best subset of the input feature set with respect to some target. Typically this is classification accuracy. Clearly, the examination of all possible candidate subsets via an exhaustive search is unfeasible and impractical even for a moderate number of domain features. In fact Cover [CVC77] showed that no non-exhaustive sequential feature selection procedure could be guaranteed in general to produce the optimal subset. However, a number of suboptimal feature selection techniques have been developed and they essentially trade-off the optimality of the selected subset for the computational efficiency [JZ97]. Nevertheless, there have been some impressive performance gains in attacking large dimensionality with many irrelevant features, examples of which can be found in [DB00, Kah94, TC+03 and Ng98].

There are two modes of selection. Forward selection, where a growing set of features are evaluated to maximize some criterion function, and Backward selection where a shrinking set of features are evaluated. The strategies that have been generally used to evaluate alternative subsets of features fall into two main classes, Wrapper and Filter.

Wrapper strategies imply that the selection algorithm searches for a good subset of features using the induction algorithm itself as a part of the evaluation function. Examples can be found in [JKP94, KJ97, KS95, Hsu04, ZW+03 and IL+00]. Filter strategies on the other hand do not use the learning algorithm in the evaluation function but use the intrinsic properties of the data to asses the goodness of the feature subset. Kira [KR92] for example developed a system called RELIEF which uses a statistical method to select relevant features, however its application was limited to binary classes. Witten [WF00] overcame this limitation with an updated version, REIEF-F. Other examples of Filter approaches can be found in [BW00, KS96 and DK03]. A comprehensive overview of many methods can be found in [DL97] with specific feature selection approaches studied within the machine learning (ML) literature in [Lan94].

Blum [BL97] argues that any feature selection method (Wrapper or Filter) must take into consideration four basic issues that determine the nature of the search process.

The four issues are :

- *The starting point in space.*

This determines the direction of the search. Options available are to start with no features and successively add them or to start with all features and successively remove them. The alternative is a variant of both, starting in the middle.

- *Organisation of the search.*

This is the strategy of the search, for example sequential forward or backward selections.

- *The evaluation function.*

The evaluation function is a measure of the effectiveness of a particular subset of features after the search algorithm has made its selection. This could be based on information in respect of that gained from adding a feature or in the case of a Wrapper, perhaps the classifier error rate.

- And lastly the *Criterion for halting the search.*

This determines when to stop the search process. A typical criterion is when no further improvement of the evaluation function value of the alternative subsets is obtained.

5.2 'Selective' MIM Classifier- Selecting Features for the Class MB

In the previous section we identified two of the most widely used strategies for evaluation of the alternative subsets of features, namely Wrapper and Filter. For the MIM classifier enhancement we propose a 'Wrapper' type method and as Blum [BL97] suggests, start by considering the four issues in respect of this approach, essentially, our preliminary design considerations.

- *The starting point in space.*

In the introductory paragraphs to this chapter we suggested that the class MB defined by the CL algorithm could be considered as an 'initial MB'. Since the algorithm is effectively working as an implied feature selector, we can take this subset as representing a reasonable starting point in constructing the 'Selective' MIM classifier (SMIM). As we are only interested in improving classification performance here, the class MB is the only area of interest and thus the focus of expansion.

- *Organisation of the search.*

With the 'initial MB' representing the starting point we will expand the class MB by adopting a sequential forward selection process. In order to determine which edge to add to the 'initial MB' feature subset, the edge MI value or branch weight will be first ordered by size and then assessed. Other examples that have utilised information based algorithms for feature selection are Zaffolon [FZ00] where MI was used in a Filter approach to select features, Kleiter [KJ96] who used MI to learn BNs from data and Last [LKM01] where an information-fuzzy neural network was proposed for feature selection.

Since we are only interested in the class MB expansion, only those edges corresponding to the class variable C-Z and not Z-Z will be considered. As we are assuming that the starting point class MB is a good representation (confirmed by the results obtained in Chapter 4) we will not remove any features from the 'initial MB'. The limitation of this approach is that we will not be able to achieve an 'optimal' solution.

- *The evaluation function.*

The strategy to be adopted is a Wrapper type, and in this implementation the MIM classifier will be used as the evaluation function. Since the classifier is a 'tree' based representation and will be transformed into a network as new edges are added, the classifier will require modification in order to deal with the expanded MB. We will consider this in more detail in the following sections.

- *Criterion for halting the search.*

In respect of the Wrapper approach and the use of the MIM classifier as the evaluation function, the halting criterion will be when the addition of features offers no further improvement to the evaluation function value.

From the results of Chapter 4, we demonstrated the comparable performance of the MIM classifier against the NB classifier and in some cases showed it to be better. By design the NB uses all the domain features to define the class MB. In general, the MIM classifier required only a subset, therefore we expect most of the relevant features to have already been selected by the CL algorithm. Thus our choice of the Wrapper strategy should in general not be constrained by the consequential computational expense. Despite the need to call the induction algorithm (here the MIM classifier) for each feature set considered, it will only be in respect of the remaining subset, excluding the 'initial'

MB, which in the case of most data sets will be of a manageable size. Moreover, as a result of the work carried out by other researchers previously cited, when the size of the problem allows the use of the Wrapper approach, in terms of predictive accuracy, the Wrapper approach has been shown to have superiority over the Filter approaches.

Construction of the 'Selective' MIM classifier (SMIM) consists of two phases. First we will define the process for determining the 'optimised'[16] subset selection, then we will discuss the modifications to the MIM classifier in finalising the SMIM. As the classifier will also be used as the (Wrapper) evaluation function, the modifications will apply to both the resulting SMIM and the evaluation function as the selection process progresses.

For class labels in C namely $C_1,\ldots,C_m$ and features $Z=\{Z_1,\ldots,Z_n\}$ where $C \notin Z$, consider a subset of features defined by an application of the CL algorithm. This subset, the class MB, will represent the *lower bound* Z_{CL} and comprise the class variable related feature associations. Let Mp_o represent the measure of performance for this *lower bound*, this value being essentially the optimum error rate of the MIM classifier prior to any modifications (as defined in Chapter 3 and evaluated in Chapter 4). The feature selection process begins by evaluating each feature, not in Z_{CL}, and its corresponding contribution to the new subset.

If we denote a specific feature by Z_i which implies that $Z_i \in Z \setminus Z_{CL}$ and let k be a subset of features, then initialisation of the *lower bound* will be $k \leftarrow \{Z_{CL}\}$ and $Mp(k) \leftarrow Mp_o$.

Each Z_i will be selected on the basis of the maximal $I(C,Z_i)$.

Step 1: Compute $k \leftarrow \{Z_i\}$ | added a feature to the *lower bound*.

Step 2: Compute $g(k)$ | intermediate measure of performance of subset using MIM classifier.

Step 3: **IF** $g(k) > Mp(k)$

THEN

$Mp(k) \leftarrow g(k)$ | update current optimum of network.

$Ku \leftarrow \{k\}$ | update u*pper bound* Ku subset.

---

[16] Only 'Optimised' as we cannot guarantee that the final subset will be optimal.

Repeat for all Z_i steps 1-3.

IF $Ku = \{Z_{CL}\}$ at the end of the process

THEN

There is no improvement in performance and the *upper bound = lower bound.* {Essentially, the original MIM classifier performance and feature subset ('initial MB').}

ELSE

Ku{} will contain the new subset of features.

This approach unfortunately assumes that the selection of features results in a monotonic performance that improves or degrades in correspondence with the additions. However, both local minima and maxima are possible and if they occur would halt the selection process prematurely. For this reason it is necessary to evaluate all Z_i sequential additions. This could easily be considered a drawback, particularly if a domain has a large number of features and the 'initial MB' is small. However, our assumption, which is supported by the results achieved in Chapter 4, is that the 'initial' class MB is a good subset which already contains the most relevant features for classification (or at least the majority). Nevertheless to avoid the possibility of carrying out an exhaustive search in the case of large featured domains, two heuristic halting criterions are proposed.

- A reduction in the feature subset space can be achieved by applying a Kullback threshold to remove low values of branch MI (refer Appendix, Chapter 4, section 4.10), thus terminating the overall search earlier. In this approach we are not removing any edges or vertices but excluding potential features *C-Z* from being added to the class MB.
- Halting the search earlier can also be achieved when the measure of performance is found to have a succession of degradation as features are added to the subset. In this case the possibility of finding a local minima or maxima needs to be considered and the precise number of measurement assessments will be based upon an arbitrary choice.

5.3 Classification - SMIM Classifier/Evaluation Function

In the previous section, we proposed a Wrapper approach for selecting a subset of features, thereby expanding the MIM classifier class MB. In this section, we will show that the MIM classifier can still be used as the evaluation function, as defined in Chapter 3, even though the structure will change from a 'tree' based one into a network.

Consider the 'tree' structure shown in Figure 5.1, which represents a MWST for some domain consisting of a class $C = \{C_1, \ldots, C_m\}$ and a set of features $Z = \{Z_1, \ldots, Z_5\}$. The structure can be defined by two components, Z_{CL} which represents the class MB and Z_R the remaining edges. If we now consider the MWST in terms of its Mutual Information, then the 'overall' MI for the structure can be similarly represented by these two components. That is: $I(C,Z_1)+I(C,Z_2)$ defining Z_{CL} and $I(Z_1,Z_3)+I(Z_2,Z_4)+I(Z_1,Z_5)$ defining Z_R.

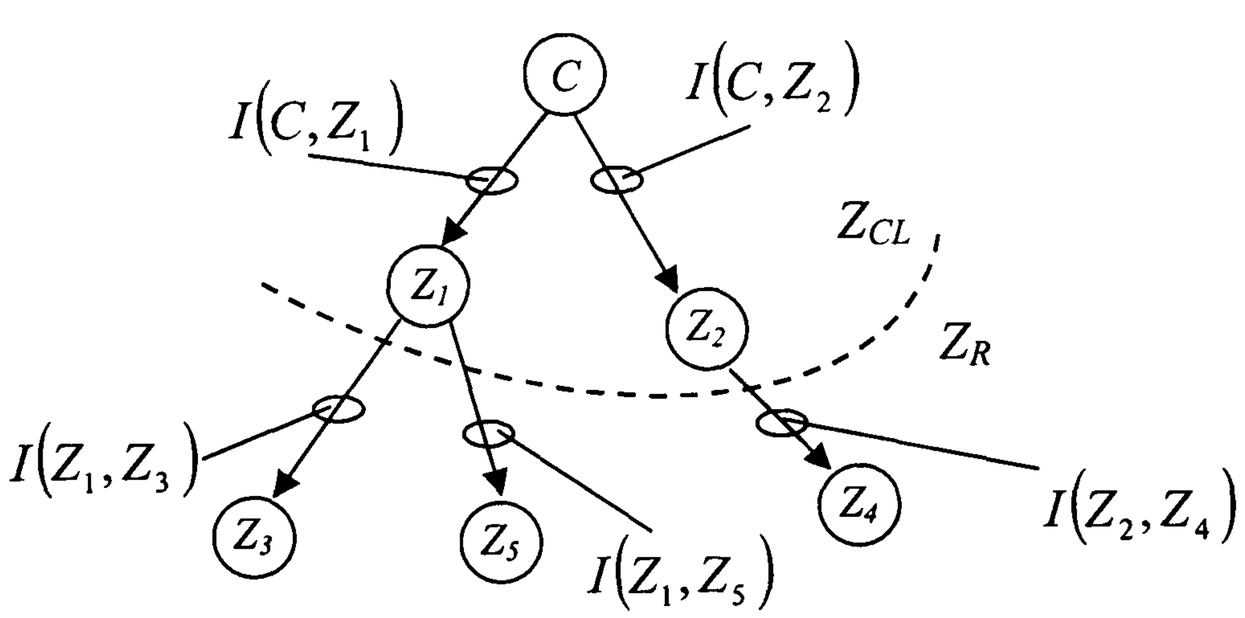


Figure 5.1. MIM 'tree' based Representation.

For the SMIM classifier the class MB is expanded by adding edges directed by the maximal MI values. For Figure 5.1, the possible considerations for inclusion will be the edges: $C \rightarrow Z_3$, $C \rightarrow Z_4$ and $C \rightarrow Z_5$. Figure 5.2, represents the SMIM with a new edge, $C \rightarrow Z_5$, added to the existing class MB. Z_{CL} in terms of the 'overall' MI will now be: $I(C,Z_1)+I(C,Z_2)+I(C,Z_5)$ whilst Z_R will remain unchanged. Since we have added the edge $C \rightarrow Z_5$ to the class MB, the structure has now changed from the 'tree' based one defined by the CL algorithm, to one of a

'network'. In the previous section however, the Wrapper approach proposed for the SMIM classifier used the MIM classifier as its evaluation function, which now needs to be able to handle a network rather than a tree structure.

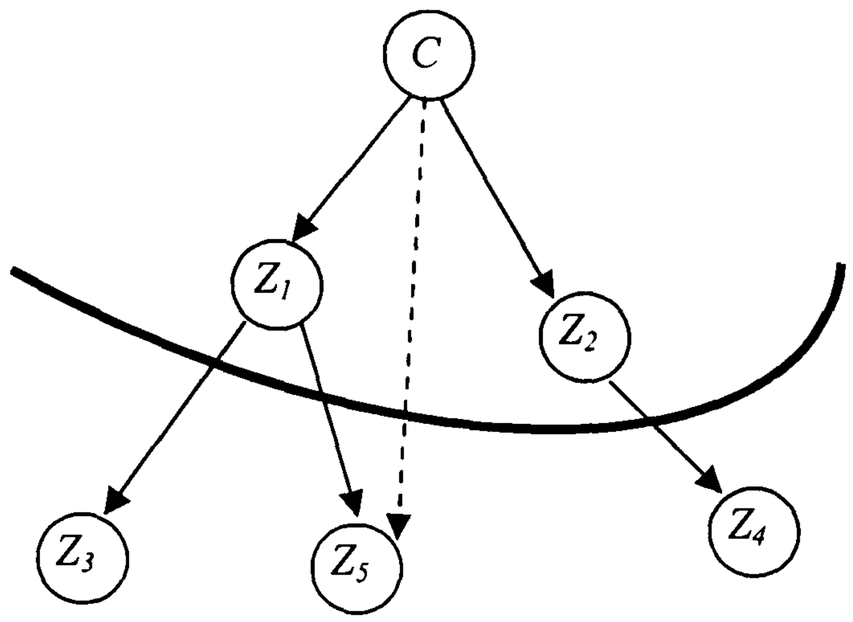


Figure 5.2. SMIM 'network' Representation.

If we consider an observation defined by a feature vector $\{C, Z_1, \ldots, Z_5\}$ and a class label with assignment C_1, then the 'overall' MI will be updated to reflect this observation. In the case of Figure 5.2, where the feature Z_5 has been selected for illustration purposes, $I(C, Z_1) + I(C, Z_2) + I(C, Z_5)$ will be updated, corresponding to Z_{CL} and similarly $I(Z_1, Z_3) + I(Z_2, Z_4) + I(Z_1, Z_5)$ corresponding to Z_R, as we described in Chapter 3. Now consider the same observation but now with a class label C_2 assigned. Since only Z_{CL} has class variable associations, then the update will only be evident for Z_{CL}. The component Z_R will remain unchanged from the previous 'update' corresponding to class label C_1. This is to be expected as the Z_{CL} component represents the class MB and will be the only area that has an influence on the classification of the observation. As we described in Chapter 3, the MIM classifier discriminates between class labels by measuring the change in 'overall' MI in correspondence with the class MB. For the SMIM variant, instead of evaluating the information change for each of the possible 'trees' corresponding to the class label instantiations, the SMIM classifier does the same but for 'networks'. Moreover, in both cases we focus only on the class MB, and this will be applicable no matter what the topology.

The following example demonstrates that edges can be added to expand the class MB and that the same classification technique, as we proposed for the MIM classifier, can also be applied for the SMIM variant.

As was shown in Chapter 3, the training sample of the domain can be viewed as a series of class partitions characterising samples belonging to a particular class, as in Figure 5.3.

Each partition is described by a vector of class attributes $Z = \{Z_1, \ldots, Z_3\}$ and this will be the case for each class label where $C = \{C_1, C_2\}$. An instantiation of an evidence vector $\{Z_1 = 0, Z_2 = 0, Z_3 = 1\}$ in position C_1 will increase the marginal $P(C_1)$ and update the joint probabilities $P(C_1, Z)$ in respect of the evidence vector $Z = \{Z_1, \ldots, Z_3\}$ and their values, similarly, for an instantiation in position C_2.

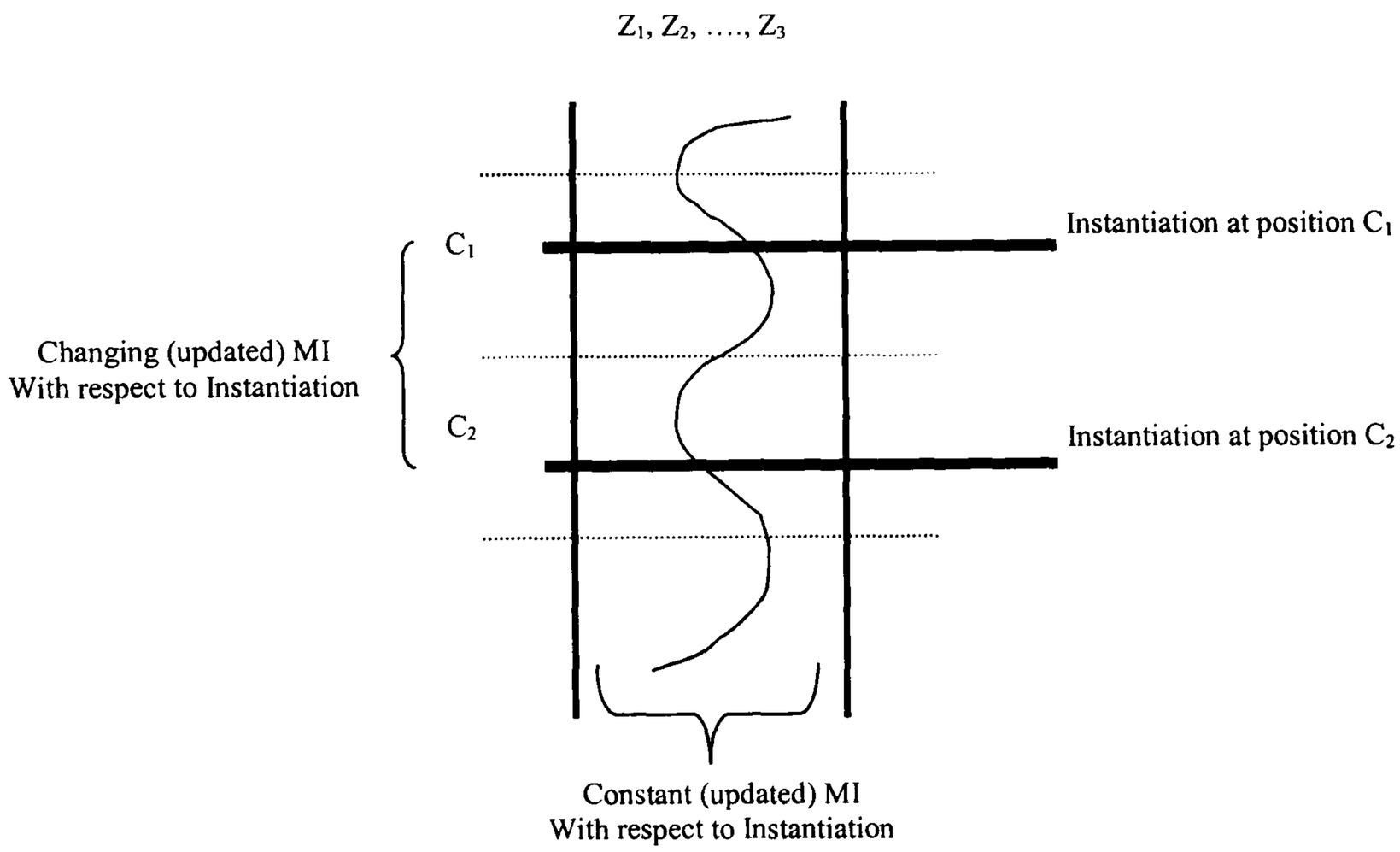


Figure 5.3. Domain Data set Representation example.

Since the evidence vector corresponding to $\{Z_1 = 0, Z_2 = 0, Z_3 = 1\}$ will be common for the C_2 instantiation position, the marginal probabilities $P(Z)$ for each value of Z, due to the evidence, will be updated but remain at a constant value.

In terms of our SMIM structure, depicted in Figure 5.4, this implies that any changes of information, that is the branch weights, due to observing an evidence vector $\{Z_1 = 0, Z_2 = 0, Z_3 = 1\}$, will only be measurable on edges that are directly associated with the class vertex.

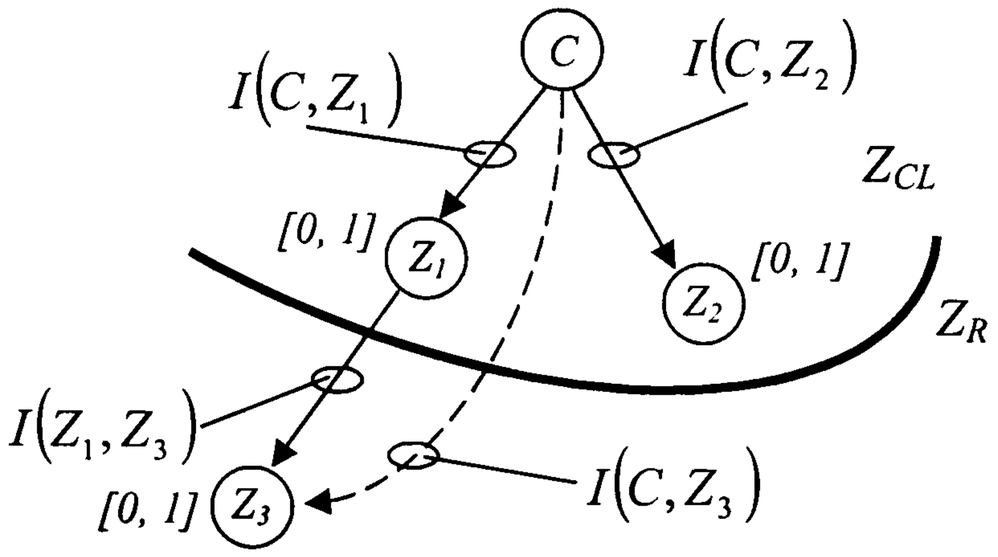


Figure 5.4. SMIM Structure Example.

The corresponding information on edges not associated with the class will remain at a constant value, for each instantiation position $C = \{C_1, C_2\}$. Figure 5.5 shows the corresponding updates in respect to the two observations, shown as case (i) and case (ii).

$Case(i)\{C_1, Z_1 = 0, Z_2 = 0, Z_3 = 1\}$
$Case(ii)\{C_2, Z_1 = 0, Z_2 = 0, Z_3 = 1\}$

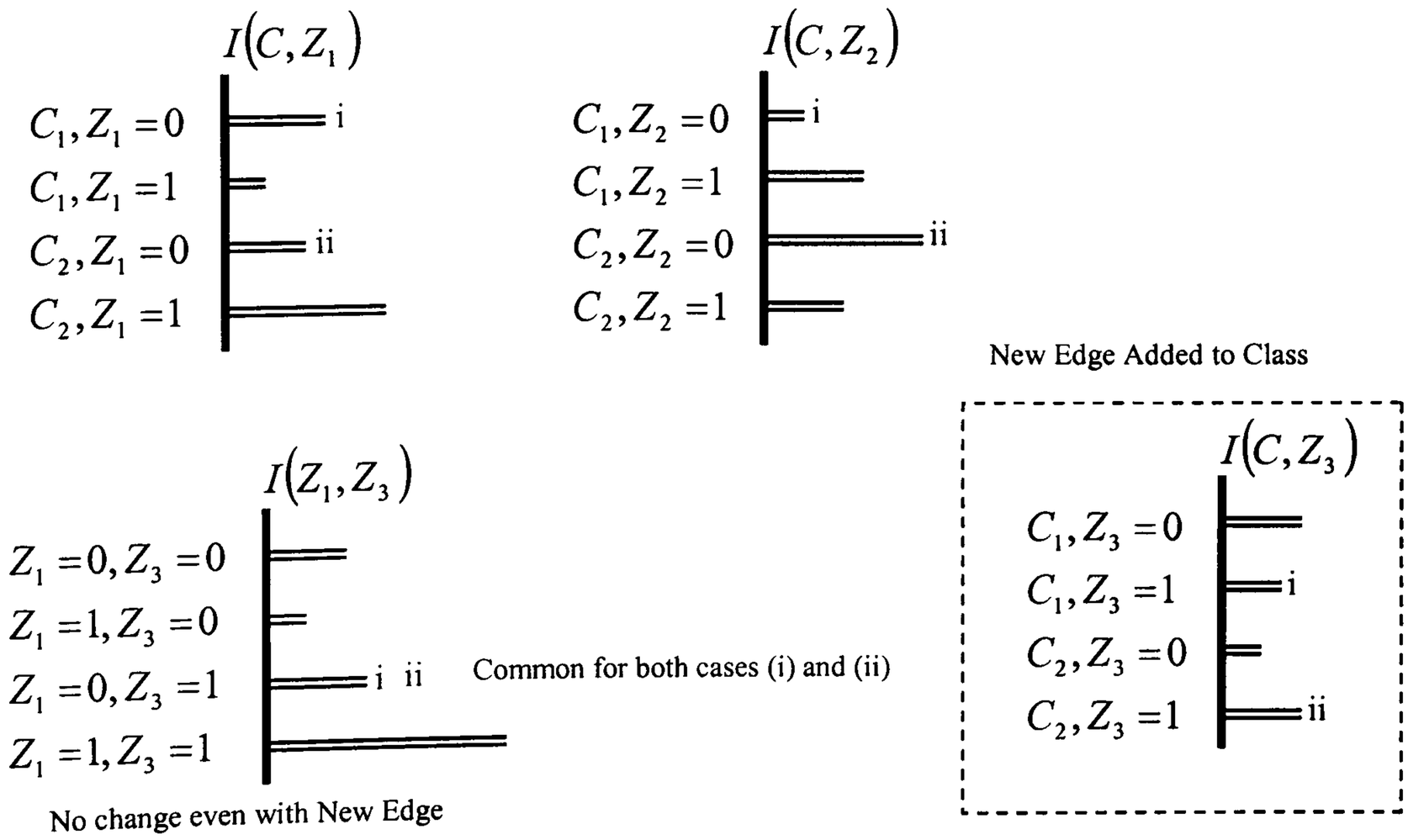


Figure 5.5. SMIM Classifier Working Example.

Thus, the classification can be achieved locally using a subset of the domain features, defined by the class MB, as was the case for the MIM classifier.

5.4 Related Work

Mutual Information has played an important role in the selection of relevant features from a domain, particularly one in which there is a high feature dimensionality. Examples can be found in [YG04, BS96, SS94, LWY04, AD02, SR+04 and KC02].

In Cheng's [CH+02] work a Filter approach discards irrelevant features on the basis of low values of MI in respect of the class variable associations, whilst, in Gammerman [GL91] structures are pruned removing irrelevant features using a Kullback thresholding technique. The method similarly targets low values of MI but unlike Cheng's approach, does not just involve the class variable associations but any feature associations. Both approaches use the CL algorithm for determining the underlying domain structure as does the MIM classifier. The problem with these approaches is that they assume the removal of features, in respect of irrelevancy, is valid for all class-states. However, for some domains this assumption may not be correct with, as Chapter 7 will show, irrelevancy only applicable for a subset of the class-states. In respect of the 'selective' MIM classifier variant, the SMIM approach in contrast does not actually remove any features but merely excludes additional class variable associations from contributing to the class MB. In general the SMIM expands the 'initial MB' or *lower bound*, with irrelevant features outside the MB being rejected. Unlike Cheng and Gammerman's approaches the SMIM selects features with the highest information (MI value) from the features that are not already part of the class MB.

Sucar [SP+97] like the MIM classifier, also utilises the CL algorithm to discover the domain structure, which similarly represents an 'initial MB' with respect to the task of classification. However, as a BN the selection choice of node ordering defining the class MB means the actual class MB may not be the same as that of the MIM classifier. Sucar selects candidate features with respect to their MI value in the same way as the selective variant of the MIM classifier, but here their inclusion is controlled by the existing branch directionality in order to maintain the DAG. The use of node ordering to control edge addition, together with a stopping criteria based on performance accuracy can however, lead to the construction of dense structures. As many branches will be introduced outside the class MB, the resulting topology may be fairly complex with large CTPs and

the consequential problems associated with the estimation of probabilities, especially where data sets are sparse. By focusing on the class MB, the SMIM's alternative approach leads to a less complex structure. Driven by the magnitude of the MI values and not node ordering the 'initial' class MB is potentially enhanced and in general will require only a marginal number of additional edges. For the task of classification, instead of updating a possible unrealistic and conceivably unreliable number of probabilities, the SMIM requires to update at most $m(n-1)$ branch weights for m class-states. Although both Sucar's approach and that of the SMIM classifier use the measure of performance as a halting criterion, the latter does not result in a possible intractable solution.

Whilst MI has been used to select edges for removal or inclusion, some researchers have considered the 'subset' of features as representing a distribution of information.

Koller [KS96] demonstrates a cross-entropy Filter approach. Essentially the cross-entropy between two distributions measures the extent of error if one distribution is substituted by another. In the case of the SMIM classifier, the 'overall' MI distribution of the 'initial MB' is compared to that of the newly increased subset (class MB + feature), with the differences implied by a performance improvement or degradation. The subset which essentially leads to the 'optimised' classifier defines the new subset or *upper bound*. For Koller's approach, the original feature set is compared to the reduced set, but here in terms of the class-entropy associated with the two feature sets.

Al-Ani [ADC03] not only showed that MI could be used in feature selection but that it performs best (for classification) when represented as a 'subset' of features. An evaluation function is used to measure how well the feature subset distinguishes between class labels by looking at the amount of information in the subset. The SMIM classifier's 'initial MB' represents a similar feature subset and by use of the MIM classifier, distinguishes between class labels by measuring the 'overall' MI content with respect to the class MB, as new features are added. Al-Ani's approach represents a Filter which only selects features. The MI is used to determine how much discriminating power exists in the choice of subset for distinguishing between class labels. The actual task of classification is performed by use of an Artificial Neural Network (ANN). In contrast the SMIM not only assesses the discrimination power but also represents the actual evaluation function. By using the branch weights in respect of the class MB, discrimination between class labels can be achieved for each sequential addition of features. Battiti [Bat94] also used MI to evaluate a set of features and thus

select the most informative subset for use as input data for a Neural Network classifier. This approach is similar to that of Al-Ani but in Battiti's implementation, just as for the SMIM, the maximal MI for only class variable associations are added, whereas Al-Ani found subsets by features that combined, looking at the amount of collective information that was contained within the subset. Whilst the notion of feature 'subset' can be demonstrated to be a viable approach to improved classification, the dependency on MI to define the subset may have a misleading effect. As we discussed in Section 3.3, feature associations may actually be due to 'commonality' rather than domain specific characterisation. As a consequence, some Z-Z branch associations may appear to have strong relationship measures, but in fact offer a poor contribution to the feature 'subset'. Although the approach taken by the SMIM classifier does alleviate this problem, by only targeting the C-Z branch associations for its equivalent feature subset (the class MB), it has the unfortunate drawback of not being able to guarantee to find the 'optimal' subset.

5.5 Summary

In this chapter, we proposed a 'selective' MIM classifier (SMIM) based on the expansion of the class MB. We suggested that the 'initial MB' derived from the CL algorithm represented a *lower bound* and thus could potentially be modified for improved performance. In section 5.3, we showed that the MIM classifier could be used as the evaluation function in a Wrapper strategy without modification to the original algorithm defined in Chapter 3. Despite the computational expense of this approach and its difficulty in determining an 'optimal' solution, it does have the advantage of being able to incorporate any potential bias associated with the MIM classifier during feature selection, and thus offer an improvement to the classifier. In the next chapter, we evaluate the SMIM classifier using the UCI benchmark databases previously studied in Chapter 4, and demonstrate that in some cases, performance of the MIM classifier can be improved.

Chapter 6

Evaluation – Selective MIM Classifier

Chapter 4, section 4.4, identified some drawbacks concerning the class MB and the qualitative measure of the CL algorithm derived 'tree' structure. In Chapter 5, we proposed a 'selective' variant of the MIM classifier which focussed on the expansion of the class MB in order to address these drawbacks. In this chapter, we evaluate the performance of the selective MIM classifier (SMIM) by carrying out a series of experiments on the same benchmark databases as detailed in Chapter 4, section 4.2. In the next section, we describe our main objectives and aims with section 6.2 and section 6.3, describing the experimental methodology and design respectively. In section 6.4, the results of comparing the SMIM classifier to a selective variant of NB (SNB), together with the previously generated models described in Chapter 4, are reviewed. In section 6.5, we consider some implementations of the results with section 6.5 reviewing the contributions and section 6.6 summarising the chapter.

6.1 Objectives

Feature selection has been widely used to deal with domain complexity, with many approaches achieving performance levels that even exceed non-selective methods, as discussed in section 5.1. In Chapter 4, section 4.7, we considered the possibility that the implied feature selection of the 'initial' MB, derived by use of the CL algorithm, may not be the best for the task of classification, thus limiting its performance capability. Our first experimental objective was therefore to determine whether the expansion of the class MB, using a Wrapper approach, could improve the performance of the MIM classifier. Moreover, we are interested in determining whether the initial MB assumptions discussed in Chapter 5, section 5.2 are correct. With the advent of more features defining the MB, our second objective was to assess the consequential cost for any performance gain, in terms of model complexity.

From the results obtained in Chapter 4, we observed the effect of high dimensional domains on small data sets. For the SMIM classifier this issue is of particular importance. Since the MIM classifier

class MB is to be expanded, it will characterise high dimensionality and may have problems dealing with small data sets, for which the previous MIM classifier did not. To investigate this aspect we carried out comparisons of the selective and non-selective variants performance as a function of the sample size for learning the classifiers. As the SMIM transforms the 'tree' into a 'network', our comparisons also include the GBN.

In Chapter 4, we compared the MIM classifier performance with the simple and efficient NB classifier. However, NB's assumption of extreme CI is known to be influenced by features that are highly correlated [Paz96]. Langley [LS94] proposed a selective variant of the NB to deal with this by trying to improve the NB performance by removing the highly correlated edges from the class MB. This approach (SNB), in contrast to the SMIM, reduces the features in the class MB and therefore represents a even simpler and potentially 'optimised' model of the domain. Our final objective was thus to compare the performance of the selective MIM classifier to that of the selective NB, and to additionally determine if they portray any 'similar' characteristics.

6.2 Experimental Methodology

For these experiments we used the same UCI benchmark databases [MA95, BM00] as detailed in Chapter 4, Table 4.1. Missing and continuous features were dealt with in the same way as described in section 4.3, that is, during the evaluation of the non-selective variants.

The SMIM classifiers were constructed as defined in Chapter 5, section 5.2 and 5.3. In the case of the SNB, we used the implementation provided by the utility MLC++ [KJ+94].

6.3 Experimental Design

For the experiments described in section 4.4, we evaluated the larger datasets by applying a hold-out technique, whilst for the smaller ones a cross-validation with 5-folds. In order to construct the 'selective' MIM classifier we adopted a similar method to that used in Chapter 4, but for these experiments we made a slight modification. In the case of the larger data sets we further divided the 2/3 'learn' partition into two sub-partitions. A sub-learn comprising 2/3 of the learn partition and a 1/3 sub-test partition similar to the original hold-out approach, but in this instance, an 'internal' partitioning as depicted by Figure 6.1.

The procedure initially applies the SMIM algorithm to the two sub-partitions (sub-learn, sub-test) to discover the 'optimised' SMIM feature subset. Once established the new structure, along with its

corresponding branch 'weight' assignment, is evaluated to determine its performance accuracy by use of the original 'test' partition. Branch 'weights were calculated using the whole 'learn' partition, that is, the combination of the two sub-partitions, once the final feature subset had been identified.

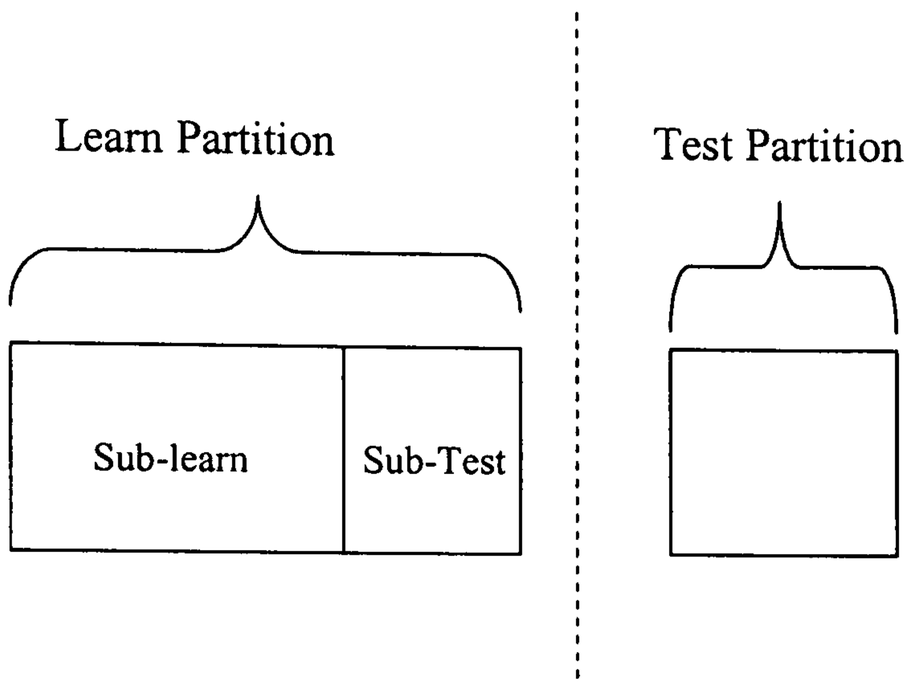


Figure 6.1. Data set partitioning technique – Hold-out example.

For each of the two selective classifiers, the structure was learned/constructed using the training data set (learn partition) and the classifier accuracy determined on the test data set.

Just as we carried out in section 4.4, this process was repeated over a series of 25 trial runs in order to gain a sample average together with the standard deviation for the predictive accuracy using the test partition. The statistical significance of the differences in classification accuracy was measured using an Analysis of Variance (one-way ANOVA). To further determine all pair-wise differences, that is, the magnitude and direction between each pair of methods being compared, we followed ANOVA by Post Hoc Tukey comparisons with an overall confidence level 95%. As for Chapter 4, prior to applying ANOVA we first established the validity of the assumptions.

The classification accuracy was determined as a percentage of the test cases that identified the correct class. For the smaller data sets we used cross-validation with 5-folds instead of the hold-out technique and similarly repeated the process for 25 trial runs. For each data set studied we applied a stratified distribution as we did in Chapter 4, to minimise bias due to the effects of differing class sample sizes.

To measure the rate of improvement of the two classifiers, we conducted experiments similar to those defined in Chapter 4, concerning the accuracy measurement for different quantities of the learning cases. In order to evaluate the effect of consequential MB expansion and thus

dimensionality, we compared the results of the selective models to the learn rates of the previous non-selective models, including the GBN and 'polytree'.

In Chapter 5, section 5.2, we hypothesised the possibility of local minima occurring during the feature subset selection process. During our preliminary experimental work we observed this to be a real anomaly. Figure 6.2 shows the performance accuracy measured against the number of features within the class MB for the data set 'Vehicle'. The initial performance level is the *lower bound* and is defined by the CL 'tree' MB. Subsequent performance is in respect of the addition of features to the *lower bound* class MB, as described in Chapter 5, section 5.2.

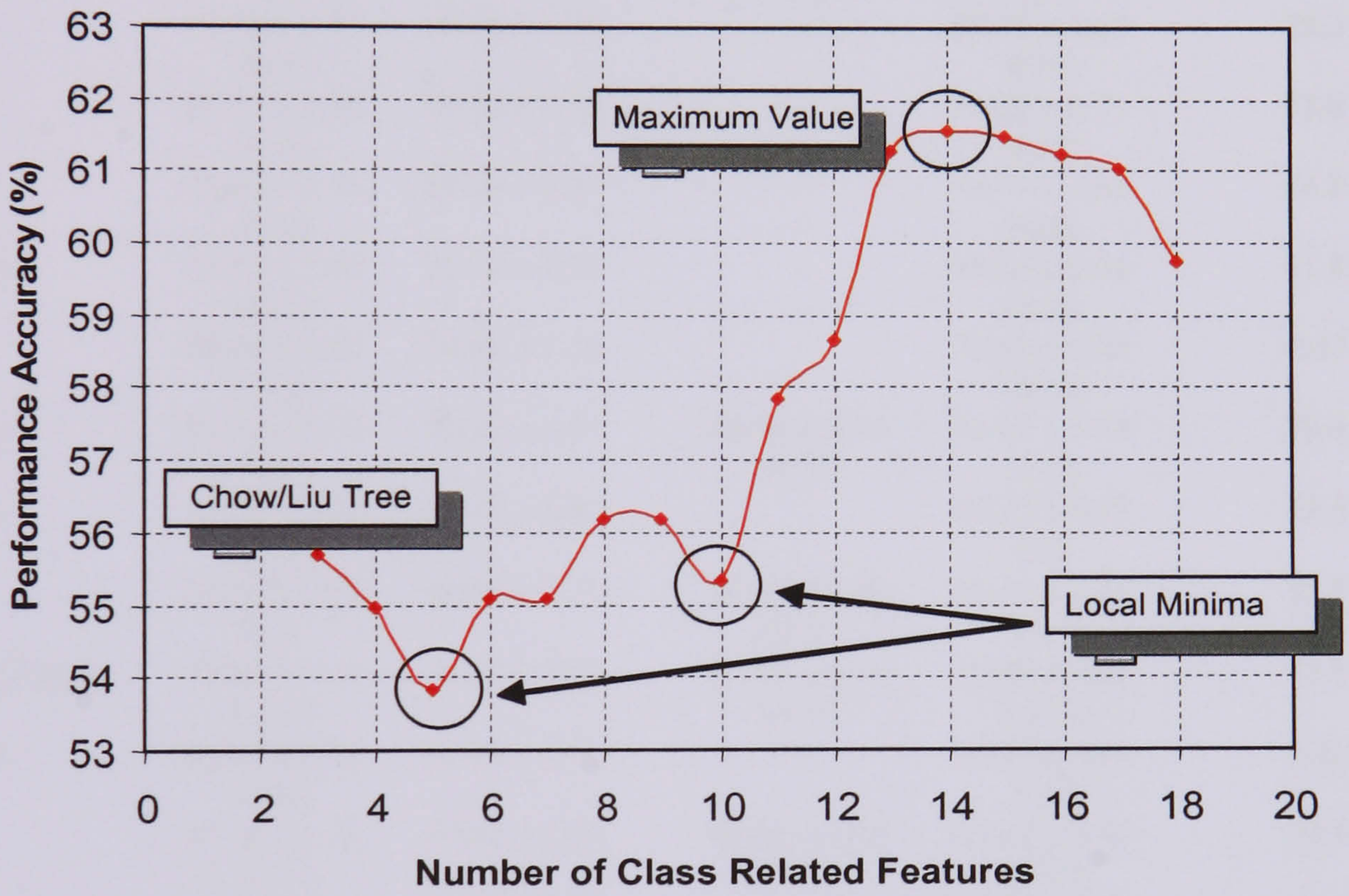


Figure 6.2. SMIM 'Local' Minimums for data set 'Vehicle'.

Figure 6.2 displays two local minima with a steep improvement of performance observed once the class MB contains eleven features, whereby the 'optimised' level is attained. Also shown are local maxima occurring with feature additions eight and nine. As this characteristic is possible to occur for all data sets under investigation, the examination in respect of all feature additions (outside the class MB) implies a potentially intractable exhaustive search. However, as defined in section 5.2, we applied the two heuristic 'stopping criteria' for each data sets studied in order to avoid this scenario.

6.4 Experimental Results

The average predictive accuracies taken over 25 runs for the two 'selective' classifiers are shown in Table 6.1. Each entry describes the average accuracy along with the sample standard deviation for predictive accuracy variations from sample to sample.

Table 6.1: Average Predictive Accuracy

DB Name	MIM	NB	SMIM	SNB	Default (overall)
Vehicle	55.66 ± 1.51 3(18)	58.28 ± 1.79	**61.53 ± 0.91** 14(18)	61.36 ± 2.96 10(18)	25.8
DNA	***95.58 ± 0.42*** 15(60)	94.97 ± 0.29	***95.80 ± 0.36*** 19(60)	93.59 ± 0.71 23(60)	51.9
Car_Evaluation	86.11 ± 0.74 5(6)	***86.58 ± 1.78***	***86.43 ± 0.85*** 6(6)	85.01 ± 0.83 6(6)	70.0
Flare	82.93 ± 1.26 2(10)	80.99 ± 1.28	- -	**83.40 ± 1.67** 3(10)	79.2
Chess	**96.27 ± 3.56** 6(36)	87.34 ± 1.02	- -	94.28 ± 0.71 9(36)	52.0
Vote	**95.40 ± 2.41** 2(16)	89.89 ± 5.29	- -	94.71 ± 1.63 3(16)	54.8
Mushroom	98.56 ± 1.06 1(22)	95.79 ± 0.39	- -	**99.96 ± 0.04** 8(22)	51.8
Letter	**80.26 ± 0.37** 11(16)	74.96 ± 1.10	- -	78.82 ± 0.52 12(16)	4.07
Hepatitis	84.00 ± 7.22 3(19)	81.20 ± 3.70	**85.76 ± 7.14** 5(19)	82.69 ± 5.29 9(19)	79.4
Nursery	95.78 ± 0.30 8(8)	94.76 ± 0.45	- -	**96.33 ± 0.27** 8(8)	33.3
CRX	85.00 ± 0.52 6(15)	86.60 ± 0.71	**87.90 ± 0.90** 12(15)	85.22 ± 1.10 8(15)	55.5
Soybean_Large	91.29 ± 0.10 33(35)	90.78 ± 0.72	91.54 ± 0.10 34(35)	**92.08 ± 2.01** 24(35)	13.5
Segment	**94.49 ± 0.64** 5(19)	91.95 ± 1.10	- -	93.25 ± 0.82 7(19)	4.80
Vote1	88.51 ± 1.90 4(15)	87.60 ± 2.10	**90.15 ± 1.53** 5(15)	89.34 ± 2.70 5(15)	61.4
Cars	***99.23 ± 0.70*** 1(8)	98.98 ± 0.47	- -	***99.17 ± 1.08*** 1(8)	62.5
Austria	85.07 ± 0.91 6(14)	86.38 ± 1.10	**87.10 ± 1.65** 11(14)	86.52 ± 2.17 7(14)	55.5
Heart	85.83 ± 2.10 6(13)	85.00 ± 1.13	**86.39 ± 0.89** 10(13)	85.28± 2.50 7(13)	55.6
Promoter	**87.97 ± 1.31** 4(57)	82.00 ± 2.02	- -	86.89 ± 1.30 3(57)	50.0
Glass	**69.37 ± 2.08** 7(9)	68.31 ± 1.98	- -	66.20 ± 1.99 6(9)	35.5
Ann-Thyroid	97.17 ± 0.08 2(21)	99.11 ± 0.31	98.27 ± 0.26 16(21)	**99.30 ± 0.25** 11(21)	92.6

Key: MIM – Mutual Information Measure Classifier, **NB** – Naive Bayes Classifier, **SNB** – selective Naive Bayes Classifier, **SMIM** – Selective Mutual Information Measure Classifier.

Values in **bold** type indicate the highest model performance achieved by the classifier in respect of each database. ***Bold italic*** values highlight performance levels that are close to the highest level achieved.

For the purpose of comparison the corresponding results for the previous non-selective variants, namely the MIM classifier and NB, have also been included (previously recorded in Table 4.2, Chapter 4) along with the data sets 'overall' default values.

Table 6.1 also includes the number of relevant features, which are those defined by the class MB, for each classifier's 'optimised' MB. These are shown as items under the performance levels i.e. 3(18) depicts 3 features in the class MB from a maximum of 18 domain features, shown in brackets. The items marked with (-) indicate that there was no performance improvement for the MIM classifier and thus the performance for the SMIM corresponds to the *lower bound*.

In the following sections, we review the results obtained from classifying the samples taken from the databases studied using the SMIM classifier and the SNB. For completeness comparisons have been made between 'selective' and 'non-selective' variants, as recoded in Chapter 4, as well as the selective variants themselves.

The objectives of our experiments, discussed in section 6.1, focus on the structure's class MB. We consider this important because classification of a domain is related to this feature subset and the selective variants modify it either by expansion or reduction.

- **Comparison of Selective and Non-selective 'tree' based classifiers**

From the results shown in Table 6.1, we observed that the SNB improved the performance of NB for all high dimensional domains with features greater than fifteen, presumably due to the removal of those that were highly correlated. This was demonstrated by the data sets 'Mushroom', 'Chess', 'Hepatitis', 'DNA', 'Vehicle', 'Soybean_Large', 'Segment', 'Promoter', and 'ANN'. This observation was also apparent for the SMIM which improved the MIM classifier by expanding the class MB, and observed for the data sets 'CRX', 'Hepatitis', 'DNA', 'Vehicle', 'Soybean_Large', 'Vote1', and 'ANN'.

Figure 6.3 and Figure 6.4 show the error rates of the SMIM compared to the MIM and SNB compared to NB for the 20 data sets studied.

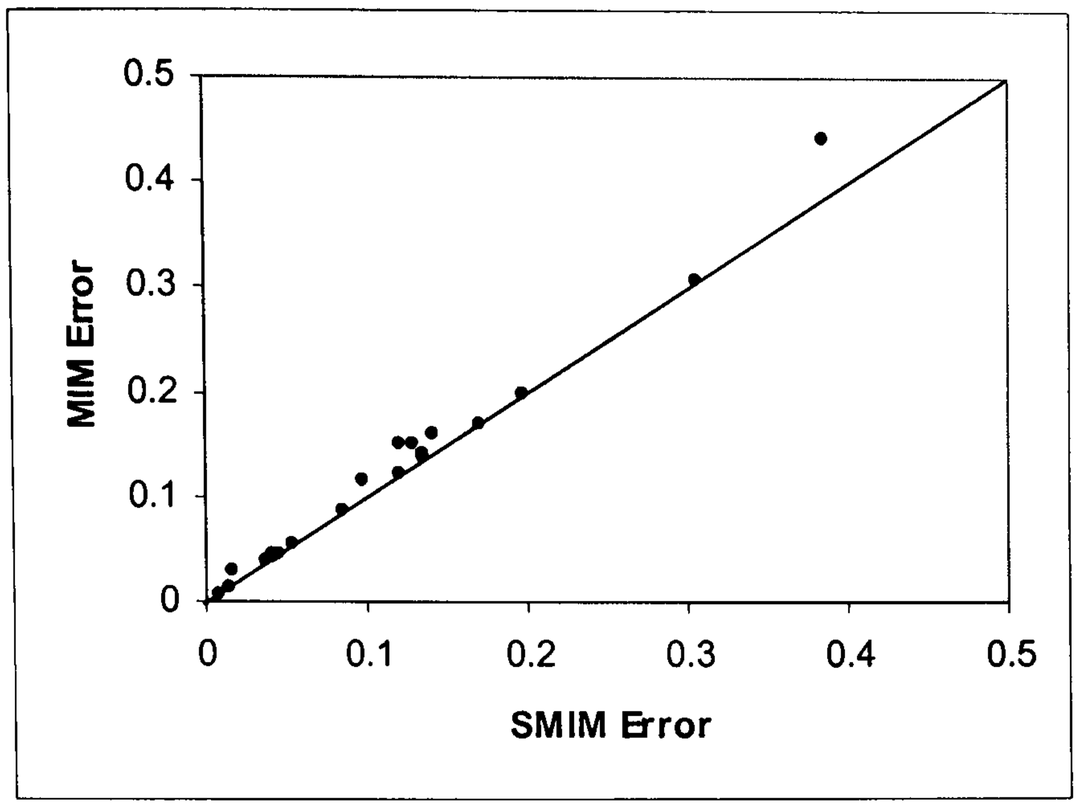


Figure 6.3. Scatter Plots Comparing Error Rates of SMIM with MIM

Note: As some of the data sets resulted in no change from applying the SMIM algorithm to the MIM classifier, error points will thus appear on the diagonal line.

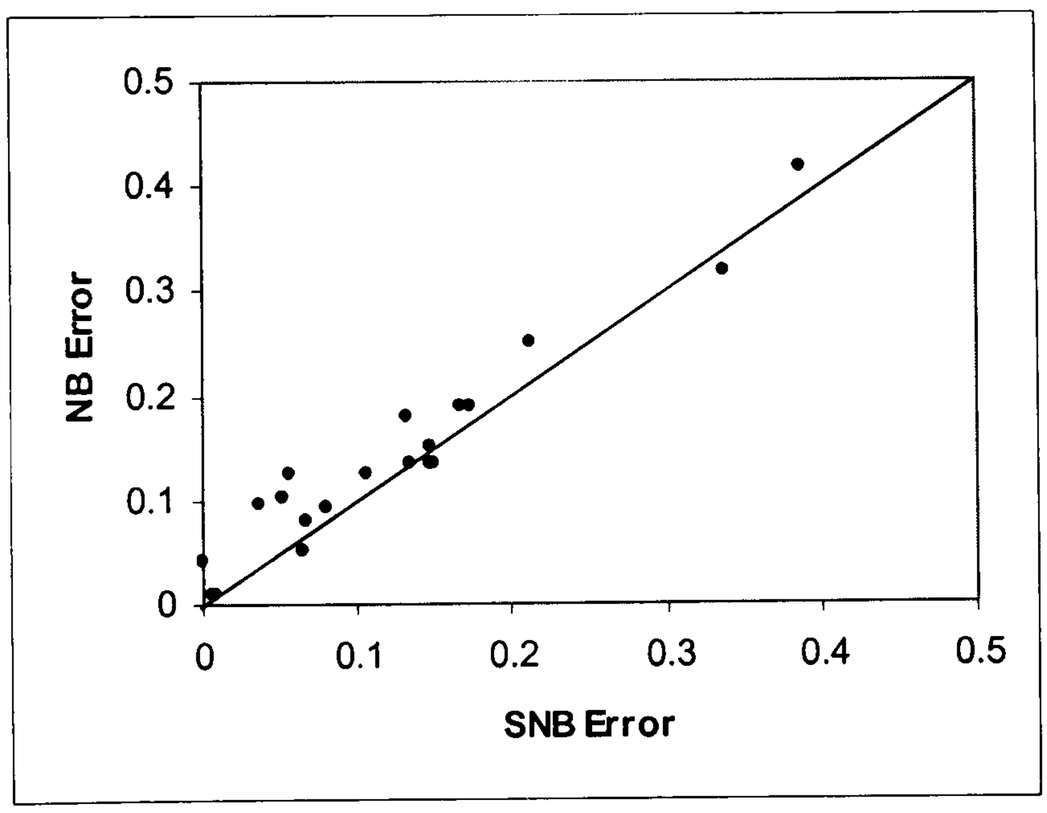


Figure 6.4. Scatter Plots Comparing Error Rates of SNB with NB

In both cases the selective variant improved performance, with SMIM and SNB having error rates less than MIM and NB. Figure 6.5 and Figure 6.6 display the corresponding differences in accuracies.

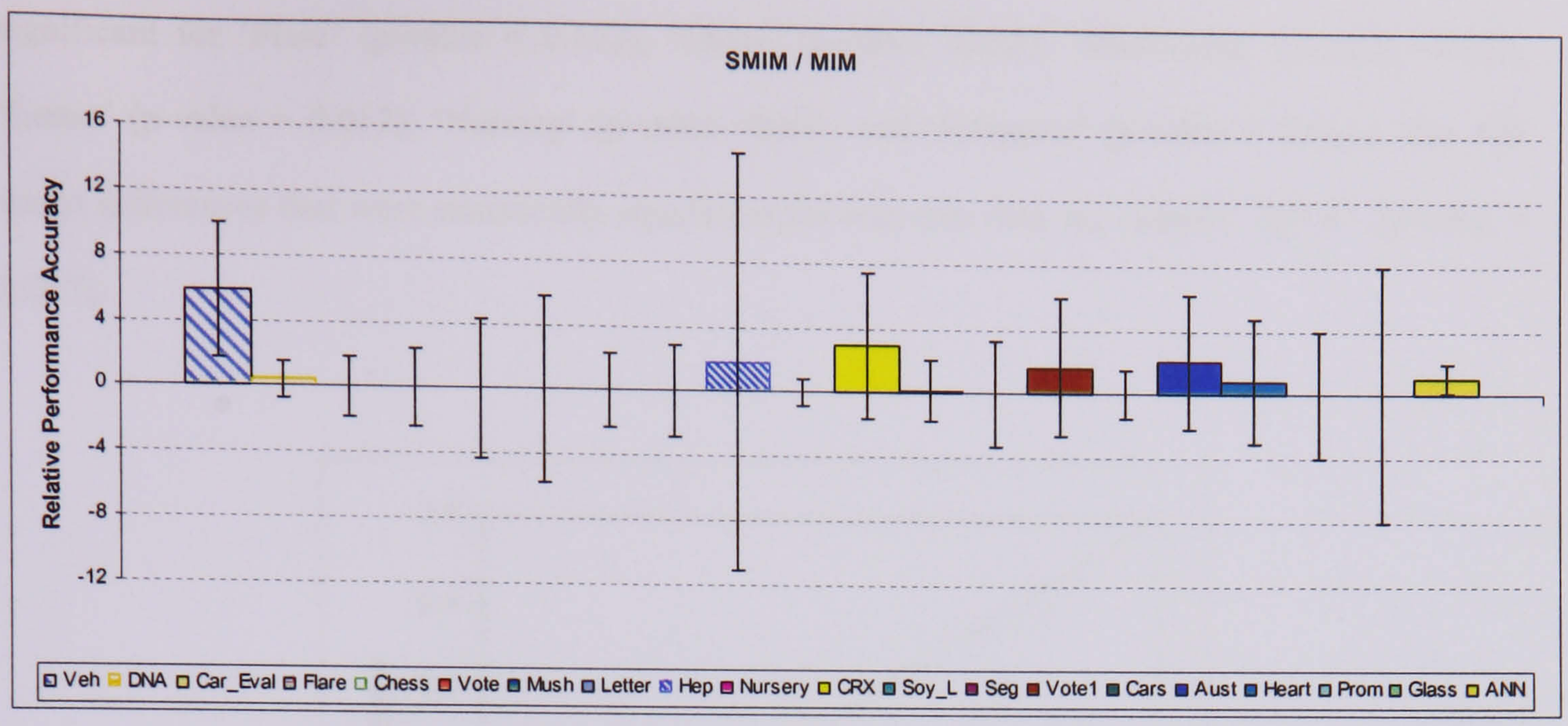


Figure 6.5. Predictive Accuracy relative to SMIM Classifier (MIM)

Note: As some of the data sets resulted in no change from applying SMIM algorithm to the MIM classifier no differences will be illustrated.

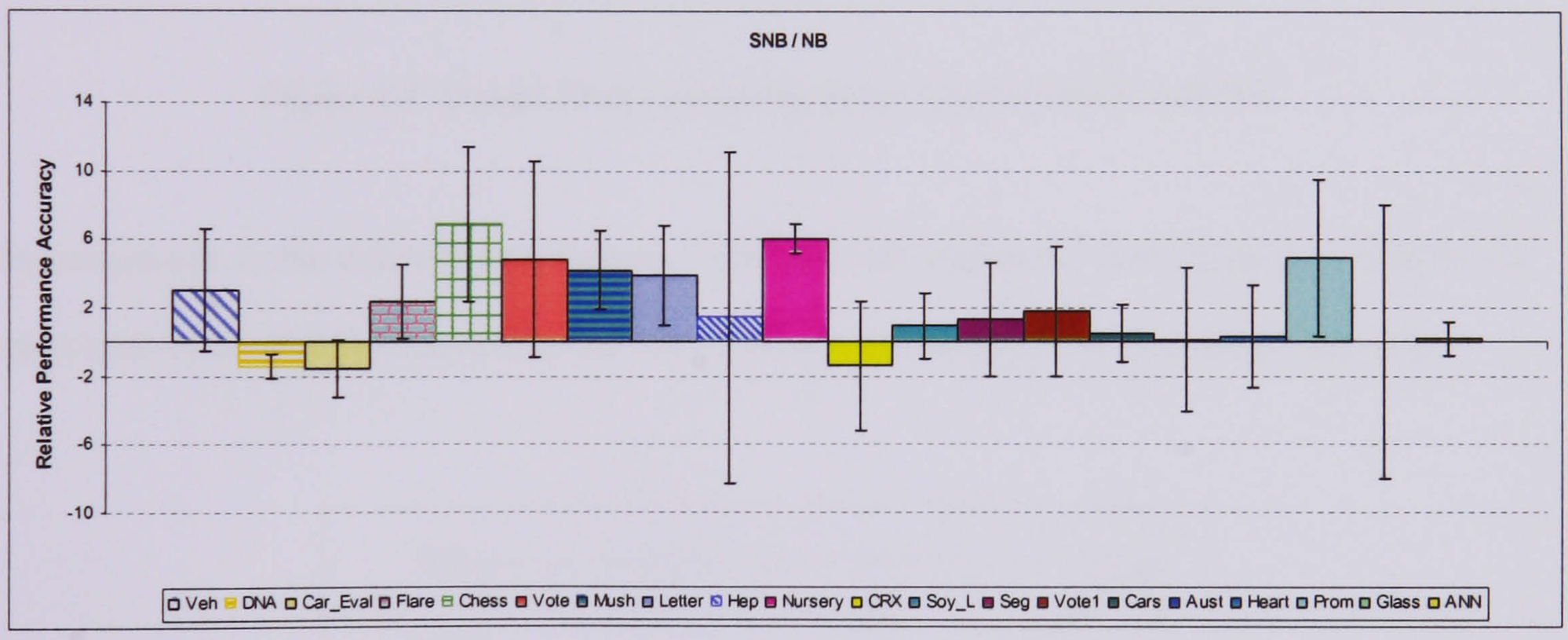


Figure 6.6. Predictive Accuracy relative to SNB Classifier (NB).

As was the case in Chapter 4, a positive value for an algorithm indicates the SMIM or SNB performed better for that particular data set, with the error bars representing the Post Hoc Tukey comparisons with overall 95% confidence for the relative differences. From Table 6.1 SMIM improved the performance of the MIM classifier for ten data sets with differences found to be statistically significant for 'Vehicle' (p-value = 0.005) and 'ANN' (p-value = 0.007). In contrast, SNB improved the performance of NB for sixteen data sets with differences that were statistically

significant for 'Flare' (p-value = 0.022), 'Chess' (p-value <0.05), 'Mushroom' (p-value <0.05), 'Letter' (p-value = 0.012), 'Nursery' (p-value <0.05), and 'Promoter' (p-value = 0.041). The NB found differences that were statistically significant for only one data set, namely, 'DNA' (p-value = 0.017).

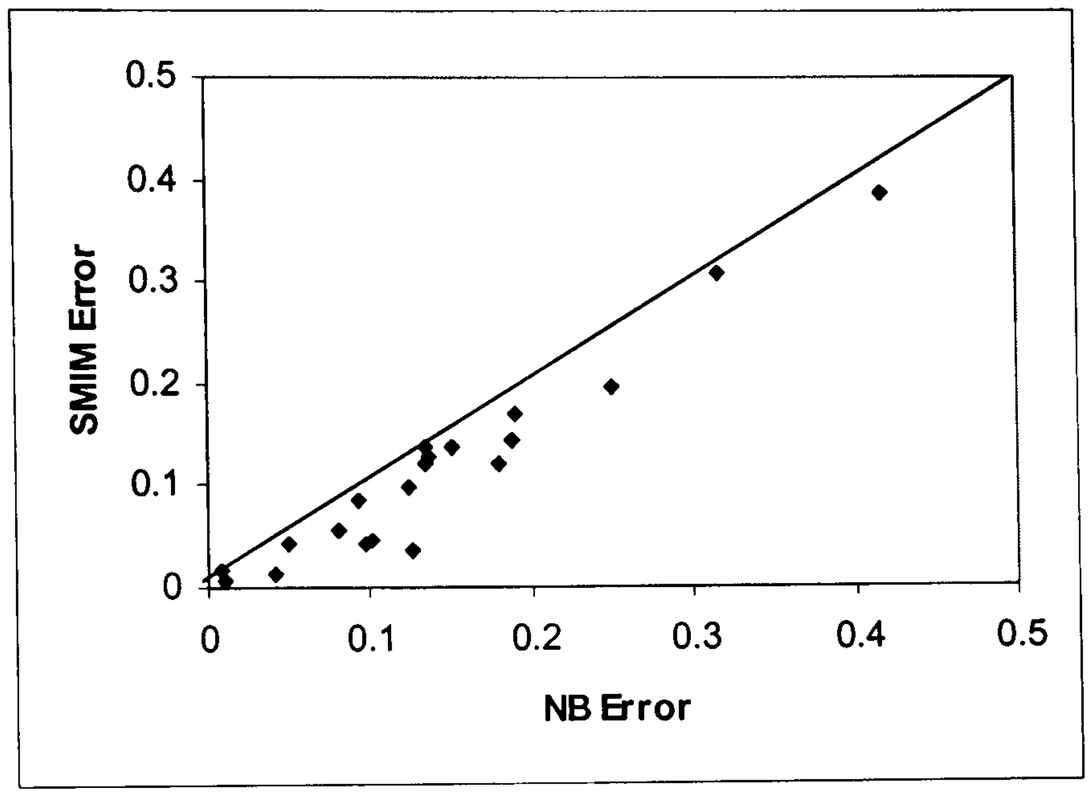


Figure 6.7. Scatter Plots Comparing Error Rates of SMIM with NB

In comparison to the non-selective variants, MIM and NB, Figure 6.7 and Figure 6.8 illustrate the error rates of the SMIM compared to NB and SNB compared to MIM for the same 20 data sets.

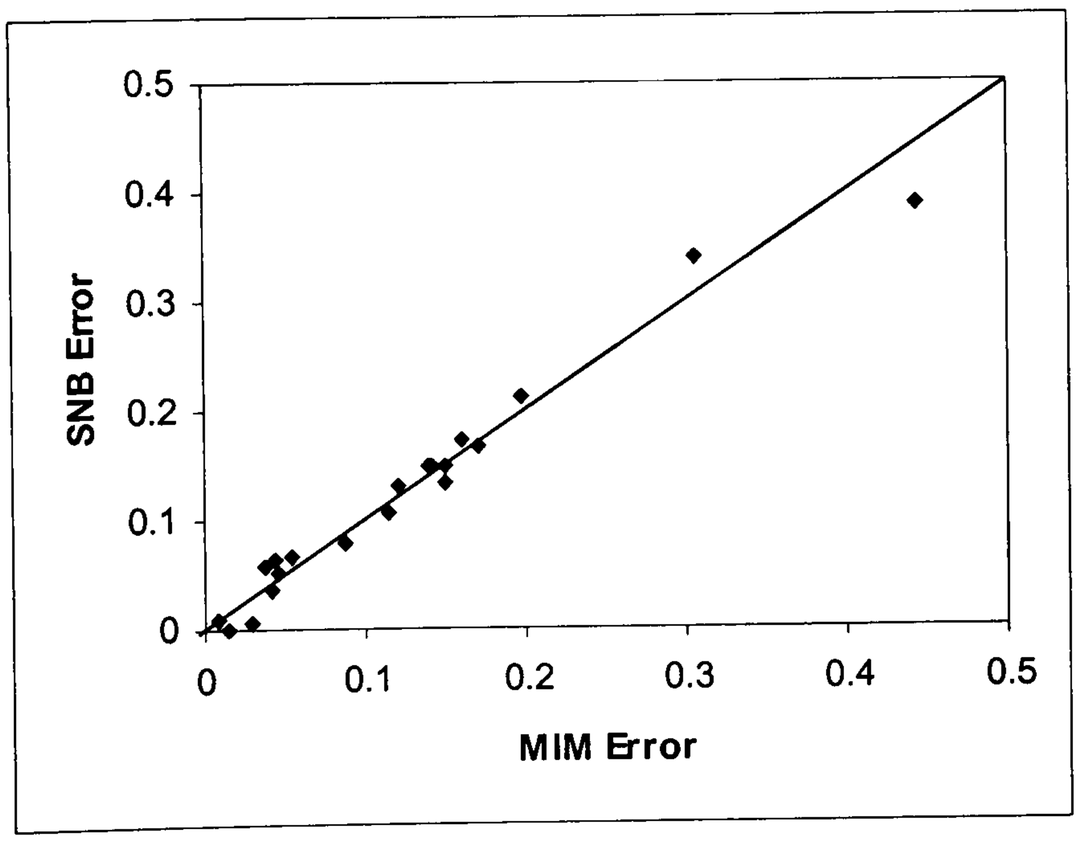


Figure 6.8. Scatter Plots Comparing Error Rates of SNB with MIM

The SMIM shows an error rate less than that of NB, whilst the error rate for the SNB compared to the MIM classifier are similar, with clustering on the diagonal line evident. The corresponding differences in accuracies are displayed in Figure 6.9 and Figure 6.10.

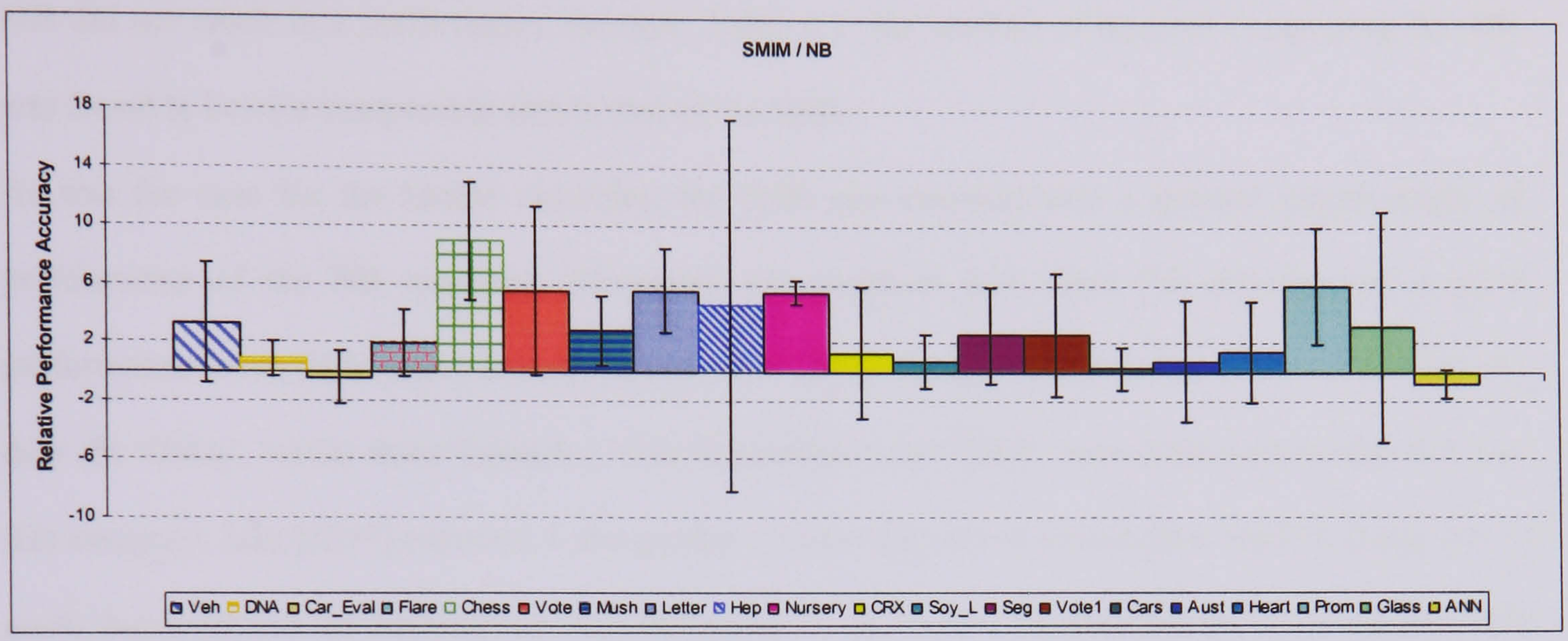


Figure 6.9. Predictive Accuracy relative to SMIM Classifier (NB).

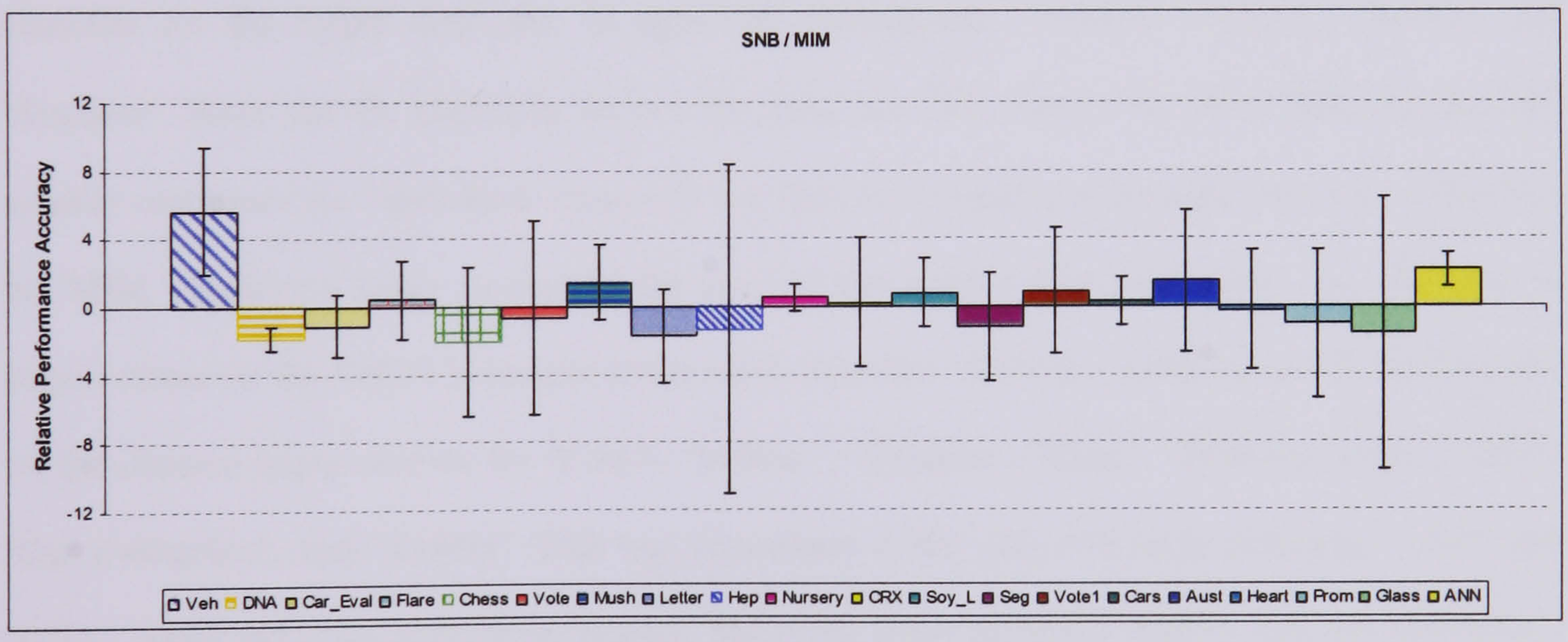


Figure 6.10. Predictive Accuracy relative to SNB Classifier (MIM).

For the SMIM compared to NB, differences found to be statistically significant were observed in data sets 'Chess' (p-value <0.05), 'Mushroom' (p-value = 0.021), 'Letter' (p-value = 0.002), 'Nursery' (p-value <0.05), and 'Promoter' (p-value = 0.018), whilst for SNB compared to MIM only 'Vehicle' (p-value = 0.002) and 'ANN' (p-value = 0.003) were found to have differences that were

statistically significant. In the case of the MIM classifier there was only one data set that had differences that were found to be statistically significant, namely, 'DNA' (p-value = 0.026).

In general, we observed that the SMIM not only improved the performance of the MIM classifier but also was better than that previously achieved by NB. Where the addition of features to the MIM class MB did not result in a performance increase, Table 6.1, the number of features comprising the MB was found to be of a comparable size to that of the SNB.

As was the case for the SMIM classifier, the SNB also demonstrated a general improvement of performance of the NB classifier. Moreover, we observed that when NB already had a good performance level, SNB improvements to NB were small. Langley [LS94] also observed this for the data set 'DNA', whilst from Table 6.1 'Car_Evaluation' and 'CRX' were observed to also fall into this category. Jain [JZ97] reported "*..the quality of selected feature subsets for small training sets is poor, but improves as training set size increases...*". In respect of the class MB, as we are only interested in the task of classification here, we observed a similar trend. For the SMIM, we observed from the results in Table 6.1, that there were generally no improvements in performance for the MIM classifier for the larger data sets. In particular 'Mushroom', 'Nursery', 'Chess', 'Letter', and 'Segment'. Since the CL algorithm defines the class MB, this implies for these data sets, the MB actually represents the 'optimised' class MB and thus the original performance levels as defined by the MIM classifier's *lower bound*. In the case of the smaller data sets, there was evidence of improvement for the SMIM in respect of the MIM classifier, Table 6.1. Although not in all data sets, we did observe improvements for 'Vote1', 'Vehicle', 'Hepatitis', 'Heart', 'Soybean_Large', 'CRX', 'Car_Evaluation', and 'Austria'. This was in contrast to the only two large data sets, 'ANN' and 'DNA', which did show some improvement. Moreover, these particular smaller data sets showed the greatest gain in performance accuracy. Expanding the class MB for smaller data sets implies that the issues stated by Jain [JZ97] are being addressed by a form of compensation. Since the class MB is defined by the CL algorithm, any constraints due to the domain characteristics have been overcome by the addition of more relevant features to the MB, leading to an improved performance.

For completeness, Figure 6.11 and Figure 6.12 illustrate the error rates of the SMIM compared to the 'polytree' and the SNB compared to the 'polytree'. The SMIM has error rates less than those of the 'polytree' which is similar to that observed for the MIM classifier in Chapter 4.

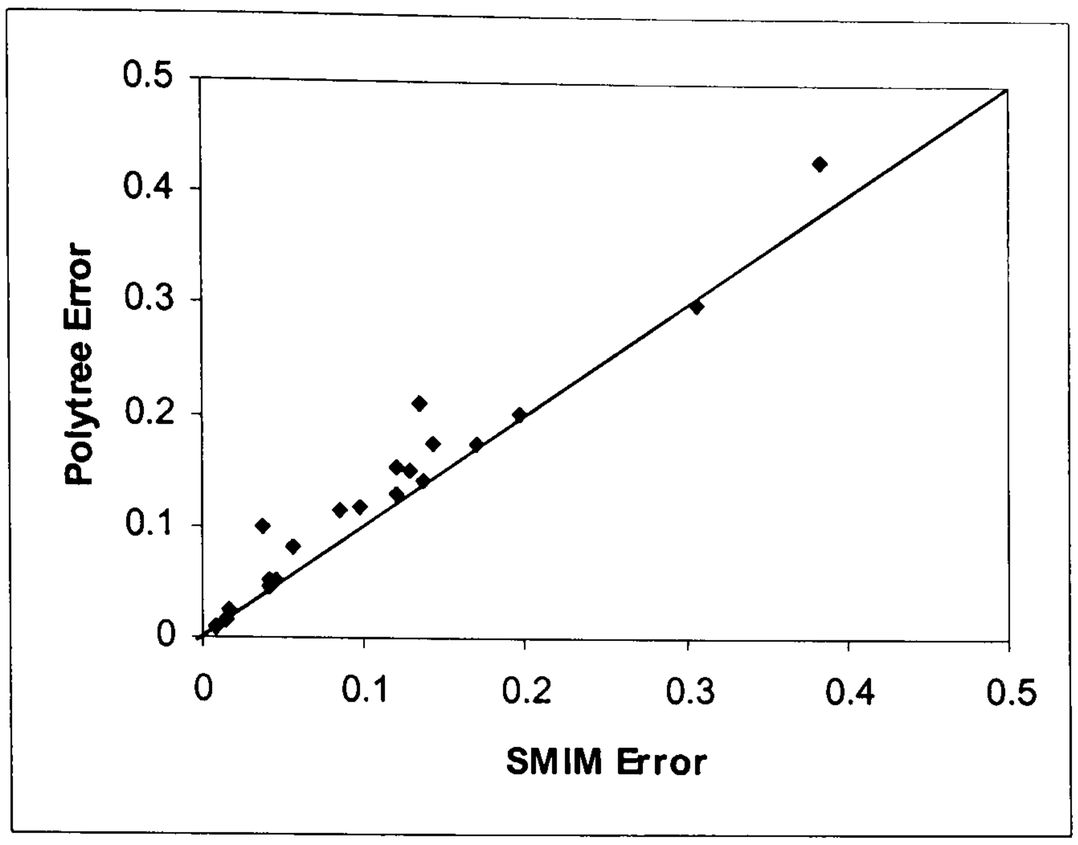


Figure 6.11. Scatter Plots Comparing Error Rates of SMIM with Polytree

In the case of the SNB, the error rate is generally lower than that of the 'polytree', however compared to the previous results of Chapter 4, there is an improvement. In Chapter 4, Figure 4.11, the NB shows a higher error rate than 'polytree', whereas SNB demonstrates an improved performance previously achieved by NB, which effectively reverses this result.

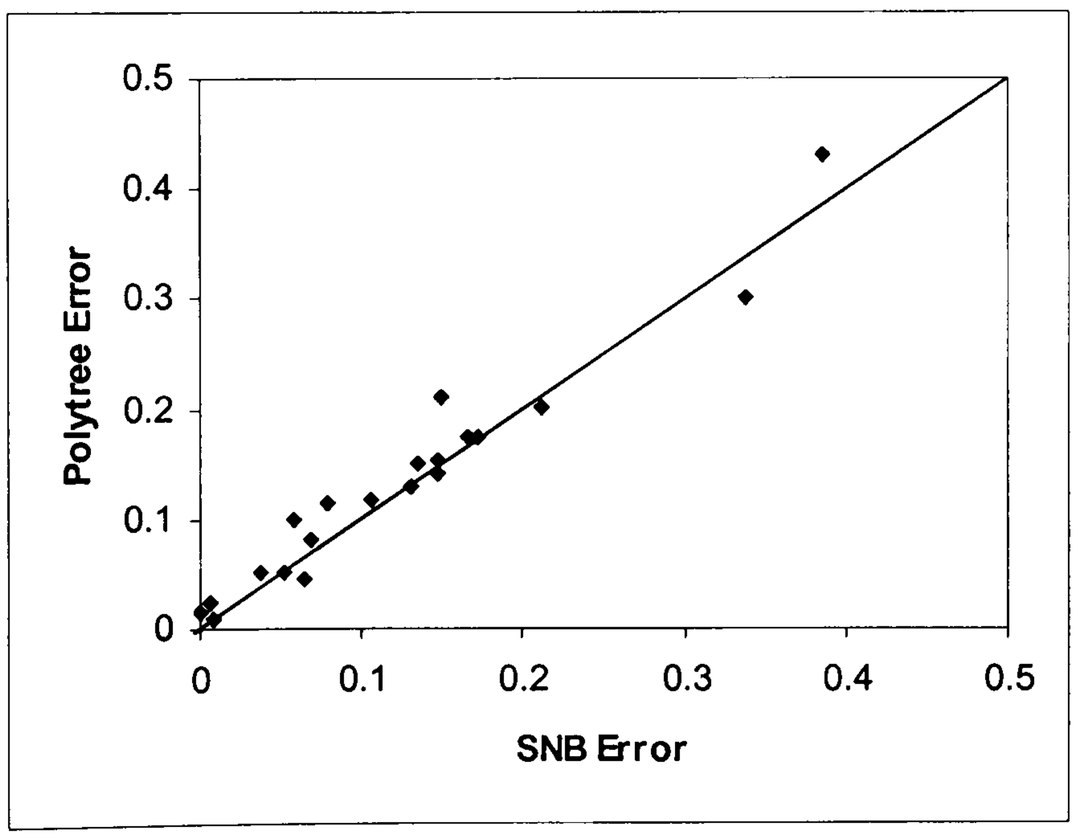


Figure 6.12. Scatter Plots Comparing Error Rates of SNB with Polytree

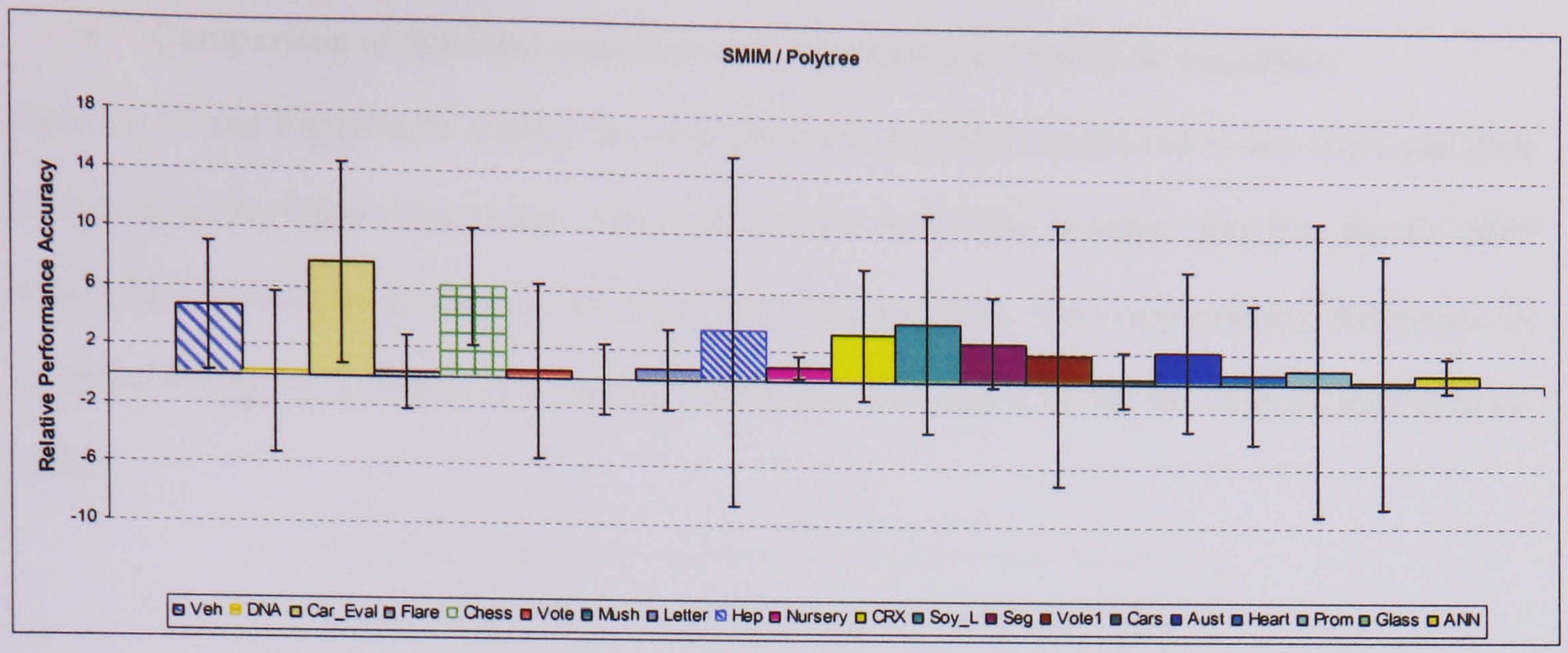


Figure 6.13. Predictive Accuracy relative to SMIM Classifier (poly).

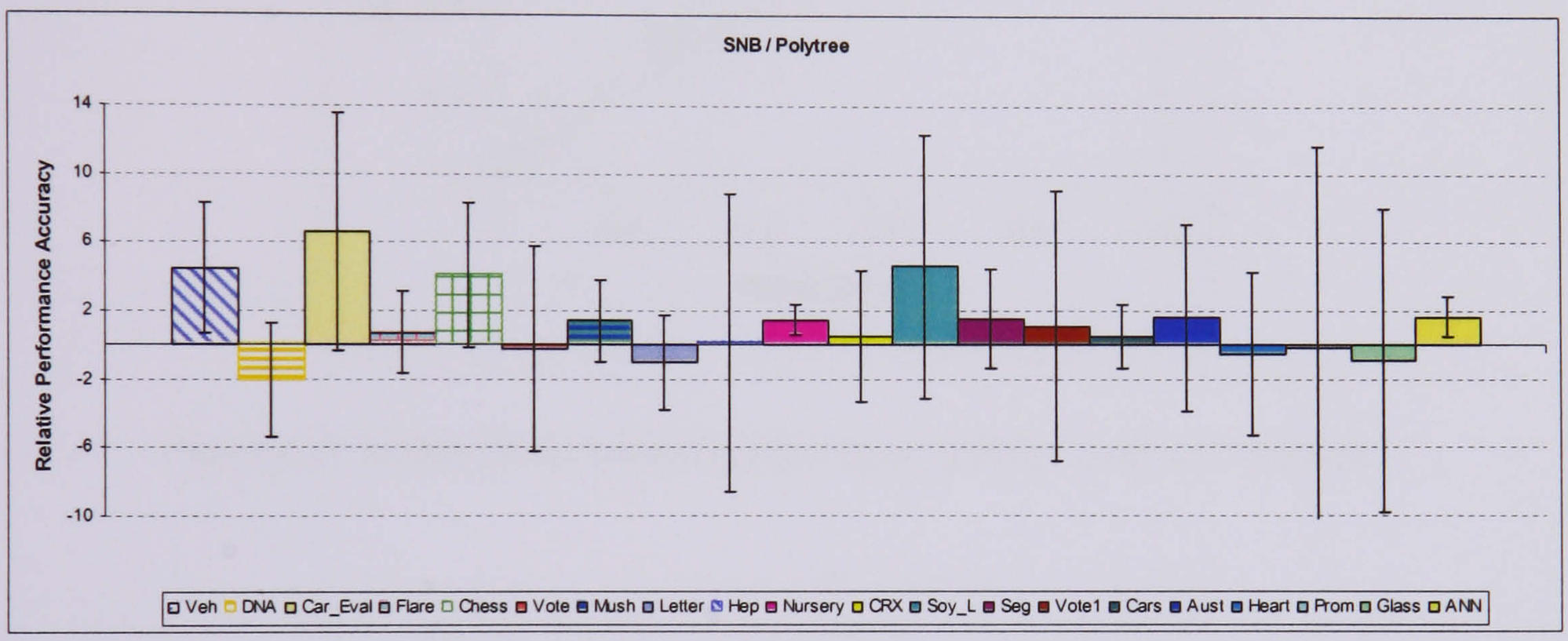


Figure 6.14. Predictive Accuracy relative to SNB Classifier (poly).

The differences in accuracies are shown in Figure 6.13 and Figure 6.14. For the SMIM there were differences that were found to be statistically significant compared to 'polytree' for 'Vehicle' (p-value = 0.036), 'Car_Evaluation' (p-value = 0.022), 'Chess' (p-value = 0.001), 'Nursery' (p-value = 0.037), with 'Vehicle' a new addition compared to the previous MIM and 'polytree' comparisons, as illustrated in Figure 4.16. In the case of SNB compared to 'polytree', differences were found to be statistically significant for data sets 'Vehicle' (p-value = 0.017), 'Nursery' (p-value = 0.001), and 'ANN' (p-value = 0.012), with 'Vehicle', as for the MIM classifier, also a new addition compared to the previous NB and 'polytree' comparisons, as show in Figure 4.12.

- **Comparison of Selective classifiers and Non-selective 'Network' classifiers**

Figure 6.15 and Figure 6.16 display the error rates for the SMIM compared to the GBN and SNB compared to the GBN respectively. The error rate for the SMIM is lower than that for the GBN whilst for the SNB, error rates are generally less than the GBN. The corresponding differences in accuracy are shown in Figure 6.17 for the SMIM and in Figure 6.18 for the SNB compared to the GBN.

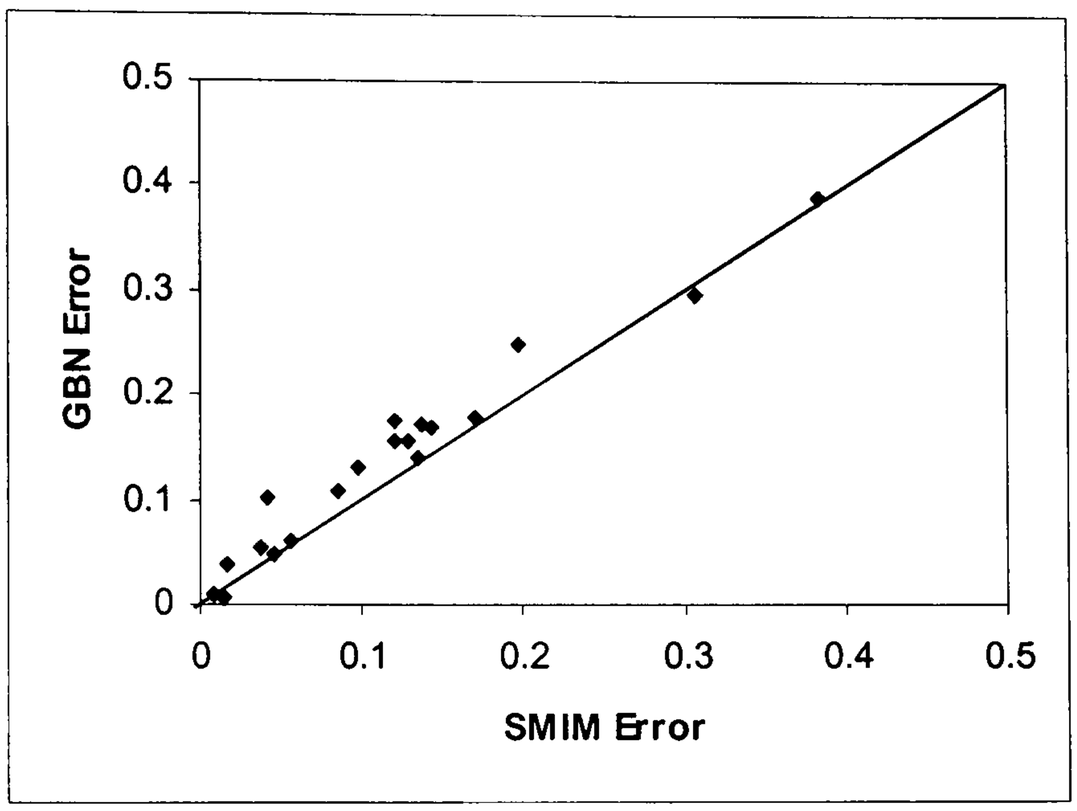


Figure 6.15. Scatter Plots Comparing Error Rates of SMIM with GBN Classifier

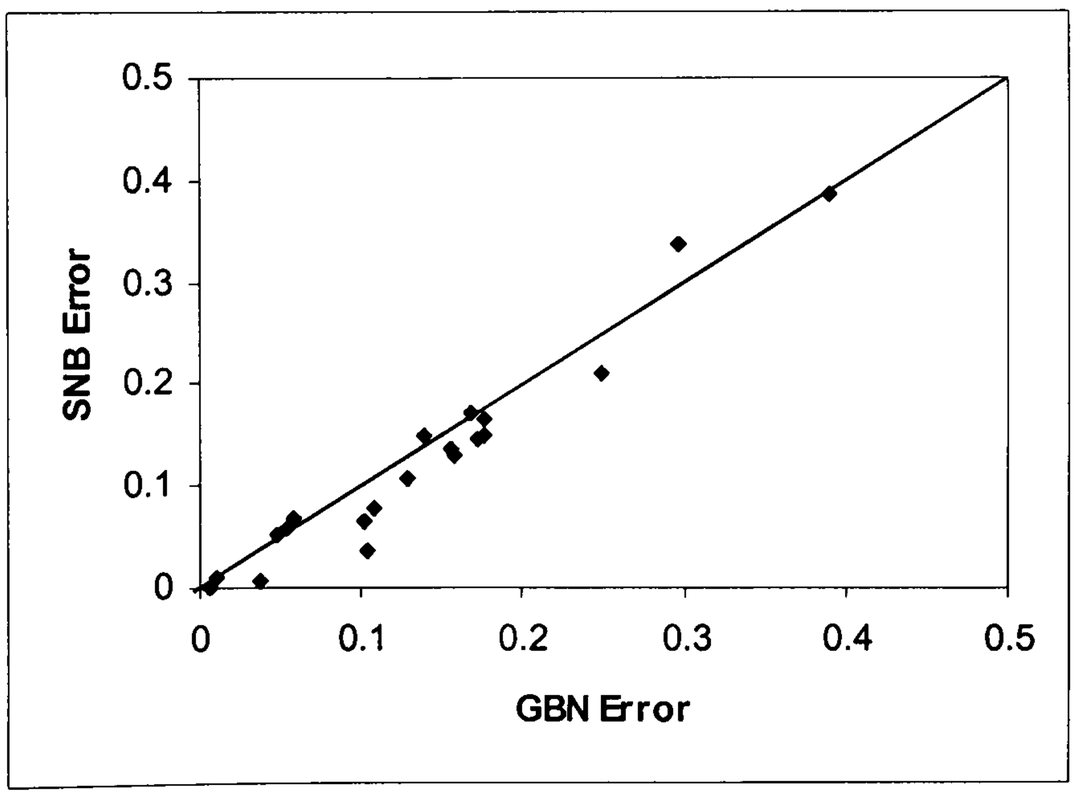


Figure 6.16. Scatter Plots Comparing Error Rates of SNB with GBN Classifier

In comparison to the previous results observed in Figure 4.5 and Figure 4.7, the performance of the SNB has improved compared to that achieved by the NB classifier against the GBN. For the SMIM

classifier, Figure 4.9 and Figure 4.10 previous results illustrate that although there is still an improvement in performance, it is not as significant as that demonstrated by the SNB. However, unlike SNB's improvement, the MIM classifier was already observed to outperform the GBN, thus the SMIM classifier's result further supports the performance achieved by the MIM classifier.

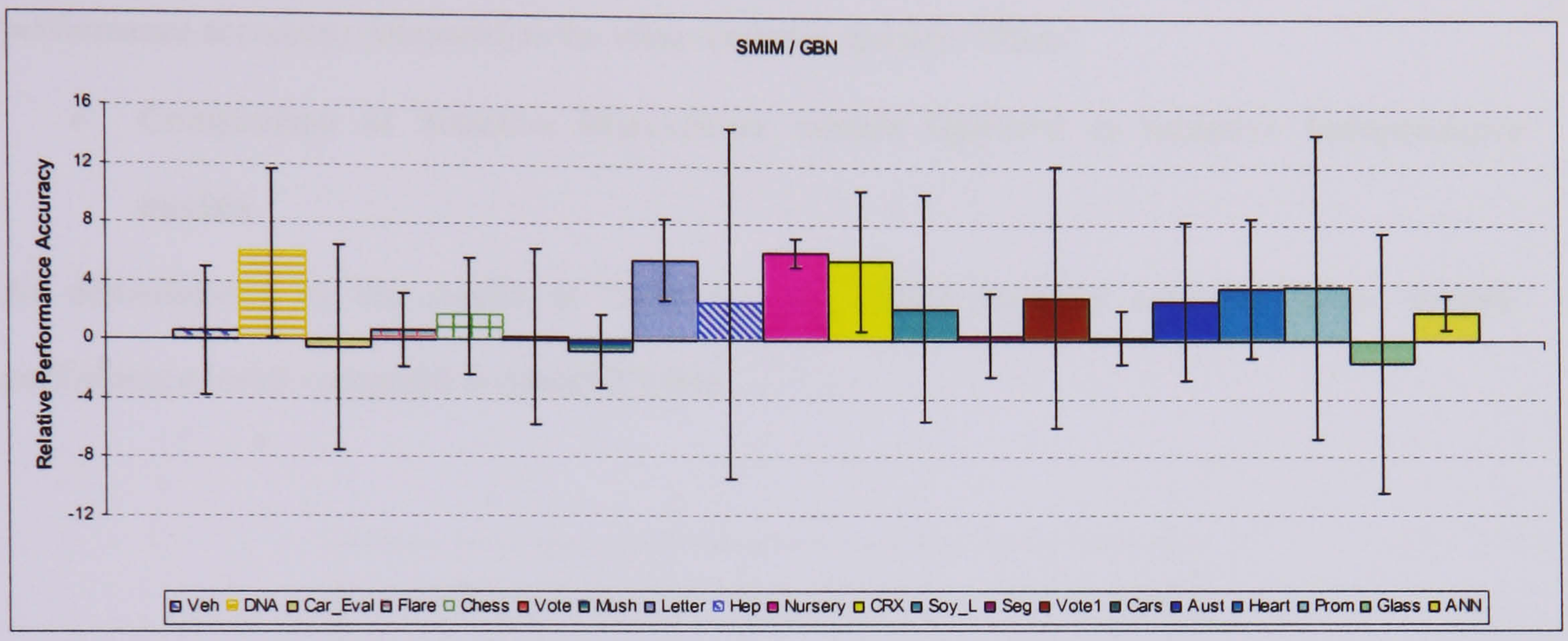


Figure 6.17. Predictive Accuracy relative to SMIM Classifier (GBN)

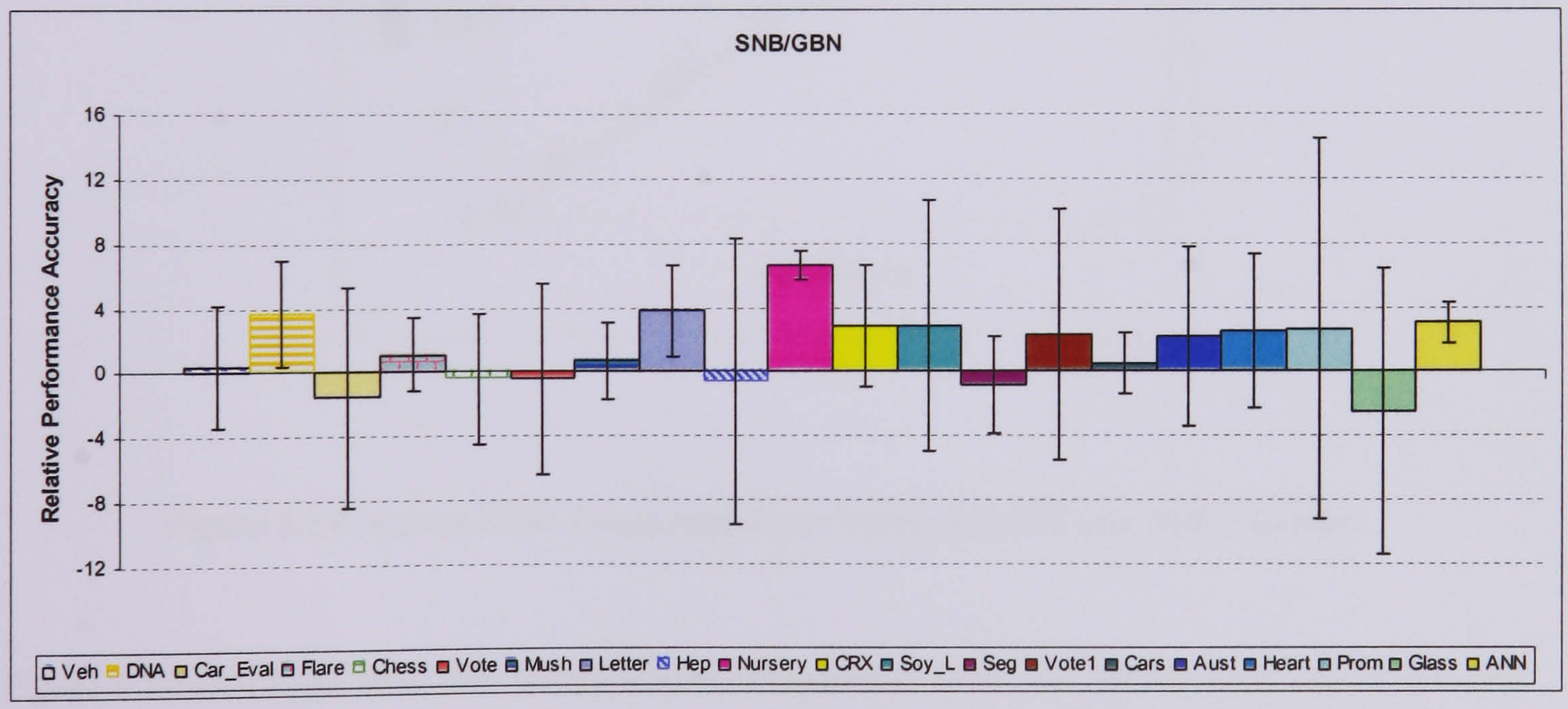


Figure 6.18. Predictive Accuracy relative to SNB Classifier (GBN)

The SMIM compared to the GBN found differences that were statistically significant for the data sets 'DNA' (p-value = 0.045), 'Letter' (p-value = 0.002), 'Nursery' (p-value <0.05), 'CRX' (p-value = 0.018), and 'ANN' (p-value = 0.005) with 'CRX' and 'ANN' new additions to those found

previously for the MIM/GBN comparisons. In the case of the SNB compared to GBN, differences were found to be statistically significant for data sets 'DNA' (p-value = 0.026), 'Letter' (p-value = 0.013), 'Nursery' (p-value <0.05), and 'ANN' (p-value = 0.001) with 'Letter' and 'Nursery' added to those found with differences that were statistically significant from the previous NB/GBN comparisons. As illustrated in Table 6.1, GBN has only one 'overall' winner (in terms of performance accuracy) compared to the other methods, namely, 'Glass'.

- **Comparison of Selective Dependence models Opposed to Selective Independence models**

As demonstrated by the results in Table 6.1, the SMIM classifier achieved seven 'overall' performance levels compared to the SNB's five.

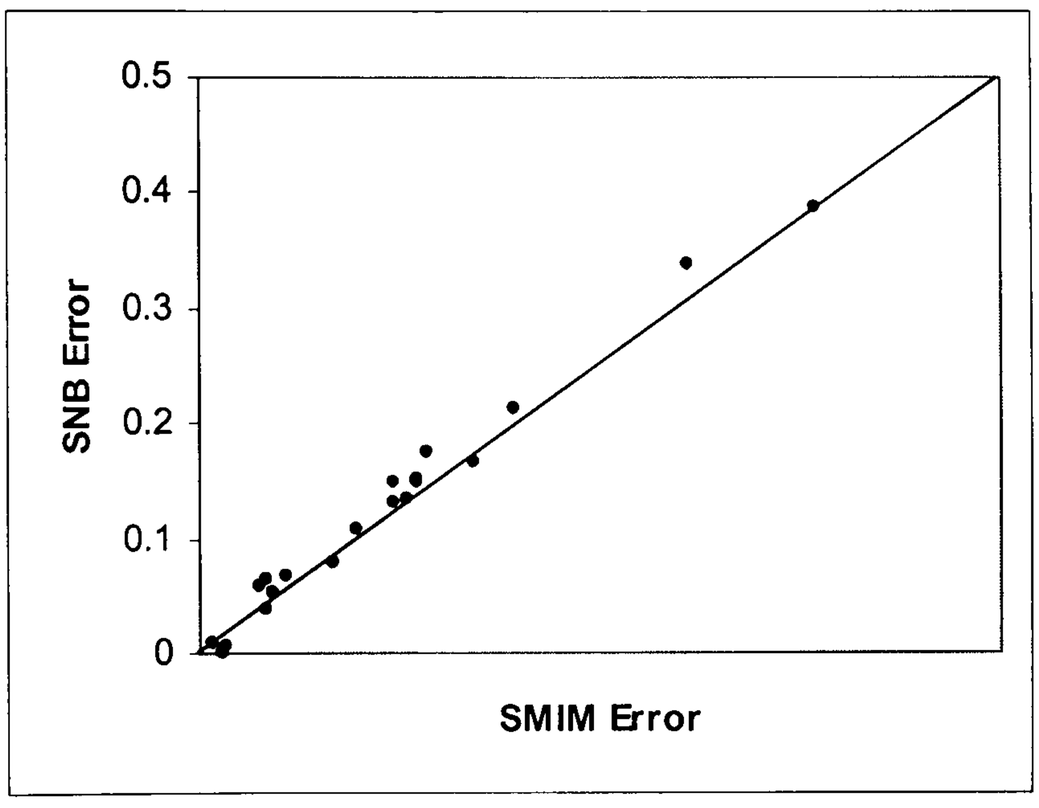


Figure 6.19. Scatter Plots Comparing Error Rates of SMIM with SNB Classifier

Figure 6.19 shows the error rate for the SMIM compared to SNB. The SMIM error rate is less than SNB with most points being above the diagonal line. The corresponding differences in accuracies are displayed in Figure 6.20 with differences found to be statistically significant in one data set, namely, 'DNA' (p-value = 0.045). In the case of SNB only data set 'ANN' (p-value = 0.009) was found to have differences that were statistically significant.

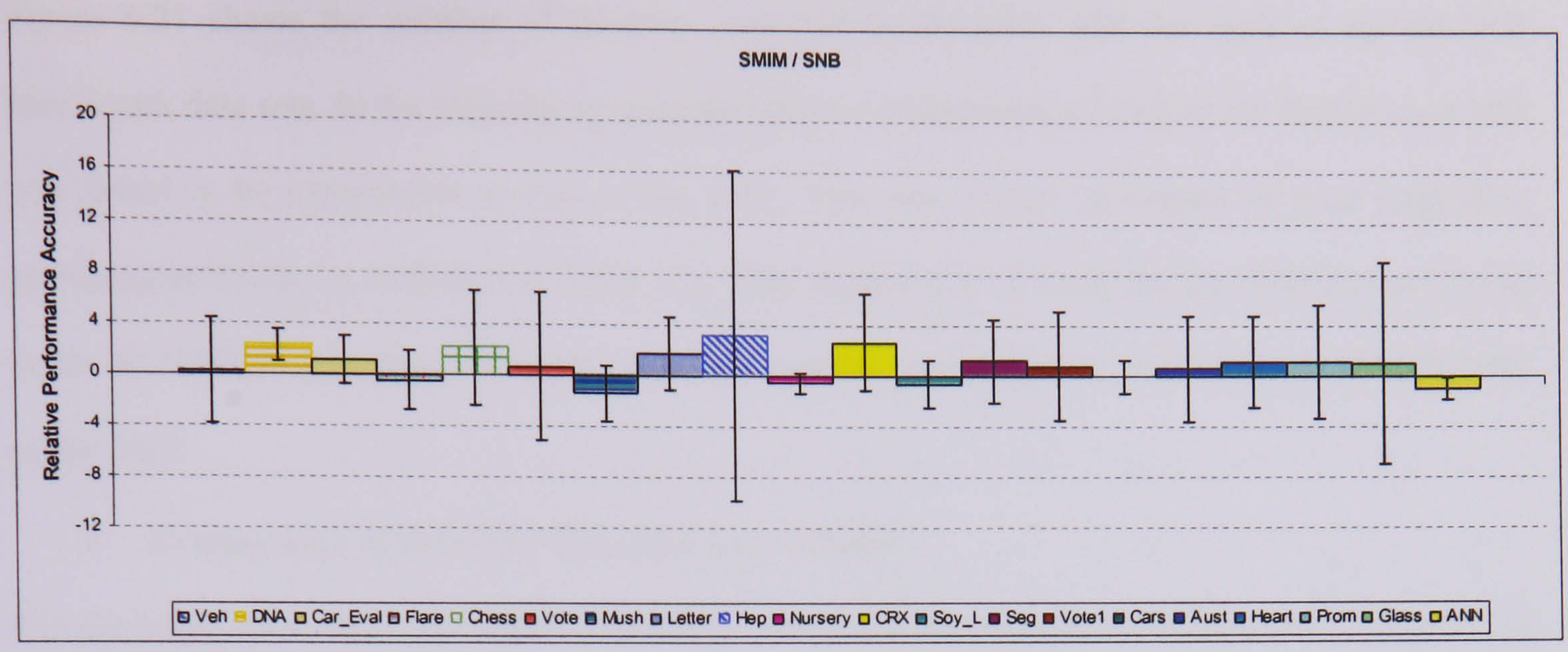


Figure 6.20. Predictive Accuracy relative to SMIM Classifier (SNB)

In general, the predictive levels for the SMIM and SNB were comparable for most data sets studied. From Table 6.1 the SMIM improved performance was better than the SNB for eight data sets with performance accuracy differences ranging from 0.17% to 3.07%, whilst the SNB, in contrast, achieved only two better than the SMIM classifier, namely 'Soybean_Large' and 'ANN'.

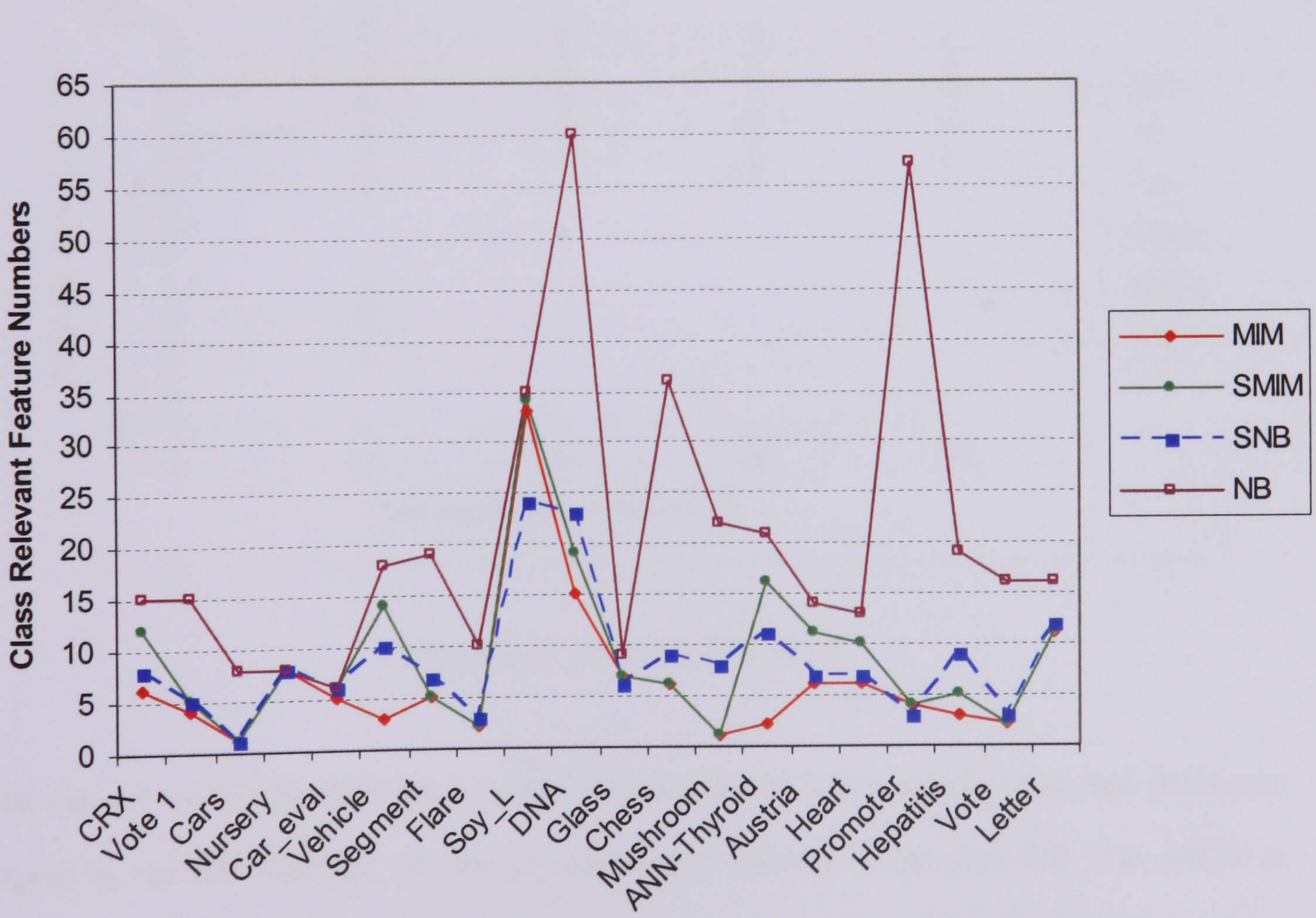


Figure 6.21. Class MB Feature Size For Each Data Set.

Figure 6.21 shows the number of features contained in the class MB for each of the 20 UCI benchmark data sets. In the majority of cases the number of features required by the SMIM class MB was found to be comparable to that of the SNB. This was similar in respect of their respective performance levels, as indicated in Table 6.1. This implies that in order for the SMIM classifier to better the SNB performance, the MIM class MB needs to generally have more features than the MB of the SNB.

- **Comparison of Selective Classifier Learn Rates**

For the larger data sets, only 'ANN' and 'DNA' showed improvements in performance. Figure 6.22 and Figure 6.23 display the learning curves for the data sets 'ANN' and 'DNA' respectively, averaged over 25 trial runs. Included for comparison are the results of the non-selective methods which were previously investigated in Chapter 4.

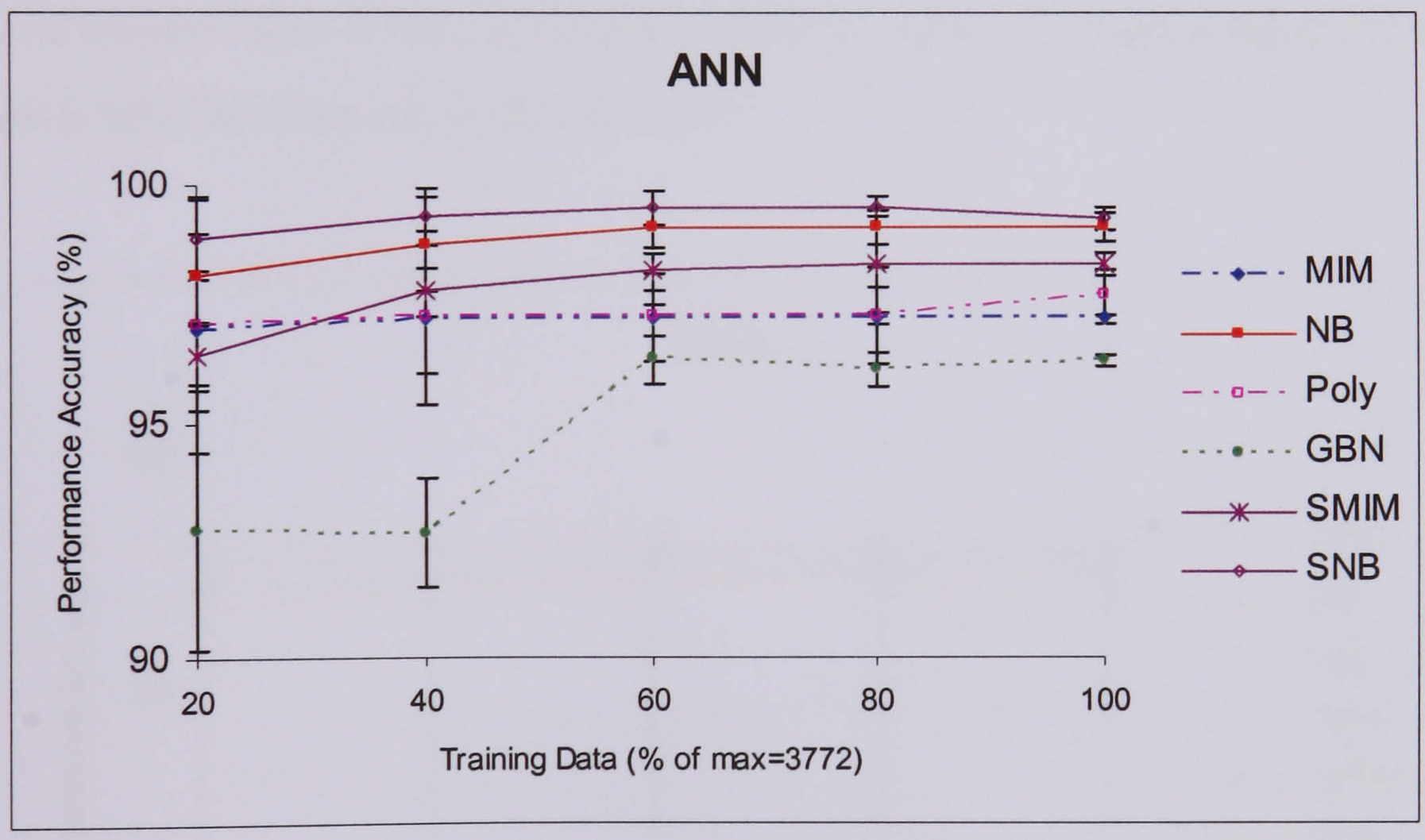


Figure 6.22. Learning Plot: ANN

In the case of 'ANN' the SNB has a better learn rate for the lower sample sizes than previously achieved by the NB classifier, due to the reduction of features in the class MB. The SMIM in contrast, shows a much steeper learn rate at the lower sample size than the MIM classifier, however, the performance levels for the SMIM are better after 40% sample size. This result was not unexpected, as we observed similar trends in Chapter 4. The large number of features and

corresponding small sample size does appear to have had an impact on performance levels. Here the SMIM expands the class MB and so the learn rate becomes steeper, whilst SNB reduces the class MB and improves the learn rate.

Both the SMIM and SNB reach stable performance levels at the 60% sample size. Figure 6.23 illustrates similar characteristics for the 'DNA' data set, with both SNB and SMIM learn rates as observed in Figure 6.22. However, unlike the learn rates observed for data set 'ANN', the SMIM maintains a performance level greater than the MIM classifier for all sample sizes. From Figure 6.23, the SNB and SMIM classifiers reach stable performance levels at a lower 40% sample size unlike their non-selective variants at 60%.

Whilst for both data sets SMIM performance levels are better than those of the MIM classifier, this was not observed for SNB. For the 'DNA' data set, NB maintained a higher performance level than SNB. The reduction in features defining the class MB for this high dimensional domain has clearly effected the performance of SNB, as illustrated in Table 6.1, and as can be seen in Figure 6.23, even begins to fall as the sample size increases to 100%.

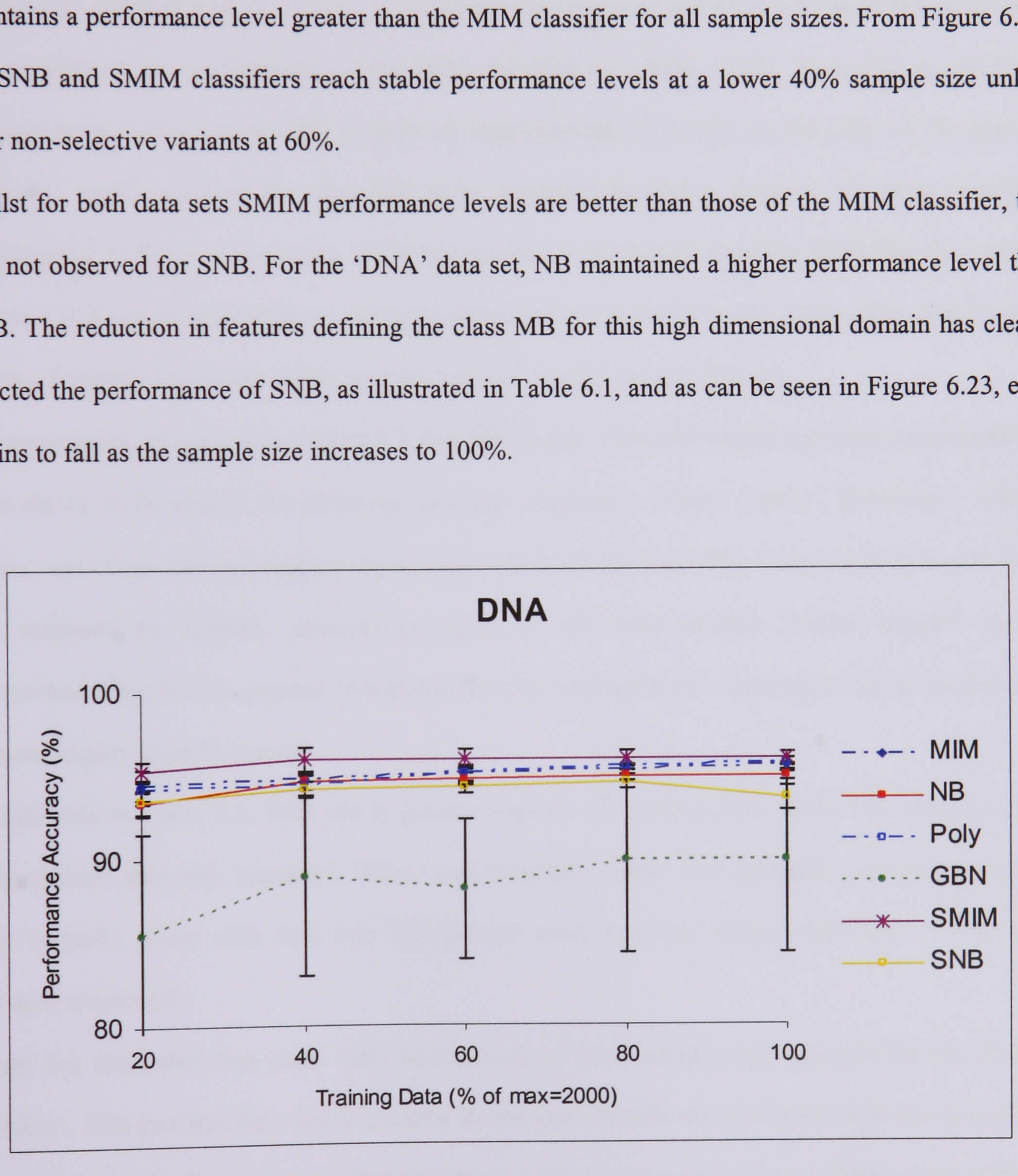


Figure 6.23. Learning Plot: DNA

6.5 Discussion

In general SMIM did improve performance of the MIM classifier with ten data sets demonstrated as improving with performance increases ranging from 0.22% to 5.87%. 'Overall' SMIM achieved seven data sets, and an additional one close to NB, better than all the other models investigated, as shown in Table 6.2.

Figure 6.21 demonstrates that the performance increases achieved by SMIM do not, for the majority of data sets, come at a price. 'Vote1' for example only required one feature to be added to the class MB to improve the performance of the MIM classifier by 1.64%, whilst 'Hepatitis' required two features to be added to the MB to gain an improvement of 1.76%. In the case of the data set 'Vehicle', which required the class MB to be expanded by eleven features, the improvement in performance not only increased by 5.87% but resulted in the SMIM classifier becoming the 'overall' winner (in terms of performance accuracy) compared to the performance of the other classifiers, as Table 6.2 shows. Previously, GBN was the 'overall' winner for this data set.

Our assumption, discussed in section 5.2, that the 'initial' class MB would represent a good start has been shown to be viable. We observed data sets 'Segment', 'Vote', 'Letter', 'Promoter', 'Chess', 'Glass' and 'Cars' did not improve by adding features to the class MB, however as shown in Table 6.2, remained as 'overall' winners compared to the other models. Further support can be demonstrated by the low number of features actually required to be added for some of the data sets, in order to gain in performance.

As indicated in Table 6.2, SNB did in general improve the performance of the NB classifier. SNB claimed three data sets 'Nursery', 'Flare' and 'Soybean_Large' that the MIM classifier previously won 'overall', along with data sets 'Mushroom' form previous winner GBN and 'ANN' from previous winner NB.

Figure 6.2 illustrated that there were occurrences of local minima and maxima for the data set 'Vehicle'. This justified the need to process all features (outside the class MB) that would be added to the class MB. In the majority of data sets that actually saw improvement by adding edges (SMIM), we observed there was a slight degradation in performance prior to any subsequent improvements. This was more apparent where a high value MI edge was introduced to the class MB and less for the small MI valued edges.

Table 6.2: 'Overall' Classifier Predictive Accuracies

DB Name	Average Performance (%)	Classifier	Default (overall)
Vehicle	**61.53 ± 0.91** 14(18)	SMIM	25.8
DNA	***95.58 ± 0.42*** 15(60)	***MIM***	51.9
----------	***95.80 ± 0.36*** 19(60)	***SMIM***	
Car_Evaluation	***86.43 ± 0.85*** 6(6)	***SMIM***	70.0
----------	***86.58 ± 1.78*** 6(6)	***NB***	
Flare	**83.40 ± 1.67** 3(10)	SNB	79.2
Chess	**96.27 ± 3.56** 6(36)	MIM	52.0
Vote	**95.40 ± 2.41** 2(16)	MIM	54.8
Mushroom	**99.96 ± 0.04** 8(22)	SNB	51.8
Letter	**80.26 ± 0.37** 11(16)	MIM	4.07
Hepatitis	**85.76 ± 7.14** 5(19)	SMIM	79.4
Nursery	**96.33 ± 0.27** 8(8)	SNB	33.3
CRX	**87.90 ± 0.90** 12(15)	SMIM	55.5
Soybean_Large	**92.08 ± 2.01** 24(35)	SNB	13.5
Segment	**94.49 ± 0.64** 5(19)	MIM	4.80
Vote1	**90.15 ± 1.53** 5(15)	SMIM	61.4
Cars	***99.23 ± 0.70*** 1(8)	***MIM***	62.5
----------	***99.17 ± 1.08*** 1(8)	***SNB***	
Austria	**87.10 ± 1.65** 11(14)	SMIM	55.5
Heart	**86.39 ± 0.89** 10(13)	SMIM	55.6
Promoter	**87.97 ± 1.31** 4(57)	MIM	50.0
Glass	**69.37 ± 2.08** 7(9)	MIM	35.5
Ann-Thyroid	**99.30 ± 0.25** 11(21)	SNB	92.6

Key: 'Overall' winning (in terms of performance accuracy) classifier is denoted by **Bold** type face, where ***italic Bold*** indicates similar performance accuracies between classifiers, as shown above for data sets: DNA, Car_Evaluation and Cars.

This implies that the initial addition as a single feature was not able to maintain the discrimative power of the *lower bound* and needed support from further feature additions. This is an issue that is further discussed in Chapter 8.

SMIM and SNB classifiers have been demonstrated as offering a benefit in the improvement of performance for the MIM and NB classifiers. In the case of SNB, this was achieved by the removal of the highly correlated features, whilst for SMIM the addition of relevant features to the class MB, particularly evident for the small data sets. Despite the MIM classifiers' class MB expansion, the classification complexity remains at $O(N)$ as for the NB classifier. SNB's model construction is however more complex and slower than NBs' as a forward process, with a greedy method to traverse the space, has at worst case time complexity $O(N^2)$.

Our use of the two heuristic 'stopping criteria', particularly in relation to performance degradation, did prevent an 'exhaustive' search from being carried out, especially for the high dimensional domains like 'DNA'.

The addition of features to the 'initial' class MB has been shown to improve performance, particularly in some of the small data sets, Table 6.1. This may have been as a consequence of the use of the CL algorithm (previously discussed in Chapter 4 as a drawback). In this approach however, the removal of irrelevant features is not permitted and is probably a drawback to achieving an 'optimal' solution in terms of performance.

Expanding the class MB by adding features transforms the 'tree' structure into a network. This is essentially what Sucar [SP+97] wanted to achieve, but just as for the SMIM, in respect of classification, the proposal does not provide an 'optimal' solution. A better approach is to derive the network in a similar fashion to that proposed by Cheng [CBL97]. This would offer a method of constructing the 'optimal' class MB as removal and additions of features are allowed by use of the 'thinning' and 'thickening' phases. Unfortunately there is a complexity issue associated with Cheng's approach of $O(N^4)$, as no node ordering would be supplied prior to construction. The question here is do we need to consider 'optimality' ? Since the MIM 'tree' structure demonstrates its ability to provide a viable 'initial' class MB and has the efficiency associated with CL algorithm for its derivation, the demonstrated performance shown in Table 6.1 implies that striving for further improvement might be too costly with very little to be gained.

6.6 Conclusion

In general, the SMIM did improve the performance of the MIM classifier with the greatest benefit observed for the smaller data sets studied. The empirical studies showed that this improvement was not at the price of increased complexity as in most cases only a marginal number of features were actually required to be added to the 'initial' class MB. For the larger data sets, the experimental results show that in the majority of cases no improvement was possible implying that the class MB derived via the CL algorithm was already a satisfactory representation.

Overall the empirical studies confirm the classification technique is not restricted to 'tree' based structures (with respect to the class MB) and is thus independent of the underlying topology of the domain being modelled. However, whilst focusing on the class MB does allow improvement in some domains without creating potentially dense networks, the technique has a drawback in that it does not guarantee to find an 'optimal' solution.

6.7 Summary

Chapter 5 proposed a method for improving the performance of the MIM classifier. In this chapter we evaluated a 'selective' variant of the MIM classifier which focussed on the expansion of the class MB. Our experimental results demonstrate that the approach taken can improve the performance of the MIM classifier and when there was improvement it only required, in general, a few additional features to be added to the class MB. 'Overall' the SMIM was better for seven data sets compared to the other methods, with two direct improvements to the MIM classifier and the remaining five sufficiently high to better the previous winning performance of NB 'polytree' and GBN. The combined 'overall' winning performance (in terms of performance accuracy) for the SMIM and MIM classifier was thirteen data sets compared to SNB and NB's six.

Our initial assumption, discussed in section 5.2, regarding the class MB with respect to the MIM classifier, appears to be viable. The results indicate that for the majority of the data sets studied, the SMIM either required a few features to be added to the class MB or that no further improvements in performance were possible. This implies that the class MB, as derived from the use of the CL algorithm, was indeed a good initial representation. Where improvements were observed the data sets were generally the smaller ones, with the majority of the larger data sets remaining at the MIM classifier's *lower bound*.

The expansion of the class MB did highlight issues in respect of the increase in MB and the learning rates. As expected we observed the SMIM having a steeper learning rate for the small sample size and a less steep one for the SNB having reduced its MB feature size. In comparison with the non-selective variants however, the performance levels of the selective models was shown to be greater for these sample sizes than achieved by their corresponding non-selective variants, namely the MIM and NB classifier.

In Chapter 4, we discussed the issues relating to the use of the CL algorithm and that it was prone to generating 'trees' that have irrelevant features added to the class MB. This could be construed as a disadvantage or even a limitation as the SMIM's *lower bound* is defined by the CL algorithm. Since we expand the class MB to improve performance, any irrelevant features already contained within the initial class MB would not be removed. Our experimental results however, did not confirm this, with the SMIM and MIM classifier performing better than the other methods for the majority of the data sets studied. Nevertheless removal of irrelevant features would potentially offer a better classifier, and in the case of the MIM classifier, would be most efficiently addressed during the 'tree' construction phase. The main issue however, lies in the actual identification of 'irrelevance'. In Chapter 7 we discuss this aspect further and show for the domain of AAP, that the removal of what are considered 'irrelevant' features can sometimes be inappropriate.

Evaluation of the MIM classifier, Chapter 4, and the 'selective' variant within this chapter have utilised the UCI benchmark database for our experimental studies. In the next chapter, we investigate the robustness of these classifiers on a 'real' data set and evaluate their performance in respect of the task of diagnosing a medical domain, Acute Abdominal Pain.

Chapter 7

Diagnosing Acute Abdominal Pain – Case Study

In this chapter we apply the techniques developed in Chapter 3 and Chapter 5 to a 'real' world domain, namely the diagnosis of Acute Abdominal Pain (AAP). This domain is renowned for its difficulty [TS94, OM+96] for both doctors and many Machine Learning (ML) approaches. In the next section, we review some of the current research findings which provide the motivation for our investigation, followed by our specific objectives and aims. In section 7.3, we describe the two data sets that will be used in the study and how we deal with anomalies. Section 7.4 and 7.5 define the experimental methodology and design respectively, with section 7.6, reviewing the experimental results. In section 7.7, we discuss the results and implications and finally in section 7.8, summarise the chapter.

7.1 Introduction

Acute Abdominal Pain (AAP) is the commonest surgical emergency in Europe and in most other parts of the World [Dom93]. Although some causes of AAP don't require admittance to hospital, other conditions such as appendicitis require urgent surgical treatment. An inflamed appendix may perforate raising the risk of death and with one in every sixteen people expected to suffer from it at some point in life [PH88] it is thus a relatively important disease group to identify. Clearly, early and accurate diagnosis is essential, but few doctors and even fewer patients realise just how difficult such early diagnosis can be. The domain of AAP is well known to be both difficult and challenging [LE93] with the diagnosis of appendicitis complicated by other diagnoses like Non-Specific Abdominal Pain (NSAP) which often presents similar signs and symptoms.

Tackling 'real' world problems in complex domains such as AAP has resulted in the development of more and more decision analytic models. Extracting knowledge from experts however, is arising as a major obstacle in model building. Adopting automated / semi-automated techniques, deriving the model directly from the data, can overcome some of these obstacles. In fact [TS94] AAP is one of

the most widely studied applications of computer aided diagnostics examples of which are [OY+95, PEJ98, Kur87, AC+86, WK01, EZ+01, OM+96, TC+03, EOL97, and WK01].

According to Provan and Clarke [PC93] probabilistic reasoning is crucial for diagnosing AAP, as the uncertainties involved cannot be adequately captured given that two patients with the same symptoms may have different diseases. Examples of approaches taking up this challenging domain can be found in [NJ75, Fry78, Ser86, GT91, and Sin98]. However, despite these models attempts to capture the domain dependencies, the empirical evidence in support of diagnostic accuracy and the capturing of dependencies in Bayesian models remains inconclusive.

During comparisons made by Todd and Stamper [TS94] of an 'expert' built GBN and the Naive Bayes, the results suggested that there were no significant improvements in accuracy by taking interactions into account. Work carried out by researchers [ED84, Dom91, and TS94] even suggested from their results that Naive Bayes was probably optimal. The research that followed de Dombal's et al [DL+72] successful application of Naive Bayes led to many approaches, which attempted to avoid making this violation of conditional independence. Here the classifier assumes that the attributes are conditionally independent given the class variable (each attribute has only the class node as a parent). One such example was the G&T system [GT90] that applies Bayes rule strictly. However, this too found Naive Bayes to outperform their dependency model. Ohman [OM+96] even compared Naive Bayes to more complex representations such as rule-based systems and found here too that there was no major 'overall' difference. Further support for Naive Bayes success and its performance in respect of AAP can be found in [PS96, HY01, GG90, and Tho99].

In consideration of these studies and in particular the optimality claim, the following section outlines our investigation aims which we address by applying the MIM classifier and its optimised variant to the AAP domain.

7.2 Objectives

As highlighted in the previous section, several researchers have argued that the NB classifier is the optimal model for this domain. However, experts have identified strong dependencies between symptoms and therefore NB should not be so efficient. As we demonstrated in Chapter 4, models that capture dependencies such as GBN and the 'tree' based dependency models outperform NB in many of the data sets studied and thus should provide a more accurate diagnosis of AAP than NB.

Despite this supporting evidence, as discussed in section 7.1, there appear to be no empirical studies that substantiate this view.

From the review in section 4.6.1, we observed that the poor performance of the GBN was due to problems in modelling high dimensional and small sample set domains. As such, GBN models may have problems with the AAP database as there are class-state imbalances with some states having small sample sizes. However, as we demonstrated in Chapter 4, the MIM classifier was not effected by this domain characteristic and was found to have a comparable performance with that achieved by the NB classifier. Since the MIM classifier does model dependencies between features, although with some loss of representation, it should perform better than the NB in this domain. In addition, as the experimental results showed in Chapter 6, the 'optimised' variant (SMIM) may be able to overcome any modelling constraints and potentially offer further performance improvements.

The first objective of the study was to investigate the optimality claim of NB for classifying in the medical domain of AAP, against three models, which do not assume extreme conditional independence. These are the GBN, 'polytree' and the MIM classifier along with NB's and the MIM classifier's optimised 'selective' variants. In particular we wanted to address three questions:

- For AAP does the Naive Bayes classifier really perform better than the dependency models?
- Is Naive Bayes (as considered by other researchers) really optimal for AAP or is it just good at identifying NSAP[17]?
- Do the dependency models offer more than the Naive Bayes irrespective of its overall accuracy performance?

Our second objective was to investigate the effect of modifying the class MB of the NB and MIM classifier. Ohmann [OM+95] noted from their study that the high dimensionality of AAP was a major problem when attempting to improve the predictive accuracy. The approach taken was to find a feature subset with the best predictive accuracy for a certain classifier. Our aim was to optimise the performance of the two classifiers (NB and MIM) and determine if NB's use of all the features of the domain is a contributory factor to its claimed optimal success. Since NB uses all features by design the SNB variant attempts to optimise the classifier by reducing its MB. In contrast the MIM classifier

---

[17] NSAP is defined by the Doctors as "*a miscellaneous set of non-significant pathologies and thus a group of exclusion*".

starts with an MB defined by the CL algorithm and thus SMIM expands the MB. We are therefore interested in determining how many features each model requires in order to achieve an optimal performance and whether the performance of the NB, in this particular domain, can be matched or even bettered by the SMIM variant.

7.3 Description of the data sets (AAP) used

Two data sets were used in this study, defined in Table 7.2. The first consists of 9867 patient records comprising 33 features, covering 135 feature states, and a class variable having 9 possible states or diseases. The data was originally collected and maintained by Mr AA Gunn [Gun76] at Bangour General Hospital and is currently retained by staff at St John's Hospital in Livingston, Edinburgh.[18] The resulting database addresses the domain of Acute Abdominal Pain (AAP) recording information gathered both during the examination and subsequent audit administration. The structure is based upon a patient's examination on arrival to the Accident and Emergency (A&E) department. Each completed record stores the doctor's 'initial' diagnosis and the 'actual' diagnosis group a patient was subsequently determined as really belonging to, on their discharge from hospital. The full contents of the database far exceed our requirements and mainly provide information necessary for hospital audits. The precise format relevant to our study is defined in Appendix A, with Table 7.1 detailing the AAP diagnostic groups and corresponding abbreviations that will be used throughout this thesis.

Table 7.1: Diagnostic Groups and Codes

Diagnostic Groups	
Value	**Disease Code**
Appendicitis	APP
Diverticulitis	DIV
Perforated Peptic Ulcer	PPU
Non Specific Abdominal Pain	NSAP
Cholecystitis	CHO
Intestinal Obstruction	INO
Pancreatitis	PAN
Renal Colic	RCO
Dyspepsia	DYS

The second data set comprises 5373 case samples again describing examination records of patients suffering from acute abdominal pain. In this case however, the data has been collected at a different

[18] CADA (Computer Assisted Diagnostic and Audit) data base which is considered the largest database of AAP in Europe. Courtesy of St John's Hospital, Livingston, Edinburgh.

geographical location, namely Leeds.19 This data was gathered over a period of 30 years concerning the diagnosis of AAP and is currently retained at the Professorial Surgical Unit and Accident and Emergency Department, at the General Infirmary. We will label the first data set 'CADA' and the second one 'LEEDS' in order to distinguish between them.

Table 7.2: AAP Data sets used in the experiments

Dbase Name	Attribute size	Class size	Sample size	Train size	Test size
CADA	33	9	9867	6959	2908
LEEDS	33	9	5373	-	5373

Both data sets have been standardised by collaboration between the two hospitals under the direction of Professor Tim de Dombal. In the following sections, we describe the experimental work and present the corresponding results.

7.4 Experimental Methodology

With the exception of one of the features namely 'AGE', which is strictly a continuous variable, all of the other 32 features represent discrete variables. In this data set the doctors themselves have provided the discretisation for the feature 'AGE' based upon their own judgements. The group Non-Specific Abdominal Pain (NSAP) is not actually a diagnostic group rather a 'catch all' category into which the doctors assign a patient whom they cannot fit into one of the other 'true' eight diagnostic groups. In a sense this can be considered as a 'don't know' category, but only in respect to the 'true' eight known categories. For this study we employed the hold-out approach partitioning the data base into a 'learn' and test sample set, as defined in Table 7.2. The training partition was approximately 2/3 of the database whilst the test partition the remaining 1/3 of the sample set. Both partitions are the result of performing a 'random' but stratified split, in order to compensate for the imbalances in respect of the nine class-state distributions.

On examination of the database, records were found to have multiple or composite parameter values stored in respect of some of the symptoms and in other cases none of the symptom parameters were recorded (missing). To deal with these two anomalies we have introduced two additional parameter values, which are appended to each symptom. For example Symptom 21 : MOOD will be described

[19] Courtesy of General Infirmary, Leeds, UK.

by parameters : normal (21/1)[20], distressed (21/2), anxious (21/3) plus composite[21] (88) and missing (99). This approach ensures that the Naive Bayes model does not have an advantage over the Bayesian models (as complete data sets are required) with the AAP data set effectively standardised for all models under study.

For the two data sets 'selective' and 'non-selective' networks were constructed. 'Non-selective' experiments were performed using the methods described in Chapter 4, section 4.3 for constructing the MIM classifier, NB, 'polytree' and GBN applications also in Thomas [THS05b]. To learn the 'selective' variants we applied the methods defined in Chapter 5, section 5.2 and Section 5.3, for constructing the SMIM and SNB classifiers.

In order to obtain an expert comparison of performance accuracy, we further extracted from the two data sets the doctors 'initial' diagnosis, as described in section 7.3.

7.5 Experimental Design

For each of the six classifiers, the structure was learned/constructed using the 2/3 training partition and each classifier's accuracy determined on the 1/3 test partition. The main performance measure used was the classification accuracy of a model on the test data, the classification accuracy being the percentage of test cases that were diagnosed correctly (identified in the data sets as 'actual' diagnosis).

This process was repeated over a series of runs in order to obtain a sample average together with the standard deviation for the predictive accuracy using the test partition. The statistical significance of the differences in classification accuracy was measured using a Analysis of Variance (one-way ANOVA) followed by Post Hoc Tukey comparisons with overall confidence level 95%. Prior to applying ANOVA, as was the case for Chapter 4 and Chapter 6, we first established the validity of the assumptions.

In the domain of AAP, where there are numerous class-states, the comparison of the six methods does not provide an accurate measure using only the classification accuracy. To address this we have

---

[20] (21/1) represents symptom number 21 and corresponding symptom parameter number 1, as defined in Appendix A, Table A-1.

[21] Some composites have been 'grouped' and added to symptoms as new parameter values. Where the frequency of occurrence for combinations was below a set threshold (arbitrarily set) these were assigned to the default composite value '88'

computed additional statistics, which are generally used for comparing 'alternative' tests with respect to medical diagnosis [CH90]. In this thesis we utilise this approach to make comparisons of our 'alternative' classifiers and thus access their ability to effectively discriminate between the individual class-states or diseases. Assuming the positive/negative value for a disease to represent its presence/absence, the different statistics we computed can be described, adopting the notation taken from Singh [Sin98], as follows.

Sensitivity: This is the ability of a classifier to correctly predict the presence of a disease in a patient with that disease. Also know as the True Positive Rate, it is defined as: $\frac{TP}{TP+FN}$ where TP is the number of true positives while FN represents the number of false negatives.

Specificity: This is the ability of a model to correctly identify patients that do not have a given disease. Thus, it is the proportion of people who do not have a given disease, and correctly predicted so by the classifier. As such: $\frac{TN}{TN+FP}$ where TN represents the number of true negatives and FP represent the number of false positives.

Likelihood Ratio: This measures the ability of a classifier to discriminate between alternative diseases.

The higher the value, the greater is the discriminating ability of the method. It is defined as follows:

$$\frac{TP(FP+TN)}{FP(TP+FN)} = \frac{sensitivity}{1-specificity}$$

Predictive Value: This measures the accuracy of a classifier on a given disease, and is the probability that a patient actually has the certain disease, given that the classifier has so predicted. It is defined as: $\frac{TP}{TP+FP}$

In addition, we also computed the discriminant matrices, for each method, describing the performance of each technique with respect to the individual diseases. This provides a mechanism for us to compare different approaches with respect to their ability to correctly identify the individual class-states (diseases).

In this investigation, the 'CADA' database was used to both construct and test the six methods, whilst the 'LEEDS' data set was only used for testing the six methods. This latter data set represents

a truly 'external' sample set as its data distribution, thus its characteristics, do not have an influence on the classifier's structure as it has been independently gathered from the 'CADA' data set.

7.6 Experimental Results

7.6.1 Results 'Non-selective' Experiments

The average predictive accuracies, taken over 10 runs, of the classifiers generated for each of the four 'non-selective' methods are shown in Table 7.3 and Table 7.4. In Chapter 4, we ran 25 trials to provide sufficient coverage of the diversity of the 20 UCI benchmark data sets studied. Since the AAP domain essentially comprises only one data set, our preliminary studies found in this case, 10 to be adequate. Each entry in the tables describes the average accuracy along with the sample standard deviation illustrating variations in the predictive accuracy from sample to sample. For completeness the doctor's predictions are also included[22]. The default value represents the majority classifiers' predictive accuracy. For the AAP data bases this is the error rate associated with the diagnostic group NSAP. As was the case in Chapter 4, Table 4.2, the 'overall' value is in respect of the 'entire' datasets (CADA/LEEDS) and not the individual test partitions.

Table 7.3: Average Predictive Accuracy 'CADA' – error rates

Doctor	MIM	NB	GBN	Polytree	Default (overall)
0.2834±0.28	0.3349±0.82	**0.2617±1.16**	0.3583±1.56	0.3566±0.66	0.5495

Key: MIM – Mutual Information Measure Classifier, **NB** – Naive Bayes Classifier, **GBN** – General Bayesian Network Classifier, **Polytree** – Pearl's Model. Values in **bold** type indicate the highest model performance achieved by the classifier in respect of the CADA database.

Table 7.3 and Table 7.4 display the 'overall' predictive values for the CADA and LEEDS databases respectively. In general the NB outperforms the BN models and in the case of the CADA database, even performs better than the doctors.

22 These are already recorded within the two data sets in respect of each test case used in the study.

Table 7.4: Average Predictive Accuracy 'LEEDS' – error rates

Doctor	MIM	NB	GBN	Polytree	Default (overall)
0.3413±0.0	0.4569±0.52	0.4489±0.53	0.4882±0.44	0.4770±0.15	0.6382

Key: MIM – Mutual Information Measure Classifier, **NB** – Naive Bayes Classifier, **GBN** – General Bayesian Network Classifier, **Polytree** – Pearl's Model. Values in **bold** type indicate the highest model performance achieved by the classifier in respect of the LEEDS database.

The GBN, Figure 7.1, 'polytree' (SCN), Figure 7.2, and the MIM, Figure 7.3, models provide a qualitative structure as shown, in contrast to the NB model, which offers only a trivial representation. In the case of the GBN the structure is a more complex DAG, whilst the SCN and MIM structures correspond to a less complex 'tree' representation. The 'tree' structures of SCN and MIM are essentially the same with the interpretation governed by edge directionality. For the SCN, Figure 7.2, there is a 'multi parented class node, whereas for MIM classifier, Figure 7.3, the class node represents the root vertex and thus acts as a lone parent. In correspondence with NB the MIM structure represents a subset of the NB. That is, the implied feature selection of MIM in respect to the class's children. However, MIM unlike NB does not make the same extreme assumptions of conditional independence.

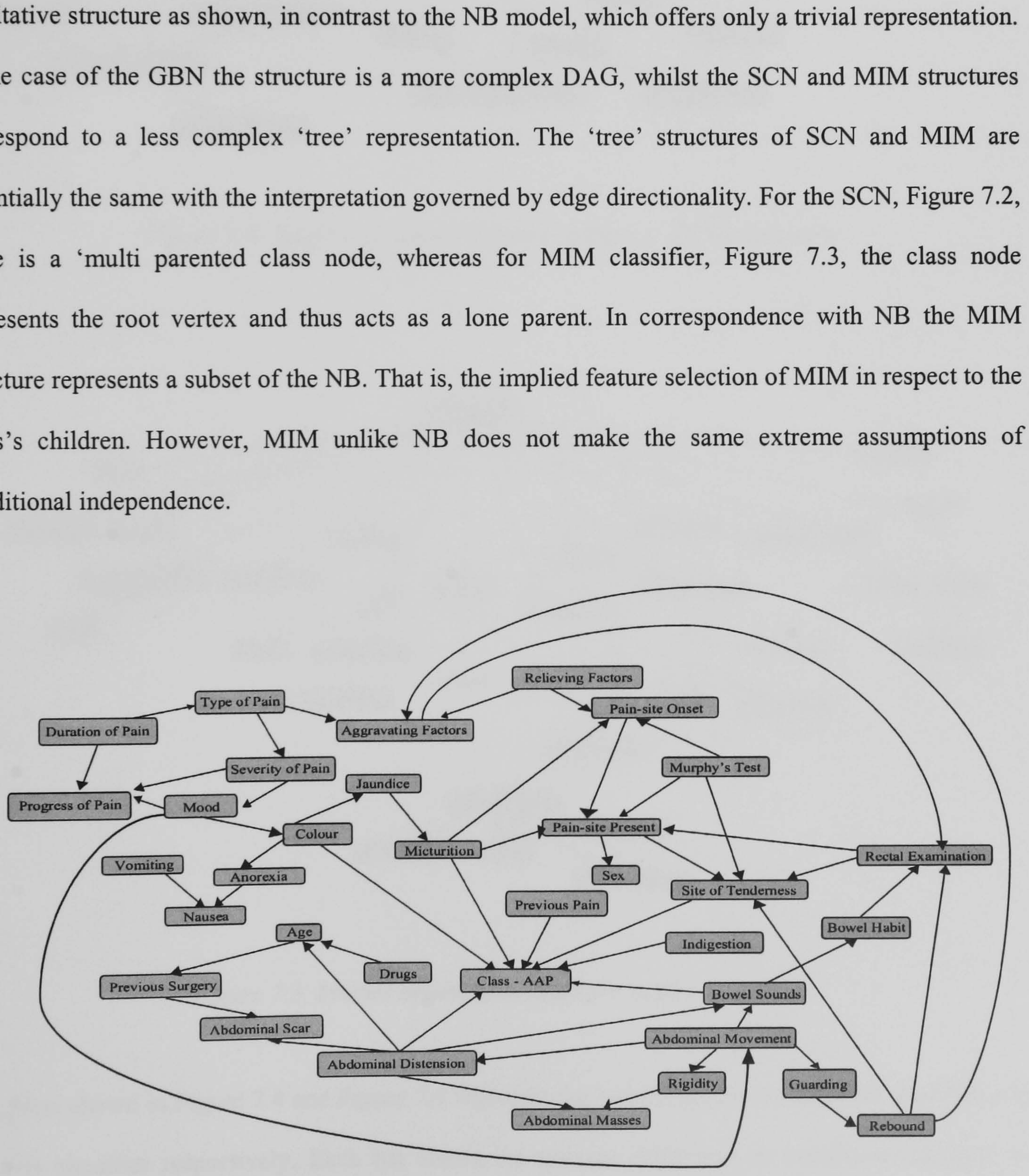


Figure 7.1. General Bayesian Network (GBN) Structure

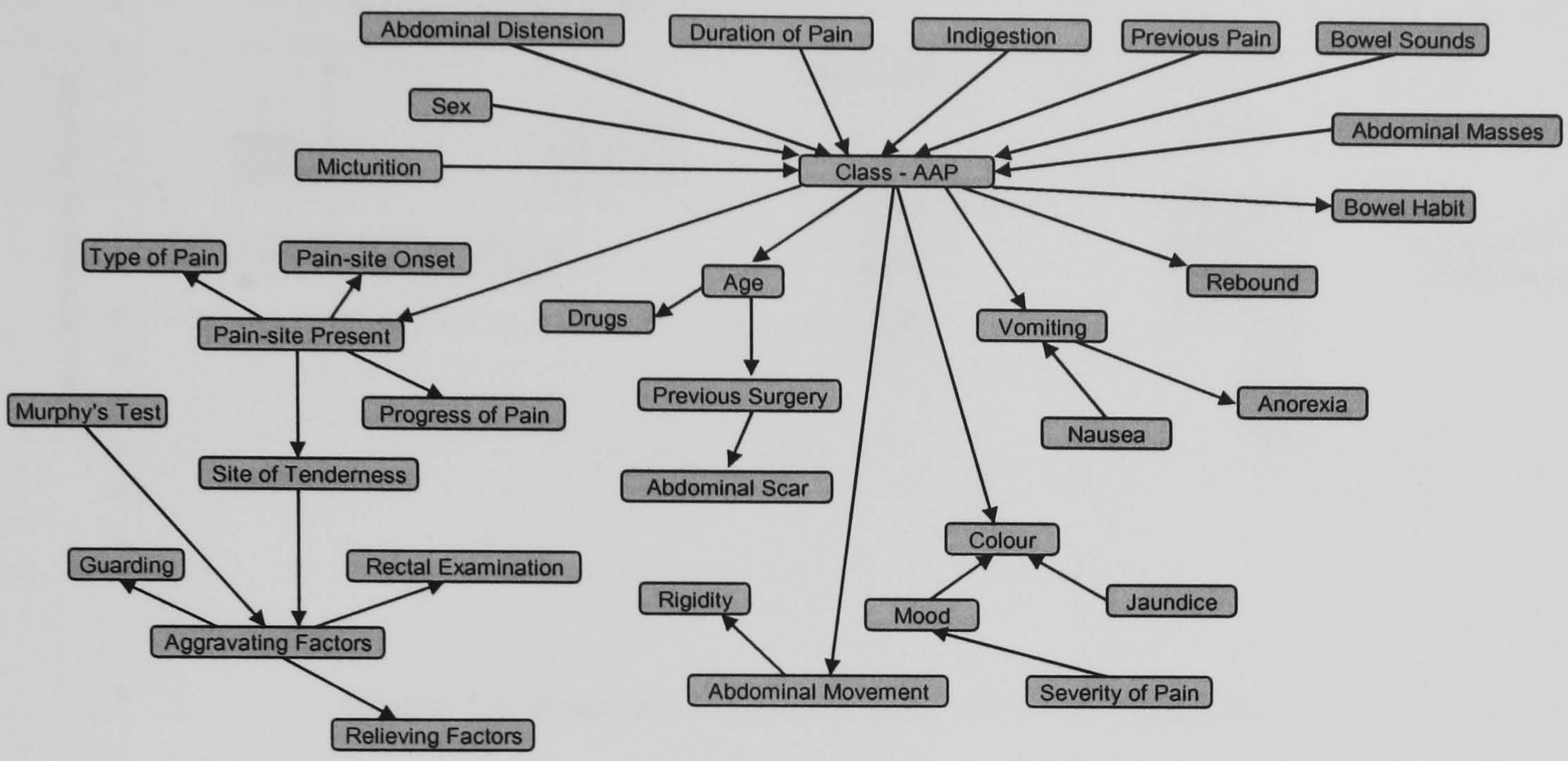


Figure 7.2. Singly Connected Network 'polytree' (SCN) Structure

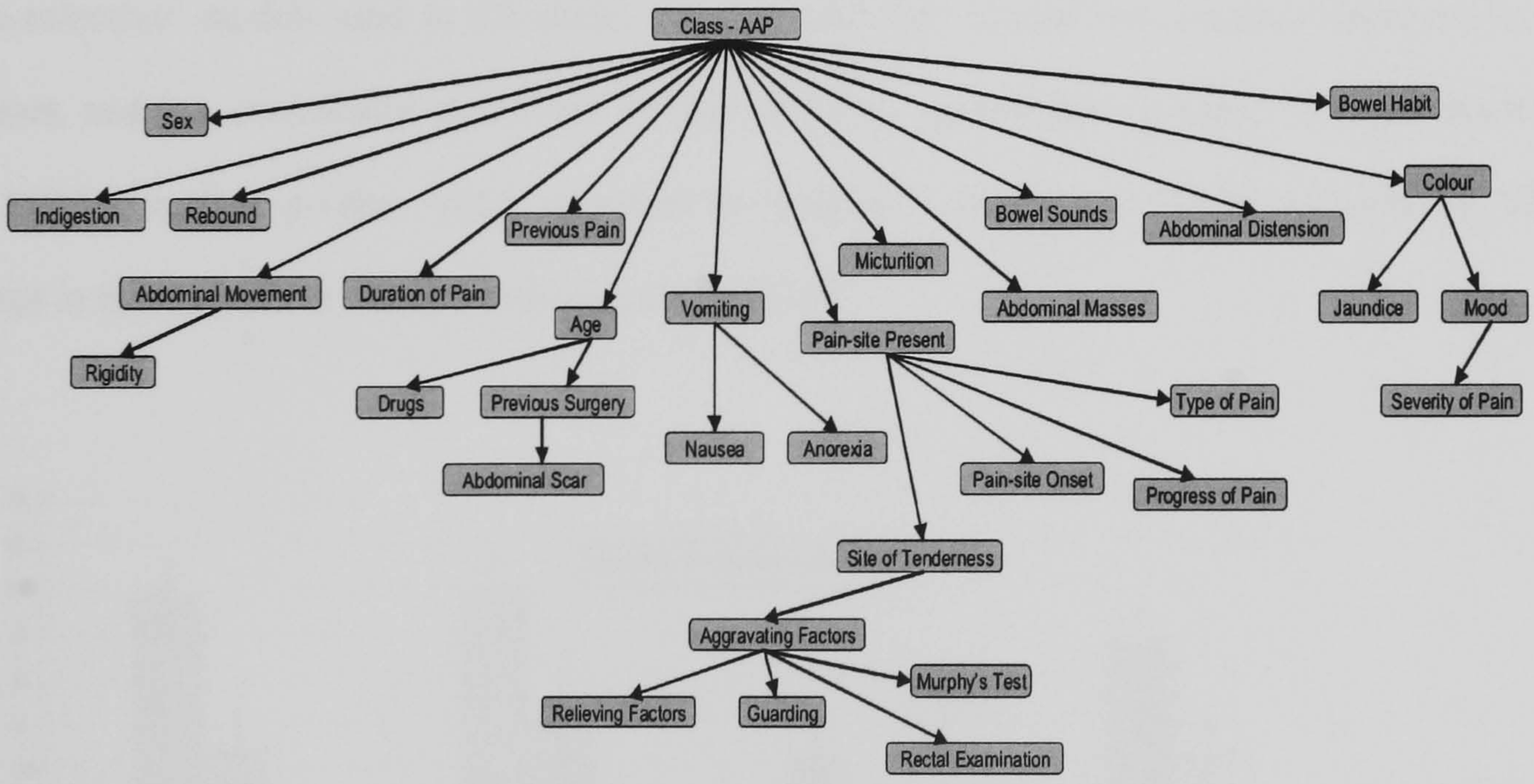


Figure 7.3. Mutual Information Measure (MIM) Structure

The plots shown in Figure 7.4 and Figure 7.5 represent the results relative to the MIM classifier and the NB classifier respectively. Each bar shows the average difference in predictive accuracy. A positive value for an algorithm indicates that the MIM or NB classifier performed better on the CADA and LEEDS data sets.

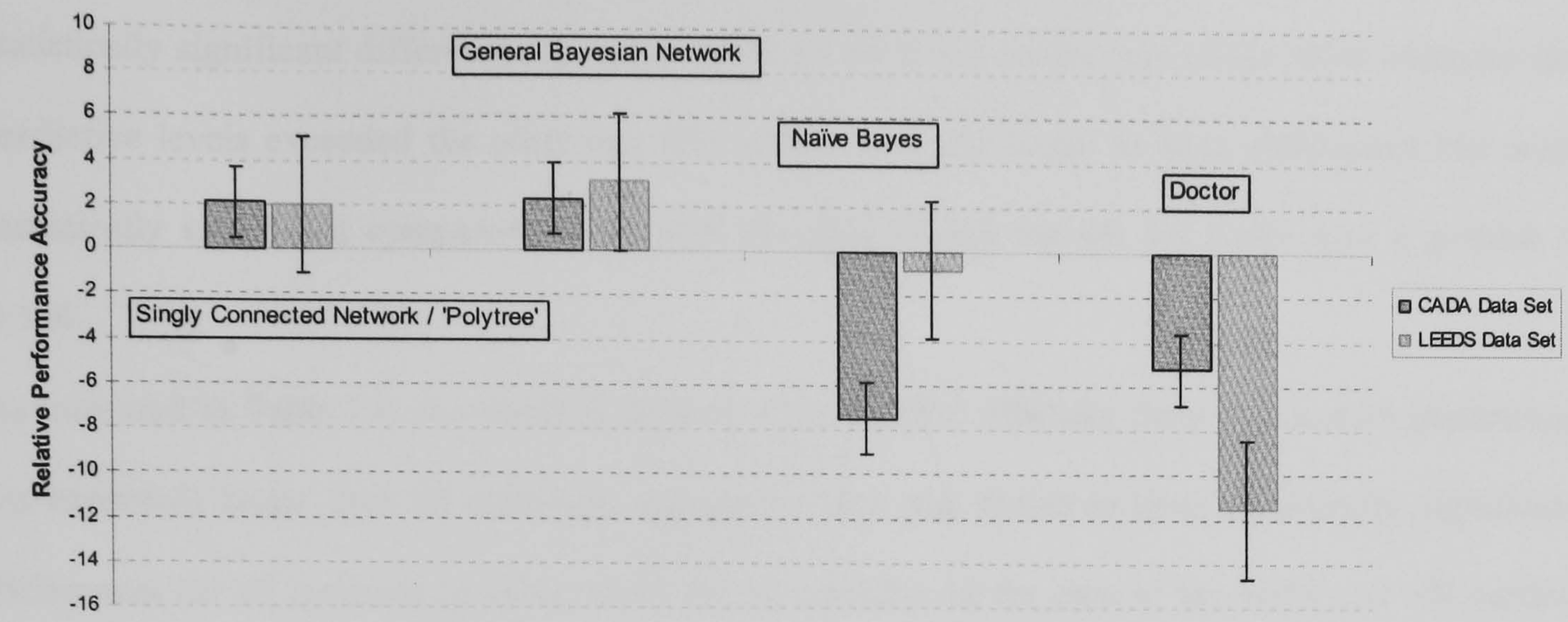


Figure 7.4. Predictive Accuracy relative to MIM Classifier.

The error bars represent the Post Hoc Tukey comparisons with overall 95% confidence for the relative differences.

For the CADA database, Table 7.3, the NB classifier has the best predictive accuracy of the four 'non-selective' models used in the study. This includes the 'overall' performance achieved by the doctors, and has statistically significant differences in all applications (p-value <0.05 compared to the MIM classifier, p-value <0.05 compared to 'polytree', and p-value <0.05 compared to GBN) except in the case of the doctors with a p-value = 0.137.

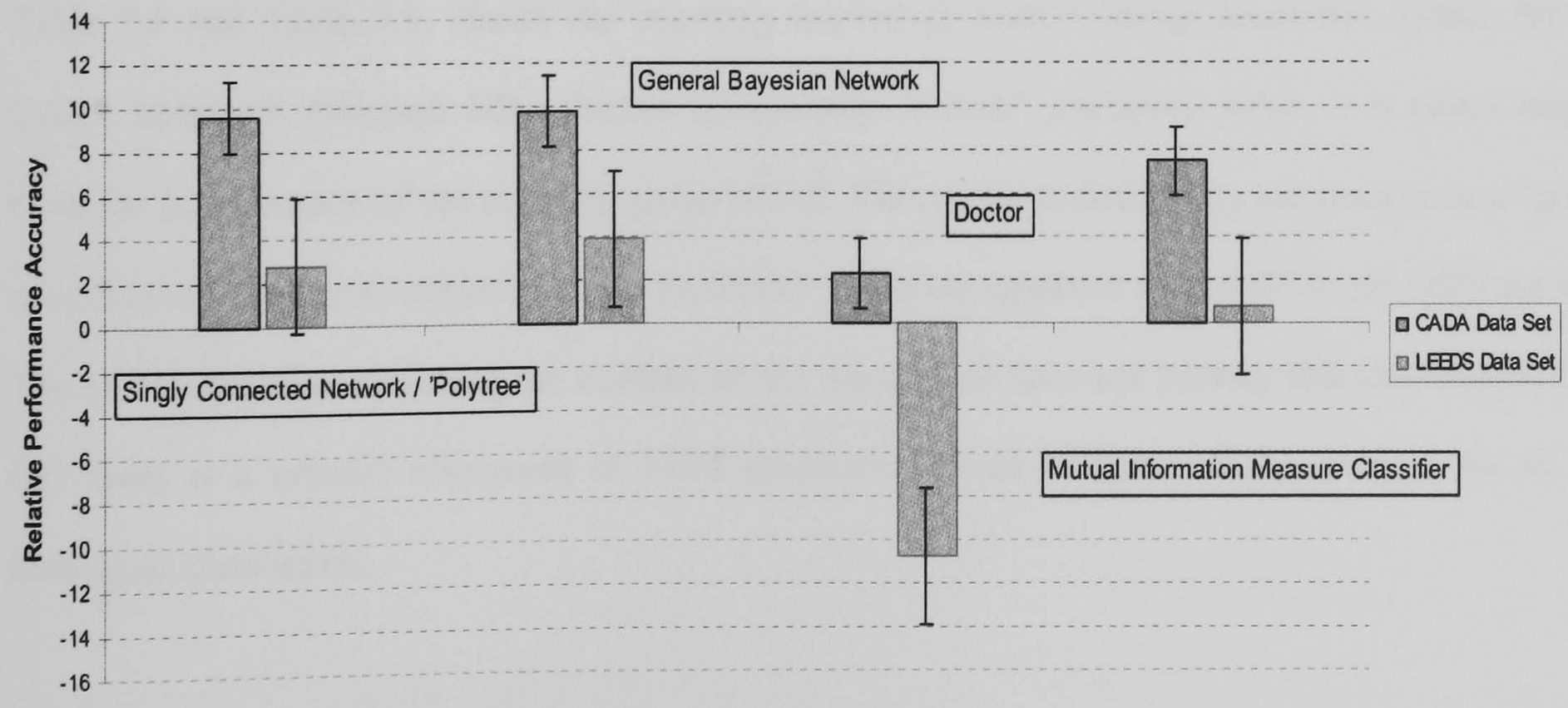


Figure 7.5. Predictive Accuracy relative to NB Classifier.

The doctors achieved the greatest predictive accuracy compared to all the BN models, and also has statistically significant differences (p-value <0.05 for all BNs). In the case of the MIM classifier the predictive levels exceeded the other two BN models and was found to have differences that were statistically significant compared to the SCN (p-value <0.05) but not the GBN with a p-value = 0.164.

As indicated in Table 7.4, the result in respect of the LEEDS database show the doctors performing (as expected) better than all statistical approaches and was found to have statistically significant differences for all methods (p-value <0.05 for all models). In the case of the MIM and NB models the predictive accuracy was found to be comparable, with the NB having differences that were statistically significant in respect of the GBN model (p-value <0.05) and the MIM classifier statistically significant differences for the GBN (p-value = 0.006) but not the SCN with a p-value = 0.105.

In deriving the structures of the BN models, Figure 7.1, Figure 7.2, and Figure 7.3, we identified some symptom-symptom relationships, which appeared meaningless probably due to some 'commonality' or 'correspondence' rather than causal interpretation. Examples are : Pain-site present / Pain-site Onset / Site of Tenderness, Vomiting / Nausea, and Previous Surgery / Abdominal Scar. Gammerman and Luo [GL91] also observed this commonality. Whilst these were identified by all the BN models the NB model lost these relationships due to the assumptions of conditional independence.

Table 7.5 and Table 7.6, shows the resulting individual disease group predictive values for the CADA database. Although NB achieves the greatest 'overall' predictive value, it is based largely upon the performance of the majority group NSAP. This group is defined by the doctors as a 'group of exclusion'. The same majority group predictive levels are apparent for the SCN and GBN models. The MIM Classifier including the doctors on the other hand does not portray this characteristic. In fact there is a relative alignment of MIM predictive scores to those of the doctors for all the individual class-states.

Table 7.5a: CADA Predictive Values

Final Diagnosis	# Cases	% Cases	MIM Predictive Value %	Doc Predictive Value %	NB Predictive Value %	GBN Predictive Value %	Poly Predictive Value %
APP	385	13.239	77.468	79.870	76.234	59.156	51.688
DIV	92	3.1637	68.750	48.913	53.261	15.489	26.902
PPU	56	1.9257	76.786	67.857	66.071	57.143	10.714
NSAP	1100	37.827	62.386	66.273	80.136	82.159	84.818
CHO	235	8.0812	62.340	68.936	59.1489	53.192	48.830
INO	228	7.8404	56.908	76.864	64.474	54.167	45.833
PAN	72	2.4759	42.014	62.500	31.944	12.153	2.7778
RCO	287	9.8693	81.969	84.321	81.446	63.415	62.108
DYS	453	15.578	66.391	75.055	75.994	54.415	67.881

Table 7.5b: CADA Likelihood Ratios

Final Diagnosis	# Cases	% Cases	MIM Likelihood Ratio	Doc Likelihood Ratio	NB Likelihood Ratio	GBN Likelihood Ratio	Poly Likelihood Ratio
APP	385	13.239	19.612	15.359	23.424	12.614	11.681
DIV	92	3.1637	30.777	38.396	28.849	14.830	17.469
PPU	56	1.9257	63.606	92.337	67.537	79.444	18.008
NSAP	1100	37.827	4.4545	3.9183	7.0518	4.7352	5.7289
CHO	235	8.0812	19.229	28.078	20.034	16.325	13.925
INO	228	7.8404	14.849	37.877	18.861	14.529	8.7469
PAN	72	2.4759	12.980	55.521	11.746	9.8565	6.3626
RCO	287	9.8693	33.920	51.697	41.236	19.313	18.709
DYS	453	15.578	12.660	18.989	17.033	7.4093	11.283

Table 7.6a: CADA Specificity value

Final Diagnosis	# Cases	% Cases	MIM Specificity	Doc Specificity	NB Specificity	GBN Specificity	Poly Specificity
APP	385	13.239	0.9649	0.9697	0.9642	0.9397	0.930
DIV	92	3.1637	0.9894	0.9832	0.9846	0.9729	0.9764
PPU	56	1.9257	0.9953	0.993	0.9933	0.9916	0.9827
NSAP	1100	37.827	0.8047	0.7994	0.8820	0.8669	0.8866
CHO	235	8.0812	0.9670	0.9749	0.9646	0.9595	0.9558
INO	228	7.8404	0.9632	0.9801	0.9695	0.9611	0.9534
PAN	72	2.4759	0.9848	0.9903	0.9825	0.9779	0.9758
RCO	287	9.8693	0.9793	0.9828	0.9802	0.9607	0.9593
DYS	453	15.578	0.9395	0.9539	0.9557	0.9175	0.9406

Table 7.6b: CADA Sensitivity values

Final Diagnosis	# Cases	% Cases	MIM Sensitivity	Doc Sensitivity	NB Sensitivity	GBN Sensitivity	Poly Sensitivity
APP	385	13.239	0.6876	0.5151	0.8380	0.7604	0.8156
DIV	92	3.1637	0.3260	0.6360	0.4434	0.4014	0.4125
PPU	56	1.9257	0.299	0.5846	0.4540	0.6667	0.3117
NSAP	1100	37.827	0.8698	0.7506	0.8330	0.630	0.6499
CHO	235	8.0812	0.6356	0.7606	0.7092	0.6605	0.6153
INO	228	7.8404	0.5463	0.7473	0.5759	0.5646	0.4074
PAN	72	2.4759	0.1967	0.5310	0.2058	0.2174	0.1539
RCO	287	9.8693	0.7014	0.8381	0.8147	0.7599	0.7609
DYS	453	15.578	0.7662	0.7460	0.7553	0.6109	0.6703

Clearly, the doctors are reluctant to assign a patient to a class-state which essentially represents 'don't know'. This reduced level of predictive value for NSAP was also observed in the G&T system [GG90] where it too had difficulty in identifying the majority group NSAP. One explanation of this may be due to the technique used by the G&T and MIM models to classify new observations. In the NB model only features presented as 'present' are used to obtain appropriate probabilities for the calculation of each class probability. For the G&T and the MIM classifier the 'absent' feature values are also used. Although not explicitly observed within the feature vector itself, they contribute to the overall calculation of the final probabilities in respect of the possible class outcomes. The G&T model uses both symptom (present) and ¬ symptom (absent) in determining relevant combinations, whilst the MIM model branch weights $I(X_1, X_2)$, relate to all $X_1 \, and \, X_2$ (present) parameter values which includes the $\neg X_1 \, and \, \neg X_2$ (absent) values. For NSAP the distribution of features is more generalised as it does not actually characterise a 'real' disease (or at least a single group). This means for NSAP identification, both the MIM and G&T models use of the 'absent' feature values will have the effect of reducing the calculated individual class probabilities used to discriminate between the disease groups. This in turn will increase the possibility of misclassification. Since NB only uses the 'present' feature values there is less or no reduction in this class probability and as NSAP is a generalised characterisation, this means it will capture a significant number of the observations more readily.

From Table 7.5 and Table 7.6, the doctor's individual group predictive values are 6/9 better than NB. The MIM classifier (best of BN models) similarly achieves 6/9 groups better than NB. For the LEEDS database Table 7.7 and Table 7.8, the doctor's performance is optimal at 9/9 compared to NB, with the MIM classifier achieving 5/9 group predictive values better than NB.

In addition, from Table 7.5 and Table 7.6, despite NB's 'overall' performance in respect of the CADA database exceeding that of the doctors, the Likelihood Ratio is in general lower for NB than the doctors. This indicates that the doctors have a greater ability to discriminate between the disease groups. The Likelihood Ratio for the LEEDS database, Table 7.7 and Table 7.8, shows a similar result, however for this particular data set the doctors are already overall winners.

Table 7.7a: LEEDS Predictive Value

Final Diagnosis	# Cases	% Cases	MIM Predictive Value %	Doc Predictive Value %	NB Predictive Value %	GBN Predictive Value %	Poly Predictive Value %
APP	1213	22.58	65.95	78.07	62.86	46.74	39.28
DIV	222	4.130	59.91	54.05	45.72	17.57	30.63
PPU	150	2.790	68.67	75.33	58	47.33	16.00
NSAP	1944	36.18	42.77	52.67	52.67	64.12	70.63
CHO	555	10.33	60.09	73.15	51.89	51.17	58.83
INO	338	6.290	54.59	75.15	60.65	50.00	42.31
PAN	224	4.170	32.14	56.25	28.57	12.50	4.910
RCO	377	7.020	67.37	79.84	70.69	44.30	48.54
DYS	350	6.510	58.86	70.86	62.86	51.14	58.29

Table 7.7b: LEEDS Likelihood Ratios

Final Diagnosis	# Cases	% Cases	MIM Likelihood Ratio	Doc Likelihood Ratio	NB Likelihood Ratio	GBN Likelihood Ratio	Poly Likelihood Ratio
APP	1213	22.58	7.360	8.720	6.830	4.980	4.750
DIV	222	4.130	16.15	36.90	17.05	9.570	16.88
PPU	150	2.790	23.74	86.61	12.71	10.05	10.53
NSAP	1944	36.18	2.980	2.830	2.920	2.580	2.460
CHO	555	10.33	16.22	23.80	13.78	11.88	16.36
INO	338	6.290	11.04	38.74	13.57	11.11	6.840
PAN	224	4.170	5.670	30.400	7.520	5.630	4.090
RCO	377	7.020	22.81	49.640	29.23	13.91	15.38
DYS	350	6.510	16.70	30.810	19.85	9.950	13.27

One reason why NB may outperform BN models is the possibility of high problem dimensionality. In both the CADA and LEEDS databases the number of domain variables is high, however for some of the disease groups there is very little data in order to adequately learn the model. As a consequence overfitting may occur due to spurious dependencies and unreliable probability estimates. The use of 'tree' structures may alleviate this problem as they offer less complex structures. This is demonstrated by the results shown in Table 7.3 and Table 7.4 where the MIM classifier performs better than the GBN model. In respect of the SCN, although a 'tree' structure, it is dependent upon edge directionality and full recovery from data alone is not always possible [Pea88]. From the results obtained node ordering seems to have had an effect on its final predictive performance.

Table 7.8a: LEEDS Sensitivity values

Final Diagnosis	# Cases	% Cases	MIM Sensitivity	Doc Sensitivity	NB Sensitivity	GBN Sensitivity	Poly Sensitivity
APP	1213	22.58	0.7149	0.6078	0.7146	0.7048	0.7399
DIV	222	4.130	0.2923	0.7229	0.4012	0.3333	0.4964
PPU	150	2.790	0.2269	0.6175	0.1652	0.1610	0.2513
NSAP	1944	36.18	0.7721	0.6750	0.6910	0.5675	0.5170
CHO	555	10.33	0.7306	0.7355	0.7385	0.6521	0.7567
INO	338	6.290	0.3498	0.6530	0.3741	0.3811	0.2750
PAN	224	4.170	0.1739	0.5780	0.2357	0.2106	0.1642
RCO	377	7.020	0.5695	0.7582	0.6481	0.5749	0.5894
DYS	350	6.510	0.4859	0.6310	0.5213	0.3499	0.3985

Table 7.8b: LEEDS Specificity values

Final Diagnosis	# Cases	% Cases	MIM Specificity	Doc Specificity	NB Specificity	GBN Specificity	Poly Specificity
APP	1213	22.58	0.9029	0.9303	0.8954	0.8586	0.8443
DIV	222	4.130	0.9819	0.9804	0.9765	0.9652	0.9706
PPU	150	2.790	0.9905	0.9929	0.9870	0.9840	0.9761
NSAP	1944	36.18	0.7410	0.7614	0.7636	0.7804	0.7899
CHO	555	10.33	0.9550	0.9691	0.9464	0.9451	0.9538
INO	338	6.290	0.9683	0.9831	0.9724	0.9657	0.9598
PAN	224	4.170	0.9694	0.9810	0.9686	0.9626	0.9599
RCO	377	7.020	0.9750	0.9847	0.9778	0.9587	0.9617
DYS	350	6.510	0.9709	0.9795	0.9737	0.9648	0.9699

Since the MIM model is not constrained by node ordering it provides an ideal middle ground between the NB and BN approaches.

As demonstrated by the use of the LEEDS database, which represents a truly 'external' test sample, the NB and the MIM classifier 'overall' predictive performance was comparable. Individually the MIM classifier performed better in identifying disease groups compared to the NB. For the CADA database 67% of the disease groups were identified by the MIM classifier compared to NB with NB's 'overall' predictive performance reflected by the majority group NSAP. In the case of the LEEDS database this was not the case.

The group NSAP is not a 'real' group and its sample distribution is thus a generalisation that represents several sub-groups. For the CADA and LEEDS data sets this 'characterisation' may differ due to geographical population anomalies. As the classifier models are derived from the CADA sample set, the corresponding CADA test samples will be classified better because they have a similar 'characterisation' and sample distribution. However, for the LEEDS test samples NSAP will have, in general terms, some similar aspects but on the whole be sufficiently different to make classification of NSAP samples harder to identify using the CADA generated models. Clearly from the results, the remaining eight 'real' disease groups have a 'common' and well characterised description and so their predictive performance, in the case of the CADA and LEEDS databases, align relatively well.

Table 7.9 details the symptom parameters that the experts look for when deciding to assign a patient to NSAP[23]. Those marked by a (x) are new items not recorded within the existing 135 points defining the CADA database. These additional NSAP identifying group characteristics provide the doctors with an advantage that the statistical Classifiers do not have and perhaps explains the doctor's performance. As a general point, doctors found that patients, on most occasions, who were assigned to NSAP got better within 24 hours.

---

[23] No information was supplied for the groups DIV, PAN and RCO

Table 7.9: 'NSAP' Identification Parameters – Suggested by the Experts

Diagnostic Group	Symptom Parameters
APP	Very High temperature (x) No increase in pulse rate (x) History of viral illness (x) Normal Urine Normal white cell count (x) Absence of rebound tenderness Absence of anorexia Symptoms spontaneously reoccur over 24-48 hours Changing Pain site Patient looks well Pain not progressively worsening No rigidity
PPU	Pain reducing
CHO	Other pathology ruled out
INO	Normal blood level white cell count (x) Normal biochemistry (x) Normal X-ray abdomen No guarding No rigidity No mass All investigations normal, no associated symptoms Normal Urine
DYS	Abdomen pain settles within 24 hours Signs and symptoms don't fit any of the other groups

From the discriminant matrices, Tables 7.10 to 7.14, in conjunction with, Table 7.7 and Table 7.8.

The following are observed in respect of the LEEDS database.

In general, the high frequency group misclassifications are lost to other high frequency groups (similarly observed in CADA). Low frequency group misclassifications for the SCN and GBN are lost to high frequency groups, however for MIM and NB these are generally lost to low frequency groups. The exception is the disease group diverticulitis (DIV) whose misclassification is directed to a high frequency group.

The predictive value for MIM is generally higher than those of NB, GBN and SCN for the high frequency groups with the exception of NSAP (similarly observed in CADA). In the case of the low frequency groups the MIM predictive values are greater than those obtained by NB, GBN and SCN. From Table 7.7 and Table 7.8, the sensitivity values for disease groups perforated peptic ulcer (PPU) and pancreatitis (PAN) are lower than those of the doctors for CADA, Table 7.5 and Table 7.6. Clearly, the doctors have used some heuristics to diagnose groups perforated peptic ulcer and pancreatitis as it is known that the group pancreatitis in particular has a very poor data definition stored within the database.

From the CADA discriminant matrices, Tables 7.15 to 7.19, in conjunction with Table 7.5 and Table 7.6. The following are observed in respect of the CADA database.

The predictive value for MIM is greater than that of the NB for low frequency groups. In most cases, MIM and NB values are higher than those of GBN and SCN. For high frequency groups the predictive value for NB and MIM are similar, with the exception of NSAP, where the MIM and doctors levels fall below those of the SCN, GBN and NB.

On the whole, high frequency groups misclassify into other high frequency groups for the NB, SCN and GBN, whereas for the MIM, this model misclassifies into low frequency groups. From Table 7.5 and Table 7.6, the sensitivity values for disease groups perforated peptic ulcer and pancreatitis are lower compared to those of the doctors, again illustrating the doctor's use of heuristics.

Table 7.10: Doctors Discriminant Matrix (LEEDS)

	APP	DIV	PPU	NSAP	CHO	INO	PAN	RCO	DYS	TOTAL
APP	947	9	5	229	5	6	1	5	6	1213
DIV	23	120	5	41	3	23	1	4	2	222
PPU	6	0	113	15	6	1	6	0	3	150
NSAP	539	23	16	1024	76	64	36	77	89	1944
CHO	10	3	10	52	406	23	23	5	23	555
INO	14	6	6	46	2	254	3	3	4	338
PAN	3	2	20	17	26	10	126	2	18	224
RCO	12	2	0	53	7	1	1	301	0	377
DYS	4	1	8	40	21	7	21	0	248	350
TOTAL	1558	166	183	1517	552	389	218	397	393	5373

Table 7.11: MIM Classifier Dsicriminant Matrix (LEEDS)

	APP	DIV	PPU	NSAP	CHO	INO	PAN	RCO	DYS	TOTAL
APP	800	60	85	150	8	61	20	19	10	1213
DIV	8	133	18	9	3	32	5	11	3	222
PPU	3	5	103	3	7	9	16	-	4	150
NSAP	267	154	87	831	55	155	122	130	143	1944
CHO	3	26	49	13	334	21	72	9	28	555
INO	16	45	29	11	6	185	26	6	14	338
PAN	5	8	48	6	28	37	72	6	14	224
RCO	15	16	21	34	6	10	18	254	3	377
DYS	3	8	14	20	10	17	62	10	206	350
TOTAL	1120	455	454	1077	457	527	413	445	425	5373

Table 7.12: NB Classifier Discriminant Matrix (LEEDS)

	APP	DIV	PPU	NSAP	CHO	INO	PAN	RCO	DYS	TOTAL
APP	763	18	162	226	3	26	4	6	5	1213
DIV	9	102	18	55	2	27	2	4	3	222
PPU	13	7	87	4	9	7	21	-	2	150
NSAP	240	79	108	1024	46	178	60	109	100	1944
CHO	6	7	46	31	288	35	75	12	55	555
INO	16	24	28	32	3	205	13	8	9	338
PAN	5	2	47	18	26	31	64	6	25	224
RCO	14	12	8	56	3	10	4	267	3	377
DYS	1	3	23	36	11	28	28	-	220	350
TOTAL	1067	254	527	1482	391	547	271	412	422	5373

Table 7.13: GBN Classifier Discriminant Matrix (LEEDS)

	APP	DIV	PPU	NSAP	CHO	INO	PAN	RCO	DYS	TOTAL
APP	567	20	159	391	21	23	6	9	17	1213
DIV	13	39	12	92	6	43	5	5	7	222
PPU	25	2	71	15	13	7	8	1	8	150
NSAP	145	29	77	1246	52	126	22	78	169	1944
CHO	7	6	34	102	284	27	26	13	56	555
INO	19	10	16	80	10	169	11	5	18	338
PAN	15	5	31	50	34	18	28	3	40	224
RCO	10	4	26	125	6	12	8	167	19	377
DYS	4	2	14	97	9	18	18	9	179	350
TOTAL	805	117	440	2198	435	443	132	290	513	5373

Table 7.14: SCN – 'polytree' Classifier Discriminant Matrix (LEEDS)

	APP	DIV	PPU	NSAP	CHO	INO	PAN	RCO	DYS	TOTAL
APP	477	7	8	680	2	28	2	3	6	1213
DIV	5	68	6	81	2	43	-	11	6	222
PPU	11	2	24	43	12	28	13	2	15	150
NSAP	123	32	9	1373	41	141	14	79	132	1944
CHO	4	3	14	74	327	46	9	13	65	555
INO	11	18	11	112	6	143	6	8	23	338
PAN	3	-	17	50	29	50	11	5	59	224
RCO	7	4	3	156	2	15	4	183	3	377
DYS	3	2	4	85	11	26	8	7	204	350
TOTAL	644	136	96	2654	432	520	67	311	513	5373

Table 7.15 : MIM Classifier Discriminant Matrix (CADA)

	APP	DIV	PPU	NSAP	CHO	INO	PAN	RCO	DYS	TOTAL
APP	298	7	13	46	3	5	4	7	2	385
DIV	2	63	5	1	3	13	1	2	2	92
PPU	-	-	43	1	3	3	5	-	1	56
NSAP	111	60	16	687	27	47	30	75	47	1100
CHO	1	9	13	6	147	16	17	7	19	235
INO	11	25	13	16	4	130	18	3	8	228
PAN	1	6	12	2	3	5	30	1	12	72
RCO	7	6	6	11	7	5	9	235	1	287
DYS	3	17	23	19	34	13	40	3	301	453
TOTAL	434	193	144	789	231	237	154	333	393	2908

Table 7.16 : NB Classifier Discriminant Matrix (CADA)

	APP	DIV	PPU	NSAP	CHO	INO	PAN	RCO	DYS	TOTAL
APP	294	2	5	73	2	2	2	1	4	385
DIV	-	49	5	19	1	12	1	2	3	92
PPU	1	-	37	-	4	2	11	-	1	56
NSAP	36	28	5	880	12	47	8	45	39	1100
CHO	-	1	6	11	139	16	26	3	33	235
INO	8	14	6	22	6	147	15	-	10	228
PAN	2	3	9	1	7	7	23	2	18	72
RCO	7	4	1	25	6	6	2	234	2	287
DYS	3	9	8	27	20	17	23	2	344	453
TOTAL	351	110	82	1058	197	256	111	289	454	2908

Table 7.17 : GBN Classifier Discriminant Matrix (CADA)

	APP	DIV	PPU	NSAP	CHO	INO	PAN	RCO	DYS	TOTAL
APP	228	2	3	133	3	7	1	2	6	385
DIV	4	15	1	42	2	20	1	3	4	92
PPU	7	-	32	3	6	2	2	-	4	56
NSAP	37	6	4	904	13	34	7	37	58	1100
CHO	1	2	3	42	125	7	10	3	42	235
INO	6	4	2	66	5	124	5	3	13	228
PAN	5	1	2	22	10	7	9	-	16	72
RCO	6	1	1	69	8	5	1	182	14	287
DYS	6	4	1	153	17	14	2	9	247	453
TOTAL	300	35	49	1434	189	220	38	239	404	2908

Table 7.18: Doctors Discriminant Matrix (CADA)

	APP	DIV	PPU	NSAP	CHO	INO	PAN	RCO	DYS	TOTAL
APP	308	2	1	66	1	1	3	2	1	385
DIV	1	45	6	23	3	12	-	1	1	92
PPU	1	-	38	5	5	2	1	-	4	56
NSAP	271	13	1	729	12	26	4	24	20	1100
CHO	1	2	4	24	162	7	13	2	20	235
INO	5	3	3	26	1	179	3	1	7	228
PAN	-	2	3	10	4	4	45	1	3	72
RCO	6	-	-	27	4	1	1	247	1	287
DYS	3	3	9	58	19	5	15	1	340	453
TOTAL	596	70	65	968	211	237	85	279	397	2908

Table 7.19: SCN – 'polytree' Classifier Dsicriminant Matrix (CADA)

	APP	DIV	PPU	NSAP	CHO	INO	PAN	RCO	DYS	TOTAL
APP	199	1	1	172	2	4	2	-	4	385
DIV	1	25	2	31	3	25	-	3	2	92
PPU	1	2	6	9	7	17	1	-	13	56
NSAP	30	11	1	933	16	35	1	33	40	1100
CHO	1	-	4	34	115	27	1	11	42	235
INO	4	13	2	68	6	105	3	5	22	228
PAN	2	1	3	20	9	8	2	-	27	72
RCO	3	4	-	85	7	6	1	178	3	287
DYS	2	3	1	82	22	29	2	4	308	453
TOTAL	243	60	20	1434	187	256	13	234	461	2908

7.6.2 Results 'Selective' Experiments

Table 7.20 and Table 7.21 show the average prediction accuracies for the selective MIM and selective NB classifiers, for the CADA and LEEDS data sets respectively. As was the case in section 7.6.1, each table entry describes the average accuracy along with the sample deviation. For comparison, the resulting accuracies for the MIM classifier, NB and doctors have also been included.

Table 7.20: Average Predictive Accuracy 'CADA' – error rates

SMIM	SNB	Default (overall)	Doctor	MIM	NB
0.2937±1.04	0.2906±1.49	0.5495	0.2834±0.28	0.3349±0.82	**0.2617±1.16**

Key: MIM –Mutual Information Measure Classifier, **NB** – Naive Bayes Classifier, **SMIM** – Selective MIM Classifier, **SNB** Selective NB Classifier. Values in **bold** type indicate the highest model performance achieved by the classifier in respect of the CADA database.

Table 7.21: Average Predictive Accuracy 'LEEDS' – error rates

SMIM	SNB	Default (overall)	Doctor	MIM	NB
0.4336±0.11	0.4395±1.32	0.6382	**0.3413±0.0**	0.4569±0.52	0.4489±0.53

Key: MIM –Mutual Information Measure Classifier, **NB** – Naive Bayes Classifier, **SMIM** – Selective MIM Classifier, **SNB** Selective NB Classifier. Values in **bold** type indicate the highest model performance achieved by the classifier in respect of the LEEDS database.

The plots in Figure 7.6 and Figure 7.7 represent the performance results relative to the MIM classifier and the NB classifier for the two selective variants. Whilst Figure 7.8. shows the predictive accuracy of the SNB relative to the SMIM classifier. The error bars are the Post Hoc Tukey comparisons with overall 95% confidence for the relative differences.

The SMIM performed better than the MIM classifier in both the CADA and LEEDS data set with the differences found to be statistically significant in both. (CADA p-value = 0.014, LEEDS p-value = 0.001). The NB classifier did not have differences that were statistically significant in respect of the LEEDS data set compared to the MIM classifier with a p-value = 0.28, and this was also the case when compared to the SMIM classifier with a p-value = 0.965, however unlike the MIM classifier the SMIM was marginally better in predictive performance than NB.

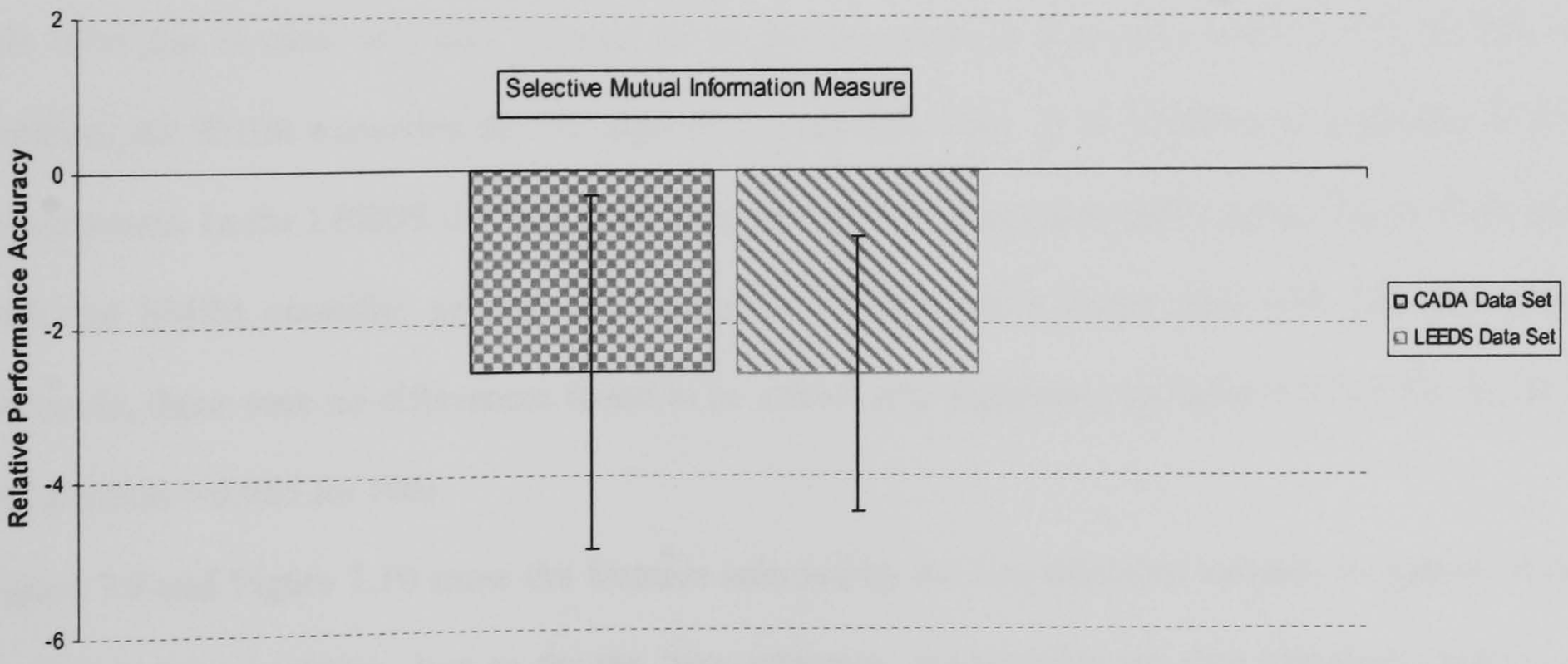


Figure 7.6. Predictive Accuracy relative to MIM Classifier (SMIM).

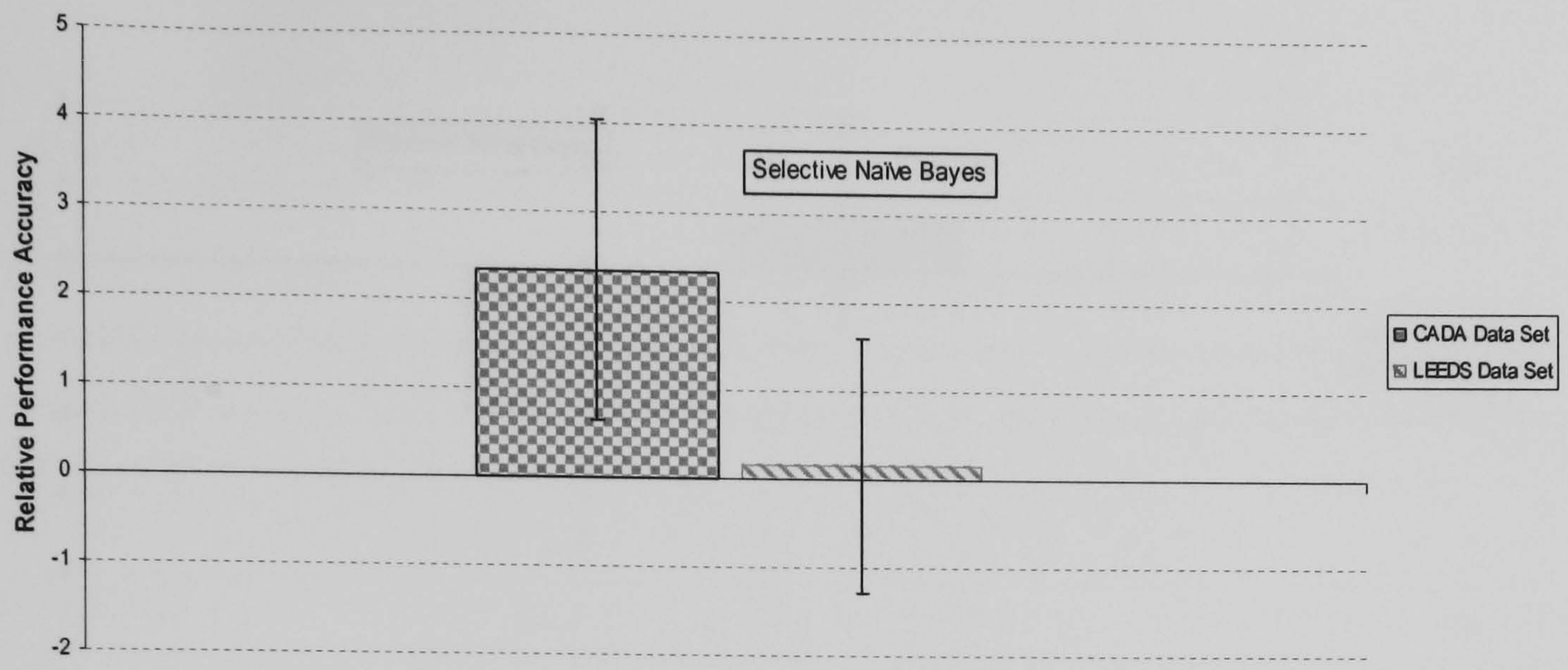


Figure 7.7. Predictive Accuracy relative to NB Classifier (SNB).

For the CADA data set although NB has a better predictive accuracy to that of the doctors, it did not have differences that were statistically significant, with p-value = 0.221. As previously observed in section 7.6.1 when compared to the MIM classifier, the NB maintained its predictive performance outperforming the SMIM with differences found to be statistically significant with a p-value < 0.05. Despite the increase in predictive accuracy of the SMIM, overall it did not exceed that achieved by NB. However, in respect of the SNB, its performance was found to be almost equal. During our investigation we observed that the optimised SNB required 20 features of the 33 that NB used, but this reduction in class MB also reduced its predictive accuracy from that achieved by the NB. In contrast, the SMIM expanded the CL algorithm class MB from 15 to 30 features, matching SNB's performance. In the LEEDS data set there were no obvious changes in performance for the NB/SNB with the SMIM classifier performance found to be marginally better than both NB and SNB. However, there were no differences found to be statistically significant (p-value = 0.727 for the SNB and p-value = 0.965 for NB).

Figure 7.9 and Figure 7.10 show the features selected by the two selective variants in respect of the CADA training partition. Just as for the 'non-selective' experiments we also calculated additional statistics as shown in Table 7.22 and 7.23.

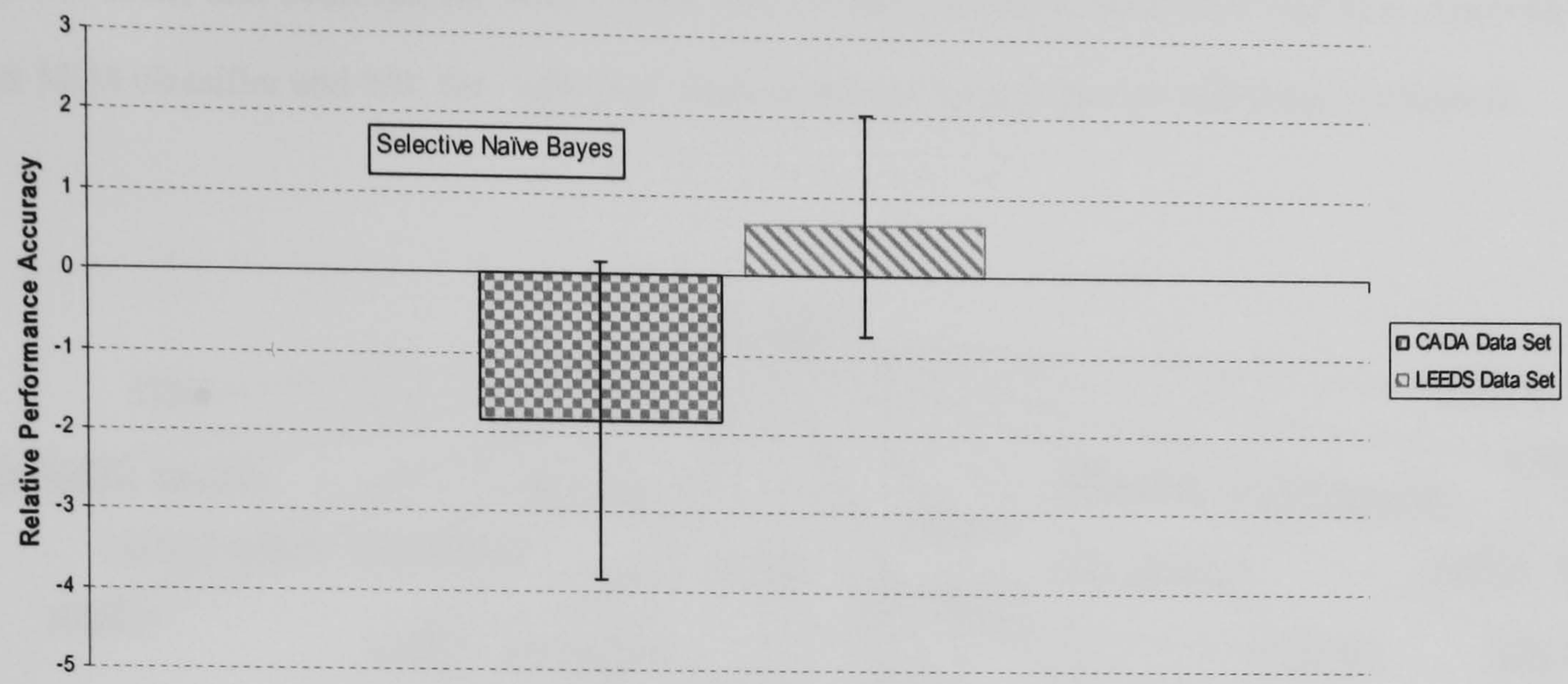


Figure 7.8. Predictive Accuracy relative to SMIM Classifier (SNB).

For the individual diseases the SMIM identified 7/9 better than the SNB. This was two groups better (INO/RCO) than that achieved by the MIM classifier compared to the NB when tested with the LEEDS data set.

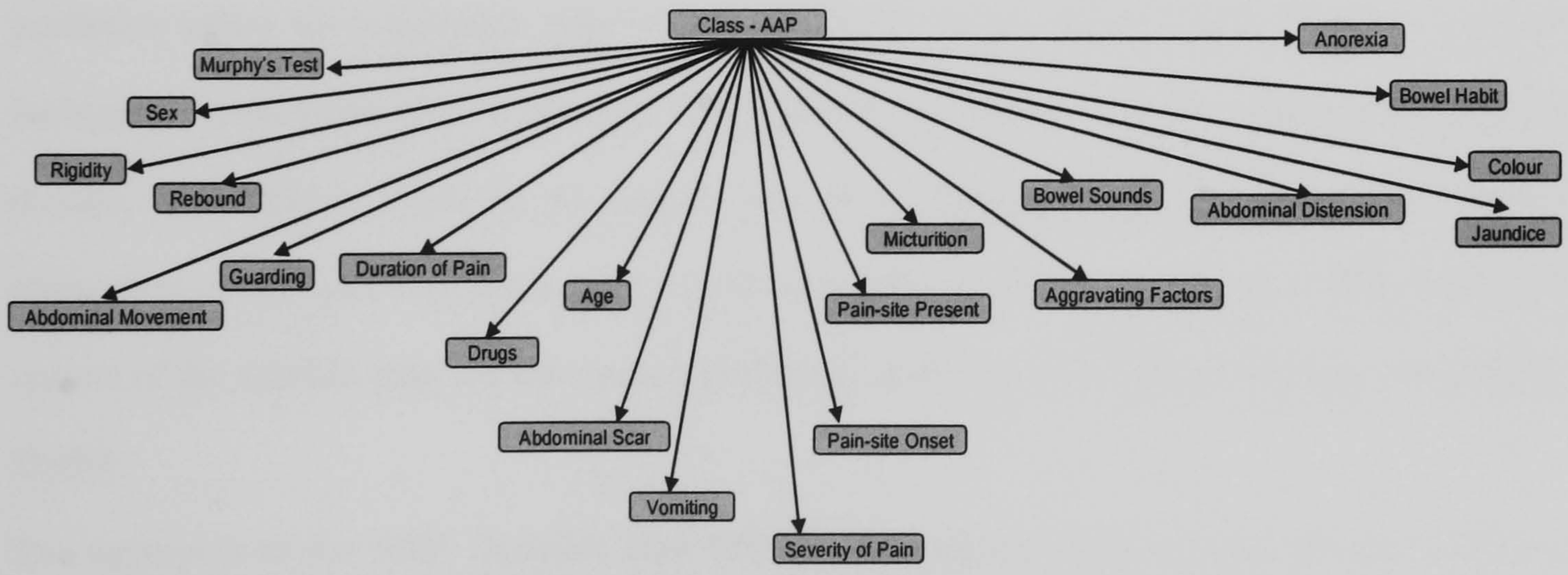


Figure 7.9. Selective Naive Bayes (SNB) Structure

For the CADA data set, this remained at 6/9 as previously achieved by the MIM classifier when compared to NB. From the Tables 7.22 and 7.23 and Tables 7.24 to 7.27, the discriminant matrices, the SMIM misclassifies high frequency diseases to other high frequency diseases. This was the same

for the SNB, and observed for both CADA and LEEDS data sets. Since this was also observed for the MIM classifier and NB, the 'selective' variants appear to have had no additional influences.

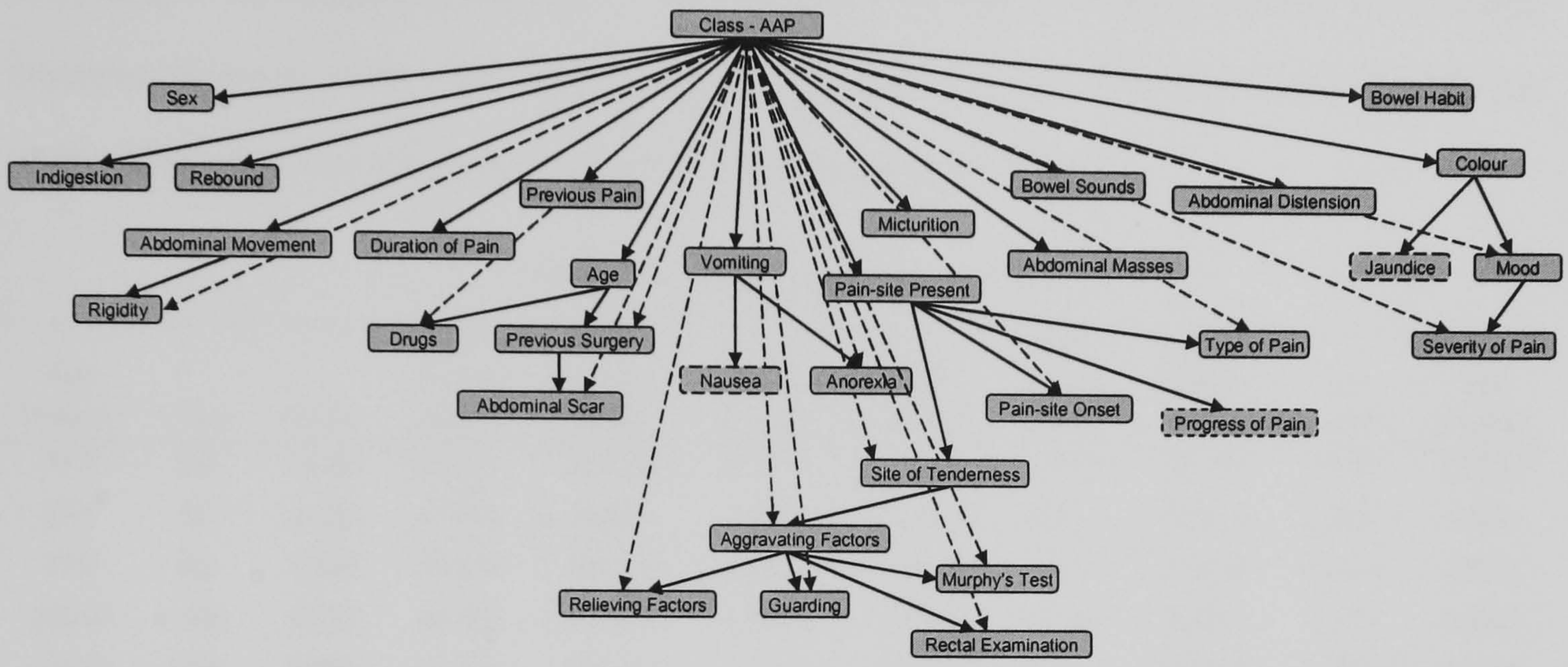


Figure 7.10. Selective Mutual Information Measure (dashed lines) Structure

The individual disease predictive values for the SMIM, Tables 7.22 and 7.23, are generally higher than those of the SNB in respect of the low frequency diseases, whereas for high frequency diseases, predictive values are comparable with SNB except for the disease group NSAP. This was observed for both the CADA and LEEDS data sets. When SMIM was compared to the results achieved by the doctors, the predictive levels for all diseases were found comparable except for PAN and internal obstruction (INO), with SMIM achieving a better appendicitis (APP) value for the CADA data set. In respect of the LEEDS data set, the doctors performed generally better for all diseases compared to SMIM.

The expansion of the MIM classifier class MB (SMIM) appears to have improved the predictive performance in both data sets. In the case of the SNB, although its performance was reduced for CADA there were no 'real' changes in performance for LEEDS. The NBs use of all features is supported by SMIM and probably implies the CL algorithm defined class MB was not the best for classifying this domain. As previously observed, the majority group NSAP contributed heavily to both the 'overall' predictive accuracy of NB and SNB. However, with respect to the individual

diseases, the MIM and SMIM classifiers identified more than both NB and SNB on the LEEDS 'external' test samples.

The improvement in performance of the MIM classifier, that is SMIM, is probably due to better modelling of the domain, resulting in an improvement in predicative accuracy for both low and high frequency diseases, whilst maintaining a comparable accuracy to NB. This was not observed for the SNB which resulted in a degradation of performance compared to NB.

Table 7.22: CADA Statistical values

Final Diagnosis	# Cases	% Cases	SMIM Predictive Value %	SMIM Likelihood Ratio	SMIM Sensitivity	SMIM Specificity	SNB Predictive Value %	SNB Likelihood Ratio	SNB Sensitivity	SNB Specificity
APP	385	13.24	82.251	25.9589	0.7189	0.9723	62.078	13.797	0.7749	0.9438
DIV	92	3.160	67.754	26.921	0.2961	0.9890	46.424	22.811	0.4013	0.9824
PPU	56	1.930	79.464	93.556	0.3853	0.9959	57.643	50.418	0.4224	0.9916
NSAP	1100	37.83	64.803	4.9076	0.8985	0.8169	81.506	6.4344	0.758	0.8821
CHO	235	8.080	61.135	21.110	0.7124	0.9663	61.702	20.579	0.6868	0.9666
INO	228	7.840	57.237	15.733	0.5724	0.9636	57.140	15.471	0.5647	0.9635
PAN	72	2.480	47.917	13.699	0.1885	0.9863	21.028	12.338	0.2465	0.9800
RCO	287	9.870	86.121	48.040	0.7429	0.9845	82.878	42.945	0.8075	0.9812
DYS	453	15.58	69.978	14.496	0.7869	0.9457	70.294	13.334	0.7265	0.9455

Table 7.23: LEEDS Statistical values

Final Diagnosis	# Cases	% Cases	SMIM Predictive Value %	SMIM Likelihood Ratio	SMIM Sensitivity	SMIM Specificity	SNB Predictive Value %	SNB Likelihood Ratio	SNB Sensitivity	SNB Specificity
APP	1213	22.58	67.752	7.8085	0.7214	0.9076	49.793	5.6221	0.7497	0.8667
DIV	222	4.130	65.165	21.477	0.3360	0.9844	38.288	15.094	0.4007	0.9735
PPU	150	2.790	75.222	26.118	0.2017	0.9923	61.427	13.905	0.1669	0.9880
NSAP	1944	36.18	44.865	3.1116	0.7830	0.7484	57.657	2.4368	0.5819	0.7612
CHO	555	10.33	62.913	18.219	0.7629	0.9581	60.386	16.459	0.7359	0.9553
INO	338	6.290	56.016	13.209	0.4007	0.9697	55.749	13.029	0.3978	0.9695
PAN	224	4.170	49.256	9.9785	0.2316	0.9768	20.027	6.7888	0.2347	0.9655
RCO	377	7.020	68.789	28.343	0.6689	0.9764	64.836	26.694	0.7041	0.9736
DYS	350	6.510	53.238	17.077	0.5549	0.9675	56.203	15.475	0.4781	0.9691

Table 7.24 : SMIM Classifier Discriminant Matrix (CADA)

	APP	DIV	PPU	NSAP	CHO	INO	PAN	RCO	DYS	TOTAL
APP	317	4	12	41	1	2	6	1	1	385
DIV	1	62	5	3	2	13	2	3	1	92
PPU	0	1	45	0	3	1	5	0	1	56
NSAP	100	77	11	711	16	50	22	69	44	1100
CHO	3	4	8	4	144	12	32	4	24	235
INO	9	37	10	11	2	131	18	5	5	228
PAN	1	4	11	1	5	4	35	2	9	72
RCO	8	9	2	8	5	5	3	247	0	287
DYS	3	10	11	14	25	12	59	2	317	453
TOTAL	442	208	115	793	203	230	182	333	402	2908

Table 7.25 : SNB Classifier Discriminant Matrix (CADA)

	APP	DIV	PPU	NSAP	CHO	INO	PAN	RCO	DYS	TOTAL
APP	239	2	7	122	2	6	1	1	5	385
DIV	2	43	2	20	3	16	1	3	2	92
PPU	1	5	32	2	4	3	5	1	3	56
NSAP	46	20	3	897	16	31	4	40	43	1100
CHO	1	3	6	13	145	16	12	4	35	235
INO	9	20	5	37	5	130	8	3	11	228
PAN	1	6	12	2	3	5	30	1	12	72
RCO	5	3	1	30	4	4	1	238	1	287
DYS	4	5	9	55	27	17	15	3	318	453
TOTAL	308	107	77	1178	209	228	77	294	430	2908

Table 7.26: SMIM Classifier Discriminant Matrix (LEEDS)

	APP	DIV	PPU	NSAP	CHO	INO	PAN	RCO	DYS	TOTAL
APP	822	35	168	128	2	40	10	5	3	1213
DIV	11	145	18	15	3	20	5	3	2	222
PPU	7	4	113	2	7	2	13	0	2	150
NSAP	265	158	98	872	59	157	125	100	110	1944
CHO	5	8	43	11	349	23	91	8	17	555
INO	13	49	30	17	3	189	24	8	5	338
PAN	4	4	49	9	15	19	110	3	11	224
RCO	13	22	20	37	2	8	15	259	1	377
DYS	0	5	19	23	18	15	83	1	186	350
TOTAL	1140	430	558	1114	458	473	476	387	337	5373

Table 7.27: SNB Classifier Discriminant Matrix (LEEDS)

	APP	DIV	PPU	NSAP	CHO	INO	PAN	RCO	DYS	TOTAL
APP	604	16	174	369	6	24	10	4	6	1213
DIV	7	85	19	64	3	36	2	4	2	222
PPU	12	2	92	5	10	9	16	0	4	150
NSAP	143	65	113	1221	47	126	38	73	118	1944
CHO	4	8	42	38	335	32	47	8	41	555
INO	14	19	26	56	7	188	10	7	11	338
PAN	7	3	47	25	32	33	45	5	27	224
RCO	13	9	12	82	3	6	4	244	4	377
DYS	2	4	25	67	14	19	20	2	197	350
TOTAL	806	211	550	1927	457	473	192	347	410	5373

7.6.2.1 Related Work – 'Selective'

Applying feature selection to reduce model dimensionality not only promotes a rise in interpretability and comprehensibility, but as these models require less input they can easily be used by clinicians. From our experimental results, Figure 7.2, we observed that the class variable had several parent nodes leading to a large number of probabilities to be both estimated and updated. This was also found to be the case by Luo [GL91] who constructed a SCN employing the same CL algorithm as used for building the MIM classifier. The approach taken to reduce dimensionality and thus make the class MB more tractable was to apply a Kullback χ^2 equivalent threshold to identify and remove symptoms found to be statistically irrelevant. Their experimental results suggested that there was no loss of performance, although when compared to NB it was out performed despite modelling domain dependencies. In contrast, the SMIM classifier adopts an expansion of the class MB and is not influenced by 'multi' parented class nodes. As demonstrated by the results shown in

Table 7.20 and Table 7.21, the increase in features with respect to the class MB actually results in an increase of performance. The Tables B1-B8, Appendix B, however indicate that the approach taken by Luo may not be valid. Whilst statistically symptoms may appear to be irrelevant, their removal from the class MB for all diseases is not appropriate. From the Tables B1-B8, Appendix B, the distribution of symptom values differs from disease to disease. This is further supported by the questionnaire responses supplied by experts identifying 'key' symptoms in respect of each disease. The removal from the class MB implying that a particular symptom is irrelevant for all nine diseases is clearly wrong.

To illustrate the effect of this approach we applied a similar Kullback threshold (χ^2 with a level of confidence set to 0.99) to the CADA data set. Table 7.28, shows the symptoms that were found to be statistically irrelevant. If we include in this table those symptoms that the experts have specified as important in assigning a disease to a patient, then it is easy to see that many 'key' items will be removed.

Table 7.28: Kullback Thresholding – Symptom/Disease Removal

Symptoms identified as statistically 'irrelevant'	**Doctors 'key' symptoms taken from Questionnaires**	**Diseases most appropriately associated with the symptom**
Progress of Pain	N/A	-
Duration of Pain	Duration of Pain	APP (12-24 Hours), DIV (>48 Hours), PPU (12-24 Hours), CHO (>48 Hours).
Severity of Pain	N/A	-
Nausea	N/A	-
Vomiting	Vomiting	CHO (present), INO (present), PAN (present), DYS (present)
	Vomiting	DIV (No Vomiting)
Anorexia	N/A	-
Colour	Jaundice	CHO (Jaundice)
Bowel Habit	Bowel Habit	INO (Constipated), DIV(Diarrhoea)
Micturition	Micturition	RCO (Haematuria)
Previous Pain	Previous Pain	DIV (Similar Pain Before), PPU (No Similar Pain Before)
Drugs	N/A	-
Rigidity	N/A	-
Abdominal Masses	Abdominal Masses	INO (present), DIV (present)

The approach taken by Luo represents a 'top-down' optimisation. Singh's Selective BN [Sin98] and Gu's G&T [GG90] system on the other hand adopt a 'bottom-up' selective approach. In these two methods symptoms are incrementally added to the model until the design criteria is satisfied. Unlike Luo's approach, symptoms deemed 'irrelevant' are excluded. In the case of the G&T system combinations of symptoms are determined for each disease and assigned a probability value for the corresponding grouping using Bayes theory. As was previously shown, each disease is represented by a varying 'set' of symptoms (values), which for the G&T represents a combination of symptom values. Essentially new evidence is matched to a particular 'combination' and its probability read-off. With nine possible values, one for each disease, the highest probability is used to classify a patient's symptoms into one of the mutually exclusive diseases. Like the MIM/SMIM classifier, both the observed and unobserved symptom values are used to evaluate a patient's illness. In the case of the G&T, unobserved symptom values are considered during the selection of symptom combinations, just as those actually recorded in the database. This is not to be confused by 'missing' which represents symptom values that have not been recorded at all, but the compliment of the recorded symptom value. For example, if an observed symptom has a value 'present' the unobserved value would be 'absent'. For the MIM/SMIM classifier, these values form part of the calculation of MI branch weights. As we discussed in section 7.6.1, this approach highlighted a limitation for both the MIM/SMIM classifier and the G&T system in classifying samples of NSAP.

In the G&T system the setting of the χ^2 threshold level determines the cut-off for 'combination size'. That is, the numbers of symptom values that will be considered statistically relevant that constitute a 'set' or group of symptoms with respect to a disease. This allows the system to optimise its 'selective' process and potentially increase its performance accuracy. Unlike the SMIM classifier, which can only expand its class MB, the G&T has some flexibility to both expand and reduce its symptom combination size. Despite the additional modelling of dependencies and flexibility, the NB classifier was found to outperform the G&T system. As was also reported by Singh [Sin98] the NB classifiers 'overall' performance was largely influenced by the performance of the majority group NSAP.

7.7 Discussion

Two of the key findings of this study are the effect of feature selection in this domain, and the use of restricted 'tree' based classifiers. The results show that 'trees' which use only a small fraction of the features (class MB specific), selected by using the CL algorithm, display a much higher accuracy compared to GBNs unrestricted modelling of all the features. This is more evident for the SMIM classifier which expands the class MB whilst maintaining the efficiency of the 'tree' based MIM classifier. Although not so obvious perhaps in performance levels, the 'polytree' also outperformed the GBN. In the case of the 'polytree', despite being computationally simpler than the GNB, it is sensitive to the choice of node ordering, which in terms of branch directionality means potentially large CPTs, and is still difficult to fully recover. As demonstrated by the results, the performance of the SCN seems to have been influenced by this choice, just as was observed for other datasets used in Chapter 4.

For the high frequency diseases the GBN performed comparably to NB/SNB at identifying them correctly and for some diseases marginally better. In the case of the low frequency diseases the GBN was poorer, with the NB and SNB (although SNB was slightly less than NB) much better. This was also observed for the 'polytree' which aligned with that of the GBN. This could possibly be attributed to the lack of sufficient data in this high dimensional domain.

For the diseases that occurred more frequently (NSAP, DYS, CHO) GBN/SCNs are able to easily detect the appropriate relationships and get accurate estimates of the parameters, thus yielding higher accuracies from these diseases. For the low frequency diseases GBNs in particular, pick up spurious dependencies with inaccurate parameter estimates, which in turn leads to a poor performance on these diseases (PAN, DIV, PPU). For the SCN, the 'tree' based approach should alleviate the problem however, the class MB will be dependant upon the node ordering and thus as the results indicate, only marginally outperforms the GBN.

The implication is that the performance of complex representations such as the GBN will probably improve in domains were there is sufficient available data. Nevertheless, for even the small amount of data, the experimental results show that the MIM/SMIM classifiers are viable alternatives to the extremely simple NB classifiers as well as the more complex GBNs. The results indicate that the SMIM classifier was comparable to the NB at correctly identifying low frequency diseases and to the

GBN for correctly identifying high frequency diseases. For the LEEDS (external) data set the SMIM classifier identified correctly 7/9 diseases compared to the SNB.

Supported by the results we observed that our approach to treating missing values as an extra value of the domain symptoms was probably appropriate for this domain. As stated by Friedman [Fri98], in some cases it is the very absence of a value that is informative, and not the actual missing value. This appears to be the case for AAP, which has many classes and even more features. As we have shown, several features are relevant only for certain diseases (Tables B1-B8, Appendix B) and, thus, their values may not be recorded in circumstances in which the examining doctor has considered them to be irrelevant or redundant in respect of the current known information. In this scenario, the particular value is no longer important. The implication is that the absence of a value for a particular feature should increase the likelihood of diseases for which that feature is not relevant (similarly decrease for which it is). Only by treating missing values as an extra value, can this relationship be readily modelled [Sin98]. Undoubtedly, the best way to deal with missing items is to use domain knowledge.

In the objectives, section 7.2, we stated some aims of the study we wanted to specifically investigate. From our experimental results it seems likely that the NB classifier is 'optimal' for this domain. However, this is only in respect of the 'overall' performance accuracy which in turn is due to the majority group NSAP (for CADA). When evaluated with 'truly' external data (LEEDS) the NB classifier still performed well, but was matched by the SMIM classifier which did not have an 'overall' performance influenced by NSAP. What was evident was the need to expand the class MB to virtually include all 33 features just as NB's default feature set. The degraded result achieved by the SNB, which reduced the class MB, further supported the apparent need to include the full 33 features.

The advantage offered by the dependency models over the NB is the qualitative structure, providing a recognisable representation of the domain. Moreover, the MIM classifier not only compares favourably with the NB but demonstrates a viable alternative without making extreme CI assumptions. In respect of the effect of modification to the class MB, the results indicate that the 'initial' CL algorithm derived class MB was probably not 'optimised'. Although due to exponential constraints it was not possible to find the optimal solution, the expansion of the class MB (SMIM)

did increase performance accuracies of the MIM classifier which closed the gap on the NB classifiers performance.

In general, the experts viewed the MIM classifier's diagnostic proposition, as a prompt to carry out further investigations or to serve to enforce what they already believed. They did however, confirm many of the statistically derived symptoms (in terms of parameter values as listed in Appendix B) aligned with their own perceptions. From the questionnaire, (specifically Q2, Appendix C) there was evidence that provided support for this view with an indication that additional analysis was almost a necessity. On one hand, their responses related to the use of some heuristics. For example, sweating, smoking and a high alcohol intake pointed towards PPU whilst observing a patient getting on/off a trolley towards APP. On the other hand, their responses indicated the need to carry out further investigations and/or tests such as White Blood Cell (WBS) counts, X-rays and temperature measurements. This may be an indication as to the reason for achieving an 'overall' 70% predictive accuracy, with both the experts and the statistical methods having to make a decision based purely on the analysis of the 'initial' patient observation record (Appendix D).

The MIM classifier's qualitative representation, along with the symptom parameters, not only provides an insight to the domain but also offers a mechanism for knowledge discovery. The experts regarded this aspect as a very effective aid for visualising the domain of AAP and in turn, considered this knowledge representation a benefit to casualty officers and junior house officers. What they also found interesting was the symptom-symptom 'commonality' aspect. Whilst the initial format of the data set was originally considered adequate for this domain, this characteristic may well have identified some redundancy in the data definition and perhaps a need to re-define the domain variables. Despite this anomaly, the use of the class MB does enable the experts to 'focus' on the specific questions that need to be asked in respect of a suspected disease, rather than routinely going through all. Since recording patient details is considered a labour intense activity, particularly for a busy A&E department, any reduction offered was viewed as beneficial.

7.8 Summary

In this chapter, we investigated the claims that the NB classifier was optimal in respect of the medical domain AAP. The main contribution of the study lies in showing that, with respect to the dependency model representations, the MIM classifier and its selective variant SMIM, can be effectively applied to the domain of AAP. Unlike NB the MIM classifier does so without making the assumption of extreme conditional independence providing a qualitative structure of the domain recognisable by the doctors. In the main part of our study we compared the Naive Bayes with two other 'tree' modelling approaches, namely the MIM classifier and a 'polytree' (SCN) as defined by Rebane and Pearl [RP87], along with a general Bayesian network approach.

The MIM classifier performed 'overall' better than the 'polytree' and GBN. When evaluated with a truly 'external' database of the domain, the MIM classifier's 'overall' predictive performance was found to be comparable to that achieved by the NB classifier. Moreover, we observed that the apparent 'optimality' of the NB classifier's success, particularly in the CADA data set, was largely due to its ability to successfully identify the majority group NSAP. This observation was confirmed in respect of the domain individual disease groups with the MIM classifier identifying 5/9 class values better than that achieved by NB for the LEEDS data set and 6/9 class values for the CADA data set.

In the second part of the study we expanded the class MB of the MIM classifier and compared it to the performance of the SNB. Our experimental results indicated that the 'initial' class MB, defined by the CL algorithm, was probably not optimised for this domain and more features were required. The loss of performance shown by the SNB in reducing the class MB further supported the need to include more, rather than less, domain features.

By modelling the domain using an efficient 'tree' structuring algorithm we have avoided the issues of complexity and overfitting to which networks are prone.

Our experimental results on the two AAP databases have demonstrated that the MIM/SMIM classifier's performance was comparable to that of the NB classifier when evaluated with 'external' data of the domain, namely the LEEDS database, and provides an ideal middle ground between the NB and GBN approaches.

Chapter 8

Conclusion

In Chapter 1, we discussed two main objectives of this research. Our motivation was to better support the acceptance of BNs, specifically in the task of classification. First, we wanted to overcome some of the issues that limit the GBN/SCN by exploring the application of an information theory based method for inducing a Classifier from a BN. Second, we wanted to demonstrate the feasibility and effectiveness of the proposed approach by applying it to a selection of difficult benchmark domains. Since previous chapters provided individual summaries, this final chapter reviews in the sections that follow, the main contributions of the thesis, with a discussion of some proposals for further research in section 8.2.

8.1 Summary of Contributions

The main issue addressed in this thesis is the inference complexity due to large CPTs and the corresponding probability estimations, often associated with high dimensional 'real' world domains. In section 8.1.1, we summarise a new method to deal with this problem, while in section 8.1.2, we further explore a solution to improve and overcome the drawbacks of the proposed method. Section 8.1.3, briefly discusses the application of the two approaches to the 'real' world domain of Acute Abdominal Pain, with section 8.1.4 providing a summary of the section.

8.1.1 The Mutual Information Measure (MIM) Classifier

To address inference complexity in BNs, we proposed a new classifier, described in Chapter 3, which explored a new direction of applying information theory based methods to induce a Classifier from a BN in the form of a 'tree' structure. Our aim was to avoid the dependence upon prior node ordering and the consequential inference complexity associated with the estimation of probabilities in large CPTs. In addition, we wanted to maintain as accurate qualitative representation of the domain as possible by modelling all the feature associations. Specifically, our intention was to avoid making strong conditional independence assumptions as adopted by the NB classifier.

In this thesis, we have developed a method for inducing a classifier from data and carried out extensive experiments to demonstrate that the new classifier generally performs better than the unrestricted GBN, 'polytree' and NB methods. We also show the method can learn a BN classifier (as a 'tree') that is smaller and more computationally efficient for inference, thus making the proposed approach more feasible for 'real' world applications that might have otherwise been avoided. The method, called the MIM classifier, as it corresponds to the restricted class of trees built from MI, uses the efficient CL algorithm [CL68] for constructing an undirected 'tree' structure. Since node ordering choice and subsequent CPT dimensionality are known to impact on a BN's ability to perform well as a classifier, the new method applies a node ordering heuristic in order to complete the qualitative aspect of the method. This circumvents the possibility of making a bad choice as doing so may not only result in a topology which leads to an intractable solution but also to the possibility that CPTs will require an unrealistic number of probabilities to both estimate and subsequently update on receipt of new evidence. By adopting this heuristic, we ensure directionality is assigned outwards from the root vertex to all other domain features of the tree structure. Moreover, by configuring the class variable to be the root of the tree, the CL algorithm can additionally be used to define the classifier's class MB with consistency.

In order to classify new evidence from the domain, the MI values are considered as branch 'weights', representing a measure of strength, which are then used in conjunction with the class MB to 'profile' the characteristics of the individual class-states and thus provide a mechanism for discriminating between them. Classification in BNs makes use of the CPT estimated probabilities, however, in situations where domain feature dimensionality is high and data sets sparse, these probability estimates may be unreliable. In contrast, by using only pair-wise marginals to calculate 'weights', the MIM classifier requires, at most, $(n-1)$ MI values to be updated in order to classify new evidence and therefore overcomes the issues of CPT dimensionality that limit the 'polytree' and GBN models. Relative to the methods that depend upon CPTs to classify (GBN and 'polytree') the MIM classifier was shown in Chapter 4 to perform comparably, confirming that neither the restricted topology, that is the assumption that the underlying structure was a 'tree', nor the node ordering heuristic, affected its performance. Moreover, the results further demonstrated that the MIM classifier outperformed significantly the NB classifier, one of the most widely studied methods.

In general, when compared to the dependency methods, the MIM classifier was better than the GBN and SCN for 13/20 data sets, whilst compared to the NB classifier, 15/20. In respect of the GBN, the 'tree' based structure of the MIM classifier meant that there would be a reduction in parameterisation and thus lead to a model with less complexity than the unrestricted BN. Our expectation was that as a 'tree' structure, the class MB would comprise only a subset of the available domain features and therefore perform better, especially in the high dimensional, small sample sized domains. From the results in Chapter 4, this was confirmed, with the MIM classifier achieving 16/20 data sets better than the GBN with, in general, a smaller class MB.

In the case of the SCN, our implementation meant that structurally (skeleton) the 'polytree' would have essentially the same topology as the MIM classifier, with the only potential difference being its class MB. As this is determined by the resulting directionality assignment the class MB for the 'polytree' could be (as it will potentially include child nodes with parents) more complex than that of the MIM classifier. This latter aspect represents a form of node ordering and as we stated earlier, a bad choice could easily impact on its performance as a classifier. For the MIM classifier however, the MI branch value is independent of the actual directionality assigned to a branch, and therefore the node ordering heuristic was not expected to impact on its classification performance, other than in deriving its class MB. The experimental results, Chapter 4, bear this out. In comparison with the 'polytree' the node ordering choice did appear to have an influence, with the 'polytree' only achieving 6/20 in direct comparison to that achieved by the MIM classifier.

During the process of investigation we also discovered some properties of the data sets for which the classifier is best suited, along with some limitations that can be used to assist in the design of improved methods (these are further discussed in section 8.2). Moreover, from the results shown in Chapter 4, several important ramifications have also emerged.

Firstly, there is an indication that the MIM classifier provides a way to apply BNs to problems where previously it was not possible due to large CPT size and poor probability estimates. This is particularly evident for high dimensional, small sample sized domains.

Secondly, in overcoming the issues that limit the 'polytree' and GBN models, the MIM classifier represents a good middle ground model (lying between the NB and GBN). Essentially, by using pair-wise marginals to calculate branch 'weights' we avoid the issues associated with CPT

dimensionality and by application of a node ordering heuristic, we further avoid the consequences of making a poor choice. By using an efficient 'tree' structuring algorithm we also avoid making strong CI assumptions, thus maintaining a qualitative representation of the domain (even if restricted).

Thirdly, by identifying various properties of a particular data set, the results provide an aid to the determination of whether the MIM classifier approach is best suited for a specific domain and whether it offers any significant benefits in comparison to the other competitive methods. In Table 8.1 we provide a summary guide, in respect of each non-selective approach, indicating what domain properties best suit a particular method. In the case of the MIM classifier and in the context of the databases studied, we observed that it was generally more beneficial than the GBN for all types of data sets, especially those with high dimensionality and 'multi' parented class variables. In contrast, where data sets have only a few features and the sample size is large, the GBN is a preferable option. For domains that have large sample sizes and a medium dimensionality (14-34 features), the MIM classifier is better suited, however for smaller sample sizes the NB should be selected.

8.1.2 The 'Selective' MIM Classifier

A drawback, identified in Chapter 4, section 4.7, highlighted an issue concerning the induction of the MIM classifier. Since the CL algorithm is prone to generating 'trees' with missing relevant features or adding irrelevant features to the structure, the MIM classifier's derived class MB optimality is uncertain. In order to investigate this we considered the issue as a feature selection problem and derived a 'Wrapper' type selective variant of the MIM classifier, described in Chapter 5. Our aim was to determine whether the CL algorithm could be considered a viable approach for determining the class MB and as an 'initial' MB, whether it could be improved.

Since high dimensional domains represent exponential feature selection search problems, our approach considered the 'initial' MB as a *lower bound*. Taking this view we used the 'Wrapper' to expand the class MB, selecting candidate edges having a maximal MI value in respect of their $C-Z$ branch associations. As edges were added, the 'selective' variant was evaluated by measuring its performance accuracy. For the evaluation function we used the same efficient 'tree' based classifier (namely the MIM classifier), which has the advantage of incorporating any potential bias associated with the classifier during feature selection. Although using the 'initial' MB does provide a good starting point, for the high dimensional domains the approach proposed still requires

a potentially unrealistic number of edges to be considered in order to reach an 'optimised' performance. Whilst we could select a limiting criteria to circumvent this possibility, the occurrence of local maxima/minima as shown in Chapter 6, forces the approach to consider all $C-Z$ associations. To deal with this, we introduced two heuristics which enabled the algorithm to terminate safely and alleviate the inevitable computational expense.

To evaluate the 'selective' MIM classifier, presented in Chapter 5, we performed several experiments using the same data sets as for the MIM classifier and demonstrated that the approach in general did improve its performance. The results showed that the combined 'overall' performance of the SMIM/MIM classifiers was 13/20 data sets better than all other methods studied. During our evaluation several specific aims were addressed. First, we wanted to determine whether the expansion of the class MB improved the MIM performance accuracy. The results demonstrate that by focusing on the class MB there is improvement in performance for domains with greater than 15 features. The most prominent of enhancements was observed for the smaller data sets in contrast to the larger ones where, in general, no improvements were observed. Second, the CL algorithm is an implied feature selector and as such we were interested in whether the class MB or *lower bound* represented an 'optimised' MB, and if not, was it a good starting point for expansion. The results of Chapter 6, indicate that this was the case in both instances. Firstly, for the larger data sets where there was very little improvement in performance accuracy and thus the *lower bound*, the MIM classifier maintained superior performance levels over the other methods for many of the data sets studied. This implies that the MB could not be further improved upon. In the second instance, where there was evidence of improvement by MB expansion for some of the smaller data sets, performance of the SMIM classifier was again found to be superior to the other approaches. The drawback however, is that whilst focusing on the class MB does allow improvement in some domains, the 'Wrapper' approach does not guarantee to find an 'optimal' solution.

Our third and fourth aims were concerned with the consequences of the MB expansions. As the SMIM classifier expands the class MB, we wanted to evaluate the cost in terms of model complexity and the effect on learn rates. The results of Chapter 6 indicate that where performance accuracies were enhanced, it was not at the price of increased complexity as in most cases only a marginal number of features actually required to be added to the 'initial' class MB. Thus the complexity of

the SMIM classifier was not much more than that of the MIM classifier. As expected there was an impact on the learn rates, particularly evident for small sample sizes. However, the 'overall' performance levels achieved were generally found to be better than the non-selective variant with performance levels stabilising at the same 60% sample size after an initial steep climb at 20% of the sample size.

The implication of these results is that the classification technique, proposed in section 8.1.1, is not restricted to 'tree' based structures (with respect to the class MB) and is thus independent of the underlying topology of the domain being modelled.

At first glance this may appear to offer a step closer to achieving a GBN representation, that is, use a network rather than a 'tree' as the underlying structure. However, as stated in Chapter 1, deriving networks is, in the general case, NP-hard. In making a decision as to what the underlying topology should be, we need to consider first what the overall goals are. If the objective is to retain a qualitative aspect, then as demonstrated in Chapter 4, the 'tree' based algorithms are far more viable and efficient. On the other hand, if classification performance alone is the goal, then deriving a class MB with minimal irrelevance is a better approach.

Clearly, the tree based CL algorithm offers a good compromise, however as pointed out in the beginning of this section, using it places an uncertainty in the 'optimality' of the derived class MB, particularly if the sample set is small. In section 8.2.2, we explore this issue further and discuss possible ways to overcome the limitations of the CL algorithm.

8.1.3 Case Study – Diagnosing Acute Abdominal Pain (AAP)

The investigations concerning the MIM classifier (Chapter 4) and its 'selective' variant (Chapter 6) utilise the benchmark data sets of the UCI repository. However, one of our motivations was to demonstrate a viable solution for BN classification to 'real' world domains. As described in Chapter 7, we applied both methods to the task of diagnosing Acute Abdominal Pain. This is known to be a very difficult and challenging domain [LE93] because it is a high dimensional problem characterised by sparse data, several features, many class-states and for our data sets comprises samples with missing and composite feature states.

We have carried out detailed experiments on two data sets of the domain and compared the performance of the MIM classifier along with its 'selective' variant to that of the GBN, 'polytree'

NB and SNB. As a measure of confidence in the levels of performance achieved, the classifiers were additionally compared to results obtained from the experts, both contained within the data sets and qualitatively from questionnaires completed by the A&E Doctors at St John's Hospital, Edinburgh, as shown in Appendix B. Our aim was to investigate the optimality claim of researchers in respect of the NB for this domain, with the expectation, supported by the empirical evidence described in Chapter 4, that since the experts have identified strong dependencies between symptoms that the dependency models should outperform the NB classifier.

The experiments showed that for the dependency models, the MIM and SMIM classifiers were found to be the most effective of the BN models. Unlike the NB, this was achieved without making the extreme assumption of conditional independence given the class variable. The qualitative structure in respect of the MIM classifier, as the SMIM focuses on performance accuracy improvements, was recognisable by the experts who also confirmed many of the class – symptom associations aligned with their own beliefs. Our investigation into the specific optimality claim of NB in this domain did not quite meet with our expectations. The results indicate that the NB did appear to be 'optimal' for the domain. However, this was only in respect to its 'overall' winning performance accuracy. This result was largely due to the performance and contribution of the majority class-state NSAP, which the experts define as a group of exclusion. Moreover, when evaluated with a 'truly external' test data set, the removal of the influence of NSAP showed that the 'overall' performance achieved by MIM/SMIM classifier was able to match that of the NB classifier.

Although classification accuracy gives an 'overall' measure of the performance of each method, we considered it important, as discussed in Chapter 7, to evaluate the performance of the various algorithms on each individual disease. For example, the class-state appendicitis can be life threatening and needs to be diagnosed quickly and correctly, whereas the class-state NSAP is not, with patients usually showing signs of improvement within 24 hours. The results demonstrate that the MIM classifier and the experts are able to discriminate between individual class-states more readily than the NB classifier. The MIM classifier identified 6/9 for CADA and 5/9 for LEEDS better than NB, whilst the experts identified 6/9 for CADA and 9/9 for LEEDS.

The ramifications of the resulting investigation where numerous. Firstly, for class-states that were poorly characterised such as pancreatitis, there was evidence (via questionnaires) that the experts

used 'heuristics' to overcome anomalies in the database. Secondly, many of the features that were recorded in the database were found to have a low frequency of occurrence. Whilst this implies these are irrelevant and can be discarded, our experimental investigations show that although these could indeed be statistically removed, the questionnaires from the experts indicate that many of these features are in fact considered 'key' in identifying a specific class-state and therefore should not be removed. Thirdly, the symptom-symptom relationships identified in the qualitative structure imply that the database definition may require modification as these infer 'commonality' rather than true symptom associations, and thus a source of redundancy. Fourthly, experiments in respect of the 'selective' variant of the MIM classifier showed that the expansion of the class MB required nearly as many features as that of the NB MB. This might suggest that the domain is better suited to the assumption of independency despite the rationale for clear dependency amongst features. This is further supported by the way experts view this domain, which maybe attributed to the fact that doctors gather as many different *independent* bits of relevant information as possible and do not include two features where one would suffice.

Whilst this observation might imply we should be modelling the classifier in a similar way to that of NB, the expert's questionnaires and resulting qualitative representation of the MIM classifier infer otherwise. Essentially, each class-state appears to have its own characterisation with associations between specific symptom parameters distinctly different for each state. This suggests that we should in fact be individually modelling the class-states. A drawback of the MIM classifier however, is that its structure is constrained by its underlying MWST, and as a consequence this aspect cannot be fully realised. In section 8.2, we pursue this matter further and suggest a method that can provide a more representative solution to that of the experts, whilst still retaining the benefits of the 'tree' based approaches.

8.1.4 Summary

In summary, this thesis has proposed a MIM classifier based on the efficient tree structuring CL algorithm. By overcoming the issues that limit the 'polytree' and GBN models, the MIM classifier represents a good middle ground model. In using pair-wise marginals to calculate branch 'weights',

it avoids the issues associated with CPT dimensionality and by application of a node ordering heuristic, avoids the consequences of making a poor choice. The classifier has been demonstrated as performing better than the unrestricted models such as the GBN and the controversial but simple NB classifier. Compared to NB, the MIM classifier performed significantly better on of the data sets studied, even with a class MB at 50% of that used by the NB classifier. Where the NB did perform well due to known independence in the data set under test, the MIM classifier performed comparably.

The advantages of the classifier is that it provides a way to classify with BNs on domains previously not possible due to dependence upon prior node ordering and the subsequent inference complexity when the network topology leads to large conditional probability tables. It has further been shown that the MIM classifier and its 'selective' variant can effectively be applied to a difficult 'real' world domain. In Table 8.1, we provide a summary of the thesis findings along with a guide for suitability of methods, in respect of domain application, in the context of the data sets studied in this thesis.

Table 8.1: Summary of Findings

MIM Classifier[1]	
	For large sample sized domains, the CL algorithm can be used to derive a satisfactory class MB. This class MB may however not be 'optimal'.
	Classification based on MI edge 'weights' does not require prior node ordering or CPT probability estimates.
	For small sample sized domains, the derived class MB is unlikely to be 'optimal'.
	The MIM classifier technique (Chapter 3) is independent of the underlying topology and can be applied (via class MB) to 'trees' and networks.
	Classification of new evidence of the domain requires no more than a maximum of *(n-1)* edge 'weights'.
	Focusing on expanding the class MB can improve performance accuracy. In general, only a marginal number of features are required to be added. Domains of greater than 15 features will gain the most benefit, as will small sample sized domains.
Case Study – APP[2]	
	NB is arguably optimal for this domain, but only in respect of its 'overall' performance accuracy.
	The 'overall' performance is due to the contribution of majority class-state NSAP, which the experts define as a group of exclusion.
	Experts use 'heuristics' to overcome anomalies in database.
	The MIM classifier/Experts are able to discriminate between individual class-states more readily than NB.
	BNs using the majority of domain features (with respect to the class MB) perform best. This implies AAP is better modelled by an assumption of independence, despite the rationale for dependence.

Method	**MIM classifier**	**NB**	**'Polytree'/SCN**	**GBN**
Suitable Domains[1]	High/medium dimensional, small sample size (over GBN). Medium dimensional, large sample size (over NB).	Medium dimensional, small sample sizes (over MIM). High/medium dimensional, small sample size (over NB). GBN).	High dimensional, small sample size (over GBN). Medium dimensional, large sample size (over	Small/medium dimensional, large sample sizes (over MIM/polytree). Small/medium dimensional, small samples size - general.

Key: In the context of the data sets studied in this thesis. High dimensional ≡ 35+ features, Medium dimensional ≡ 14+ features (max 34), Small dimensional ≡ under 14 features.

Notes: [1]With respect to the UCI data sets studied.
[2]In respect of the CADA and LEEDS databases.

8.2 Further work

Despite the favourable results demonstrated by of the MIM classifier, in comparison to the other methods, there are a number of options that can be considered to improve it further. In the previous sections, we highlighted some of the limitations in taking the approach proposed within this thesis. In the sections that follow, we review these in more detail and suggest further investigations to address them.

Section 8.2.1 begins by considering how continuous features may be supported in the MIM classifier along side the discrete features. In section 8.2.2, we focus our attention to inducing an 'optimal' class MB, with section 8.2.3 considering how the experts view of the domain can be utilised to better model the data set, leading to improved classification.

8.2.1 Dealing with Continuous Features

The approach adopted in constructing the MIM classifier benefits from using the CL algorithm to derive its underlying structure as it can be achieved efficiently in polynomial time. The empirical studies show that for larger data sets the resulting induced 'tree' structure not only derives a satisfactory representation of the domain class MB, but also provide a qualitative structure recognisable by domain experts. However, a limitation in employing the CL algorithm is its requirement to have purely discrete data. In the case of many 'real' world domains this unfortunately is not the norm with many features being continuous. Although discretisation works well for the task of classification, as demonstrated by the empirical results, there are some detrimental consequences in taking this approach. Apart from the obvious loss of information, not only is the qualitative representation of the domain it models constrained, but also the continuous feature parameter space expanded.

A natural enhancement to consider, is the extension to the distribution that can be represented by the MIM classifier, and more specifically, the ability to be able to deal with a combination of discrete and continuous features and model the interactions between them. Indeed, work carried out by Friedman et al [FGL98] shows that modelling both discrete and continuous features together offers the best model for classification performance.

In considering this technique however, we are faced with the problem of trying to decide which domain features to discretise or leave as continuous. Unfortunately, this task requires an exponential space of options to make a selection. Friedman's Hybrid-TAN offers a solution to the problem by representing both the continuous feature and its discretised counterpart within the same TAN model. As reviewed in Chapter 3, section 3.3, TAN addresses the controversial issues associated with the NB classifier by modelling some of the feature dependencies.

The approach taken in the Hybrid-TAN uses a dual representation for each continuous feature with one discretised and the other based on fitting a parametric distribution. By taking advantage of the modelling language of the BN, interactions between the discrete and continuous versions of the feature are represented simultaneously. This has the advantage of maintaining a qualitative, though restricted, network structure of the domain. In addition, since the structure includes a mixture of discrete and continuous features the resulting structures induced are directed trees thus circumventing the requirements for prior node ordering.

In the Hybrid-TAN, the edges of the model are assigned 'weights' based upon a *score*. This *score* is calculated in respect of the conditional probability distribution of the feature and depends upon its representation. In the case of a discrete feature, these are tables using empirical frequency of events in the training data, whilst for continuous features, these are Gaussian distributions. This suggests that in order to enhance the capability of the MIM classifier to better support 'real' world domains, these *scores* require to be translated into equivalent MI representations. For discrete features, *scores* relate to the conditional mutual information between two features given the class. In the case of the continuous features, the *scores* will require a similar information theoretic interpretation and it is this aspect that represents the main focus of activities for future work.

8.2.2 Optimising the Class MB

The key to constructing an 'optimal' MIM classifier lies in determining the most representative class MB of the domain, prior to performing any classification. Despite being able to demonstrate that the CL algorithm can be used to derive a satisfactory 'initial' MB, the empirical evidence was only in support of the larger data sets, with the smaller ones requiring some MB modification. This latter situation was addressed by a 'selective' variant of the MIM classifier, which applied a Wrapper type feature selection process to expand the class MB. Whilst the technique does allow improvement in

some domains, the approach has a drawback in that it does not guarantee to find an 'optimal' solution. As discussed in Chapter 4, section 4.7, the CL algorithm is prone to adding irrelevant features which means that since, in the Wrapper approach, no irrelevant features are identified and subsequently removed, the 'initial' class MB may already contain some of these features. The occurrence of too many of these features will not only complicate the model but be detrimental to its performance.

For high dimensional domains there will undoubtedly be a measure of dependence between some of the features. Although in using the CL algorithm some of these dependencies will be captured, the approach adopted for the 'selective' MIM classifier on the other hand may not, especially for small sample sets. Whilst the selection of features based on their individual magnitude of MI *(C-Z)* may produce 'overall' a class MB that enhances performance, some of these features may actually be redundant. Unfortunately as the halting criteria is not based on 'single' edge additions but on performance accuracy, the influences of these individual additions (good or bad) can easily be lost whilst focussing on achieving the target objective.

One way to overcome this limitation is to consider the technique proposed by Cheng [CG01]. As we reviewed in Chapter 3, section 3.3, the objective of Cheng's work was to construct a BN. After deriving an 'initial' tree structure, using the CL algorithm, Cheng's procedure transforms the 'tree' into a network via a series of phases which either adds or deletes edges. In doing so, the modifications to the structure also lead to the derivation of a near 'optimal' class MB. This same approach could equally be utilised to not only allow classification, using the MIM classifier technique in networks, but also to aid in determining a better class MB representation.

Unfortunately whilst providing a possible enhancement, without prior node ordering the complexity of Cheng's algorithm is $O(N^4)$. An alternative approach is to use the joint mutual information (JMI). In the MIM classifier, the branch 'weights' are derived as pair-wise marginals, however the concept of MI can be further extended to include more than two random variables. By use of the chain rule, the JMI between a set of features $(Z_1, Z_2, \ldots, Z_n)$ and the outcome $\boldsymbol{C}$ (i.e. the desired class) is $I(Z_1, Z_2, \ldots, Z_n; C) = \sum_{i=1}^{n} I(Z_i; C \mid Z_{i-1}, Z_{i-2}, \ldots, Z_1)$.

According to Tourassi et al [TF+01] the JMI is more appropriate for feature selection because it can produce an optimal subset that contains the most relevant features with a minimum amount of feature redundancy.

The Wrapper approach discussed in Chapter 5, incrementally adds 'single' edges to the 'initial' class MB as derived by the CL algorithm. However by applying the JMI, a subset of the most relevant features to add can be identified. For the larger data sets, as indicated by the results of Chapter 6, the performance in general does not benefit from class MB expansion. However, for the smaller data sets the JMI can be used to find the most appropriate subset to add to the 'initial' class MB rather than 'single' edges. Although for the large data sets the induced class MB is a satisfactory representation of the domain, the JMI could still be used in order to determine a 'measure' of quality of the 'initial' class MB. This has the benefit of offering a way to potentially identify irrelevant features for subsequent removal.

Whilst this approach can be used to effectively counter the issues associated with the CL algorithm, the focus of the future work lies in the determination of a suitable terminating criteria, particularly for large dimensional domains. In the case of these data sets, finding an 'optimal' subset (constrained to the class MB) may require an exponential number of feature subsets to measure against before identifying the maximal JMI output.

8.2.3 Modelling Individual Class-States

From the completed Doctor's questionnaires, Appendix B, there is an indication that AAP class-states have characterisations that are distinctly different involving a diverse combination of symptom parameters. This implies that perhaps we should be modelling the domain more accurately by considering individual class-states having their own representative class MB. Whilst the MIM classifier does this via 'profiling', it is somewhat constrained by its 'overall' underlying topology. In this approach, we assign different configurations of symptom parameters to a particular class-state however, it is in respect of the fixed MWST derived by the CL algorithm (which is essentially the class MB for the MIM classifier). In Chapter 3, section 3.3 we reviewed work of Friedman [FGG97] concerning the 'multinet' which does construct individual class-state related structures. Whilst it is possible for these sub-structures to be networks, in doing so means dealing with the difficulties of learning a general graphical structure. The approach taken by Friedman overcomes this by using

'tree' structures, however as reviewed in Chapter 3, section 3.3 the technique has the disadvantage in that it derives class-state representations using the individual class data partitions. As well as being computationally expensive to learn, 'real' world class-state imbalances can also lead to a biased classification performance.

As demonstrated by the MIM classifier and supported by the empirical results, classification using 'trees' provides for an efficient and viable approach in comparison to networks. In modelling acyclic pair-wise dependencies, the representational power of 'tree' is somewhat limited, however this is overcome by combining them into mixtures. A generalisation of tree distributions is offered by work carried out by Meila [Mei99] called mixture of trees. This is basically a probabilistic mixture of a set of graphical components, each of which is a tree. Essentially the JPD is represented by a finite mixture of tree distributions, each modelling the class-conditional density of one of the class-sates.

In Meila's approach the mixture of trees provides a reasonable compromise between the simplicity of tree distributions and expressive power of the Bayesian 'multinet'. Unlike 'multinet' as proposed by Friedman, Meila's mixture of trees treats the class variable as another input feature which allows the classifier to utilise all the training data in order to train/learn the model as does the MIM classifier.

The difference between the latter two approaches however, is that Meila's approach yields a mixture model for each class-state and not a fixed MWST as represented by the MIM classifier.

In the same way the MIM classifier derives its underlying structure, the mixture of trees also uses the CL algorithm over discrete variables to construct trees, controlling the complexity by the number of trees in the mixture. In the case of the mixture of trees, an accelerated CL algorithm is employed which has the additional benefit of offering a partial solution to the consequential computational expense of this approach.

An interesting proposition to investigate further is whether the MIM classifier's technique for using MI branch 'weights' can be similarly applied to the class-state specific trees. In doing so it will possible to determine if the modelling of the individual class-state MBs provides any further improvements to the MIM classifier's fixed class MB performance. Moreover, although for classification the class MB is the only area of interest the advantage the MIM classifier offers over the NB classifier is the qualitative representation (even if restricted) of the domain. From the questionnaires, the experts confirm many symptom parameters align with those identified by the

algorithm for each class-state. However, the experts do not provide sufficiently detailed qualitative information in respect of the symptom inter-relationships. Since each class-state will now be characterised by its own structure (tree based) a further consideration will be to also investigate what qualitative aspects can be additionally discovered.

Bibliography

[AC+86] I. D. Admas, M. Chan, P. C. Clifford, W. M. Cooke, V. Dallos, F. T. de Dombal, M. H. Edwards, D. M. Hancock, D. J. Hewett, N. McIntyre, P. G. Somerville, D. J. Spiegelhalter, J. Wellwood, D. H. Wilson, Computer-aided diagnosis of acute abdominal pain : a multicentre study, British Medical Journal, **293**:800-804, 1986.

[AC+91] S. Acid, L. M. de Campos, A. Gonzalez, R. Molina, N. Perez de la Blanca, Learning with CASTLE, in : R. Kruse, P. Siegel, eds., Symbolic and Quantitative Approaches to Uncertainty, Lecture notes Computing Science 548, Springer Verlag, 99-106, 1991.

[ACH01] S. Acid, L. M. de Campos, J. F. Huete, The search of causal orderings: A short cut for learning belief networks, Lecture notes in computer science, vol 2143, Springer-verlag, pp. 216-227, 2001.

[AD02] A. Al-Ani, M. Deriche, Feature Selection using Mutual Information Based Measure, in: Proceedings of the 16th International Conference on Pattern Recognition, **4**:82-85, 2002.

[ADC03] A. Al-Ani, M. Deriche, J. Chebil, A new mutual Information based measure for feature selection, Intelligent Data Analysis, **7**(1) :43-57, 2003.

[Bat94] R. Battiti, Using Mutual Information for selecting Features in Supervised neural net learning, IEEE Transactions on Neural Networks 15, 537-550, 1994.

[BCS04] C. Barbacioru, D. J. Cowden, J. Saltz, An algorithm for reconstruction of Markov blankets in Bayesian networks of gene expression datasets, in Proceedings of the 2004 IEEE Computational Systems Bioinformatics Conference (CBS 2004), 597-598, 2004.

[BL+01] R. Blanco, P. Larrañaga, I. Inza, B. Sierra, Selection of highly accurate genes for cancer classification by estimation of distribution algorithms, in: European Conference on Artificial Intelligence in Medicine (AIME'01): working notes for workshop, pp. 29-34, 2001.

[BL97] A. L. Blum, P. Langley, Selection of relevant features and examples in machine learning, Artificial Intelligence, **97**(1-2):245-271, Special Issue in relevance, 1997.

[BM00] C. Blake, C. J. Merz, UCI repository of machine learning databases, Department of Information and Computer Science, University of California, Irvine, CA, 2000. http://www.ics.uci.edu/~mlearn/MLRepository.html

[Boe95] B. Boerlage, Link strengths in Bayesian networks, Master's Thesis, Department of Computer Science, University of British Columbia, 1995.

[BS96] G. L. Barrows, J. C. Sciorino, Jr, A Mutual Information Measure For Feature Selection with Application to Pulse Classification, in: Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time Analysis, **18-21**: 249-252, 1996.

[Bun96] W. Buntine, A guide to the literature on learning probabilistic networks from data, IEEE Transactions on Knowledge Data Engineering. **8**:195-210, 1996

[BW00] D. A. Bell, H. Wang, A formalism for relevance and its application in feature subset selection, Machine Learning, **41**:175-195, 2000.

[Cam96] L. M. de Campos, Independency Relationships and Learning Algorithms for Singly Connected Networks, Technical Report DECSAI-96-02-04, Department of Computer Science, University of Granada, 1996.

[CBL97] J. Cheng, D. A. Bell, W. Liu, An algorithm for Bayesian belief network construction from data, In Proceedings of Artificial Intelligence and Statistics' 97, 83-90, 1997.

[CC90] R. Chavez, G. Cooper, A randomized approximation algorithm for probabilistic inference on Bayesian belief networks, Networks 20: 681-685, 1990.

[CG01] J. Cheng, R. Greiner, Learning Bayesian belief network classifiers: Algorithms and system, in: Proceedings. 14th Biennial Conference of the Canadian Society for Computational Studies of Intelligence, Ottawa, ON, 2001.

[CG99] J. Cheng, R. Greiner, Comparing Bayesian Network Classifies, In Proceedings of Uncertainty in Artificial Intelligence, UAI-99, 1999.

[CGK02] J. Cheng, R. Greiner, J. Kelly, Learning Bayesian Networks from Data: An Information-Theory Based Approach, Artificial Intelligence Journal 137, (1-2):43-90, 2002.

[CH+02] J. Cheng, C. Hatzis, H. Hayashi, M. Krogel, S. Morishita, D. Page, J. Sese, KDDcup2001 report, ACM SIGKDD Explorations **3**(2), 2002.

[CH90] J. R. Clarke, C. Z. Hayward, Workshop on surgical decision making: a scientific approach to surgical reasoning, Theoretical Surgery, **5**:129-132, 1990.

[CH92] G. F. Cooper, E. Herskovits, A Bayesian method for the induction of probabilistic networks from data, Machine Learning **9**: 309-347, 1992.

[Che98] J. Cheng, PowerConstructor System, 1998. *http://www.cs.ualberta.ca/~jcheng/bnpc.htm*.

[Chi96] D. M. Chickering, Learning Bayesian Networks is NP-Complete, Learning from data: AI and Statistics V, 121-130, Springer, 1996.

[CL68] C. K. Chow, C. N. Liu, Approximating Discrete Probability Distributions with Dependence Trees, IEEE Transactions On information Theory, Vol. IT-14, 462-467, 1968.

[CLR90] T. H. Cormen, C. E. Leiserson, R. L. Rivert, Introduction to Algorithms, MIT Press, 1990.

[CMN00] N. Cruz-Ramirez, J. May, R. Nicolson, Algorithms based on Information Theory that build Bayesian Networks from data, Internal report, Department of Psychology, University of Sheffield, Sheffield, 2000.

[Coo90] G. Cooper, The Computational Complexity of Probabilistic inference using Belief networks, Artificial Intelligence, **42**:393-405, 1990.

[CP+03] J. Cao, R. Panetta, S. Yue, A. Steyaert, M. Young-Bellido, S. Ahmad, A naïve Bayes model to predict coupling between seven transmembrane domain receptors and G-proteins, Bioinformatics 19 (2), Oxford University Press, pp. 234-240, 2003.

[CS+03] I. Cohen, N Sebe, F. G. Cozman, M. C. Cirelo, T. S. Huang, Learning Bayesian Network Classifiers for Facial Expression Recognition using both Labeled and Unlabeled Data, in: Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'03), 1:595-601, 2003.

[CVC77] T. M. Cover, J. M. Van Campenhout, On the possible orderings in the measurement selection problem, IEEE Trans. On Systems, Man and Cybernetics, 7(9):657-661, 1977.

[DAA94] T. L. Dean, J. Allen, J. Aloimunous, Artificial Intelligence: Theory and Practice, Benjamin/Cummings, 1994.

[DAm91] B. D'Ambrosio, Local expression languages for probabilistic dependencies, In Proceedings of the seventh Annual Conference on Uncertainty in Artificial Intelligence, 95-102, Morgan Kaufman, 1991.

[Das99] S. Dasgupta, Learning Polytrees, in: Proceedings of the UAI'99, 1999.

[DB00] J. G. Dy, C. E. Brodley, Feature subset selection and order identification for unsupervised learning, In Proceedings of the 17th International Conference on Machine Learning, 247-254, 2000.

[DG00] M. J Druzdzel, L. C. van der Gaag, Building probabilistic networks: 'Where do the numbers come from ?' guest editors' introduction, IEEE Trans. Knowledge Data Engineering, 12, 481-486, 2000.

[DH01] R. O. Duda, P. E. Hart, Pattern Classification and Scene Analysis, John Wiley & Sons, 2nd Edition, 2001.

[DH73] R. Duda, P. Hart, Pattern Classification and Scene Analysis, John Wiley & Sons, 1973.

[DK03] M. Dong, R. Kothari, Feature subset selection using a new definition of classifiability, Pattern Recognition Letters, **23**:(9-10):1215-1225, 2003.

[DL+72] F. T. De Dombal, D. J. Leaper, J. R. Staniland, A., McCann, J. Horrecks, Computer aided diagnosis of acute abdominal pain, British Medical Journal, **2**:9-13, 1972.

[DL93] P. Dagum, M. Luby, Approximating probabilistic inference using Bayesian belief networks is NP-hard, Artificial Intelligence, **60**(1):141-153, 1993.

[DL97] M. Dash, H. Liu, Feature selection for classification, Intelligent Data Analysis, **1**:131-156, 1997.

[Dom91] F. T. De Dombal, The diagnosis of acute abdominal pain with computer assistance, Annals Chir., **45**:273-277, 1991.

[Dom93] F. T. De Dombal, editor, Surgical Decision Making. Oxford: Butterwoth Heinemann, 1993.

[DP97] P. Domingos, M. Pazzani, On the optimality of simple Bayesian classifier under zero-one loss, Machine Learning 29, 103-130, 1997.

[Dra95] D. Draper, Localized partial evaluation of belief networks, PhD Thesis, Department of Computer Science, University of Washington, 1995.

[ED84] F. Edwards, R. Davis, Use of a Bayesian algorithm in the computer-assisted diagnosis of appendicitis, Surgical Gynaecology Obstetrics, **158**:219-222, 1984.

[EN95] K. Ezawa, S. Norton, Knowledge discovery in telecommunication services data using Bayesian network models, in: Proc. 1st International Conference on Knowledge Discovery and Data Mining, 1995.

[EOL97] H. P. Eich, C. Ohmann, K. Lang, Decision Support in Acute Abdoninal Pain using an expert system for different knowledge bases, in: Proceedings of 10th IEEE Symposium on Computer-based medical systems, 2-7, 1997.

[EZ+01] M. Zorman, H. P. Eich, P. Kokol, C. Ohmann, Comparison of three databases with a decision tree approach in the medical field of acute appendicitis, Proceedings of the 10th World Congress on Medical Informatics MEDINFO 2001, London, pp 1414-1418, 2001.

[FF03] L. Frey, D. Fisher, Identifying Markov Blankets with Decision Tree Induction, in Proceedings of the Third IEEE International Conference on Data Mining (ICDM'03), 59-66, 2003.

[FF95] R. Fung, B. D. Favero, Applying Bayesian networks to information retrieval, Communications of the ACM **38** (3), 1995.

[FG96] N. Friedman, M. Goldszmidt, Learning Bayesian networks with local structure, in: Proceedings of the 12th International Conference on Uncertainty in Artificial Intelligence, 1996.

[FGG97] N. Friedman, D. Geiger, M. Goldszmidt, Bayesian Network Classifiers, Machine Learning, **29**:131-161, 1997.

[FGL98] N. Friedman, M. Goldszmidt, T. J. Lee, Bayesian Networks classification with continuous attributes : getting the best of both discrete and parametric fitting, in: Jude Shavlik (ed) International Conference on Machine Learning, 1998.

[Fri98] N. Friedman, The Bayesian Structural EM Algorithm, in: Proceedings of Conference of Uncertainty in Artificial Intelligence, UAI-98, Morgan Kaufmann, 1998.

[Fry78] D. G. Fryback, Bayes' theorem and conditional nonindependence of data in medical diagnosis, Computers and Biomedical Research, **11**(5):429-435, 1978.

[FZ00] E. Faginoli, M. Zaffalon, Tree-augmented naïve creedal classifier, In IPMU 2000, proceedings of the 8th Information Processing and Management of Uncertainty in Knowledge-Based Systems Conference, 1320-1327, 2000.

[Gal94] S. I. Gallent, Neural Network Learning and Expert Systems, MIT Press, Cambridge, Massachusetts, 1994.

[GD04] D. Grossman, P. Domingos, Learning network classifiers by maximizing conditional likelihood, Twenty-first International Conference on Machine Learning, article 46, 2004.

[Gei92] D. Geiger, An entropy-based learning algorithm of Bayesian conditional trees, in: UAI'92, pp. 92-97, 1992.

[GG90] Y. Gu, A. Gammerman, A computer-aided diagnostic system and its application to a large medical database, In IEE Colloquium on AI in medical decision making, London, 1990.

[GGS97] R. Greiner, A. Grove, D. Schuurmans, Learning Bayesian nets that perform well, In Proceedings of UAI-97, 1997.

[GL91] A. Gammerman, Z. Luo, Constructing Causal Trees from a medical database, Technical Report TR91002, Department of Computer Science, Heriot-Watt University, Edinburgh, UK, 1991.

[GM04] A. Goldenberg, A. Moore, Tractable learning of large Bayes net structures from sparse data, Twenty-first International Conference on Machine Learning, article 44, 2004.

[GP+02] H. Guo, B. B. Perry, J.A. Stilson, W. H. Hsu, A Generic Algorithm for Tuning Variable orderings in Bayesian network structure learning, 18th National Conference on Artificial Intelligence, 951-952. Albeta, Canada, 2002.

[GT90] A. Gammerman, A. R. Thatcher, Bayesian Inference in an expert system without assuming independence, In. Golumbi, M, editor, Advances in AI, Natural Languages and Knowledge Based Systems, pp 182-218, Springer-Verlag, 1990.

[GT91] A. Gammerman, A. R. Thatcher, Bayesian diagnostic probabilities without assuming independence of symptoms, Methods of Information in Medicine, **30**:15-22, 1991.

[Gun76] A. A. Gunn, The Diagnosis of Acute Abdominal Pain with computer diagnosis, Journal of the Royal College of Surgeons, Edinburgh, **21**:170-172, 1976.

[HC93] J. F. Huete, L. M. de Campos, Learning causal polytrees, in: M. Clarke, R. Kruse, S. Moral, eds., Symbolic and Quantitative Approaches to reasoning and Uncertainty, Lecture notes Computing Science 747, Springer Verlag, 180-185, 1993.

[Hec98] D. Heckerman, A tutorial on learning with Bayesian networks, in: M. I. Jordan (Ed): Learning in Graphical models, Dordrecht, Netherlands, Kluwer, 1998.

[Hec99] D. Heckerman, A tutorial on learning with Bayesian networks, In M. Jordon (ed), Learning in Graphical Models, Cambridge, MA, MIT Pres, 1999.

[Hen88] M. Henrion, Propagating Uncertainty in Bayesian networks by probabilistic logic sampling, In Uncertainty in Artificial Intelligence 2, eds, J. Lemmer, L. Kanal, 149-163, Elsevier Science, 1988.

[Her91] E. H. Herskovits, Computer-based Probabilistic Construction, PhD Thesis, Medical Information Science, Stanford University, Stanford, CA, 1991.

[HGC94] D. Heckerman, D. Geiger, D. M. Chickering, Learning Bayesian Networks: the combination of knowledge and statistical data, Technical report MSR-TR-94-09, Microsoft Research, 1994.

[HGC95] D. Heckerman, D. Geiger, D. M. Chickering, Learning Bayesian networks: the combination of knowledge and statistical data, Machine Learning, **20**(3):197-243, 1995.

[HKL02] K. Huang, I. King, R. Lyu, Constructing a large node Chow-Liu tree based on frequent itemsets, in Proceedings of the 9th International Conference on Neural Information Processing (ICONIP'02), **1**:498-502, 2002.

[HKL03] K. Huang, I. King, M. R. Lyu, Discriminative Training of Bayesian Chow-Liu Multinet Classifiers, in Proceedings of the International Joint Conference on Neural Networks, **1**:484-488, 2003.

[HMC97] D. Heckerman, C. Meek, G. Cooper, A Bayesian approach to causal discovery, Technical report MSR-TR-97-05, Microsoft Research, 1997.

[Hsu04] W. H. Hsu, Genetic wrappers for feature selection in decision tree induction and variable ordering in Bayesian network structure learning, Information Sciences: International Journal, **163**(1-3):103-122, 2004.

[HJR00] J. Hellerstein, T. S. Jayram, I. Rish, Recognizing end-user transaction in performance management, in: Proceedings of AAAI-2000, Austin, Texas, pp. 596-602, 2000.

[HY01] D. J. Hand, K. Yu, Idiot's Bayes – Not so stupid after all?, International Statistical Review, **69:3**:385-398, 2001.

[IL+00] I. Inza, P. Larrañaga, R. Etxeberria, B. Sierra, Feature subset selection by Bayesian Networks based optimization, Artificial Intelligence, 123:157-184, 2000.

[Jen01] F. V. Jensen, Bayesian Networks and Decision Graphs, New York, Springer Verlag, 2001.

[Jen96] F. V. Jensen, An introduction to Bayesian Networks, UCL Press Ltd, London, ISBN 1-85728-332-5, 1996.

[JJ98] T. S. Jaakkola, M. I. Jordan, Learning in graphical models, chapter Improving the mean field approximations via the use of mixture distributions, Kluwer Academic Publishers, 1998.

[JKP94] G. H. John, R. Kohavi, K. Pfleger, Irrelevant features and the subset selection problem, Machine Learning: W. W. Cohen, H. Hirsh (eds) Proceedings of the 11th International Conference, 121-129, Morgan Kaufmann, 1994.

[JL+04] J. Ji, C. Liu, J. Yan, N. Zhong, Bayesian Network Structure Learning and Its Application to Personalized Recommendations in a B2C Portal, in: Proceedings of the Web Intelligence, IEEE/WIC/ACM International Conference on (WP'04), **00**, 2004.

[JLO90] F. Jenson, S. Lauritzen, K. Olesen, Bayesian updating in causal probabilistic networks by local computation, Computational Statistics Quartely **4**: 269-282, 1990.

[JN97] N. Jitnah, A. Nicholson, treenets: A framework for anytime evaluation of belief networks, in: 1st International Joint Conference on Qualitative and Quantitative Practical reasoning (ECSQARU-FAPR'97), Lecture notes in Artificial Intelligence, Springer-Verlag, 1997.

[JN99] N. Jitnah, A. Nicholson, Arc Weights for approximate evaluation of dynamic belief networks, Artificial Intelligence (AI99), Sydney, 1999.

[JZ97] A. K. Jain, D. Zongker, Feature selection: evaluation, application and small sample performance, IEEE Trans. On Pattern Analysis and Machine Intelligence, **19**(2):153-158, 1997.

[KaS04] M. J. Kane, A Savakis, Bayesian Network Structure Learning and Inference in Indoor vs. Outdoor Image Classification, in: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04), **2-02**:479-482, 2004.

[KC02] N. Kwak, C. Choi, Input Feature Selection by Mutual Information Based on Parzen Window, IEEE Trans. On Pattern Analysis and Machine Intelligence, **24**(12), 1667-1671, 2002.

[KJ97] R. Kohavi, G. H. John, Wrappers for feature subset selection, In Artificial Intelligence, **97** (1-2):273-324, 1997.

[KJ+94] R. Kohavi, G. John, R. Long, D. Manley, K. Pfleger, MLC++: A machine learning library in C++, in: Proceedings of 6th International conference on Tools with Artificial Intelligence, IEEE Computer Society, 1994.

[KJ96] G. D. Kleiter, R. Jiroušek, Learning Bayesian Networks under the control of mutual information, in: Proceedings of 6th International Conference IPMU-1996, 1996.

[Kja94] U. Kjærulf, Reduction of computational complexity in Bayesian networks through removal of weak dependencies, In Proceedings of the tenth Conference on Uncertainty in Artificial Intelligence, 374-382, 1994.

[KL51] S. Kullback, R. A. Leibler, On information and sufficiency, Annals of Statistics 22, 79-86, 1951.

[Koh94] Kohavi, Features subset selection as search with probabilistic estimates, In: AAAI Fall Symposium on Relevance, 122-126, 1994.

[Koh95] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimates and model selection, in: Proceedings 14th International Joint Conference on Artificial Intelligence, Morgan Kaufmann, pp. 1137-1143, 1995.

[KR92] K. Kira, L. Rendell, A practical approach to feature selection, In Proceedings of the 9th International Conference on machine learning, 249-256, Morgan Kaufmann, 1992.

[KS95] R. Kohavi, D. Summerfield, Feature subset selection using the wrapper model: Overfitting and dynamic search space topology, In 1st International Conference on Knowledge Discovery and Data Mining, 192-197, 1995.

[KS96] D. Koller, M. Sahami, Toward optimal feature selection, In 13th International Conference in machine learning, 1996.

[KS00] A. Kleiner, B Sharp, A New Algorithm for Learning Bayesian Classifiers from Data, in: Proceedings of 3rd IASTED International Conference on Artificial Intelligence and Soft Computing (ASC 2000), Banff, Canada, 2000.

[KS04] M. Koivisto, K. Sood, Exact Bayesian Structure Discovery in Bayesian Networks, The Journal of Machine Learning Research 5, MIT press Cambridge, MA, 2004.

[Kul68] S. Kullback, Information Theory and Statistics, Dover Publications, New York, 1968.

[Kur87] M. N. Kurzynski, Diagnosis of acute abdominal pain using three-stage classifier, Comp. Bio Med, **17**(1):19-27, 1987.

[Lan94] P. Langley, Selection of relevant features in machine learning, In R. Greiner (ed) Proceedings of AAAI Fall Symposium on Relevance, 140-144, AAAI Press, 1994.

[Lar02] P. Larrañaga, Learning Bayesian networks from data : some applications in Biomedicine, Workshop of Intelligent Data Analysis in medicine and Pharmacology, IDAMAP2002, Lyon, France, 2002.

[LB94] W. Lam, F. Bacchus, Learning Bayesian belief networks: An approach based on the MDL principle, Computational Intelligence **10** (4), 1994.

[LE93] B. T. W. Luken, C. Emerman, The National History and Clinical Findings in Undifferentiated Abdominal Pain. Annals of Emergency Medicine, **22.4**:690-696, 1993.

[LIT92] P. Langley, W. Iba, K. Thompson, An analysis of Bayesian classifiers, in: AAAI'90, pp. 223-228, 1992.

[LK+96] P. Larrañaga, C. M. H. Kuijpers, R..H. Murga, Y. Yurrassendi, Learning Bayesian network structures by searching for the best ordering with genetic algorithms, IEEE Transactions on Systems, Man and Cybernetics **26** (4), pp. 487-493, 1996.

[LKM01] M. Last, A. Kandel, O. Maimon, Information-Theoretic algorithm for feature selection, Pattern Recognition Letters, **22**:799-811, 2001.

[LS88] S. Lauritzen, D. Spiegelhalter, Local Computations with Probabilities on graphical structures and their application to Expert Systems, Journal of the Royal Statistical Society, **B50**(2): 157-224, 1988.

[LS94] P. Langley, S. Sage, Induction of Selective Bayesian classifiers, in: Proceedings of the 10th Conference on Uncertainty in AI, Morgan Kaufmann, pp 399-406, 1994.

[LTD01] G. Li, F. Tong, H. Dai, Evolutionary Structure Learning Algorithms for Bayeisan Network and Penalized Mutual Information Metric, in: Proceedings of the 2001 IEEE International Conference on Data Mining, 2001.

[LWY04] A. W. Liew, Y. Wu, H. Yan, Selection of Statistical Features Based on Mutual Information for Classification of Human Coding and Non-coding DNA Sequences, in: Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04), 766-769, 2004.

[MA95] P. .M. Murphy, D. W. Aha, UCI Repository of Machine Learning databases, 1995. *http://www.ics.edu/$\sim$mlean/MLRepository.html*.

[Mac98] D.J.C. Mackay, Learning in Graphical models, chapter Introduction to Monte Carlo methods, Kluwer Academic Publishers, 1998.

[Mad02] M. G. Madden, Evaluation of the Performance of the Markov Blanket Bayesian Classifier Algorithm, Technical Report NUIG-IT-011002, Department of Information Technology, National University of Ireland, Galway, Ireland, 2002.

[MC+01] K. McNaught, S. Clifford, M. Vaughn, A. Fogg, M. Foy, A Bayesian Belief network for lower back pain diagnosis, in: European Conference on Artificial Intelligence in Medicine (AIME'01): working notes for workshop, pp. 53-58, 2001.

[Mei99] M. Meila-Predoviciu, Learning with mixture of trees, PhD thesis, Massachusetts Institute of Technology, 1999.

[Mor74] M. J. Moroney, Facts from Figures, Mathematics & Statistics, ISBN 01402.02366, 1974.

[MR94] D. Madigan, A. Rafferty, Model selection and accounting for model uncertainty in graphical models using Occam's window, Journal of the American Statistical Association, 1994.

[MST94] (Eds.) D. Michie, D. J. Spiegelhalter, C. C. Taylor, Machine Learning, Neural and Statistical Classification, Ellis Horwood Series in Artificial Intelligence, 1994.

[MT00] D. Margaritis, S. Thrun, Bayesian Network Induction via local neighborhoods, In Advances in Neural Information Processing Systems 12:505-511, S. A. Solla, T. K. Leen, K. R. Müller (eds), MIT Press, 2000.

[Nea90] R. Neapolitan, Probabilistic Reasoning in Expert Systems, John Wiley & Sons, New York, 1990.

[Nea03] R. E. Neapolitan, Learning Bayesian Networks, Prentice Hall, 2003.

[Ng98] A. Y. Ng, On feature selection learning with exponentially many irrelevant features as training examples, In Proceedings of 15th Conference on Machine Learning, 404-412, 1998.

[NJ75] M. Norusis, J. Jacquez, Diagnosis I: symptom nonindependence in mathematical models for diagnosis, Computers and Biomedical Research, **8**:156-172, 1975.

[NJ98] A. E. Nicholson, N. Jitnah, Using mutual Information to determine relevance in Bayesian networks, Pacific RIM International Conference on Artificial Intelligence, pp 399-410, 1998.

[OM+96] C. Ohmann, V. Moustakis, Q. Yang, K. Lang, Evaluation of automatic knowledge acquisition techniques in the diagnosis of acute abdominal pain, Artificial Intelligence in Medicine, **8**:23-36, 1996.

[OY+95] C. Ohmann, Q. Yang, V. Moustakis, K. Lang, P. J. Van Elk, Machine Learning Techniques applied to diagnosis of acute abdominal pain, In Pedro Barahona and Mario Stefanelli, Editors, Lecture notes in Artificial Intelligence: Artificial Intelligence in Medicine AIME95, **934**:276-281, Springer, 1995.

[Pan02] H. P. Pan, Learning Bayesian networks: II – a computational algorithm, in: 5th International Conference on Information Fusion, (submitted), Anapolis, Maryland, USA, 2002.

[Paz96] M. J. Pazzani, Searching for Dependencies in Bayesian Classifiers, Learning from Data: AI and Statistics V, 239-248, Edited by D. Fisher, H. J. Lenz, Springer-Verlag, 1996.

[PC93] G. M. Provan, J. R. Clarke, Dynamic Network Construction and Updating Techniques for the Diagnosis of Acute Abdominal Pain, IEEE Transactions on Pattern Analysis and Machine Intelligence, **15:3**:299-307, 1993.

[PD03] H. Peng, C. Ding, Structure Search and Stability Enhancement of Bayesian Networks, in: Proceedings of the Third IEEE International Conference on Data Mining, 2003.

[Pea86] J. Pearl, Fusion, propagation and structuring in belief networks, Artificial Intelligence, **29**: 241-288, 1986.

[Pea87] J. Pearl, Evidential Reasoning using stochastic simulation of causal models, Artificial Intelligence **32**(2): 245-258, 1987.

[Pea88] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann Publishers, INC., San Mateo, Californian. ISBN 0-934613-73-7, 1988.

[Ped98] T. Pedersen, Naïve Bayes as a satisficing model, Working notes of the AAAI Spring Symposium on satisficing models, Palo Alto, CA, pp. 60-67, 1998.

[PEJ98] E. Pesonen, M. Eskelinen, M. Juhola, Treatment of missing data values in a neural network based decision support system for acute abdominal pain, Artificial Intelligence in Medicine, **13**(3):139-146, 1998.

[PH88] H. I. Pass, T. D. Hardy, The appendix, in Hardy's Textbook of Surgery. Philadelphia: JB Lippincott, pp 574-581, 2nd edition, 1988.

[PS96] G. M. Provan, M. Singh, , Data Mining and model simplicity: A case study in diagnosis, Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KKD), 1996.

[Qui93] J. R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, San Francisco, 1993.

[Ris01a] I. Rish, An empirical study of the Naive Bayes classifier, in: 17th International Joint Conference on Artificial Intelligence, Seattle, Washington, 2001.

[Ris01b] I. Rish, An analysis of data characteristics that affect Naive Bayes performance, Technical Report RC21993, IBM T. J. Watson Research Center, 2001.

[RP87] G. Rebane, J. Pearl, The recovery of causal polytrees from statistical data, in: Proceedings of 3rd conference On Uncertainty in Artificial Intelligence. Seattle, WA, 1987.

[SB00] R. Sangüesa, P Burrell, Application of Bayesian Network Learning Methods to Waste Water Treatment Plants, Applied Intelligence, **13**(1), 2000.

[Ser86] B. Seroussi, Computer-aided diagnosis of acute abdominal pain when taking into account interactions, Methods of Information in Medicine, **25**:194-198, 1986.

[SGS00] P. Spirtes, C. Glymour, R. Sheines, Constructing Bayesian network models of gene expression networks from microarray data, In Proceedings of the Atlantic Symposium on Computational Biology, Genome Information Systems and Technology, 2000.

[SGS90] P. Spirtes, C. Glymour, R. Scheines, Causality from probability, in: Proceedings of Advanced Computing for social sciences, Williamsburgh, VA, 1990.

[SGS91] P. Spirtes, C. Glymour, R. Scheines, An algorithm for fast recovery of sparse causal graphs, Social Science Computer review 9, 1991, pp. 62-72.

[SGS96] P. Spirtes, C. Glymour, R. Scheines, Causation, Prediction, and Search (book), http://hss.cmu.edu/html/departments/philosophy/TETRAD.BOOK/book.html, 1996.

[Sha88] R. Shachter, Probabilistic Inference and Influence Diagrams, Operational Research **36**(6): 589-604, 1988.

[Shi00] B. Shipley, Cause and correlation in Biology, A Users' Guide to Path Analysis, Structural Equations and causal inference, Cambridge, 2000.

[Sin98] M. Singh, Learning Bayesian Networks for solving real-world problems, PhD Thesis, Department of Computer and Information Science, University of Pennsylvania, 1998.

[SJJ96] L. K Saul, T. S. Jaakkola, M. I. Jordon, Mean field theory for sigmoid belief networks, JAIR **4**: 61-76, 1996.

[SP+97] L. E. Sucar, J. Pérez-Brito, J. C Ruiz-Suárez, E. F. Morales, Learning Structure from Data and its application to Ozone Prediction, Applied Intelligence, **7**(4): 327-338, 1997.

[SP+02] S. Souafi-Bensafi, M. Parizean, F. Lebourgeois, H. Emptuz, Bayesian network classifiers applied to documents, in : Proceedings of the 16th International Conference on Pattern Recognition, 1:483-486, 2002.

[SP89] R. D. Shachter, M. A. Peot, Simulation Approaches to general probabilistic inference on belief networks, In Proceedings of the fifth Conference on Uncertainty in Artificial Intelligence (UAI-89), 606-612, 1989.

[SR+04] V. Sindhwani, S. Rakshit, D. Deodhare, D. Erdogmus, J. C. Principe, P. Niyogi, Feature Selection in MLPs and SVMs Based on Maximum Output Information, IEEE Trans. On Neural Networks 15, (4):937-948, 2004.

[SRA90] S. Srinivas, S. Russell, A. Agogino, Automated construction of sparse Bayesian networks from unstructured probabilistic models and domain information, In M. Henrion, R. D. Shachter, L. N. Kanal, J. F. Lemmer, eds., Uncertainty in Artificial Intelligence 5, 1990.

[SS94] N, Sheikholesiami, D. Stashuk, Information based feature selection for supervised motor unit action potential classification, in: Engineering in Medicine and Biology, : Engineering Advances: New Opportunities for Biomedical Engineers, Proceedings of the 16th Annual International Conference of IEEE, **2**:1350-1351, 1994.

[Ste00] T. A. Stephenson, An Introduction to Bayesian Network Theory and Usage, Dalle Molle Institute for Perceptual Artificial Intelligence, Research Report, IDIAP-RR 00-03, 2000.

[Suz96] J. Suzuki, Learning Bayesian belief networks based on the minimum description length principle: an efficient algorithm using the b&b technique, In Proceedings of 13th International Conference on Machine Learning, 1996.

[SV93] M. Singh, M. Vatonta, An Algorithm for the construction of Bayesian network structures from data, in: D. Heckerman, E. Mamdani, eds, Uncertainty in Artificial Intelligence: Proceedings of the ninth Conference, pp. 259-265, San Mateo, CA. Morgan Kaufmann, 1993.

[SV95] M. Singh, M. Valtorta, Construction of Bayesian Network Structure from Data: a Brief Survey and an Efficient Algorithm, International Journal of Approximate Reasoning, **12**:111-131, 1995.

[TA03] I. Tsamardinos, C. F. Aliferis, Towards Principled Feature Selection : Relevancy, Filters and Wrappers, In the Proceedings of the 9th International Workshop on Artificial Intelligence and Statistics, 3-6, Key west, Florida, 2003.

[TC+03] A. Tsymbal, P. Cunningham, M. Pechenizkiy, S. Puuronen, Search strategies for ensemble feature selection in medical diagnosis, Proceedings of the 16th Annual IEEE Symposium on computer-based medical systems, New York, pp 124-129, 2003.

[TF+01] G. D. Tourassi, E. D. Frederick, M. K. Markey, C. E. Floyd Jr, Application of the mutual Information criterion for feature selection in computer-aided diagnostics, Medical Physics, **28**(12), 2001.

[Tho96] C. S Thomas, A Mutual Information Measure Classifier, Technical Report CSM-137, Department of Computing Science and Mathematics, University of Stirling, UK, 1996.

[Tho99] C. S. Thomas, Classifying acute abdominal pain by assuming independence: A study using two models constructed from data, Technical Report CSM-153, Department of Computing Science and Mathematics, University of Stirling, Stirling, UK, 1999.

[THS05a] C. S. Thomas, C. A. Howie, L. S. Smith, A New Singly Connected Network Classifier Based on Mutual Information, Intelligent Data Analysis, **9**(2), 2005.

[THS05b] C. S. Thomas, C. A. Howie, L. S. Smith, A Case Study in Diagnosis: Classifying Acute Abdominal Pain without Assuming Extreme Conditional Independence, Submitted to Statistics in Medicine, 2005.

[TS94] B. S. Todd, R. Stamper, The relative accuracy of a variety of medical diagnostic programs, Methods of Information in Medicine, **33**(4): 402-416, 1994.

[VP92] T. Verma, J. Pearl, An algorithm for deciding if a set of observed independencies has a causal explanation, in: Proceedings of the 8th Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, pp. 323-330, 1992.

[WF00] I. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Technologies with Java Implementations, Morgan Kaufmann, 2000.

[WK01] M. Woziak, M. Kurzynski, Generating classifier for acute abdominal pain diagnosis problem, Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Vols 1-4 building new bridges at the frontiers of engineering and medicine, pp 3819-3821, 2001.

[YG04] S. Yang, J. Gu, Feature selection based on mutual information and redundancy-synergy coefficient, Journal of Zhejiang University Science, **5**(11), 1382-1391, 2004.

[Yor92] J. York, Bayesian methods for the analysis of misclassified or in-complete multivariate discrete data, Ph.D thesis, Department of Statistics, University of Washington, Seattle, 1992.

[YZ01] Z. R. Yang, M. Zwolinski, Mutual Information Theory for Adaptive Mixture Models, IEEE Transactions On Pattern Analysis and Machine Intelligence **23** (4), 396-403, 2001.

[ZF03] M. Zaffalon, E. Faginoli, Tree-based credal networks for classification, Reliable Computing, **9**(6): 487-509, 2003.

[ZL02] H. Zhang, Y. Lu, Learning Bayesian network classifiers from data with missing values, in: Proceedings of IEEE TENCON'02, 35-38, 2002.

[ZW+03] L. Zhang, J. Wang, Y. Zhao, Z. Yang, A novel hybrid feature selection algorithm: using RELIEFF estimation for GA_Wrapper search, in: Proceedings of the second International Conference on Machine Learning and Cybernetics, Xi'an, **2-5**:380-384, 2003.

APPENDIX A - Diagnostic and Symptom Codes AAP

Table A1-1: Symptom Parameters and Codes

Symptom	Value
SEX	male(1/1), female(1/2)
AGE	0-9(2/1), 10-19(2/2), 20-29(2/3), 30-39(2/4), 40-49(2/5), 50-59(2/6), 60-69(2/7), 70 +(2/8)
Pain-site Onset	right upper quadrant(3/1), left upper quadrant(3/2), right lower quadrant(3/3), left lower quadrant(3/4), upper half(3/5), lower half(3/6), right half(3/7), left half(3/8), central(3/9), general(3/10), right loin(3/11), left loin(3/12), epigastric(3/13), right upper quadrant + epigastric(3/14), right lower quadrant + left lower quadrant(3/15), right lower quadrant + right loin(3/16)
Pain-site Present	right upper quadrant(4/1), left upper quadrant(4/2), right lower quadrant(4/3), left lower quadrant(4/4), upper half(4/5), lower half(4/6), right half(4/7), left half(4/8), central(4/9), general(4/10), right loin(4/11), left loin(4/12), epigastric(4/13), pain settled(4/14), right upper quadrant + epigastric(4/15), right lower quadrant + central(4/16), right lower quadrant + right loin(4/17), left lower quadrant + left loin(4/18), right half + right loin(4/19), left half + left loin(4/20), central + epigastric(4/21)
Aggravating Factors	movement(5/1), coughing(5/2), inspiration(5/3), food(5/4), other(5/5), nil(5/6), movement + coughing(5/7), movement + inspiration(5/8), movement + food(5/9), movement + other(10), movement + coughing + inspiration(5/11), coughing + inspiration(5/12)
Relieving Factors	lying still(6/1), vomiting(6/2), antacids(6/3), milk/food(6/4), other(6/5), nil(6/6), lying still + vomiting(6/7), lying still + other(6/8)
Progress of Pain	getting better(7/1), no change(7/2), getting worse(7/3)
Duration of Pain	under 12 hours(8/1), 12-24 hours(8/2), 24-48 hours(8/3), over 48 hours(8/4)
Type of Pain	steady(9/1), intermittent(9/2), colicky(9/3), sharp(9/4), steady + intermittent(9/5), steady + colicky(9/6), steady + sharp(9/7), steady + colicky + sharp(9/8), intermittent + colicky(9/9), intermittent + sharp(9/10), intermittent + colicky + sharp(9/11), colicky + sharp(9/12)
Severity of Pain	moderate(10/1), severe(10/2)
Nausea	nausea present(11/1), no nausea(11/2)
Vomiting	present(12/1), no vomiting(12/2)
Anorexia	present(13/1), normal appetite(13/2)
Indigestion	history of dyspepsia(14/1), no history of dyspepsia(14/2)
Jaundice	history of jaundice(15/1), no history of jaundice(15/2)
Bowel habit	no change(16/1), constipated(16/2), diarrhoea(16/3), blood(16/4), mucus(16/5), constipated + diarrhoea(16/6), diarrhoea + blood(16/7)
Micturition	normal(17/1), frequent(17/2), dysuria(17/3), haematuria(17/4), dark urine(17/5), frequent + dysuria(17/6)
Previous Pain	similar pain before(18/1), no similar pain before(18/2)
Previous surgery	yes(19/1), none(19/2)
Drugs	being taken(20/1), not being taken(20/2)
Mood	normal(21/1), distressed(21/2), anxious(21/3), distressed + anxious(21/4)
Colour	normal(22/1), pale(22/2), flushed(22/3), jaundiced(22/4), cyanosed(22/5)
Abdominal Movement	normal(23/1), poor/nil(23/2), visible peristalsis(23/3)
Abdominal scar	present(24/1), absent(24/2)
Abdominal Distension	present(25/1), absent(25/2)
Site of Tenderness	right upper quadrant(26/1), left upper quadrant(26/2), right lower quadrant(26/3), left lower quadrant(26/4), upper half(26/5), lower half(26/6), right half(26/7), left half(26/8), central(26/9), general(26/10), right loin(26/11), left loin(26/12), epigastric(26/13), none(26/14), right upper quadrant + epigastric(26/15), right lower quadrant + left lower quadrant (26/16), right lower quadrant + right half(26/17), right lower quadrant + central(26/18), right lower quadrant + right loin(26/19), right lower quadrant + epigastric(26/20), left lower quadrant + left loin(26/21), left half + left loin(26/22)
Rebound	present(27/1), absent(27/2)
Guarding	present(28/1), absent(28/2)
Rigidity	present(29/1), absent(29/2)
Abdominal Masses	present(30/1), absent(30/2)
Murphy's test	positive(31/1), negative(31/2)
Bowel sounds	normal(32/1), decreased/absent(32/2), increased(32/3)
Rectal Examination	tender left side(33/1), tender right side(33/2), generally tender(33/3), mass felt(33/4), normal(33/5)

APPENDIX B – Doctors' Suggested Symptoms (Taken from Questionnaires)

Table B1-1: Symptom Parameters for APP (Age : Generally Young)

Symptom name	Symptom Parameter Name/Group APP	Parameter (code)	MIM Tree
Sex	-	-	1(1)
Age	Age 0-9	2(1)	2(2)
Pain-site Onset	Right Lower Quadrant	3(9)	3(3)
Pain-site Present	Right Lower Quadrant	4(3)	4(3)
Aggravating Factors	Movement	5(1)	-
Aggravating Factors	Coughing	5(2)	5(2)
Relieving Factors	Lying Still	6(1)	6(1)
Progress of Pain	Getting worse	7(3)	7(3)
Duration of Pain	12-24 hours	8(2)	8(2)
Type of Pain	Steady	9(1)	9(1)
Type of Pain	Sharp	9(4)	-
Severity of Pain	Moderate	10(1)	10(2)
Nausea	Nausea Present	11(1)	11(1)
Vomiting	Present	12(1)	12(1)
Anorexia	Present	13(1)	13(1)
Indigestion	No History of Dyspepsia	14(2)	14(2)
Jaundice	No History of Jaundice	15(2)	15(2)
Bowel Habit	Diarrhoea	16(3)	16(3)
Micturition	Normal	17(1)	17(1)
Previous Pain	No Similar Pain Before	18(2)	18(2)
Previous Surgery	No	19(2)	19(2)
Drugs	Not Being Taken	20(2)	20(2)
Mood	Distressed	21(2)	-
Mood	Anxious	21(3)	21(3)
Colour	Flushed	22(3)	22(3)
Abdominal Movements	Poor/nil	23(2)	23(2)
Abdominal Scar	Absent	24(2)	24(2)
Abdominal Distension	Absent	25(2)	25(2)
Site of Tenderness	Right Lower Quadrant	26(3)	26(3)
Rebound	Present	27(1)	27(1)
Guarding	Present	28(1)	28(1)
Rigidity	Absent	29(2)	29(1)
Abdominal Masses	Absent	30(2)	30(2)
Murphy's Test	Negative	31(2)	31(1)
Bowel Sounds	Normal	32(1)	32(2)
Rectal Examination	Tender Right Side	33(2)	33(2)

Notes: 4-15% over 55, 5% under 55. Patient generally looks ill, Fever.

Key: Field 'MIM Tree' represents the MIM classifier structure symptom values with respect to the disease APP.

Table B2-1: Symptom Parameters for DIV (Age : Generally Old)

Symptom name	Symptom Parameter Name/Group DIV	Parameter (code)	MIM Tree
Sex	-	-	1(2)
Age	Age 60-69	2(7)	2(8)
Pain-site Onset	Left Lower Quadrant	3(4)	3(3)
Pain-site Present	-	-	4(4)
Aggravating Factors	Movement	5(1)	-
Aggravating Factors	Coughing	5(2)	5(2)
Relieving Factors	Lying Still	6(1)	6(1)
Progress of Pain	Getting worse	7(3)	7(3)
Duration of Pain	Over 48 hours	8(3)	8(4)
Type of Pain	Steady	9(1)	9(1)
Severity of Pain	-	-	10(1)
Nausea	Nausea Present	11(1)	11(1)
Vomiting	No Vomiting	12(2)	12(2)
Anorexia	-	-	13(1)
Indigestion	No History of Dyspepsia	14(2)	14(1)
Jaundice	No History of Jaundice	15(2)	15(2)
Bowel Habit	Constipated	16(2)	16(2)
Bowel Habit	Diarrhoea	16(3)	-
Micturition	Normal	17(1)	17(2)
Previous Pain	Similar Pain Before	18(1)	18(2)
Previous Surgery	None	19(2)	19(1)
Drugs	Not Being Taken	20(2)	20(1)
Mood	Distressed	21(2)	21(2)
Colour	Flushed	22(3)	22(2)
Abdominal Movements	Normal	23(1)	23(2)
Abdominal Scar	Absent	24(2)	24(2)
Abdominal Distension	Absent	25(2)	25(1)
Site of Tenderness	Left Lower Quadrant	26(4)	26(4)
Rebound	Present	27(1)	27(1)
Guarding	Present	28(1)	28(1)
Rigidity	-	-	29(1)
Abdominal Masses	Present	30(1)	30(1)
Murphy's Test	Negative	31(2)	31(2)
Bowel Sounds	Normal	32(1)	32(3)
Rectal Examination	Generally Tender	33(2)	33(3)

Notes: Uncommon before 40. 50% - over 80, 6% over 55, <1% under 55. Fever.

Table B3-1: Symptom Parameters for PPU (Age : Generally Old)

Symptom name	Symptom Parameter Name/Group PPU	Parameter (code)	MIM Tree
Sex	Male	1(1)	1(1)
Age	Age 50-59	2(6)	2(8)
Pain-site Onset	Upper Half	3(5)	3(5)
Pain-site Present	Lower Half	4(5)	4(10)
Aggravating Factors	Movement	5(1)	5(1)
Aggravating Factors	Coughing	5(2)	-
Relieving Factors	Lying Still	6(1)	6(4)
Progress of Pain	Getting worse	7(3)	7(3)
Duration of Pain	12-24 hours	8(2)	8(1)
Type of Pain	Steady	9(1)	9(1)
Severity of Pain	Severe	10(2)	10(2)
Nausea	-	-	11(1)
Vomiting	No Vomiting	12(2)	12(1)
Anorexia	Present	13(1)	13(1)
Indigestion	History of Dyspepsia	14(1)	14(1)
Jaundice	No History of Jaundice	15(2)	15(2)
Bowel Habit	No Change	16(1)	16(2)
Micturition	Normal	17(1)	17(99)
Previous Pain	No Similar Pain Before	18(2)	18(2)
Previous Surgery	Yes	19(1)	19(1)
Drugs	Being Taken	20(1)	20(1)
Mood	Distressed	21(2)	21(2)
Colour	Pale	22(2)	22(2)
Abdominal Movements	Poor/nil	23(2)	23(2)
Abdominal Scar	Absent	24(2)	24(2)
Abdominal Distension	Absent	25(2)	25(99)
Site of Tenderness	Upper Half	26(5)	26(10)
Rebound	Present	27(1)	27(1)
Guarding	Present	28(1)	28(1)
Rigidity	Present	29(1)	29(1)
Abdominal Masses	Absent	30(2)	30(2)
Murphy's Test	Negative	31(2)	31(2)
Bowel Sounds	Decreased/Absent	32(2)	32(2)
Rectal Examination	Normal	33(5)	33(5)

Table B4-1: Symptom Parameters for CHO (Age : Generally Old)

Symptom name	Symptom Parameter Name/Group CHO	Parameter (code)	MIM Tree
Sex	Male	1(1)	1(2)
Age	Age 60-69	2(7)	2(8)
Pain-site Onset	Right Upper Quadrant	3(1)	3(1)
Pain-site Present	Right Upper Quadrant	4(1)	4(1)
Aggravating Factors	Coughing	5(2)	5(4)
Relieving Factors	-	-	6(6)
Progress of Pain	Getting worse	7(3)	7(3)
Duration of Pain	Over 48 hours	8(3)	8(1)
Type of Pain	Sharp	9(4)	9(1)
Type of Pain	Severe	10(2)	10(2)
Nausea	Nausea Present	11(1)	11(1)
Vomiting	Present	12(1)	12(1)
Anorexia	Present	13(1)	13(1)
Indigestion	No History of Dyspepsia	14(2)	14(1)
Jaundice	-	-	15(1)
Bowel Habit	No change	16(1)	16(2)
Micturition	Normal	17(1)	17(5)
Previous Pain	Similar Pain Before	18(1)	18(1)
Previous Surgery	None	19(2)	19(1)
Drugs	Not Being Taken	20(2)	20(2)
Mood	Distressed	21(2)	21(2)
Colour	Jaundiced	22(4)	22(4)
Abdominal Movements	-	-	23(2)
Abdominal Scar	-	-	24(1)
Abdominal Distension	Absent	25(2)	25(2)
Site of Tenderness	Right Upper Quadrant	26(1)	26(1)
Rebound	-	-	27(2)
Guarding	Present	28(1)	28(1)
Rigidity	-	-	29(1)
Abdominal Masses	Absent	30(2)	30(1)
Murphy's Test	Positive	31(1)	31(1)
Bowel Sounds	Normal	32(1)	32(2)
Rectal Examination	Normal	33(5)	33(5)

Notes: Over 55 12-20%, under 55 5%, Fever. Severity of Pain helps distinguish CHO from PPU.

Table B5-1: Symptom Parameters for INO (Age : Generally Old)

Symptom name	Symptom Parameter Name/Group INO	Parameter (key)	MIM Tree
Sex	-	-	1(2)
Age	Age 70+	2(8)	2(8)
Pain-site Onset	Lower Half	3(6)	-
Pain-site Onset	Central	3(9)	3(9)
Pain-site Present	Lower Half	4(6)	4(9)
Pain-site Present	Central	4(9)	-
Aggravating Factors	-	-	5(6)
Relieving Factors	Vomiting	6(2)	6(2)
Progress of Pain	Getting worse	7(3)	7(3)
Duration of Pain	-	-	8(4)
Type of Pain	Colicky	9(3)	9(3)
Severity of Pain	Moderate	10(1)	10(1)
Nausea	Nausea present	11(1)	11(1)
Vomiting	Present	12(1)	12(1)
Anorexia	Present	13(1)	13(1)
Indigestion	No History of Dyspepsia	14(2)	14(2)
Jaundice	No History of Jaundice	15(2)	15(2)
Bowel Habit	Constipated	16(2)	16(2)
Micturition	Normal	17(1)	17(1)
Previous Pain	Similar pain before	18(1)	18(1)
Previous Surgery	Yes	19(1)	19(1)
Drugs	-	-	20(1)
Mood	-	-	21(2)
Colour	Pale	22(2)	22(2)
Abdominal Movements	Poor/nil	23(2)	23(2)
Abdominal Scar	Normal	24(1)	24(1)
Abdominal Distension	Present	25(1)	25(1)
Site of Tenderness	Lower Half	26(6)	26(9)
Rebound	Absent	27(2)	27(2)
Guarding	-	-	28(2)
Rigidity	Absent	29(2)	29(2)
Abdominal Masses	Present	30(1)	30(1)
Murphy's Test	Negative	31(2)	31(2)
Bowel Sounds	Increased	32(3)	32(3)
Rectal Examination	Mass felt	33(4)	33(4)

Notes: Over 55 12%, under 55 <1%.

Table B6-1: Symptom Parameters for PAN (Age : Generally Young/Old)

Symptom name	Symptom Parameter Name/Group PAN	Parameter (code)	MIM Tree
Sex	-	-	1(1)
Age	-	-	2(5)
Site of Tenderness	Upper Half	3(5)	3(5)
Site of Tenderness	Central	3(9)	-
Pain-site Present	-	-	4(13)
Aggravating Factors	Food	5(4)	5(2)
Relieving Factors	Lying still	6(1)	6(1)
Progress of Pain	Getting worse	7(3)	7(3)
Duration of Pain	12-24 hours	8(2)	8(2)
Type of Pain	Steady	9(1)	9(1)
Severity of Pain	Severe	10(2)	10(2)
Nausea	Nausea present	11(1)	11(1)
Vomiting	Present	12(1)	12(1)
Anorexia	Present	13(1)	13(1)
Indigestion	History of Dyspepsia	14(2)	14(1)
Jaundice	History of Jaundice	15(1)	15(1)
Bowel Habit	No change	16(1)	16(3)
Micturition	Dark Urine	17(4)	17(5)
Previous Pain	Similar pain before	18(1)	18(1)
Previous Surgery	None	19(2)	19(2)
Drugs	Being Taken	20(1)	20(1)
Mood	-	-	21(2)
Colour	-	-	22(2)
Abdominal Movements	Poor/nil	23(2)	23(2)
Abdominal Scar	Absent	24(2)	24(2)
Abdominal Distension	Absent	25(2)	25(1)
Site of Tenderness	-	-	26(13)
Rebound	Present	27(1)	27(1)
Guarding	Present	28(1)	28(1)
Rigidity	-	-	29(1)
Abdominal Masses	Absent	30(2)	30(1)
Murphy's Test	-	-	31(2)
Bowel Sounds	Decreased/absent	32(2)	32(2)
Rectal Examination	Normal	33(5)	33(5)

Notes: Most often alcohol abuse – blood test, temp increases, fever, and patient looks ill.

Table B7-1: Symptom Parameters for RCO (Age : Generally Young/Old)

Symptom name	Symptom Parameter Name/Group RCO	Parameter (code)	MIM Tree
Sex	-	-	1(1)
Age	Age 30-39	2(4)	2(5)
Pain-site Onset	Right Loin	3(11)	3(11)
Pain-site Present	Right Loin	4(11)	4(11)
Aggravating Factors	Nil	5(6)	5(6)
Relieving Factors	-	-	6(6)
Progress of Pain	Getting worse	7(3)	7(3)
Duration of Pain	Under 12 hours	8(1)	8(1)
Type of Pain	-	-	9(3)
Severity of Pain	Severe	10(2)	10(2)
Nausea	Nausea present	11(1)	11(1)
Vomiting	No vomiting	12(2)	12(1)
Anorexia	Normal appetite	13(2)	13(2)
Indigestion	No history of dyspepsia	14(2)	14(2)
Jaundice	No history of jaundice	15(2)	15(2)
Bowel Habit	No change	16(1)	16(1)
Micturition	Haematuria	17(4)	17(4)
Previous Pain	Similar pain before	18(2)	18(1)
Previous Surgery	None	19(2)	19(2)
Drugs	Not being taken	20(2)	20(2)
Mood	Anxious	21(2)	21(2)
Colour	Normal	22(1)	22(2)
Abdominal Movements	Normal	23(1)	23(1)
Abdominal Scar	Absent	24(2)	24(2)
Abdominal Distension	Absent	25(2)	25(2)
Site of Tenderness	-	-	26(11)
Rebound	Absent	27(2)	27(2)
Guarding	-	-	28(2)
Rigidity	Absent	29(2)	29(2)
Abdominal Masses	Absent	30(2)	30(2)
Murphy's Test	Negative	31(2)	31(2)
Bowel Sounds	Normal	32(1)	32(1)
Rectal Examination	Normal	33(5)	33(5)

Notes: >55 4%, No fever.

Table B8-1: Symptom Parameters for DYS (Age : Generally Young/Old)

Symptom name	Symptom Parameter Name/Group DYS	Parameter (code)	MIM Tree
Sex	Male	1(1)	1(1)
Age	-	-	2(4)
Pain-site Onset	Epigastric	3(13)	3(13)
Pain-site Present	Epigastric	4(13)	4(13)
Aggravating Factor	Food	5(4)	5(4)
Relieving Factor	Antacids	6(3)	6(3)
Progress of Pain	Getting better	7(1)	7(1)
Duration of Pain	12-24 hours	8(2)	8(4)
Type of Pain	Steady	9(1)	9(1)
Severity of Pain	Moderate	10(1)	10(1)
Nausea	Nausea present	11(1)	11(1)
Vomiting	Present	12(1)	12(1)
Anorexia	Normal appetite	13(2)	13(2)
Indigestion	History of dyspepsia	14(1)	14(1)
Jaundice	No History of jaundice	15(2)	15(2)
Bowel Habit	No change	16(1)	16(4)
Micturition	Normal	17(1)	17(1)
Previous Pain	Similar pain before	18(1)	18(1)
Previous Surgery	None	19(2)	19(2)
Drugs	Being taken	20(1)	20(1)
Mood	Normal	21(1)	21(1)
Colour	Normal	22(1)	22(1)
Abdominal Movements	Normal	23(1)	23(1)
Abdominal Scar	Absent	24(2)	24(2)
Abdominal Distension	Absent	25(2)	25(2)
Site of Tenderness	Epigastric	26(13)	26(13)
Rebound	Absent	27(2)	27(2)
Guarding	-	-	28(2)
Rigidity	Absent	29(2)	29(2)
Abdominal Masses	Absent	30(2)	30(2)
Murphy's Test	Negative	31(2)	31(2)
Bowel Sounds	Normal	32(1)	32(1)
Rectal Examination	Normal	33(5)	33(5)

APPENDIX C – Questionnaire Template

1. For the Diagnostic Group --------- tick (✓) the symptom parameters that you consider are most closely associated with this Group.

Please carry out this exercise for **three** age ranges. (Indicate in the appropriate box your chosen age groups)

Symptom Parameters	**Young**	**Young/Old**	**Old**
AGE			
0-9	☐	☐	☐
10-19	☐	☐	☐
20-29	☐	☐	☐
30-39	☐	☐	☐
40-49	☐	☐	☐
50-59	☐	☐	☐
60-69	☐	☐	☐
70+	☐	☐	☐
SEX			
Male	☐	☐	☐
Female	☐	☐	☐
Either	☐	☐	☐
PAIN-SITE-ONSET			
Right Upper Quadrant	☐	☐	☐
Left Upper Quadrant	☐	☐	☐
Right Lower Quadrant	☐	☐	☐
Left Lower Quadrant	☐	☐	☐
Upper Half	☐	☐	☐
Lower Half	☐	☐	☐
Right Half	☐	☐	☐
Left Half	☐	☐	☐
Central	☐	☐	☐
General	☐	☐	☐
Right Lion	☐	☐	☐
Left Lion	☐	☐	☐
Epigastic	☐	☐	☐

Symptom Parameters	Young	Young/Old	Old
AGGRAVATING FACTORS			
Movement			
Coughing			
Inspiration			
Food			
Other			
Nil			
PAIN-SITE-PRESENT			
Right Upper Quadrant			
Left Upper Quadrant			
Right Lower Quadrant			
Left Lower Quadrant			
Upper Half			
Lower Half			
Right Half			
Left Half			
Central			
General			
Right Lion			
Left Lion			
Epigastic			
Pain Settled			
RELIEVING FACTORS			
Lying Still			
Vomiting			
Antacids			
Milk/Food			
Other			
Nil			
PROGRESS OF PAIN			
Getting Better			
No Change			
Getting Worse			

Symptom Parameters	Young	Young/Old	Old
DURATION OF PAIN			
Under 12 hours	☐	☐	☐
12 - 24 hours	☐	☐	☐
24 – 48 hours	☐	☐	☐
Over 48 hours	☐	☐	☐
SEVERITY OF PAIN			
Moderate	☐	☐	☐
Severe	☐	☐	☐
TYPE OF PAIN			
Steady	☐	☐	☐
Intermittent	☐	☐	☐
Colicky	☐	☐	☐
Sharp	☐	☐	☐
NAUSEA			
Nausea Present	☐	☐	☐
No Nausea	☐	☐	☐
VOMITING			
Present	☐	☐	☐
No Vomiting	☐	☐	☐
ANOREXIA			
Present	☐	☐	☐
Normal Appetite	☐	☐	☐
JAUNDICE			
History of Jaundice	☐	☐	☐
No history of Jaundice	☐	☐	☐

Symptom Parameters	Young	Young/Old	Old
INDIGESTION			
History of Dyspepsia	☐	☐	☐
No history of Dyspepsia	☐	☐	☐
BOWEL HABIT			
No Change	☐	☐	☐
Constipated	☐	☐	☐
Diarrhea	☐	☐	☐
Blood	☐	☐	☐
Mucus	☐	☐	☐
MICTURITION			
Normal	☐	☐	☐
Frequent	☐	☐	☐
Dysuria	☐	☐	☐
Haematuria	☐	☐	☐
Dark Urine	☐	☐	☐
PREVIOUS PAIN			
Similar Pain before	☐	☐	☐
No similar Pain before	☐	☐	☐
PREVIOUS SURGERY			
Yes	☐	☐	☐
None	☐	☐	☐
DRUGS			
Being taken	☐	☐	☐
Not being taken	☐	☐	☐
MOOD			
Normal	☐	☐	☐
Distressed	☐	☐	☐
Anxious	☐	☐	☐

Symptom Parameters	Young	Young/Old	Old
COLOUR			
Normal	☐	☐	☐
Pale	☐	☐	☐
Flushed	☐	☐	☐
Jaundiced	☐	☐	☐
Cyanosed	☐	☐	☐
ABDOMINAL MOVEMENTS			
Normal	☐	☐	☐
Poor/nil	☐	☐	☐
Visible peristalsis	☐	☐	☐
ABDOMINAL SCAR			
Present	☐	☐	☐
Absent	☐	☐	☐
ABDOMINAL DISTENSION			
Present	☐	☐	☐
Absent	☐	☐	☐
REBOUND			
Present	☐	☐	☐
Absent	☐	☐	☐
GUARDING			
Present	☐	☐	☐
Absent	☐	☐	☐
RIGIDITY			
Present	☐	☐	☐
Absent	☐	☐	☐

Symptom Parameters	**Young**	**Young/Old**	**Old**
SITE-OF-TENDERNESS			
Right Upper Quadrant	☐	☐	☐
Left Upper Quadrant	☐	☐	☐
Right Lower Quadrant	☐	☐	☐
Left Lower Quadrant	☐	☐	☐
Upper Half	☐	☐	☐
Lower Half	☐	☐	☐
Right Half	☐	☐	☐
Left Half	☐	☐	☐
Central	☐	☐	☐
General	☐	☐	☐
Right Lion	☐	☐	☐
Left Lion	☐	☐	☐
Epigastic	☐	☐	☐
None	☐	☐	☐
ABDOMINAL MASSES			
Present	☐	☐	☐
Absent	☐	☐	☐
MURPHY'S TEST			
Positive	☐	☐	☐
Negative	☐	☐	☐
BOWEL SOUNDS			
Normal	☐	☐	☐
Decreased/Increased	☐	☐	☐
Increased	☐	☐	☐
RECTAL EXAMINATION			
Tender Left side	☐	☐	☐
Tender Right side	☐	☐	☐
Generally tender	☐	☐	☐
Mass felt	☐	☐	☐
Normal	☐	☐	☐

2. Please list any additional symptom parameters or tests for the Diagnostic Group --------- that you consider or feel are important in helping you make a correct diagnostic decision.

3. Under what conditions do you decide that a diagnostic decision should be NSAP ?

Thank you for your valuable assistance and time.

Notes for Questionnaires

The questionnaire represents a 'generic' document containing three questions. For each of the eight (true) diagnostic groups, the experts were asked to anonymously complete the fields using their own knowledge. The hospital administrator was additionally requested to distribute the questionnaire to both senior and junior doctors in order to obtain a good cross-section of expertise.

The rationale of each of the questions is as follows:

Question 1: This question is designed to record the expert's list of pertinent symptom parameters with respect to each of the individual diagnostic groups. In order to assess the relationship of the diagnostic groups and patient age, three columns headed Young, Young/Old, and Old are asked to be completed.

Since the options are bounded by the CADA database symptom parameter list, the choices made by the experts will represent a subjective selection.

Question 2: For this question, the experts are asked to suggest additional symptoms or tests that are not included within the CADA database. The objective of this question is to determine what heuristics they applied and potentially what might constitute a useful addition to the existing CADA database definition. Again as for question 1, this is in respect of the eight diagnostic groups.

Question 3: The objective of this question is to determine under what circumstances a patient is assigned to the group NSAP

As this question is not limited by the particulars of the CADA database, the output will be both subjective and heuristic in content.

APPENDIX D – St. John's A&E Patient Record Sheet

Abdominal Pain Chart

NAME | REG NUMBER (Patient ID)

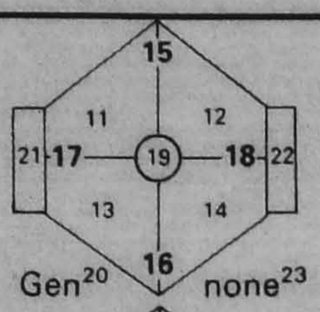
[1]MALE/[2]FEMALE AGE –9,[3] 10s,[4] 20s,[5] 30s,[6] 40s,[7] 50s,[8] 60s,[9] 70+[10] | FORM FILLED BY (Username)

PRESENTATION (999, GP, etc) | DATE | TIME

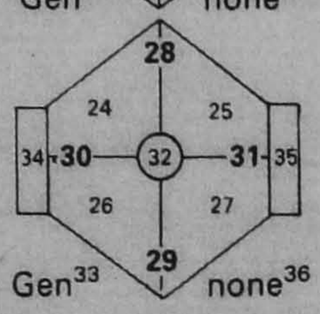
PAIN

SITE

ONSET



PRESENT



RADIATION

AGGRAVATING FACTORS
[37]movement
[38]coughing
[39]respiration
[40]food
[41]other
[42]none

RELIEVING FACTORS
[43]lying still
[44]vomiting
[45]antacids
[46]food
[47]other
[48]none

PROGRESS
[49]better
[50]same
[51]worse

DURATION
–12,[52] 12–23,[53] 24–48,[54]
2–7 days[55]

TYPE
[59]intermittent
[60]steady
[61]colicky
62 MILD

SEVERITY
[63]moderate
[64]severe

HISTORY

NAUSEA
[66]yes no[66]

VOMITING
[67]yes no[68]

ANOREXIA
[69]yes no[70]

PREV INDIGESTION
[71]yes no[72]

JAUNDICE
[73]yes no[74]

BOWELS
[75]normal
[76]constipation
[77]diarrhoea
[78]blood
[79]mucus

MICTURITION
[80]normal
[81]frequency
[82]dysuria
[83]dark
[84]haematuria

PREV SIMILAR PAIN
[85]yes no[86]

PREV ABDO SURGERY
[87]yes no[88]

DRUGS FOR ABDO PAIN
[89]yes no[90]

♀ LMP

pregnant

Vag. discharge

dizzy/faint

EXAMINATION

MOOD
[91]normal
[92]distressed
[93]anxious

SHOCKED
yes no

COLOUR
[94]normal
[95]pale
[96]flushed
[97]jaundiced
[98]cyanosed

TEMP PULSE

BP

ABDO MOVEMENT
[99]normal
[100]poor/nil
[101]peristalsis

SCAR
[102]yes no[103]

DISTENSION
[104]yes no[105]

TENDERNESS

REBOUND
[119]yes no[120]

GUARDING
[121]yes no[122]

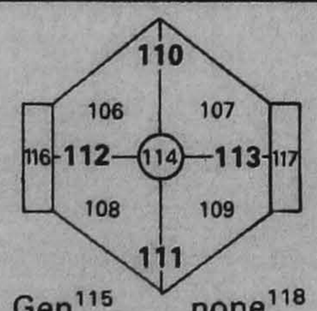
RIGIDITY
[123]yes no[124]

MASS
[125]yes no[126]

MURPHY'S
[127]+ve –ve[128]

BOWEL SOUNDS
[129]normal [130]absent [131]+++

RECTAL — VAGINAL TENDERNESS
[132]left
[133]right
[134]general
[135]mass
[136]none



INITIAL DIAGNOSIS & PLAN

RESULTS
amylase
blood count (WBC)
computer
urine
X-ray
other

DIAG & PLAN AFTER INVEST

(time)

DISCHARGE DIAGNOSIS

History and examination of other systems on separate case notes

APPENDIX E – Notation and Acronyms

Notation	Meaning
$I'()$	Mutual Information Element (subset of MI).
G	Qualitative Graph structure (DAG).
A	Number of edges in graph.
θ	Quantitative parameter.
N	Number of vertices in graph (or nodes).
C	Class variable.
$pa(Z_i)$	The parent of Z_i.
$\{Z_1,\ldots,Z_n\}$	Instantiation vector of feature values.
$\wedge$	Denotes conjunction.
$\pi(V_i)$	Causal support from parents.
$\lambda(V_i)$	Diagnostic support from $V_i's$ children.
$I(Z_i,Z_j)$	Mutual Information Measure for association of Z_i and Z_j.
D	Main data set
$Z\setminus\{C\}$	Feature set excluding class variable C .
$C-Z$	Class to feature association (undirected).
$Z-Z$	Feature to feature association (undirected).
$W_t(\ )$	Branch *Weight* value.
T_m	Maximum Weight Spanning Tree (MWST).
$T_{m_{ci}}$	MWST for class-state C_i.
$cv5$	Cross-validation , 5-folds.
$a \rightarrow b$	Feature to feature association (directed).
$x(y)$	x relevant features with respect to class MB from a maximum of (y) features.
Z_{CL}	Subset of the class MB or *lower bound* (of edges).
Z_R	Remaining edges **not** in class MB but within structure.
X,Y,Z	One-dimensional variables.

A&E: Accident and Emergency.

AAP: Acute Abdominal Pain.

ANN: Artificial Neural Network.

ANOVA: Analysis of Variance.

BAN: BN Augmented Naive Bayes.

BN: Bayesian Network.

CADA: Computer Assisted Diagnostic & Audit.

CI: Conditional Independence.

CL: Chow and Liu.

CPT: Conditional Probability Table.

DAG: Directed Acyclic Graph.

FN: False Negative.

FP: False Positive.

FS: Feature Selection.

GBN: General Bayesian Network.

JMI Joint Mutual Information.

JPD: Joint Probability Distribution.

K-L: Kullback and Leibler.

LPE: Localised Partial Evaluation.

MACIE: MAtrix Controlled Inference Engine.

MB: Markov Blanket.

MI: Mutual Information.

MIM: Mutual Information Measure classifier.

ML: Machine Learning.

MWST: Maximum Weight Spanning Tree.

NB: Naive Bayes classifier.

SCN: Singly Connected Network.

SMIM: Selective variant of MIM classifier.

SNB: Selective Naive Bayes.

TAN: Tree Augmented Naive Bayes.

TANC: Credal TAN.

TN: True Negative.

TP: True Positive.

UCI: University of California Irvine.

APPENDIX F – Publications

Intelligent Data Analysis 9(2) 2005
IOS Press

A New Singly Connected Network Classifier Based on Mutual Information

Clifford S. Thomas[1], Catherine A. Howie, Leslie S. Smith
Department of Computing Science and Mathematics, University of Stirling, Stirling, FK9 4LA, Scotland

Received 15 March 2004

Revised 21 May 2004

Accepted 16 June 2004

Abstract. For reasoning under uncertainty the Bayesian Network has become the representation of choice. However, except where models are considered 'simple' the tasks of construction and inference are provably NP hard. For modelling larger real-world problems this computational complexity has been addressed by methods that approximate the model. The Naive Bayes (NB) Classifier which has strong assumptions of independence among features is a common approach whilst the class of trees another less extreme example. The aim of this paper is to investigate the use of an information theory based technique as a mechanism for inference in Singly Connected Networks (SCN) or 'polytrees'. We call this variant a Mutual Information Measure (MIM) Classifier. We experimentally evaluate this new approach and compare the resulting classification performance of the MIM Classifier against (a) a Naive Bayes Classifier, (b) a General Bayesian Network (GBN) Classifier and (c) a Singly Connected Network, using benchmark problems taken from the UCI repository. With respect to (a) we show that the MIM Classifier generally performs better than the NB Classifier. For (b) and (c) we show that the MIM Classifier is comparable with both the GBN and SCN Classifiers and in most data sets used performs marginally better.

1. Introduction

The process of learning a Bayesian Belief Network (BBN) is defined by two activities: learning the graphical structure, and then learning the parameters for the structure [43]. Characteristic of a BBN and key to defining its representation, in respect of the domain it models, is the determination of edge directionality of the graph. Where possible a domain expert can specify the node/vertices ordering, that is, the domain knowledge used to specify a causal order of nodes or variables of the domain. However, where expertise is scarce, finding a node ordering by alternative means that will represent a useful BBN can be a difficult task.

Whilst it is possible to find a BBN for any given ordering (as the Joint Probability Distribution can be written by successive applications of the chain rule) it is clearly not practical to search among all possible orderings of nodes. Moreover, if we choose a poor order we get a more complicated network. As the topology changes more tenuous relationships can occur, which may in turn require unnatural and problematic probability judgements.

The dependence BBNs have on node ordering has led to researchers actively developing algorithms to efficiently determine edge directionality. One approach uses a search and scoring method to find the correct directions of the edges [20,34] but this was found to be slow as the search space can be large if prior node ordering is not supplied.

[1] Corresponding author: Clifford S Thomas, Tel.: + 44 (0) 131 343 4827; Fax: + 44 (0) 1786 464551; E-mail: cst@cs.stir.ac.uk

Another more common approach uses Conditional Independence (CI) tests and have been used in many edge orientation algorithms [45,51-53]. These methods are generally exponential in complexity. Singh [49] proposed a variant on the CI tests generating a "good" node ordering from data. Although offering an improvement in complexity, it was noted that the quality of the recovered network structure was very sensitive to the node ordering determined by their algorithm. Further strategies can be found in [1,9,37,42].

Singly Connected Networks (SCN or polytrees) are a restricted class of networks that can efficiently be solved in time linear in the number of nodes. However, despite this reduction in complexity the task of finding edge directionality on a skeleton tree structure, thus completing the polytree, is still as complicated [7] to resolve. Polytree recovery techniques have been proposed [7,14,45] based on CI tests, but in some situations full recovery was not always possible, leaving some edges undirected. When directionality was fully recovered it was found that even with a small number of parents, a node's conditional probability table still required an unrealistic number of values to complete its description.

The aim of this paper is to investigate the use of an information theory based technique for constructing a BBN and as a mechanism for subsequent inference. Our objective is to avoid the issues of model complexity and overfitting, together with the dependence upon prior node ordering, by taking advantage of the existing tree structuring algorithms. Our concept builds on the efficient SCN or polytree construction as proposed by Pearl [43], using the orientation of the tree edges, with respect to the class node, as a heuristic for assigning edge directionality. We call this SCN variant a Mutual Information Measure (MIM) Classifier as it corresponds to the restricted class of trees built from mutual information. Once constructed we experimentally evaluate this new approach and compare the resulting classification performance against three well known classifiers. These are a Naive Bayes classifier, a general Bayesian network and an implementation of a polytree [45].

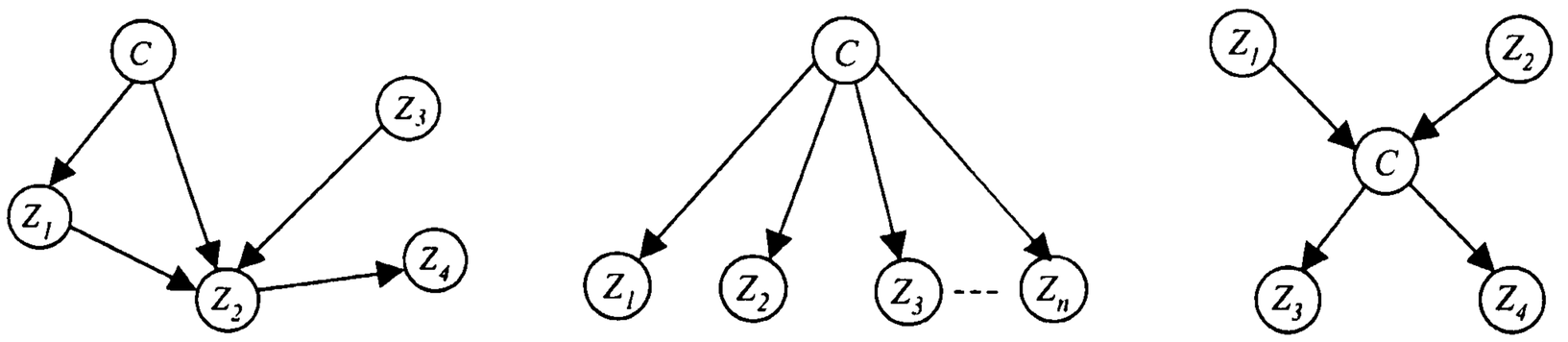
The remainder of this paper is organised as follows. In the next section we review the background of modelling BBNs and in particular BBNs in the form of 'trees'. In § 3 we consider the use of mutual information as a mechanism for inference in SCNs. In § 4 we review the experimental work with § 5 discussing the results. Finally, in § 6 we summarise our work and consider some possible improvements.

2. Bayesian Belief Networks as Classifiers

Probabilistic graphical models or Bayesian Belief Networks [43] offer a unified qualitative and quantitative framework for representing and reasoning with probabilities and independencies. One advantage is their comprehensibility. Due to their attractive features they are often used in real-world applications [3,18,21,39]. In a Bayesian Belief Network, Fig. 1a, vertices represent propositional variables in a domain, and edges between vertices represent the dependency relationships among the variables. By taking advantage of the independencies existing between subsets of variables in the domain, they model the joint densities that limit the problems of dimensionality, namely parameter space.

Modelling a Bayesian Belief Network consists of determining the qualitative graph structure G and the quantitative parameter θ. The qualitative network structure $G(N, A)$ is a directed acyclic graph (DAG). Each of the vertices $n \in N$ represents a domain variable, and each edge $a \in A$ between vertices represents a probabilistic dependency [43].

Edges in the Bayesian Network represent the dependencies among the variables $Z = \{Z_1, \ldots Z_n\}$ with the parents of Z_i, $pa(Z_i)$ the direct predecessors of Z_i in *G*. An absence of edges indicates that there is conditional independence. The qualitative parameter θ consists of the joint probability distribution $P(Z_1, \ldots Z_n)$.



(a) General Bayesian Network (b) Naive Bayes Network (c) Singly Connected Network

Fig. 1. Network Examples.

This is the general product and can be written:

$P(Z_1, \ldots Z_n) = \prod_{i=1}^{n} P(Z_i \mid pa(Z_i))$ where $pa(Z_i)$ is designated as the parent of Z_i.

If the network is built in collaboration with domain experts, the determination of the structure is often a relatively easy task, since this task usually fits well with knowledge that for example, medical experts often have about causal relationships between variables. In an automated approach a data set can be utilised but this task in general is considered to be difficult [11,13]. For the quantitative part (that is quantifying the conditional probability tables in the network) this aspect is often considered by medical experts as a much harder or even impossible task. The reason is that medical domain experts themselves often have no idea about these probabilities. When available a domain data set can provide estimates of the probabilities more readily than the experts. The reader is directed to [5,8,20,23,24,26,27,42] for further details of Bayesian Belief Networks.

Learning a Belief network structure from data is hard in general [23]. However, there are classes of structures for which both learning the structures and the parameters can be done efficiently. This is the class of 'Trees' [10,22,43].

A Tree, like any graphical model, has the ability to express the dependencies between variables separately from the detailed forms of these dependencies, contained in the parameters. In doing so it provides a property that offers excellent support for human intuition and allows for the design of inference and learning algorithms.

Trees are simple models: this is especially evident when one examines the algorithms that fit a tree to a given distribution. All the information about the target distribution that a tree can capture is contained within a small number, at most $(n-1)$, pair-wise marginals. Simplicity leads to computational efficiency where efficient inference is a direct consequence of the fact that trees are decomposable models with small clique size. Trees have a probability distribution that can be mapped perfectly as both a Bayes net and Markov net, where a Markov net is defined by a structure, which is an undirected graph with an arbitrary topology.

Tree structures require that exactly one variable be considered as a cause of another given variable. This restriction simplifies computations, but its representational power is rather limited, since it forces us to form a single vertex from all causes sharing a common consequence. For example, when a Doctor discovers evidence in favour of one disease, it reduces the likelihood of other diseases that could explain the patient's symptoms.

A Bayesian network (where a vertex may have multiple parents) in which no more than one undirected path existing between any two parents vertices, is called a causal 'polytree' or Singly Connected Network, Fig. 1c. Polytrees represent much richer dependency models than trees, as they support products of higher-order distributions. Moreover, they can be identified by a Maximum Weight Spanning Tree (MWST) algorithm, as used by Chow and Liu [10], to find the structure and thus only require second-order statistics to establish the branch weights.

Chow & Liu's algorithm [10] takes a probability distribution P as its input and constructs a Bayesian network in the form of a tree as its output. This is achieved in only $O(N^2)$ pair-wise dependency calculations with each calculation using only second-order statistics, where N is the number of nodes.

The procedure can be summarised as follows.

1. Compute the Mutual Information $I(Z_i, Z_j)$ between each pair of variables $i \neq j$.

$$I(Z_i, Z_j) = \sum_{Z_i, Z_j} \hat{P}_D(Z_i, Z_j) \log\left(\frac{\hat{P}_D(Z_i, Z_j)}{\hat{P}_D(Z_i)\hat{P}_D(Z_j)}\right) \quad (1)$$

where $Z = \{Z_1, \ldots Z_n\}$ feature set of discrete variables and $\hat{P}_D$ the measure defined by the frequencies of events in the data D.

2. Build a complete undirected graph in which the vertices are the variables in Z. Annotate the weight of an edge connecting Z_i to Z_j by $I(Z_i, Z_j)$.

3. Build a *maximum weighted spanning tree* of the graph [12, 43].

The algorithm starts with a graph without any edges and uses a *search* method to add on edges to the graph. Once found a *scoring* method is used to see if the new structure is better than the old one. If it is, the newly added edge is retained and the algorithm continues by trying to add another one. This is essentially repeated until no further new structure is better than the previous one. In the case of the Chow and Liu algorithm, the Kullback-Liebler [32,33] (K-L) cross entropy is used as the measure of best *score*.

Of special interest is the use of the Bayesian network as a classifier where the focus of interest is in predictions about a special target variable, the class variable. The classification process involves a class variable C that can take on values $C_1, \ldots C_m$, and a feature vector Z of n features that can take on a tuple of values denoted by $\{Z_1, \ldots Z_n\}$. Given a case Z represented by an instantiation $\{Z_1, \ldots Z_n\}$ of feature values, the classification task is to determine the class value C_i for Z. For simplicity, we restrict our discussion to domains with only discrete variables.

The performance of the network is measured on some set of test cases in terms of the classification accuracy, that is, the percentage of test cases for which it predicts the class correctly. Classification is very important in most data analysis tasks and has been widely studied in the Machine Learning community. Pearl [43] reviews techniques for BBN inference.

The most straightforward and widely tested classifier is the Naive Bayes Classifier. As Fig. 1b shows, the network structure is static which means there is no need to perform any structural learning. Essentially, this classifier assumes that the attributes are conditionally independent given the class variable. The technical details are not essential for this paper and the reader is directed to [17] for further information.

Despite the controversial assumption of independence this classifier has however, outperformed many state of the art classifiers [15,25,35]. Further analysis of the Naive Bayes Classifier can be found in [6,18, 35, 44, 46, 47].

3. The Mutual Information Measure Classifier

Simplified models, such as Singly Connected Networks (SCN), have been shown to represent good approaches to automatic classifier construction [7] alleviating the time consuming processes of learning and propagation compared to that required for GBNs. Despite their loss of representation capabilities SCNs gain in efficiency and simplicity as they can be built from data using only pair-wise marginals. That is, simplification is achieved by selecting a topology that allows efficient propagation, for example a SCN or 'polytree' [7,14,45].

In this section we propose a propagation technique based upon the well-known mutual information between two random variables. This is an extension to Pearl's polytree construction utilising the Chow and Liu tree building algorithm. We consider the mutual information or 'branch weight' as a measure of strength for an edge linking multi-state vertices and further demonstrate that this branch weight representation can be used to classify locally new evidence presented to the SCN. The concept of information 'weight' has been researched and used in many other approaches [4,16,28] together with applications that have utilised the mutual information measure [2,29,41,54].

Prior to propagation we first construct a SCN based upon an information theory based technique. We achieve this by building the Mutual Information Measure (MIM) classifier structure in two stages. In the first, we use the Chow and Liu tree building algorithm to build the skeleton structure. Once constructed, stage two transforms the structure to a singly connected network. In stage two we determine the node ordering from the orientation of the tree edges with respect to the class node.

The formal algorithm for constructing the Maximum Weight Spanning Tree (MWST) can be described by the pseudo code shown in the Fig. 2.

The algorithm results in $n(n-1)/2$ pairs of $I(Z_i, Z_j)$ being generated with the algorithm terminating when $(n-1)$ branches have been selected, at which point the dependency tree has been constructed. Essentially, by looking at the association of all variables in terms of couples, an $(n-1)$ undirected branched tree can be constructed, where n is the number of variables. The procedure for learning the MIM Classifier can be further summarised as detailed in Fig. 3.

Consider the structure in Fig. 4, which represents a subset of a MIM classifier structure as generated from the procedure detailed in Fig. 3.

In this example the class vertex C has $C_1,\ldots C_m$ multi-state values and the attribute Z_1 $\{Z_{1-1}, Z_{1-2}\}$ multi-state values. The mutual information measure $I(C, Z_1)$ as defined by equation (1) is a measure of the dependence between the two variables C and Z_1. The value $I(C, Z_1)$ represents the summation of the 'individual' mutual information that is associated with each pair of class-attribute state values, that make up the overall 'branch weight' $I(C, Z_1)$. We call these 'individual' values mutual information elements and denote them by $I'(\ )$. The plots in Fig. 5, using the UCI 'DNA' database, illustrate the distribution of calculated $I'(\ )$ values with respect to the three class labels describing the primate splice-junction gene sequences (DNA). For each class $C = \{C_1,\ldots C_3\}$ the

distribution of $I'(\ )$ values reveals that there is a characterisation 'profile', which is distinctly different for each class label.

During the process of classification we 'introduce' evidence in the form of a feature vector $\{Z_1,\ldots Z_n\}$ for $n=60$ attribute instantiations. To propagate this information or evidence in our SCN we update the 'branch weight' elements $I'(\ )$ in respect of each class $C=\{C_1,\ldots C_m\}$ where $m=3$.

3.1. Classification - MIM Classifier

Let T_m represent a MWST for a tree dependent probability distribution P_t, where P_t is a Markov field relative to a tree [43]. If a feature vector $Z=\{Z_1,\ldots Z_n\}$ describes a new observation of the domain then the probability distribution P_t will be updated to P_t'.

FOR i=1 to N-1 **DO**
BEGIN
FOR $j=i+1$ to N **DO**
BEGIN
Find all second-order probability distributions $P(Z_i,Z_j)$

- (A) From the given (observed) distribution $P(Z)$, compute the joint distribution $P(Z_i,Z_j)$ for all variable pairs

Calculate mutual information measures $I(Z_i,Z_j)$

- (B) Using the pair-wise distributions (A), compute all $n(n-1)/2$ branch weights and order them by magnitude.

END
END
Branches No=0
WHILE (Branches No < (N-1))

- (comment) Repeat (C) until n-1 branches have been selected.

BEGIN
Select two variables Z_i,Z_j that have largest $I(Z_i,Z_j)$

- (C) Assign the largest two branches to the tree to be constructed.

Add the branch (Z_i,Z_j) to the tree
IF (there is a loop in the tree)
Delete the branch (Z_i,Z_j)
ELSE
Branches No = Branches No + 1

END IF

- (comment) now Examine the next-largest branch, and add it to this tree.

END
Note: for those branches having equal weight, the first largest branch found will be selected to define the structure of the MWST.

Fig. 2. Maximum Weight Spanning Tree Algorithm.

1. Input Training Data of the domain $\{C,Z_1,\ldots,Z_n\}$.
2. Build the undirected tree structure using the Chow and Liu algorithm.
3. Select the domain 'Class' variable as the root of the undirected graph.
4. Transform the graph from (3) into a directed tree (SCN) by setting the direction of all edges to be outwards from the class vertex.
5. Output SCN (MIM Classifier structure) $G(N,A)$.

Fig. 3. MIM Classifier Learning Procedure.

For Z belonging to a particular class value C_i, where $i = (1,2,\ldots,m)$, the new MWST for $P_t^{'}$ can be represented by $T_{m_{Ci}}$. If we repeat this for each possible value of C_i then m MWSTs will be constructed.

In order to assign a feature vector Z to a classification value of class C_i, we need only find the maximum $T_{m_{Ci}}$ from the m MWSTs. As was shown by Chow and Liu this is equivalent to finding the maximum total branch weight for T_m, thus minimizing the K-L measure. In other words we are calculating the relative difference between each $P_t^{'}$ in respect of the m probability distributions represented by $T_{m_{Ci}}$ for $i = (1,2,\ldots,m)$. Identifying the specific class value i to which the new observation Z belongs, requires finding the MWST ($T_{m_{Ci}}$) that has the greatest total branch weight. The winning class value $i = (1,2,\ldots,m)$ will thus identify one of the mutually exclusive classes C_i that corresponds to the maximum $T_{m_{Ci}}$.

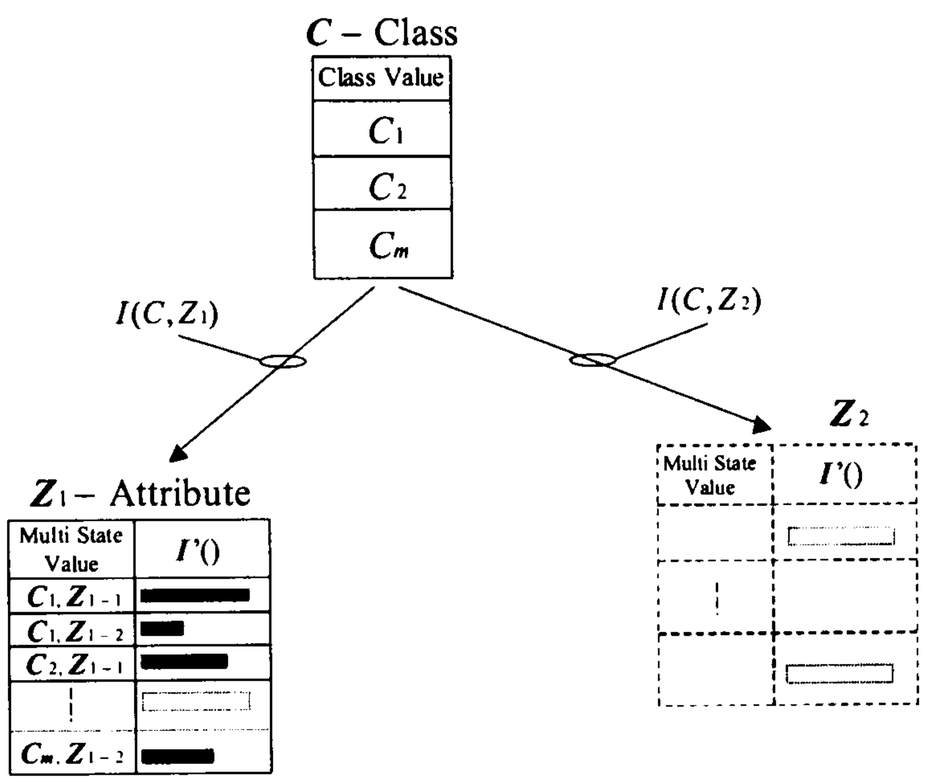
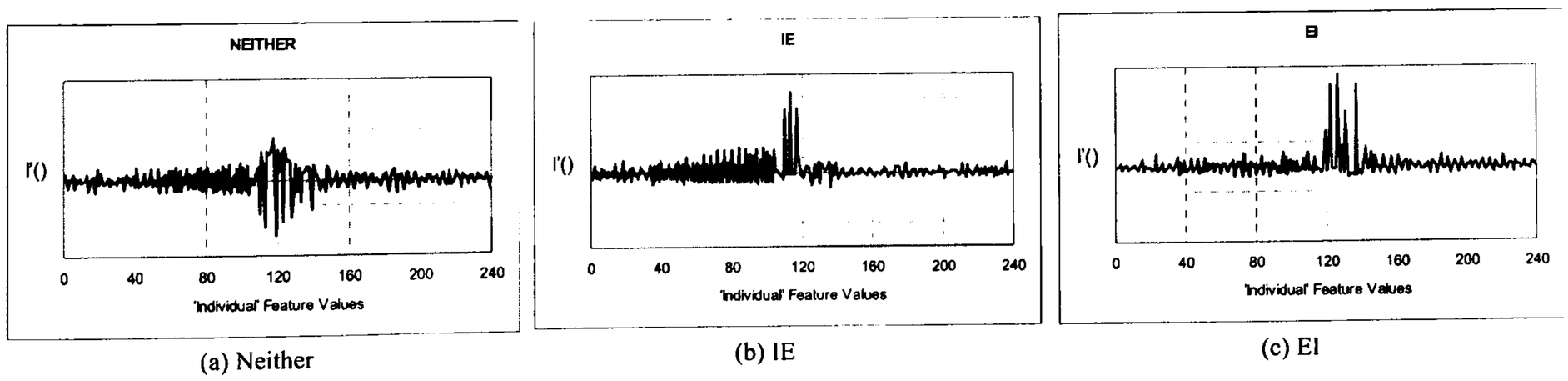


Fig. 4. MIM Classifier Tree Structure (subset).



(a) Neither (b) IE (c) EI

Fig. 5. DNA Class Distributions of $I^{'}(\,)$ values for each of the three Class-attribute measures of dependence. Where (a) represents the class 'Neither', (b) the class 'Intron/Exon' boundary or donors and (c) the class 'Exon/Intron boundary or acceptors. Each of the 60 attributes has four states (A,C,G,T) resulting in a total of 240 'individual' feature values. We omit values from the $I^{'}(\,)$ axis as only the 'profile' characterising each class is relevant.

For the representation of Fig. 4, considering the subset only, the joint probability distribution $P(C,Z_1,Z_2)$ can be written as $P(C)P(Z_1 \mid C)P(Z_2 \mid C)$ by the chain rule. If we ignore the edges $C-Z_1$ and $C-Z_2$ then the three vertices can be considered independent giving:

$$P'(C,Z_1,Z_2) = P(C)P(Z_1)P(Z_2)$$

and from Theorem 1.

$$I(C,Z_1,Z_2) = \sum_{C,Z_1,Z_2} P(C,Z_1,Z_2)\log\frac{P(C,Z_1,Z_2)}{P'(C,Z_1,Z_2)} = I(C,Z_1)+I(C,Z_2)$$

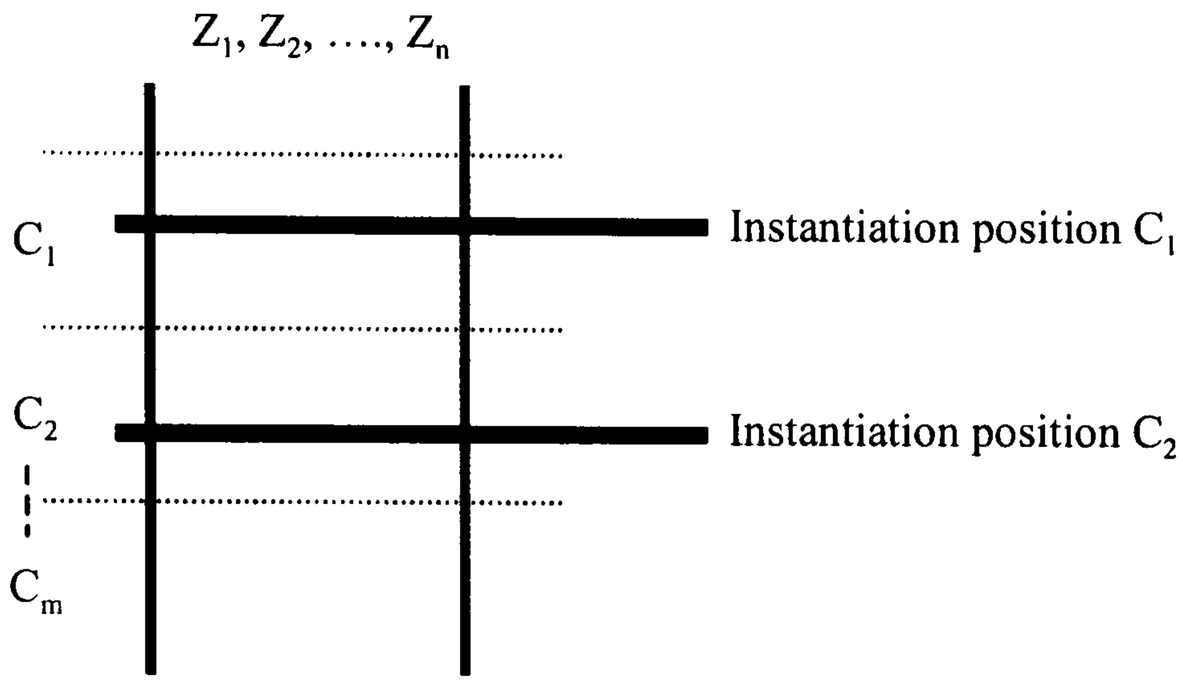


Fig. 6. Domain Data set Representation example.

This is the sum of the individual branch mutual information measure values between pairs of neighbours. More generally in terms of mutual information measure 'branch weights' $W_t(Z_0,\ldots,Z_2) = W_t(Z_0,Z_1)+W_t(Z_1,Z_2)$. So if we sum over all these branch weights $W_t(\ )$ we will have a combined measure of their affect. Since mutual information is symmetric then $C \rightarrow Z_1$ is the same as $C \leftarrow Z_1$, so that directionality in our representation will not alter the value of the branch weight $I(C,Z_1)$.

If we now consider a representation of the domain data set as depicted in Fig. 6 we can see that an instantiation of an evidence vector $Z = \{Z_1,\ldots Z_n\}$ can be classified as belonging to one of the mutually exclusive class labels $C = \{C_1,\ldots C_m\}$ by local computation.

In Fig. 6, the training sample of the domain can be viewed as a series of class partitions characterising samples belonging to a particular class. Each partition is described by a vector of class attributes $Z = \{Z_1,\ldots Z_n\}$ and this will be the case for each class label where $C = \{C_1,\ldots C_m\}$. The actual dimensions of the partitions may or may not be the same for each class and will correspond to the specifics of the domain data set.

An instantiation of an evidence vector $Z\{\ \}$ in position C_1 will increase the marginal $P(C_1)$ and update the joint probabilities $P(C_1,Z)$ in respect of the evidence vector $Z = \{Z_1,\ldots Z_n\}$ and their values. Similarly, for an instantiation in positions $C_2,\ldots,C_m$. Since the evidence vector $Z\{\ \}$ will be common for all $C\{\ \}$ instantiation positions, the marginal probabilities $P(Z)$ for each value of Z, due to the evidence, will remain at a constant but updated value.

In terms of our MIM structure this implies that any changes of information, branch weights, due to observing an evidence vector $Z\{\ \}$, will only be measurable on edges that are directly associated with the class vertex. The corresponding information on edges not associated with the class will remain at a constant value, for each instantiation position $\{C_1,\ldots C_m\}$. Thus the classification can be achieved locally using a subset of the domain features as defined by the SCN. In situations where the structure corresponds to that of Naive Bayes, the feature size will be n. However, unlike Naive Bayes the same extreme assumption of conditional independence for all features given the class is not made.

4. Experimental Work

For our experiments we used ten benchmark problems taken from the UCI repository [40]. The data sets selected are summarised in Table 1.

Table 1
Data sets used in the experiments

Database Name	Attribute size	Class size	Sample size	Train size	Test size
Vehicle #	18	4	846	Cv5	-
DNA	60	3	3186	2000	1186
Car Eval	6	4	1728	Cv5	-
Flare	10	3	1066	Cv5	-
Chess	36	2	3196	2130	1066
Vote*	16	2	435	Cv5	-
Mushroom *	22	2	8124	5416	2708
Letter	16	26	20000	14000	6000
Hepatitis *#	19	2	155	Cv5	-
Nursery	8	5	12960	8640	4320

Key: * Indicates data sets with 'missing' attribute value. # Indicates continuous valued attributes. Cv5 indicates 5 fold Cross Validation.

As we are using the Chow and Liu algorithm we have restricted our investigations to discrete data sets. All continuous features were therefore made discrete prior to application. This was achieved by use of the utility provided by MLC++ [31] on the default setting.

Residual evaluations do not provide an indication of how well the classifier will perform when required to make a prediction for data it has not already observed. This problem can be avoided by not using the entire data set when constructing each classifier structure. For the larger data sets we randomly partitioned each data set into two parts. The first part comprised 2/3 of the entire sample and was used for training/construction of the four classifier structures. The second part, the remaining 1/3, was subsequently used for evaluating the predictive accuracy of each of the classifiers constructed (this represents the simplest kind of cross validation or hold-out technique). In each case a stratified distribution was maintained in respect of the two partitions. For those data sets that were small, the hold-out technique was not applied but a 5-fold cross validation as indicated in Table 1. Our choice of folds (k = 5) is based on the recommendations of Kohavi [30]. Data sets that contained 'missing' feature values were dealt with by treating them as an additional element of that feature.

For each of the four classifiers, the structure was learned/constructed using the training data set and the classifier accuracy determined on the test data set. The classification accuracy was determined as a percentage of the test cases that identified the correct class.

This process was repeated over a series of runs in order to gain a sample average together with the standard deviation for the predictive accuracy using the test partition. The statistical significance of the differences in classification accuracy was measured using an Analysis of variance followed by Post Hoc Tukey comparisons with overall confidence level 95%.

For the purposes of this investigation we used *PowerConstructor* [9] to both learn and test the GBN classifier. In the case of the 'polytree' classifier we implemented a version based on the Rebane & Pearl [45] model as described by the pseudo code detailed in Fig. 2 with subsequent directionality discovery detailed in Fig. 7.

In the event that branches remained undirected after applying this algorithm we applied two 'rules' in order to allow the conditional probability tables to be calculated. The first rule was taken from Verma & Pearl [53] whilst the second was a heuristic derived from the partially completed polytrees. Directionality was essentially assigned to the undirected edges in conjunction to those edges that had already been successfully recovered.

The two rules applied were as follows:

FOR i=1 to N **DO**
  BEGIN
   IF Z_i has more than one neighbour
   THEN put Z_i in Multiple_Set
END
FOR each Z_i in Muliple_Set
BEGIN
 FOR any pair of neighbours Z_j, Z_k of Z_i **DO**
 BEGIN
 IF (Z_j and Z_k are independent)
 THEN $2I$ is distributed as χ^2 with (r-1)(c-1) degrees of freedom
    (Where I is the mutual information measure)
    $Z_j \rightarrow Z_i, Z_k \rightarrow Z_i$
 ELSE
    $Z_i \rightarrow Z_j, Z_i \rightarrow Z_k$
 END
END

Fig. 7. 'Polytree' Construction Algorithm for Directionality Discovery.

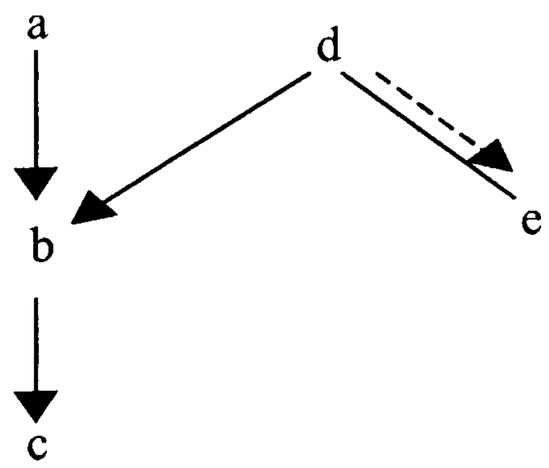


Fig. 8. Rule 2 - Directionality.

Rule 1: If $a \rightarrow b$ and a is not adjacent to c then direct $b \rightarrow c$
Rule 2: If $a \rightarrow b \rightarrow c$ and $d \rightarrow b$ then direct $d \rightarrow e$ that is as shown in Fig. 8.

5. Results and Discussion

The average predictive accuracies, taken over 25 runs, of the classifiers generated for each of the four methods are shown in Table 2. Each entry describes the average accuracy along with the sample standard deviation illustrating variations in the predictive accuracy from sample to sample.

Where the GBN model was provided with the correct node ordering the algorithm carried out $O(N^2)$ CI tests to learn an N-node network. This same time complexity applies to the Chow and Liu algorithm (calculating the $N(N-1)/2$ mutual information values) which is utilised for learning both our Polytree implementation and the MIM Classifier. In contrast the NB classifier's time complexity is $O(N)$, being proportional to the time required to read all of the training data. When the GBN model was not provided with correct node ordering, the time complexity increased to $O(N^4)$.

Table 2
Average Predictive Accuracy

DB Name	MIM	NB	GBN	Polytree	Default (overall)
Vehicle	55.66 ± 1.51	58.28 ± 1.79	**61.0 ± 2.02**	56.85 ± 1.77	25.8
DNA	***95.58 ± 0.42***	94.97 ± 0.29	89.90 ± 5.61	**95.62 ± 0.35**	51.9
Car Eval	86.11 ± 0.74	**86.58 ± 1.78**	86.11 ± 1.46	78.81 ± 8.25	70.023
Flare	**82.93 ± 1.26**	80.99 ± 1.28	82.27 ± 1.45	82.66 ± 0.79	79.2
Chess	**96.27 ± 3.56**	87.34 ± 1.02	94.65 ± 0.69	90.14 ± 1.86	52
Vote	**95.40 ± 2.41**	89.89 ± 5.29	95.17 ± 1.89	94.94 ± 3.69	54.8
Mushroom	98.56 ± 1.06	95.79 ± 0.39	**99.30 ± 0.16**	98.56 ± 1.06	51.8
Letter	**80.26 ± 0.37**	74.96 ± 1.10	75.02 ± 0.61	79.86 ± 0.80	4.07
Hepatitis	**84.00 ± 7.22**	81.20 ± 3.70	83.22 ± 1.52	82.47 ± 1.44	79.35
Nursery	**95.78 ± 0.30**	94.76 ± 0.45	89.72 ± 0.46	94.85 ± 0.27	33.3

Key: **MIM** – Mutual Information Measure Classifier, **NB** – Naive Bayes Classifier, **GBN** – General Bayesian Network Classifier, **Polytree** – Pearl's SCN Model.
Values in **bold** type indicate the highest model performance achieved by the classifier in respect of each database. ***Bold italic*** values highlight performance levels that are close to the highest level achieved.

Here the algorithm had the additional overhead of examining N^2 node pairs in order to determine the network edge orientations. In this paper both options were explored and the results in Table 2 reflect the best predictive values achieved for the GBN.

Of the four algorithms the GBN was noticeably slower to learn the network structure than the tree based algorithms, even when correct node ordering was provided, with relative times corresponding linearly with training sample sizes.

In respect of classification, testing was linear in the representation size of the structures, that is, in the number of attributes defining the class Markov blanket.

The plots shown in Fig. 9 and Fig. 10 represent the results relative to the MIM Classifier and Polytree Classifier respectively. Each bar shows the average difference in predictive accuracy. A positive value for an algorithm indicates that the MIM or Polytree

performed better on the data set under consideration. The error bars represent the Post Hoc Tukey comparisons with overall 95% confidence for the relative differences.

The MIM Classifier performed better than NB on eight of the ten databases used. In respect of the databases 'Chess', 'Mushroom', 'Letter' and 'Nursery' the MIM classifier has significantly higher performance accuracy when compared to NB. For 'Car_eval', although the NB did perform better it was not statistically significant with NB achieving on average a 0.5% better predictive level than MIM. In the case of the GBN there were two databases in which it performed better than the MIM Classifier with only 'Vehicle' being statistically significant, however, in general the overall predictive performance of the MIM Classifier aligned with that achieved by the GBN. The MIM Classifier performed better on seven of the ten databases against the GBN with three of these statistically significant; 'Nursery', 'Letter' and 'DNA'. This result was similar in respect of the polytree however none were found to be statistically significant, with the MIM Classifier performing better on seven of the ten database.

For the databases 'Nursery' and 'Car_eval' there was an indication that the features of these two databases are almost independent of each other. Since the NB Classifier makes the assumption that the features are independent, given the class variable, it was no surprise to see it perform better for these two particular domains. In modelling these databases the MIM, GBN and polytree classifiers effectively reduced to a NB structural representation.

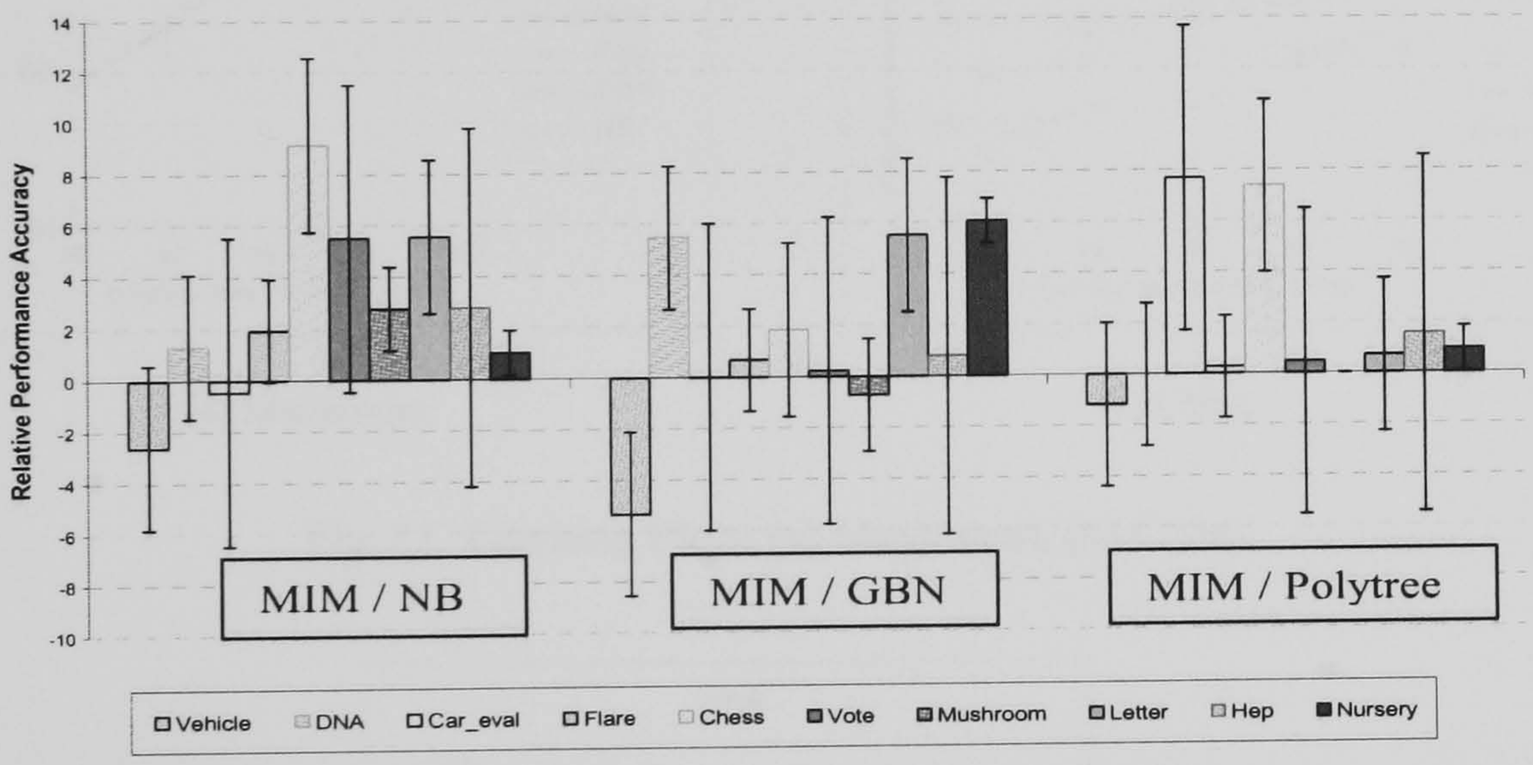


Fig. 9. Predictive Accuracy relative to MIM Classifier.

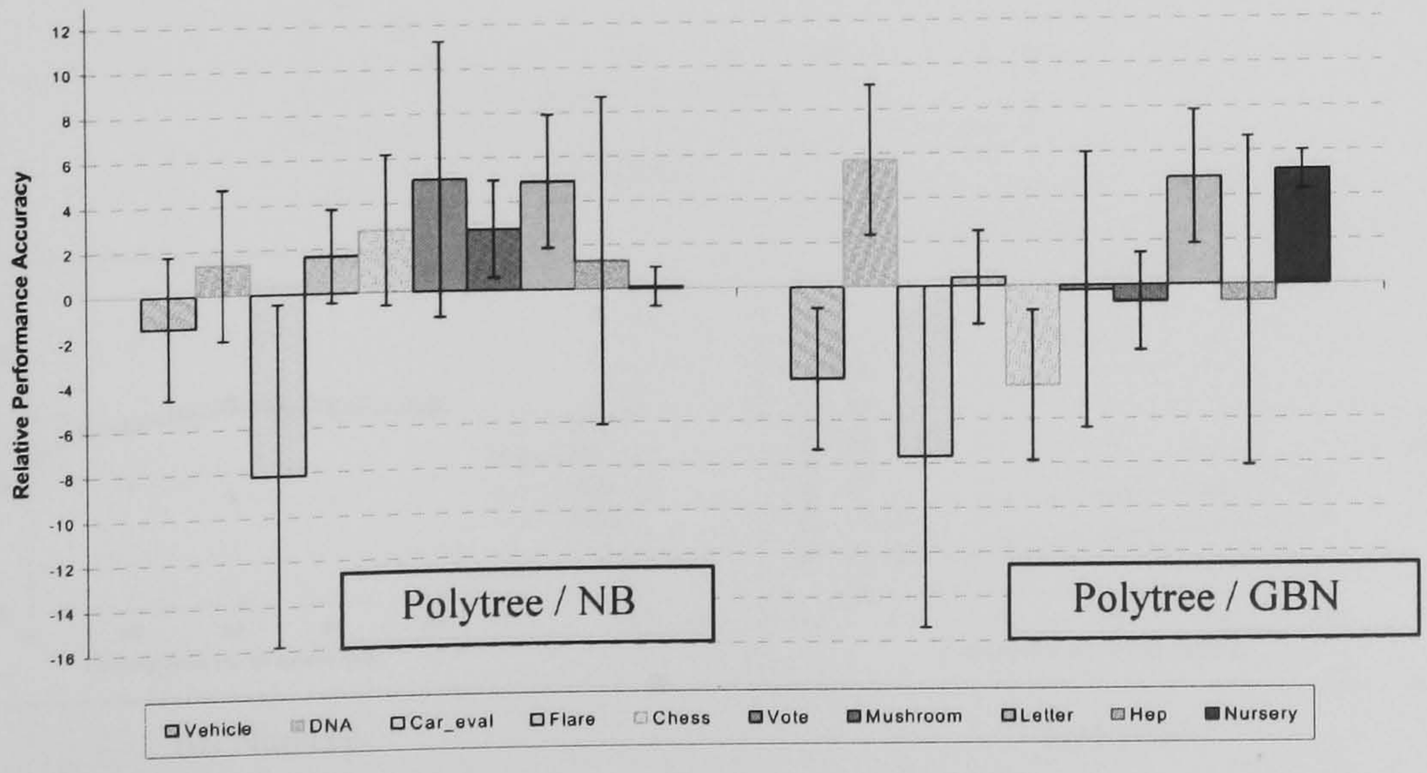
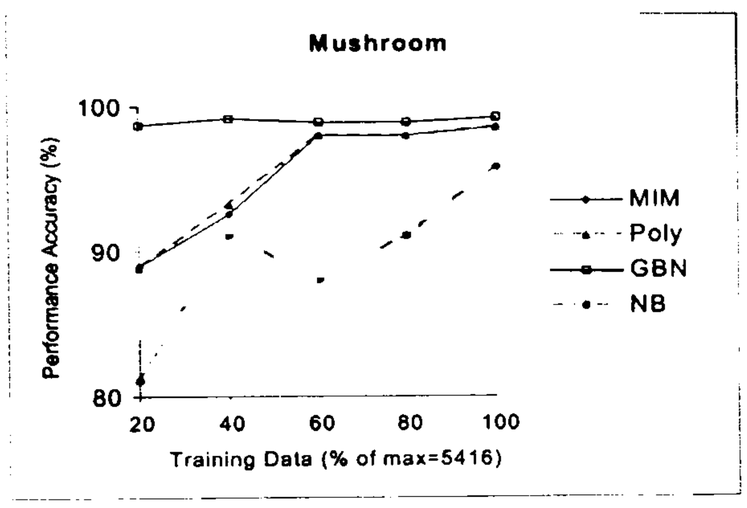


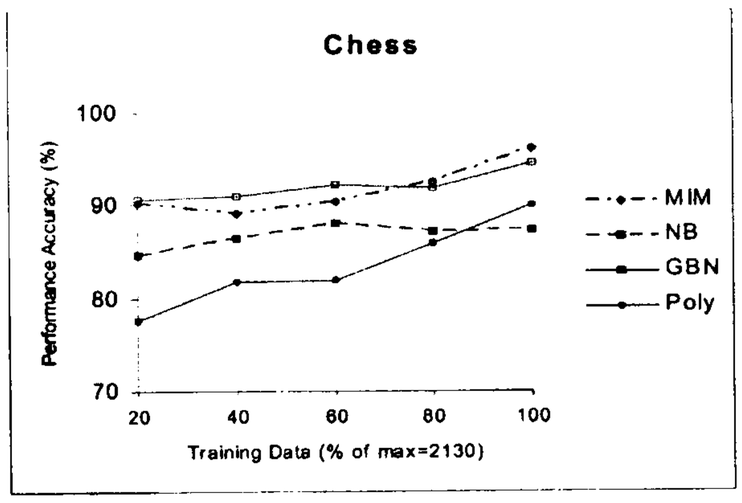
Fig. 10. Predictive Accuracy relative to 'Polytree' Classifier.

These latter however, do not make this normally invalid assumption of conditional independence. For 'Nursery', both the MIM and polytree classifiers aligned closely with the level achieved by the NB Classifier, outperforming the GBN by 5%-6% in accuracy (averaged). Neither of the 'tree' classifiers seemed to have been affected from the loss of representation even when reduced to a NB type structure. This was not as apparent for the 'Car_eval' database in respect of the polytree classifier, possibly due to its final topology as determined by the node ordering algorithm.

In general the methods that did not assume extreme conditional independence performed better than the NB classifier as shown in Table 2. This demonstrates that the additional modelling power of these methods does actually have an impact on performance. For the majority of the databases the MIM Classifier was aligned in terms of performance with the polytree. This was expected since structurally they are very similar with the only difference being attributed to node ordering defining the topology. In some cases the node ordering for the polytree modified the topology sufficiently to reduce its performance but this was not seen to be significant. Despite the reduction in complexity the 'tree' methods not assuming extreme conditional independence performed comparably with that achieved by the GBN on the selected databases.

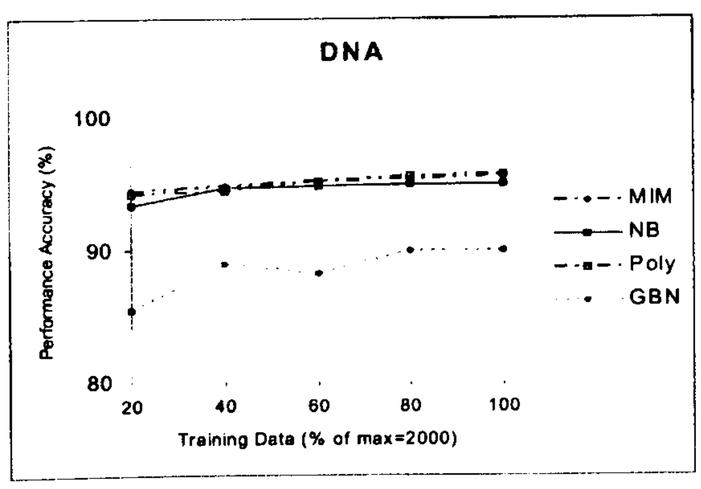


(a) Mushroom

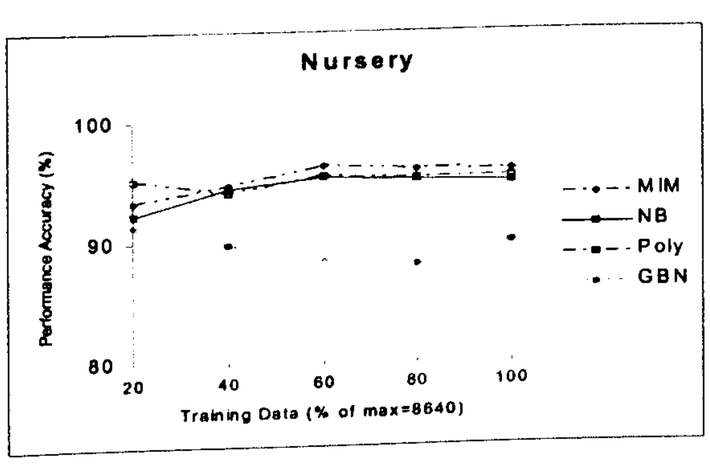


(b) Chess

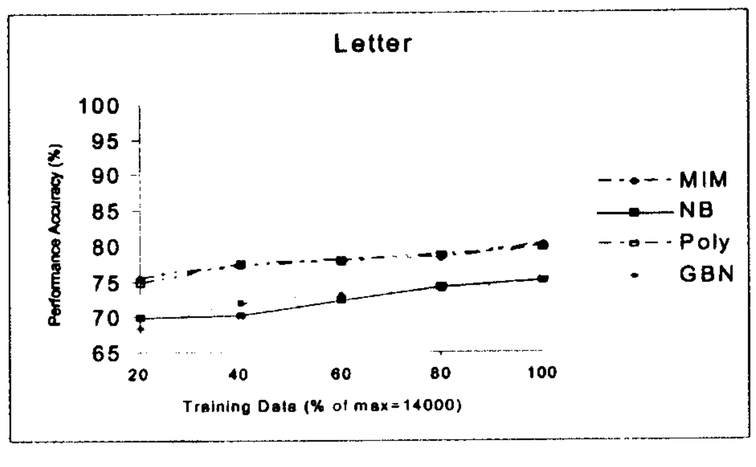
Fig. 11. Learning Plots: (a) Mushroom, (b) Chess.



(c) DNA



(d) Nursery



(e) Letter

Fig. 12. Learning Plots (c) DNA, (d) Nursery and (e) Letter.

Figures 11 and 12 illustrate the learning curves for the databases: 'Mushroom', 'Chess', 'DNA', 'Nursery' and 'Letter', based on an average of 20 runs. Only the larger databases, as shown in Table 1, were investigated, that is, those using a hold-out approach, as the smaller databases were evaluated by a 5-fold cross-validation approach.

Langley [35] showed that the NB classification performance was poor for databases 'Mushroom' and 'Chess' but good for 'DNA'. In our investigations this was also evident. The MIM and polytree classifiers performed comparably for the 'Mushroom' database and stabilised at the 60% of the sample size both statistically significant. The GBN in contrast was stable throughout the sample sizes performing slightly better and was also statistically significant. We observed that the 'Mushroom' database did not require many features to classify the majority of the test samples and it was evident that both the MIM and polytree methods did not have sufficient class – attributes in their model representations.

For the 'Chess' database both the MIM and the GBN methods improved performance as the sample size increased. However, the polytree, although structurally similar to the MIM model performed poorly compared with the MIM which was statistically significant. This may indicate a poor topology and corresponding bad choice of branch directionality, as determined by the node ordering algorithm, for a more complex structure necessary to model the chess database.

In the case of the 'DNA' database NB performs well, outperforming the GBN. Both the MIM and polytree methods were aligned in performance remaining stable throughout all sample sizes and were statistically significant along with NB. This database structure has a strong class – attribute representation which clearly favours the NB classifier. As the GBN was reduced to this NB representation a degraded performance was observed. This was not the case for the MIM or polytree methods. As the 'Nursery' database also characterised strong feature independence, it was not unexpected that the resulting plot was similar to that of the 'DNA' database. However, unlike 'DNA' only MIM and polytree at 60% sample size were statistically significant and not NB.

In respect of the 'Letter' database all four methods achieved stable performance levels at 60% of the sample size. The GBN and NB classifiers aligned at a lower level of performance accuracy than both the MIM and polytree classifiers, with the latter maintaining similar profiles in terms of performance accuracy and statistically significant.

In general where the database had strong feature independence characteristics the NB and MIM Classifier performed well. Since the structure of the GBN was forced to reduce to that of a NB structure the GBN had a slight degradation in performance. For domain modelling where the class – attribute were sparse, the MIM model required more samples to learn a model, as was seen in the 'Mushroom' database, than the GBN.

As the structure for the polytree is (the skeleton) exactly the same as that of the MIM method, its topology will only change as a consequence of node ordering applied. For most of the databases the MIM and polytree classifiers performed equally well, however on the occasions where the topology changed due to the polytrees' dependence on node ordering, performance degradation was observed. This was particularly apparent for the 'Chess' and 'Car_eval' databases, however there was no statistical significance found for any of the ten databases.

6. Conclusion

6.1. Summary of Contributions

This paper introduced a MIM approach to inference in Singly Connected Networks. The main contribution of this paper lies in showing the feasibility, advantages and effectiveness of this approach. In the main part of our investigation we compared the MIM Classifier with two other 'tree' modelling approaches, namely the Naive Bayes and a 'polytree' as defined by Pearl & Rebane [45], along with a general Bayesian network approach. The MIM classifier was statistically significantly better than NB on four of the ten databases used. For 'Car_eval', although the NB did perform better this was not statistically significant at less than 0.5% performance improvement. In respect of the GBN there were two databases in which it performed better than the MIM Classifier, however, in general the overall performance of the MIM Classifier aligned with that achieved by the GBN. The MIM Classifier performed better on seven of the ten databases with three of these improvements statistically significant. This result was similar in respect of the 'polytree', with the MIM Classifier performing better on seven of the ten databases.

The proposed use of mutual information measure 'branch weights' as a mechanism for classifying new unseen evidence has been demonstrated as feasible. The approach taken provides for both an efficient and localised method of inference in singly connected networks with comparable performance levels of less restricted methods.

By modelling the domain using efficient 'tree' structuring algorithms we have avoided the issues of complexity and overfitting prone to networks. Moreover, the utilisation of Chow and Liu's algorithm allows for tree construction to be achieved using only pair-wise marginals, and although a 'restricted' model, has not required us to make extreme conditional independence assumptions.

Our experimental results on the selected databases have demonstrated that the MIM Classifier's performance was not affected by our node ordering approach and did not show any dependence or consequences of making a bad choice as observed in the polytree representation. In addition for databases that was known to have strong feature independence properties, the reduction of the structure to that of a 'NB' representation appeared not to degrade the performance of the MIM classifier as it did for the GBN.

6.2. Future Work

Despite our encouraging experimental results, we believe there are still ways in which we might further improve the performance of the MIM Classifier. 'Branch weights', as described in this paper, allow us to focus on the most relevant nodes, given a specific query, namely a class node. However, work carried out by L de Campos [7] concerning SCN, suggests that the removal of weak links might actually improve performance. In our current representation, the MIM model is defined by the configuration determined by the Chow and Liu algorithm for constructing the tree. However, this phase of construction alone does not provide a mechanism to effectively modify the class - attribute relationship in order to better model the domain. Feature selection has been shown to be a promising area of research [50] and this is an aspect we intend to investigate.

One of the complexity issues common to BBNs relates to the number of parents a node has associated with it. In the UCI databases selected this was not an issue, however, this would not be the case for a real world problem domain which may exhibit properties such

as numerous parents. To investigate this and the effect on other methods such as the Naive Bayes, GBN and polytree, we intend to apply the MIM Classifier to the medical domain of Acute Abdominal Pain. This is well known to be a difficult domain to model due to many of the diagnostic categories having similar characterisations.

References

[1] S. Acid, L.M de Campos, J.F Huete, The search of causal orderings: A short cut for learning belief networks, Lecture notes in computer science, vol 2143, Springer-verlag, 2001, pp. 216-227.
[2] R. Battiti, Using Mutual Information for selecting Features in Supervised neural net learning, IEEE Transactions on Neural Networks 15, (1994), 537-550.
[3] R. Blanco, P. Larrañaga, I. Inza, B. Sierra, Selection of highly accurate genes for cancer classification by estimation of distribution algorithms, in: European Conference on Artificial Intelligence in Medicine (AIME'01): working notes for workshop, 2001, pp. 29-34,
[4] B. Boerlage, Link strengths in Bayesian networks, Master's Thesis, Department of Computer Science, University of British Columbia, 1995.
[5] W. Buntine, A guide to the literature on learning probabilistic networks from data, IEEE Transactions on Knowledge Data Engineering 8, (1996), 195-210.
[6] J. Cao, R. Panetta, S. Yue, A. Steyaert, M. Young-Bellido, S. Ahmad, A naïve Bayes model to predict coupling between seven transmembrane domain receptors and G-proteins, Bioinformatics 19 (2), Oxford University Press, 2003, pp. 234-240.
[7] L.M de Campos, Independency Relationships and Learning Algorithms for Singly Connected Networks, Technical Report DECSAI-96-02-04, Department of Computer Science, University of Granada, 1996.
[8] J. Cheng, R. Greiner, Learning Bayesian belief network classifiers: Algorithms and system, in: Proceedings. 14th Biennial Conference of the Canadian Society for Computational Studies of Intelligence, Ottawa, ON, 2001.
[9] J. Cheng, PowerConstructor System, 1998. *http://www.cs.ualberta.ca/~jcheng/bnpc.htm*.
[10] C.K Chow, C.N Liu, Approximating Discrete Probability Distributions with Dependence Trees, IEEE Transactions On information Theory, Vol. IT-14, (1968), 462-467.
[11] G. Cooper, The Computational Complexity of Probabilistic inference using Belief networks, Artificial Intelligence 42, 1990, pp. 393-405.
[12] P. Dagum, M. Luby, Approximating probabilistic inference using Bayesian belief networks is NP-hard, Artificial Intelligence, (1993), 141-153.
[13] T.H Cormen, C.E Leiserson, R.L Rivert, Introduction to Algorithms, MIT Press, 1990.
[14] S. Dasgupta, Learning Polytrees, in: Proceedings of the UAI'99, 1999.
[15] P. Domingos, M. Pazzani, On the optimality of simple Bayesian classifier under zero-one loss, Machine Learning 29, (1997), 103-130.
[16] D. Draper, Localized partial evaluation of belief networks, PhD Thesis, Department of Computer Science, University of Washington, 1995.
[17] R. Duda, P. Hart, Pattern Classification and Scene Analysis, John Wiley & Sons, 1973.
[18] K. Ezawa, S. Nonton, Knowledge discovery in telecommunication services data using Bayesian network models, in: Proc. 1st International Conference on Knowledge Discovery and Data Mining, 1995.
[19] J.T.A.S Ferreira, D.G.T Denison, D.J Hand, weighted naïve Bayes modelling for data mining, Technical Report, Imperial College, Department of Mathematics, 2001.
[20] N. Friedman, M. Goldszmidt, Learning Bayesian networks with local structure, in: Proceedings of the 12th International Conference on Uncertainty in Artificial Intelligence, 1996.
[21] R. Fung, B.D Favero, Applying Bayesian networks to information retrieval, Communications of the ACM 38 (3), (1995).
[22] D. Geiger, An entropy-based learning algorithm of Bayesian conditional trees, in: UAI'92, 1992, pp. 92-97.
[23] D. Heckerman, D. Geiger, D.M Chickering, Learning Bayesian networks: the combination of knowledge and statistical data, Machine Learning 20 (3), (1995), 197-243.
[24] D. Heckerman, A tutorial on learning with Bayesian networks, in: M. I. Jordan (Ed): Learning in Graphical models, Dordrecht, Netherlands, Kluwer, 1998.
[25] J. Hellerstein, J. Thatcher, I. Rish, Recognizing end-user transaction in performance management, in: Proceedings of AAAI-2000, Austin, Texas, 2000, pp. 596-602.
[26] F.V Jensen, An introduction to Bayesian Networks, UCL Press Ltd, London, ISBN 1-85728-332-5, 1996.
[27] F.V Jensen, Bayesian Networks and Decision Graphs, New York, Springer Verlag, 2001.

[28] N. Jitnah, A. Nicholson, treenets: A framework for anytime evaluation of belief networks, in: 1st International Joint Conference on Qualitative and Quantitative Practical reasoning (ECSQARU-FAPR'97), Lecture notes in Artificial Intelligence, Springer-Verlag, 1997.
[29] G.D Kleiter, R. Jiroušek, Learning Bayesian Networks under the control of mutual information, in: Proceedings of 6th International Conference IPMU-1996, 1996.
[30] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimates and model selection, in: Proceedings 14th International Joint Conference on Artificial Intelligence, Morgan Kaufmann, 1995, pp. 1137-1143.
[31] R. Kohavi, G. John, R. Long, D. Manley, K. Pfleger, MLC++: A machine learning library in C++, in: Proceedings of 6th International conference on Tools with Artificial Intelligence, IEEE Computer Society, 1994.
[32] S. Kullback, R.A Leibler, On information and sufficiency, Annals of Statistics 22, (1951), 79-86.
[33] S. Kullback, Information Theory and Statistics, Dover Publications, New York, 1968.
[34] W. Lam, F. Bacchus, Learning Bayesian belief networks: An approach based on the MDL principle, Computational Intelligence 10 (4), (1994).
[35] P. Langley, W. Iba, K. Thompson, An analysis of Bayesian classifiers, in: AAAI'90, 1992, pp. 223-228.
[36] P. Langley, S. Sage, Tractable average-case analysis of naïve Bayesian classifiers, in: Proceedings 16th International Conference on Machine Learning, Morgan Kaufmann, 1999, pp. 220-228.
[37] P. Larrañaga, C.M.H Kuijpers, R.H Murga, Y. Yurrassendi, Learning Bayesian network structures by searching for the best ordering with genetic algorithms, IEEE Transactions on Systems, Man and Cybernetics 26 (4), 1996, pp. 487-493.
[38] P. Langley, S. Sage, Induction of Selective Bayesian classifiers, in: Proceedings Conference on Uncertainty in AI, Morgan Kaufmann, 1991.
[39] K. McNaught, S. Clifford, M. Vaughn, A. Fogg, M. Foy, A Bayesian Belief network for lower back pain diagnosis, in: European Conference on Artificial Intelligence in Medicine (AIME'01): working notes for workshop, 2001, pp. 53-58.
[40] P.M Murphy, D.W. Aha, UCI repository of Machine Learning databases, 1995. *http://www.ics.edu/$\sim$mlean/MLRepository.html*.
[41] A.E Nicholson, N. Jitnah, Using mutual Information to determine relevance in Bayesian networks, Technical Report 971, Department of Computer Science, Monash University, 1997.
[42] H.P Pan, Learning Bayesian networks: II – a computational algorithm, in: 5th International Conference on Information Fusion, (submitted), Anapolis, Maryland, USA, 2002.
[43] J Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann Publishers, INC., San Mateo, Californian. ISBN 0-934613-73-7, 1988.
[44] T. Pedersen, Naïve Bayes as a satisficing model, Working notes of the AAAI Spring Symposium on satisficing models, Palo Alto, CA, 1998, pp. 60-67.
[45] G. Rebane, J. Pearl, The recovery of causal polytree from statistical data, in: Proceedings of 3rd conference On Uncertainty in Artificial Intelligence. Seattle, WA, 1987.
[46] I. Rish, An empirical study of the Naive Bayes classifier, in: 17th International Joint Conference on Artificial Intelligence, Seattle, Washington, 2001.
[47] I. Rish, An analysis of data characteristics that affect Naive Bayes performance, Technical Report RC21993, IBM T. J. Watson Research Center, 2001.
[48] C.E Shannon, W. Weaver, The mathematical theory of communication, University of Illinois Press, 1949.
[49] M. Singh, M. Vatonta, An Algorithm for the construction of Bayesian network structures from data, in: D. Heckerman, E. Mamdani, eds, Uncertainty in Artificial Intelligence: Proceedings of the ninth Conference, 1993, pp. 259-265, San Mateo, CA. Morgan Kaufmann.
[50] M. Singh, G. Provan, Efficient Learning of Selective Bayesian network Classifiers, in: International Conference on Machine Learning, 1996, pp. 453-461.
[51] P. Spirtes, C. Glymour, R. Scheines, An algorithm for fast recovery of sparse causal graphs, Social Science Computer review 9, 1991, pp. 62-72.
[52] P. Spirtes, C. Glymour, R. Scheines, Causality from probability, in: Proceedings of Advanced Computing for social sciences, Williamsburgh, VA, 1990.
[53] T. Verma, J. Pearl, An algorithm for deciding if a set of observed independencies has a causal explanation, in: Proceedings of the 8th Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, 1992, pp. 323-330.
[54] Z.R Yang, M. Zwolinski, Mutual Information Theory for Adaptive Mixture Models, IEEE Transactions On Pattern Analysis and Machine Intelligence 23 (4), (2001), 396-403.

Statistics in Medicine submitted 2004

A Case Study in Diagnosis: Classifying Acute Abdominal Pain without Assuming Extreme Conditional Independence

Clifford S. Thomas[1], Catherine A. Howie, Leslie S. Smith
Department of Computing Science and Mathematics, University of Stirling, Stirling, FK9 4LA, Scotland

SUMMARY

Acute Abdominal Pain (AAP) is the commonest surgical emergency in Europe, with conditions such as Appendicitis requiring urgent surgical treatment. In order to address the computational complexity of larger real-world problems methods that approximate the reality have been adopted. The Naive Bayes (NB) Classifier which has strong assumptions of independence among features is a common approach, whilst the class of trees another less extreme example. The aim of this paper is to investigate the optimality claim of Naive Bayes, for classifying in the medical domain of AAP, against three models, which do not assume extreme conditional independence. These are, a general Bayesian network, an implementation of a Singly Connected Network or 'polytree', and a new model called a Mutual Information Measure (MIM) Classifier because it corresponds to the restricted class of trees built from an information theory based technique. We experimentally evaluate this claim and compare the resulting classification performance of the Naive Bayes Classifier (NB) against these three Bayesian Belief Network (BBN) approaches using two datasets taken from the medical domain of Acute Abdominal Pain. Despite the loss of some representation capabilities we demonstrate that the MIM classifier can be effectively applied to the domain of AAP and that it achieves a comparable predictive performance to that of the NB classifier. We further show that this 'tree' representation of the BBN not only achieves a favourable 'overall' predictive value compared to NB, but provides a recognisable qualitative structure without violating 'real' world assertions.

Some Key words: Acute Abdominal Pain; Bayesian Networks; Naive Bayes; Mutual Information Measure; Singly Connected Networks.

1. INTRODUCTION

Acute Abdominal Pain (AAP) is the commonest surgical emergency in Europe and in most other parts of the World [1]. Although some causes of AAP don't require admittance to hospital other conditions such as appendicitis require urgent surgical treatment. An inflamed appendix may perforate raising the risk of death and with one in every sixteen people expected to suffer from it at some point in life [2] it is thus a relatively important disease group to identify. Clearly, early and accurate diagnosis is essential, but few doctors and even fewer patients realise just how difficult such early diagnosis can be. The domain of AAP is well known to be both difficult and challenging [3] with the diagnosis of appendicitis complicated by other diagnoses like Non-Specific Abdominal Pain (NSAP) which often presents similar signs and symptoms.

Tackling real world problems in complex domains such as AAP has resulted in the development of more and more decision analytic models. Extracting knowledge from experts however, is arising as a major obstacle in model building. Adopting automated / semi-automated techniques, deriving the model directly from the data, can overcome

[1] Corresponding author: Clifford S Thomas, Tel.: + 44 (0) 131 343 4827; Fax: + 44 (0) 1786 464551; E-mail: cst@cs.stir.ac.uk

some of these obstacles. In fact [4] AAP is one of the most widely studied applications of computer aided diagnostics [5. 6.7. 8.9.10.11].

According to Provan & Clarke [12] probabilistic reasoning is crucial for diagnosing AAP, as the uncertainties involved cannot be adequately captured given that two patients with the same symptoms may have different diseases. Examples of approaches taking up this challenging domain can be found in [13.14.15.16.17]. However, despite these models attempts to capture the domain dependencies, the empirical evidence in support of diagnostic accuracy and the capturing of dependencies in Bayesian models is inconclusive.

During comparisons made by Todd & Stamper [4] of an 'expert' built GBN and the Naive Bayes the results suggested that there were no significant improvements in accuracy by taking interactions into account. Work carried out by researchers [18.19.4] even suggested from their results that Naive Bayes was probably optimal. The research that followed de Dombel's et al [20] successful application of Naive Bayes led to many approaches, which attempted to avoid making this violation of conditional independence. Here the classifier assumes that the attributes are conditionally independent given the class variable (each attribute has only the class node as a parent). One such example was the G&T system [21] that applies Bayes rule strictly. However, this too found Naive Bayes to outperform their dependency model. Ohman [11] even compared Naive Bayes to more complex representations such as rule-based systems and found here too that there was no major 'overall' difference. Further support for Naive Bayes success and its performance in respect of AAP can be found in [22.23.24.25].

The aim of this paper is to investigate the optimality claim of Naive Bayes against three models, which do not assume extreme conditional independence. Namely, a general Bayesian network, an implementation of a Singly Connected Network (SCN) or 'polytree' [26] as proposed by Pearl [27], and a new model called a Mutual Information Measure (MIM) Classifier [28] as it corresponds to the restricted class of trees built from an information theory based technique.

In the following section we review the four classifier methods that will be used for this study, and further summarise their representations together with the classification task employed. The remainder of this paper is organised as follows. In Section 3 we describe the data sets used and how we deal with anomalies. In Section 4 we review the experimental work with Section 5 discussing the results. Finally, in Section 6 we summarise our work and consider some possible improvements.

2. BACKGROUND - MODELS USED IN THE STUDY

2.1. General Bayesian Networks (GBN)

A Bayesian Belief Network consists of a qualitative network structure G and the quantitative parameter θ over the network structure. The qualitative network structure $G(N, A)$ is a directed acyclic graph (DAG). Each of the vertices $n \in N$ represents a domain variable, and each edge $a \in A$ between vertices represents a probabilistic dependency [27].

Edges in the Bayesian Network represent the dependencies among the variables $Z = \{Z_1, \ldots Z_n\}$ with the parents of Z_i, $pa(Z_i)$ the direct predecessors of Z_i in G. An absence of edges indicates that there is conditional independence. The qualitative parameter θ consists of the joint probability distribution $P(Z_1, \ldots Z_n)$. This is the general product rule and can be written:

$P(Z_1, \ldots Z_n) = \prod_{i=1}^{n} P(Z_i \mid pa(Z_i))$ where $pa(Z_i)$ is the parent of Z_i.

The reader is directed to [29.30.31.32.33.34.35.36] for further details of Bayesian Belief Networks and [37.38.39.40] for various real-world applications.

The classification process involves a class variable C that can take on values $C_1, \ldots C_m$, and a feature vector Z of n features that can take on a tuple of values denoted by $\{Z_1, \ldots Z_n\}$. Given a case Z represented by an instantiation $\{Z_1, \ldots Z_n\}$ of feature values, the classification task is to determine the class value C_i to which Z belongs. In general inference is NP-hard [41]. Reviews of techniques concerning GBN inference can be found in Pearl [27], specific details are beyond the scope of this paper. Learning methods and performance of GBN for classifiers are studied in Friedman [42] and Cheng [43].

2.2. Naive Bayes Network (NB)

The simplest form of classifier is the Naive Bayesian Classifier [44.45]. This classifier assumes that the attributes are conditionally independent given the class variable, that is, each attribute has only the class vertex as a parent, Fig. 1.

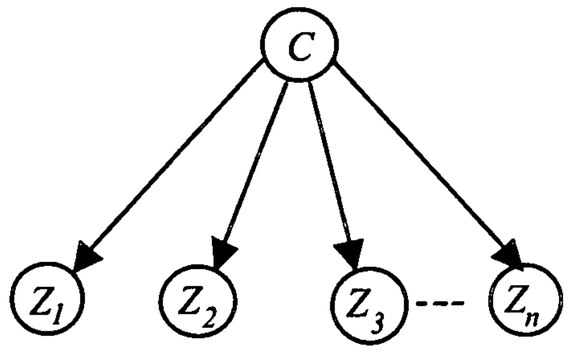


Fig. 1. A Naive Bayesian Network Example

The joint distribution is given by: $P(C, Z_1, Z_2, \ldots, Z_n) = P(C)\prod_{i=1}^{n} P(Z_i \mid C)$ where C is the class variable and $Z_1, \ldots, Z_n$ are the other domain variables.

In this case inference is straightforward. To perform this task, we assume that we have the prior probabilities, $P(c_i)$, for each value c_i of the class variable. Further, we assume that we have the conditional probability distribution for each feature value z_j given the class value c_i, $P(z_j \mid c_i)$.

Using Bayes' rule, a new case, $Z = \Lambda_j z_j$ (Λ denotes conjunction), can then be classified as: $P(c_i \mid Z) = \frac{P(c_i)P(Z \mid c_i)}{P(Z)} = \frac{P(c_i)P(\Lambda_j z_j \mid c_i)}{\sum_k P(\Lambda_j z_j \mid c_k)P(c_k)}$

2.3. Singly Connected Networks or 'polytrees'

A Bayesian network where a vertex may have multiple parents, and which is a singly connected (that is, no more than one undirected path exists between any two parents vertices), is called a causal 'polytree' or Singly Connected Network, Fig. 2.

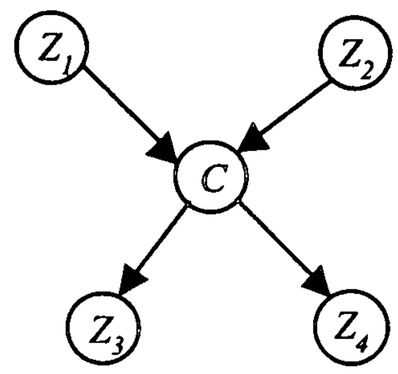


Fig. 2. 'Polytree' Example

'Polytrees' represent much richer dependency models than trees, as they support products of higher-order distributions. Moreover, they can be identified by a Maximum Weight Spanning Tree (MWST) algorithm, as described in procedure 1, to find the structure and thus only require second-order statistics to establish the branch weights.

Procedure 1 (Restricted network) [46]

The procedure of Chow and Liu can be summarised as follows.

1. Compute the Mutual Information $I(Z_i, Z_j) = \sum_{Z_i, Z_j} \hat{P}_D(Z_i, Z_j) \log\left(\frac{\hat{P}_D(Z_i, Z_j)}{\hat{P}_D(Z_i)\hat{P}_D(Z_j)}\right)$ between each pair of variables $i \neq j$.
2. Build a complete undirected graph in which the vertices are the variables in Z. Annotate the weight of an edge connecting Z_i to Z_j by $I(Z_i, Z_j)$.
3. Build a maximum weighted spanning tree of the graph [47.27].

Here $Z = \{Z_1, \ldots Z_n\}$ is the feature set of discrete variables and is $\hat{P}_D$ the measure defined by the frequencies of events in the data D.

Once a MWST has been constructed, edge directionality can be assigned thus completing the SCN. Procedure 2 describes the process we employed to determine directionality.

Procedure 2 (SCN or 'Polytree') [26]

1. Use Procedure 1 to build a MWST.
2. Determine edge directionality for the skeleton tree structure using either rules proposed in Verma & Pearl [48] or Pearl's algorithm [27] summarised in Fig. 3.

FOR i=1 to N **DO**
BEGIN
IF Z_i has more than one neighbour
THEN put Z_i in Multiple_Set
END
FOR each Z_i in Muliple_Set
BEGIN
FOR any pair of neighbours Z_j, Z_k of Z_i **DO**
BEGIN
IF (Z_j and Z_k are independent)
THEN $2I$ is distributed as χ^2 with (r-1)(c-1) degrees of freedom
(Where I is the mutual information measure)
$Z_j \rightarrow Z_i, Z_k \rightarrow Z_i$
ELSE
$Z_i \rightarrow Z_j, Z_i \rightarrow Z_k$
END
END

Fig. 3. 'Polytree' Edge Directionality Discovery Algorithm.

Where I is the mutual information measure (Procedure 1-1) and N is the number of vertices. In this method the class variable is not distinguished as a separate variable from the other features.
Any branches that remain undirected after applying the algorithm are completed by use of a domain expert.
Propagation techniques in 'polytrees' are defined in Pearl [27].

A variant of the SCN is a new classifier called the MIM Classifier as it corresponds to the restricted class of trees built from an information theory based technique. This also uses the efficient Chow and Liu 'tree' structuring algorithm to construct a skeleton structure. Procedure 3 describes the construction technique and inference mechanism.

Procedure 3 (MIM Classifier) [28]

1. Apply Procedure 1 to construct a MWST skeleton tree structure.
2. Transform into a directed tree by selecting the class node as the root variable and setting the direction of the remaining arcs from the class node.

Once the structure has been fully constructed the task of classification is performed as follows.

3. Let T_m represent the MWST for a tree dependent probability distribution P_t. If a feature vector $Z = \{Z_1, \ldots Z_n\}$ describes a new observation of the domain then the dependent probability distribution P_t will be updated to P_t'. For Z belonging to a particular class value C_i, where $i = (1, 2, \ldots, m)$, the new MWST for P_t' will be represented by $T_{m_{Ci}}$. Repeating for each possible value of C_i results in m MWSTs.

4. Classification of class C_i can be determined by finding a maximum $T_{m_{Ci}}$ for the case Z represented by an instantiation $\{Z_1, \ldots Z_n\}$ given class C_i for $i = (1, 2, \ldots, m)$.

3. DESCRIPTION OF THE DATA SETS (AAP) USED

Two data sets were used in this study, defined in Table 1. The first consists of 9867 patient records comprising 33 attributes, covering 135 features, and a class variable having 9 possible values or diseases. The data was originally collected and maintained by Mr AA Gunn [49] at Bangour General Hospital and is currently retained by staff at St John's Hospital in Edinburgh.[1] The resulting database addresses the domain of Acute Abdominal Pain (AAP) recording information gathered both during the examination and subsequent audit administration. The structure is based upon a patient's examination on arrival to the Accident and Emergency (A&E) department. Each completed record stores the doctor's 'initial' diagnosis and the 'actual' diagnosis group a patient was subsequently determined as really belonging to, on their discharge from hospital. The full contents of the database far exceed our requirements and mainly provide information necessary for hospital audits. The precise format relevant to our study is defined in Appendix A.

The second data set comprises of 5373 case samples again describing examination records of patients suffering from acute abdominal pain. In this case however, the data has been collected at a different geographical location, namely Leeds.[2] This data was gathered over a period of 30 years concerning the diagnosis of AAP and is currently retained at the Professorial Surgical Unit and Accident and Emergency Department, at the General Infirmary. Both data sets have been standardised by collaboration between the two hospitals under the direction of Professor Tim de Dombal. We will label the first data set 'CADA' and the second one 'LEEDS' in order to distinguish between them.

Table 1: AAP Data sets used in the experiments

Dbase Name	Attribute size	Class size	Sample size	Train size	Test size
CADA	33	9	9867	6959	2908
LEEDS	33	9	5373	-	5373

4. EXPERIMENTAL WORK

4.1. AAP Database

With the exception of one of the features namely 'AGE', which is strictly a continuous variable, all of the other 32 features represent discrete variables. In this data set the doctors themselves have provided the discretisation for the feature 'AGE' based upon

[1] CADA (Computer Assisted Diagnostic and Audit) data base considered the largest database of AAP in Europe. Courtesy of St John's Hospital, Livingston, Edinburgh.
[2] Courtesy of General Infirmary, Leeds, UK.

their own judgements. The group Non-Specific Abdominal Pain (NSAP) is not actually a diagnostic group rather a 'catch all' category into which the doctors assign a patient whom they cannot fit into one of the other 'true' eight diagnostic groups. In a sense this can be considered as a 'don't know' category, but only in respect to the 'true' eight known categories. For this study we employed the hold-out approach partitioning the data base into a 'learn' and test sample set, as defined in Table 1. The training partition was approximately 2/3 of the database whilst the test partition the remaining 1/3 of the sample set. Both partitions are the result of performing a 'random' but stratified split, in order to compensate for the imbalances in respect of the nine class value distributions.

On examination of the database, records were found to have multiple or composite parameter values stored in respect of some of the symptoms and in other cases none of the symptom parameters were recorded (missing). To deal with these two anomalies we have introduced two additional parameter values, which are appended to each symptom. For example Symptom 21 : MOOD will be described by parameters : normal (21/1), distressed (21/2), anxious (21/3) plus composite[3] (88) and missing (99). This approach ensures that the Naive Bayes model does not have an advantage over the Bayesian models (as complete data sets are required) with the AAP data set effectively standardised for all models under study.

4.2. Methodology

For each of the four classifiers, the structure was learned/constructed using the 2/3 training partition and each classifier's accuracy determined on the 1/3 test partition. The main performance measure used was the classification accuracy of a model on the test data, the classification accuracy being the percentage of test cases that were diagnosed correctly. This process was repeated over a series of runs in order to obtain a sample average together with the standard deviation for the predictive accuracy using the test partition. The statistical significance of the differences in classification accuracy was measured using a Analysis of Variance (ANOVA) followed by Post Hoc Tukey comparisons with overall confidence level 95%.

In the domain of AAP, where there are numerous class values, the comparison of the four methods does not provide an accurate measure using only the classification accuracy. To address this we have computed additional statistics, which are generally used for comparing 'alternative' tests with respect to medical diagnosis [50]. In this paper we utilise this approach to make comparisons of our 'alternative' classifiers and thus access their ability to effectively discriminate between the individual class values or diseases. Assuming the positive/negative value for a disease to represent its presence/absence, the different statistics we computed can be described as follows [17].

Sensitivity: This is the ability of a classifier to correctly predict the presence of a disease in a patient with that disease. Also know as the True Positive Rate, it is defined as: $\frac{TP}{TP + FN}$ where TP is the number of true positives while FN represents the number of false negatives.

---

[3] Some composites have been 'grouped' and added to symptoms as new parameter values. Where the frequency of occurrence for combinations was below a set threshold (arbitrarily set) these were assigned to the default composite value '88'

Specificity: This is the ability of a model to correctly identify patients that do not have a given disease. Thus, it is the proportion of people who do not have a given disease, and correctly predicted so by the classifier. As such: $\frac{TN}{TN+FP}$ where TN represents the number of true negatives and FP represent the number of false positives.

Likelihood Ratio: This measures the ability of a classifier to discriminate between alternative diseases. The higher the value, the greater is the discriminating ability of the method. It is defined as follows: $\frac{TP(FP+TN)}{FP(TP+FN)} = \frac{sensitivity}{1-specificity}$

Predictive Value: This measures the accuracy of a classifier on a given disease, and is the probability that a patient actually has the certain disease, given that the classifier has so predicted. It is defined as: $\frac{TP}{TP+FP}$

In addition, we also computed the discriminant matrices, for each method, describing the performance of each technique with respect to the individual diseases. This provides a mechanism for us to compare different approaches with respect to their ability to correctly identify the individual class values (diseases).

In this investigation, the 'CADA' database was used to both construct and test the four methods, whilst the 'LEEDS' data set was only used for testing the four methods. This latter data set represents a truly 'external' sample set as its data distribution, thus its characteristics, do not have an influence on the classifier's structure as it has been independently gathered from the 'CADA' data set.

4.3. Experimental Design

The objective of the study is to address three hypotheses.

1. For AAP does the Naive Bayes classifier really perform better than the Bayesian network.
2. Is Naive Bayes (as considered by other researchers) really optimal for AAP or is it just good at identifying NSAP.
3. Does the Bayesian network approach offer more than the Naive Bayes irrespective of its overall accuracy performance.

For the purposes of this investigation we used *PowerConstructor* [51] to both learn and test the GBN classifier. In the case of the 'polytree' classifier we implemented a version based on the Rebane & Pearl [26] approach. The specifics of the Naive Bayes and MIM Classifiers used for this study are as described in Section 2.

5. RESULTS AND DISCUSSION

The average predictive accuracies, taken over 10 runs, of the classifiers generated for each of the four methods are shown in Table 2 and Table 3. Each entry describes the average accuracy along with the sample standard deviation illustrating variations in the

predictive accuracy from sample to sample. For completeness the doctor's predictions are also included[4].

Table 2: Average Predictive Accuracy 'CADA' – error rates

Doctor	MIM	NB	GBN	Polytree	Default (overall)
0.2834±0.28	0.3349±0.82	**0.2617±1.16**	0.3583±1.56	0.3566±0.66	0.5495

Key: **MIM** – Mutual Information Measure Classifier, **NB** – Naive Bayes Classifier, **GBN** – General Bayesian Network Classifier, **Polytree** – Pearl's Model. Values in **bold** type indicate the highest model performance achieved by the classifier in respect of the CADA database.

Table 3: Average Predictive Accuracy 'LEEDS' – error rates

Doctor	MIM	NB	GBN	Polytree	Default (overall)
0.3413±0.0	0.4569±0.52	0.4489±0.53	0.4882±0.44	0.4770±0.15	0.6382

Key: **MIM** – Mutual Information Measure Classifier, **NB** – Naive Bayes Classifier, **GBN** – General Bayesian Network Classifier, **Polytree** – Pearl's Model. Values in **bold** type indicate the highest model performance achieved by the classifier in respect of the LEEDS database.

Table 2 and Table 3 display the 'overall' predictive values for the CADA and LEEDS databases respectively. In general the NB outperforms the BBN models and in the case of the CADA database, even performs better than the doctors.

The GBN, 'ploytree' (SCN) and the MIM models provide a qualitative structure (Appendix D) in contrast to the NB model, which offers only a trivial representation. In the case of the GBN the structure is a more complex DAG, whilst the SCN and MIM structures correspond to a less complex 'tree' representation. The 'tree' structures of SCN and MIM are essentially the same with the interpretation governed by edge directionality. For the SCN there is a 'multi parented class node, whereas for MIM the class node represents the root vertex and thus acts as a lone parent. In correspondence with NB the MIM structure represents a subset of the NB. That is, the implied feature selection of MIM in respect to the class's children. However, MIM unlike NB does not make the same extreme assumptions of conditional independence.

The plots shown in Fig 5 and Fig 6 represent the results relative to the MIM Classifier and the NB Classifier respectively. Each bar shows the average difference in predictive accuracy. A positive value for an algorithm indicates that the MIM or NB performed better on the CADA and LEEDS data sets. The error bars represent the Post Hoc Tukey comparisons with overall 95% confidence for the relative differences.

For the CADA database, Table 2, the NB Classifier has the best predictive accuracy of the four models used in the study. This includes the 'overall' performance achieved by the doctors, and was statistically significantly different in all applications except in the case of the doctors with a p-value = 0.137. The doctors achieved the greatest predictive accuracy compared to all the BBN models, and was also statistically significantly different. In the case of the MIM Classifier the predictive levels exceeded the other two BBN models and was found to be statistically significant compared to the SCN but not the GBN with a p-value = 0.164.

[4] These are already recorded within the two data sets in respect to each test case used in the study.

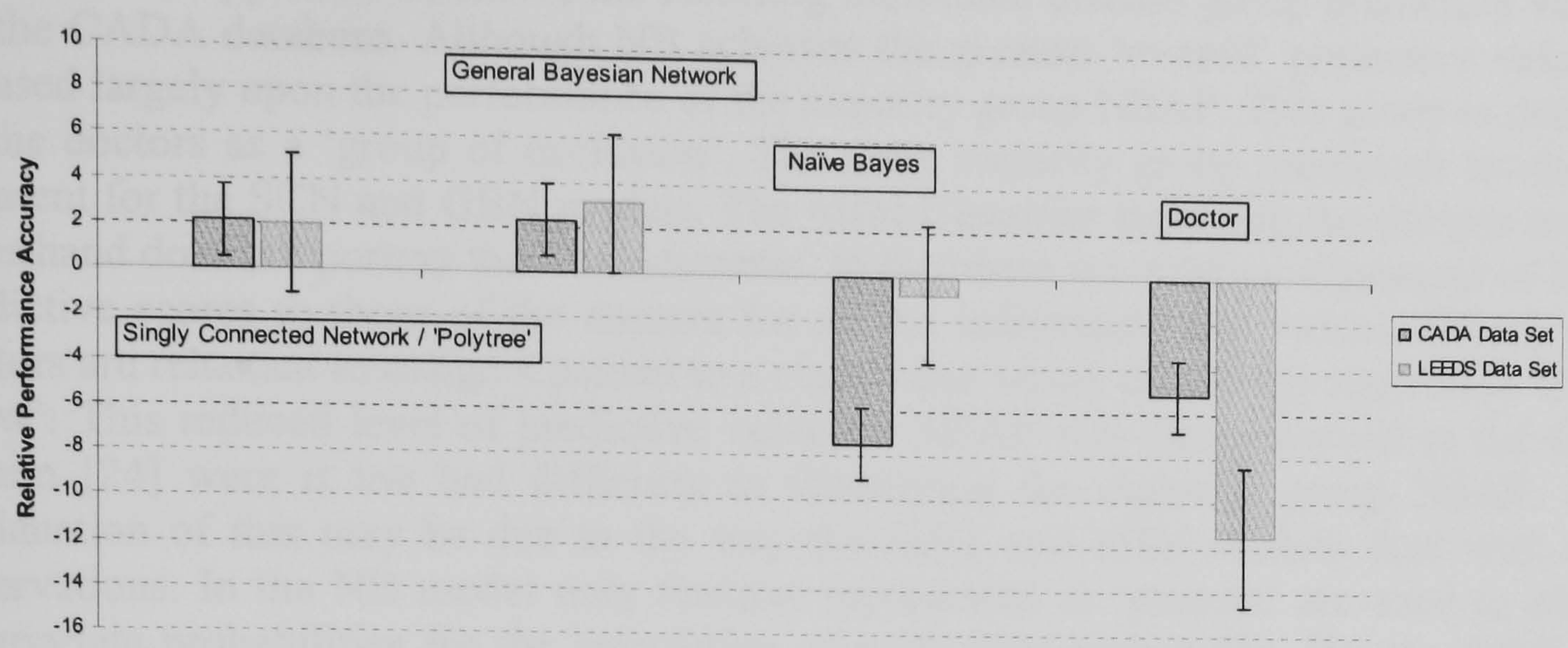


Fig 5. Predictive Accuracy relative to MIM Classifier.

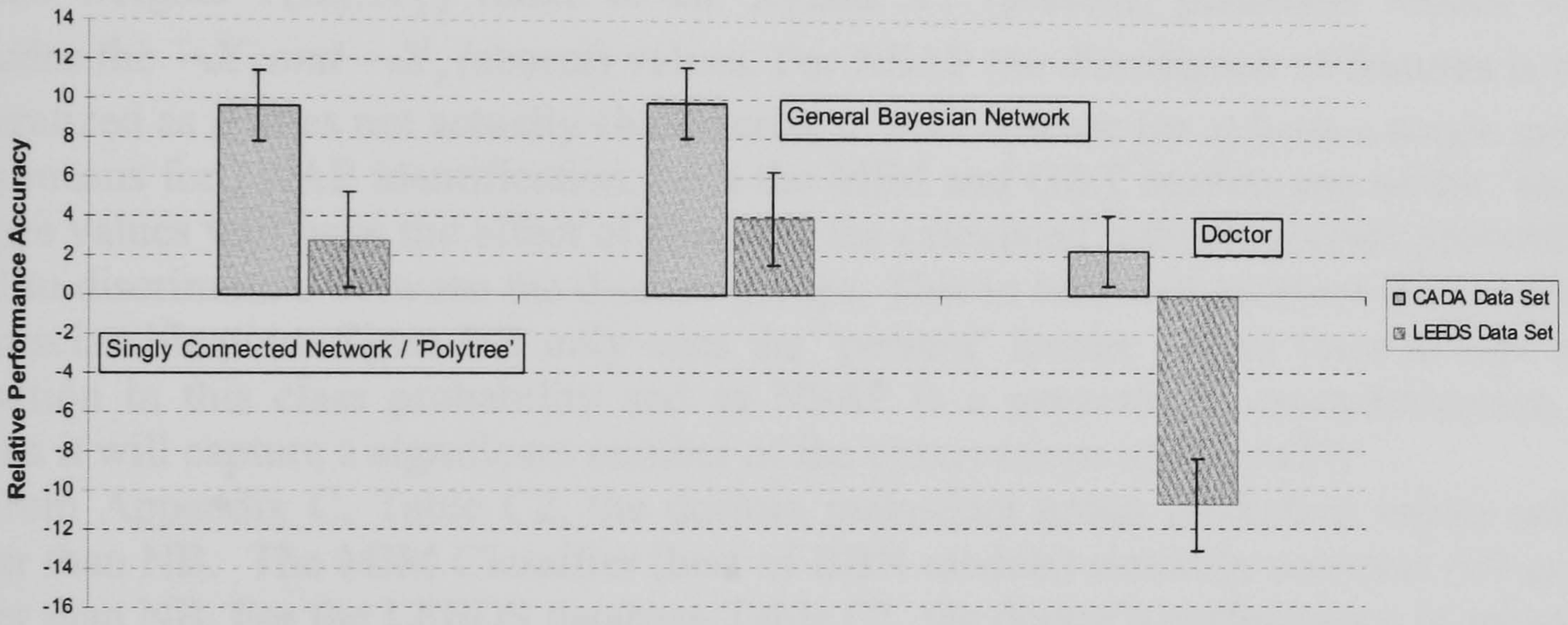


Fig 6. Predictive Accuracy relative to NB Classifier.

As indicated in Table 3, the result in respect of the LEEDS database show the doctors performing (as expected) better than all statistical approaches and was found to be statistically significantly different for all methods. In the case of the MIM and NB models the predictive accuracy was found to be comparable, with the NB statistically significant in respect of all the BBN models and the MIM statistically significantly different for both the SCN and GBN.

In deriving the structures of the BBN models, Appendix D, we identified some symptom-symptom relationships, which appeared meaningless probably due to some 'commonality' or 'correspondence' rather than causal interpretation. Examples are : Pain-site present / Pain-site Onset / Site of Tenderness, Vomiting / Nausea, and Previous Surgery / Abdominal Scar. Gammerman and Luo [52] also observed this commonality. Whilst these were identified by all the BBN models the NB model lost these relationships due to the assumptions of conditional independence.

Table C2, Appendix C, shows the resulting individual disease group predictive values for the CADA database. Although NB achieves the greatest 'overall' predictive value, it is based largely upon the performance of the majority group NSAP. This group is defined by the doctors as a 'group of exclusion'. The same majority group predictive levels are apparent for the SCN and GBN models. The MIM Classifier including the doctors on the other hand does not portray this characteristic. In fact there is a relative alignment of MIM predictive scores to those of the doctors for all the individual class values. Clearly, the doctors are reluctant to assign a patient to a class value which essentially represents 'don't know'. This reduced level of predictive value for NSAP was also observed in the G&T system [24] were it too had difficulty in identifying the majority group NSAP. One explanation of this may be due to the way the G&T and MIM models deal with new observations. In the NB model only features represented as 'present' are used to obtain appropriate probabilities for the calculation of each class probability. For the G&T and MIM the 'absent' feature values are also used. Although not explicitly observed within the feature vector itself, they contribute to the overall calculation of the final probabilities in respect of the possible class outcomes. The G&T model uses both symptom (present) and ¬ symptom (absent) in determining relevant combinations, whilst the MIM model branch weights $I(X_1, X_2)$ relate to all X_1 *and* X_2 (present) parameter values which includes the $\neg X_1$ *and* $\neg X_2$ (absent) values. For NSAP the distribution of features is more generalized as it does not actually characterize a 'real' disease (or at least a single group). This means for NSAP identification, both the MIM and G&T models use of the 'absent' feature values will have the effect of reducing the calculated individual class probabilities used to discriminate between the disease groups. This in turn will increase the possibility of misclassification. Since NB only uses the 'present' feature values there is less or no reduction in this class probability and as NSAP is a generalized characterization, this means it will capture a significant number of the observations more readily.

From Appendix C, Table C2, the doctors individual group predictive values are 6/9 better than NB. The MIM Classifier (best of BBN models) similarly achieves 6/9 groups better than NB. For the LEEDS database Table C1, the doctor's performance is optimal at 9/9 compared to NB, with the MIM Classifier achieving 5/9 group predictive values better than NB.

In addition, from Table C2, Appendix C, despite NB's 'overall' performance in respect of the CADA database exceeding that of the doctors, the Likelihood Ratio is in general lower for NB than the doctors. This indicates that the doctors have a greater ability to discriminate between the disease groups. The Likelihood Ratio for the LEEDS database, Table C1, shows a similar result, however for this particular data set the doctors are already overall winners.

One reason why NB may outperform BBN models is the possibility of high problem dimensionality. In both the CADA and LEEDS databases the number of domain variables is high, however for some of the disease groups there is very little data in order to adequately learn the model. As a consequence overfitting may occur due to spurious dependencies and unreliable probability estimates. The use of 'tree' structures may alleviate this problem as they offer less complex structures. This is demonstrated by the results shown in Table 2 and Table 3 where the MIM Classifier performs better than the GBN model. In respect of the SCN, although a 'tree' structure, it is dependent upon edge directionality and full recovery from data alone is not always possible [27]. From the results obtained node ordering seems to have had an effect on its final predictive

performance. Since the MIM model is not constrained by node ordering it provides an ideal middle ground between the NB and GBN approaches.

As demonstrated by the use of the LEEDS database, which represents a truly 'external' test sample, the NB and MIM 'overall' predictive performance was comparable. Individually the MIM performed better on 5/9 disease groups compared to the NB. For the CADA database 6/9 disease groups were identified by MIM compared to NB with NB's 'overall' predictive performance reflected by the majority group NSAP. In the case of the LEEDS database this was not the case. The group NSAP is not a 'real' group and its sample distribution is thus a generalization that represents several sub-groups. For the CADA and LEEDS data sets this 'characterization' will differ due to geographical population anomalies. As the classifier models are derived from the CADA sample set, the corresponding CADA test samples will be classified better because they have a similar 'characterization' and sample distribution. However, for the LEEDS test samples NSAP will have, in general terms, some similar aspects but on the whole be sufficiently different to make classification of NSAP samples harder to identify using the CADA generated models. Clearly from the results, the remaining eight 'real' disease groups have a 'common' and well characterized description and so their predictive performance, in the case of the CADA and LEEDS databases, align relatively well.

From the confusion matrixes, Appendix B, Table B2, in conjunction with Appendix C, Table C1. The following are observed in respect of the LEEDS database.

In general, the high frequency group misclassifications are lost to other high frequency groups (similarly observed in CADA). Low frequency group misclassifications for the SCN and GBN are lost to high frequency groups, however for MIM and NB these are generally lost to low frequency groups. The exception is the disease group diverticulitis whose misclassification is directed to a high frequency group.

The predictive value for MIM is generally higher than those of NB, GBN and SCN for the high frequency groups with the exception of NSAP (similarly observed in CADA). In the case of the low frequency groups the MIM predictive values are greater than those obtained by NB, GBN and SCN.

From Table C1, Appendix C, the sensitivity values for disease groups perforated peptic ulcer and pancreatitis are lower than those of the doctors for CADA, Table C2. Clearly, the doctors have used some heuristics to diagnose groups perforated peptic ulcer and pancreatitis as it is known that the group pancreatitis in particular has a very poor data definition stored within the database.

From the CADA confusion matrixes, Appendix B, Table B1, in conjunction with Appendix C, Table C2. The following are observed in respect of the CADA database.

The predictive value for MIM is greater than that of the NB for low frequency groups. In most cases, MIM and NB values are higher than those of GBN and SCN. For high frequency groups the predictive value for NB and MIM are similar, with the exception of NSAP, where the MIM and doctors levels fall below those of the SCN, GBN and NB.

On the whole, high frequency groups misclassify into other high frequency groups for the NB, SCN and GBN, whereas for the MIM, this model misclassifies into low frequency groups. From Table C2, Appendix C, the sensitivity values for disease groups perforated peptic ulcer and pancreatitis are lower compared to those of the doctors, again illustrating the doctors use of heuristics.

6. CONCLUSION

6.1. Summary of Contributions

In this paper we investigated the claims that the NB Classifier was optimal in respect of the medical domain AAP. The main contribution of this paper lies in showing that, with respect to the BBN representations, the MIM Classifier can be effectively applied to the domain of AAP. Unlike NB it does so without making the assumption of extreme conditional independence providing a qualitative structure of the domain recognisable by the doctors. In the main part of our study we compared the Naive Bayes with two other 'tree' modelling approaches, namely the MIM Classifier and a 'polytree' (SCN) as defined by Rebane & Pearl [26], along with a General Bayesian network approach.

The MIM Classifier performed 'overall' better than the 'polytree' and GBN. When evaluated with a truly 'external' database of the domain, the MIM Classifier's 'overall' predictive performance was found to be comparable to that achieved by the NB Classifier. Moreover, we observed that the apparent 'optimality' of the NB Classifier's success, particularly in the CADA data set, was largely due to its ability to successfully identify the majority group NSAP. This observation was confirmed in respect of the domain individual disease groups with the MIM Classifier identifying 5/9 class values better than that achieved by NB for the LEEDS data set and 6/9 class values for the CADA data set.

By modelling the domain using an efficient 'tree' structuring algorithm we have avoided the issues of complexity and overfitting to which networks are prone.

Our experimental results on the two AAP databases have demonstrated that the MIM Classifier's performance was comparable to that of the NB Classifier when evaluated with 'external' data of the domain and provides an ideal middle ground between the NB and GBN approaches.

6.2. Future Work

Despite our encouraging experimental results, we believe there are still ways in which we might further improve the performance of the MIM Classifier. In our current representation, the MIM model is defined by the configuration determined by the Chow and Liu algorithm for constructing the tree. However, this phase of construction alone does not provide a mechanism to effectively modify the class - attribute relationship in order to better model the domain. Feature selection has been shown to be a promising area of research [17.53] and this is an aspect we intend to investigate.

ACKNOWLEDGEMENTS

We would like to thank Mr AA Gunn, Mr IWJ Wallace and Mrs J Maclaren (St John's Hospital) for their assistance and permission to use the 'CADA' database, and Professor T de Dombel and Dr S Clamp (Leeds General Infirmary) for the 'LEEDS' database.

References

1. De Dombel, FT editor, (1993) Surgical Decision Making. Oxford: Butterwoth Heinemann.
2. Pass, HI. & Hardy, TD. (1988) The appendix, in Hardy's Textbook of Surgery. Philadelphia: JB Lippincott, pp 574-581, 2nd editor.
3. Luken, TW. & Emerman, C. (1993) The National History and Clinical Findings in Undifferentiated Abdominal Pain. Annals of Emergency Medicine, **22.4**:690-696.
4. Todd, BS. & Stamper, R. (1994) The relative accuracy of a variety of medical diagnostic programs, Methods of Information in Medicine, **33**(4): 402-416.
5. Ohmann, C., Yang, Q., Moustakis, V., Lang, K, & Van Elk, PJ. (1995) Machine Learning Techniques applied to diagnosis of acute abdominal pain, In Pedro Barahona and Mario Stefanelli, Editors, Lecture notes in Artificial Intelligence: Artificial Intelligence in Medicine AMIE95, **934**:276-281, Springer.
6. Pesonen, E., Eskelinen, M., & Juhola, M. (1998) Treatment of missing data values in a neural network based decision support system for acute abdominal pain, Artificial Intelligence in Medicine, **13(3):**139-146.
7. Kurzynski, MN. (1987) Diagnosis of acute abdominal pain using three-stage classifier, Comp. Bio Med, **17(1):**19-27.
8. Admas, ID., Chan, M., Clifford, PC., Cooke, WM., Dallos, V., de Dombal, FT., Edwards, MH., Hancock, DM., Hewett, DJ., McIntyre, N., Somerville, PG., Spiegelhalter, DJ., Wellwood, J. & Wilson, DH. (1986) Computer-aided diagnosis of acute abdominal pain : a multicentre study, British Medical Journal, **293**:800-804.
9. Woziak, M. & Kurzynski, M. (2001) Generating classifier for acute abdominal pain diagnosis problem, Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Vols 1-4 building new bridges at the frontiers of engineering and medicine, pp 3819-3821.
10. Zorman, M., Eich, HP., Kokol, P. & Ohmann, C. (2001) Comparison of three databases with a decision tree approach in the medical field of acute appendicitis, Proceedings of the 10th World Congress on Medical Informatics MEDINFO 2001, London, pp 1414-1418.
11. Ohmann, C., Moustakis, V., Yang, Q., & Lang, K. (1996) Evaluation of automatic knowledge acquisition techniques in the diagnosis of acute abdominal pain, Artificial Intelligence in Medicine, **8**:23-36.
12. Provan, GM & Clarke, JR. (1993) Dynamic Network Construction and Updating Techniques for the Diagnosis of Acute Abdominal Pain, IEEE Transactions on Pattern Analysis and Machine Intelligence, **15:3**:299-307.
13. Norusis, M. & Jacquez, J (1975) Diagnosis I: symptom nonindependence in mathematical models for diagnosis, Computers and Biomedical Research, **8**:156-172.
14. Fryback, DG. (1978) Bayes' theorem and conditional nonindependence of data in medical diagnosis, Computers and Biomedical Research, **11**:429-435.
15. Seroussi, B. (1986) Computer-aided diagnosis of acute abdominal pain when taking into account interactions, Methods of Information in Medicine, **25**:194-198.
16. Gammerman, A. & Thatcher, AR. (1991) Bayesian diagnostic probabilities without assuming independence of symptoms, Methods of Information in Medicine, **30**:15-22.
17. Singh, M. (1998) Learning Bayesian Networks for solving real-world problems, PhD Thesis, Department of Computer and Information Science, University of Pennsylvania.
18. Edwards, F. & Davis, R. (1984) Use of a Bayesian algorithm in the computer-assisted diagnosis of appendicitis, Surgical Gynaecology Obstetrics, **158**:219-222.
19. De Dombal, FT. (1991) The diagnosis of acute abdominal pain with computer assistance, Annals Chir., **45**:273-277.
20. De Dombal, FT., Leaper, DJ., Staniland, JR., McCann, A., & Horrecks, J. (1972) Computer aided diagnosis of acute abdominal pain, British Medical Journal, **2**:9-13.
21. Gammerman, A. & Thatcher, AR. (1990) Bayesian Inference in an expert system without assuming independence, In. Golumbi, M, editor, Advances in AI, Natural Languages and Knowledge Based Systems, pp 182-218, Springer-Verlag.
22. Provan, GM. & Singh, M. (1996) Data Mining and model simplicity: A case study in diagnosis, Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KKD).
23. Hand, DJ. & Yu, K. (2001) Idiot's Bayes – Not so stupid after all?, International Statistical Review, **69:3**:385-398.

24. Gu, Y. & Gammerman, A. (1990) A computer-aided diagnostic system and its application to a large medical database, In IEE Colloquium on AI in medical decision making, London.
25. Thomas, CS. (1999) Classifying acute abdominal pain by assuming independence: A study using two models constructed from data, Technical Report CSM-153, Department of Computing Science and Mathematics, University of Stirling, Stirling, UK.
26. Rebane, G. & Pearl, J. (1987) The recovery of causal polytree from statistical data, In Proceedings of the 3rd conf. On Uncertainty in Artificial Intelligence. Seattle, WA.
27. Pearl, J. (1988) Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann Publishers, INC., San Mateo, Californian. ISBN 0-934613-73-7.
28. Thomas, CS., Howe, CA. & Smith, LS. (2005) A New Singly Connected Network Classifier Based on Mutual Information, to appear in Intelligent Data Analysis, 9(2).
29. Jensen, F.V. (1996) An introduction to Bayesian Networks, UCL Press Ltd, London, ISBN 1-85728-332-5.
30. Heckerman, D. (1998) A tutorial on learning with Bayesian networks, In: M. I. Jordan (ed): Learning in Graphical models, Dordrecht, Netherlands: Kluwer.
31. Buntine, W. (1996) A guide to the literature on learning probabilistic networks from data, IEEE Trans Knowledge Data Engineering. **8**:195-210.
32. Cheng, J. & Greiner, R. (2001) Learning Bayesian belief network classifiers: Algorithms and system In: Proc. 14th Biennial conference of the Canadian Society for Computational Studies of Intelligence, Ottawa, ON.
33. Heckerman, D., Geiger, D., & Chickering, D.M. (1995) "Learning Bayesian networks: the combination of knowledge and statistical data", Machine Learning, **20**(3):197-243.
34. Pan, H. P. (2002) Learning Bayesian networks: II – a computational algorithm, In 5th International Conference on Information Fusion, (submitted), Annapolis, Maryland, USA.
35. Friedman, N and Goldszmidt, M. (1996) Learning Bayesian networks with local structure, In Proc of the 12th International Conf. on Uncertainty in Artificial Intelligence.
36. Jensen, F. V. (2001) Bayesian Networks and Decision Graphs, New York, Springer Verlag.
37. McNaught, K., Clifford, S., Vaughn, M., Fogg, A & Foy, M. (2001) A Bayesian Belief network for lower back pain diagnosis, In European Conf. on Artificial Intelligence in Medicine, AIME'01: working notes for workshop, 53-58.
38. Fung, R. and Favero, B. D. (1995) Applying Bayesian networks to information retrieval, Communications of the ACM **38**(3).
39. Blanco, R., Larrañaga, P., Inza, I & Sierra, B. (2001) Selection of highly accurate genes for cancer classification by estimation of distribution algorithms, In European Conf. on Artificial Intelligence in Medicine, AIME'01: working notes for workshop, 29-34,
40. Ezawa, K. & Nonton, S. (1995) Knowledge discovery in telecommunication services data using Bayesian network models, In: Proc. 1st International Conference on Knowledge Discovery and Data Mining.
41. Cooper, G. (1990) The Computational Complexity of Probabilistic inference using Belief networks, Artificial Intelligence, **42**:393-405.
42. Friedman, N., Geiger, D., & Goldszmidt, M. (1997) Bayesian Network Classifiers, Machine Learning, **29**:131-161.
43. Cheng, J. & Greiner, R. (1999) Comparing Bayesian Network classifiers, In proceedings of Uncertainty in Artificial Intelligence, UAI-99.
44. Duda, R. & Hart, P. (1973) Pattern Classification and Scene Analysis, John Wiley & Sons.
45. Langley, P., Iba, W. & Thompson, K. (1992) An analysis of Bayesian classifiers, In AAAI'90, pp 223-228.
46. Chow, C.K. & Liu, C.N. (1968) Approximating Discrete Probability Distributions with Dependence Trees, IEEE Transactions on information Theory, Vol. **IT-14**:462-467.
47. Cormen, TH. & Leiserson, CE. & Rivest, RL. (1990) Introduction to Algorithms, MIT Press.
48. Verma, T. & Pearl, J. (1992) An algorithm for deciding if a set of observed independencies has a causal explanation, In Proc. of the 8th Conf. on Uncertainty in Artificial Intelligence, Morgan Kaufmann, 323-330.
49. Gunn, AA. (1976) The Diagnosis of Acute Abdominal Pain with computer diagnosis, Journal of the Royal College of Surgeons, Edinburgh, **21**:170-172.
50. Clarke, JR. & Hayward, CZ. (1990) Workshop on surgical decision making: a scientific approach to surgical reasoning, Theoretical Surgery, **5**:129-132.
51. Cheng, J. PowerConstructor System(1998) *http://www.cs.ualberta.ca/~jcheng/bnpc.htm*.

52. Gammerman, A. & Luo, Z. (1991) Constructing Causal Trees from a medical database, Technical Report TR91002, Department of Computer Science, Heriot-Watt University, Edinburgh, UK.
53. Tsymbal, A., Cunningham, P., Pechenizkiy, M. & Puuronen, S. (2003) Search strategies for ensemble feature selection in medical diagnosis, Proceedings of the 16th Annual IEEE Symposium on computer-based medical systems, New York, pp 124-129.

APPENDIX A - Diagnostic and Symptom Codes AAP

Symptom	Value
SEX	male(1/1), female(1/2)
AGE	0-9(2/1), 10-19(2/2), 20-29(2/3), 30-39(2/4), 40-49(2/5), 50-59(2/6), 60-69(2/7), 70 +(2/8)
Pain-site Onset	right upper quadrant(3/1), left upper quadrant(3/2), right lower quadrant(3/3), left lower quadrant(3/4), upper half(3/5), lower half(3/6), right half(3/7), left half(3/8), central(3/9), general(3/10), right loin(3/11), left loin(3/12), epigastric(3/13), right upper quadrant + epigastric(3/14), right lower quadrant + left lower quadrant(3/15), right lower quadrant + right loin(3/16)
Pain-site Present	right upper quadrant(4/1), left upper quadrant(4/2), right lower quadrant(4/3), left lower quadrant(4/4), upper half(4/5), lower half(4/6), right half(4/7), left half(4/8), central(4/9), general(4/10), right loin(4/11), left loin(4/12), epigastric(4/13), pain settled(4/14), right upper quadrant + epigastric(4/15), right lower quadrant + central(4/16), right lower quadrant + right loin(4/17), left lower quadrant + left loin(4/18), right half + right loin(4/19), left half + left loin(4/20), central + epigastric(4/21)
Aggravating Factors	movement(5/1), coughing(5/2), inspiration(5/3), food(5/4), other(5/5), nil(5/6), movement + coughing(5/7), movement + inspiration(5/8), movement + food(5/9), movement + other(10), movement + coughing + inspiration(5/11), coughing + inspiration(5/12)
Relieving Factors	lying still(6/1), vomiting(6/2), antacids(6/3), milk/food(6/4), other(6/5), nil(6/6), lying still + vomiting(6/7), lying still + other(6/8)
Progress of Pain	getting better(7/1), no change(7/2), getting worse(7/3)
Duration of Pain	under 12 hours(8/1), 12-24 hours(8/2), 24-48 hours(8/3), over 48 hours(8/4)
Type of Pain	steady(9/1), intermittent(9/2), colicky(9/3), sharp(9/4), steady + intermittent(9/5), steady + colicky(9/6), steady + sharp(9/7), steady + colicky + sharp(9/8), intermittent + colicky(9/9), intermittent + sharp(9/10), intermittent + colicky + sharp(9/11), colicky + sharp(9/12)
Severity of Pain	moderate(10/1), severe(10/2)
Nausea	nausea present(11/1), no nausea(11/2)
Vomiting	present(12/1), no vomiting(12/2)
Anorexia	present(13/1), normal appetite(13/2)
Indigestion	history of dyspepsia(14/1), no history of dyspepsia(14/2)
Jaundice	history of jaundice(15/1), no history of jaundice(15/2)
Bowel habit	no change(16/1), constipated(16/2), diarrhoea(16/3), blood(16/4), mucus(16/5), constipated + diarrhoea(16/6), diarrhoea + blood(16/7)
Micturition	normal(17/1), frequent(17/2), dysuria(17/3), haematuria(17/4), dark urine(17/5), frequent + dysuria(17/6)
Previous Pain	similar pain before(18/1), no similar pain before(18/2)
Previous surgery	yes(19/1), none(19/2)
Drugs	being taken(20/1), not being taken(20/2)
Mood	normal(21/1), distressed(21/2), anxious(21/3), distressed + anxious(21/4)
Colour	normal(22/1), pale(22/2), flushed(22/3), jaundiced(22/4), cyanosed(22/5)
Abdominal Movement	normal(23/1), poor/nil(23/2), visible peristalsis(23/3)
Abdominal scar	present(24/1), absent(24/2)
Abdominal Distension	present(25/1), absent(25/2)
Site of Tenderness	right upper quadrant(26/1), left upper quadrant(26/2), right lower quadrant(26/3), left lower quadrant(26/4), upper half(26/5), lower half(26/6), right half(26/7), left half(26/8), central(26/9), general(26/10), right loin(26/11), left loin(26/12), epigastric(26/13), none(26/14), right upper quadrant + epigastric(26/15), right lower quadrant + left lower quadrant (26/16), right lower quadrant + right half(26/17), right lower quadrant + central(26/18), right lower quadrant + right loin(26/19), right lower quadrant + epigastric(26/20), left lower quadrant + left loin(26/21), left half + left loin(26/22)
Rebound	present(27/1), absent(27/2)
Guarding	present(28/1), absent(28/2)
Rigidity	present(29/1), absent(29/2)
Abdominal Masses	present(30/1), absent(30/2)
Murphy's test	positive(31/1), negative(31/2)
Bowel sounds	normal(32/1), decreased/absent(32/2), increased(32/3)
Rectal Examination	tender left side(33/1), tender right side(33/2), generally tender(33/3), mass felt(33/4), normal(33/5)

Table A1-1 Symptom Parameters and Codes

Diagnostic Groups	
Disease	**Value**
APP	Appendicitis
DIV	Diverticulitis
PPU	Perforated Peptic Ulcer
NSAP	Non Specific Abdominal Pain
CHO	Cholecystitis
INO	Intestinal Obstruction
PAN	Pancreatitis
RCO	Renal Colic
DYS	Dyspepsia

Table A1-2 Diagnostic Groups and Codes

APPENDIX B – CADA/LEEDS Discriminant Matrices

	APP	DIV	PPU	NSAP	CHO	INO	PAN	RCO	DYS	TOTAL
APP	298	7	13	46	3	5	4	7	2	385
DIV	2	63	5	1	3	13	1	2	2	92
PPU	-	-	43	1	3	3	5	-	1	56
NSAP	111	60	16	687	27	47	30	75	47	1100
CHO	1	9	13	6	147	16	17	7	19	235
INO	11	25	13	16	4	130	18	3	8	228
PAN	1	6	12	2	3	5	30	1	12	72
RCO	7	6	6	11	7	5	9	235	1	287
DYS	3	17	23	19	34	13	40	3	301	453
TOTAL	434	193	144	789	231	237	154	333	393	2908

Table B1-1 MIM Classifier (CADA)

	APP	DIV	PPU	NSAP	CHO	INO	PAN	RCO	DYS	TOTAL
APP	294	2	5	73	2	2	2	1	4	385
DIV	-	49	5	19	1	12	1	2	3	92
PPU	1	-	37	-	4	2	11	-	1	56
NSAP	36	28	5	880	12	47	8	45	39	1100
CHO	-	1	6	11	139	16	26	3	33	235
INO	8	14	6	22	6	147	15	-	10	228
PAN	2	3	9	1	7	7	23	2	18	72
RCO	7	4	1	25	6	6	2	234	2	287
DYS	3	9	8	27	20	17	23	2	344	453
TOTAL	351	110	82	1058	197	256	111	289	454	2908

Table B1-2 NB Classifier (CADA)

	APP	DIV	PPU	NSAP	CHO	INO	PAN	RCO	DYS	TOTAL
APP	228	2	3	133	3	7	1	2	6	385
DIV	4	15	1	42	2	20	1	3	4	92
PPU	7	-	32	3	6	2	2	-	4	56
NSAP	37	6	4	904	13	34	7	37	58	1100
CHO	1	2	3	42	125	7	10	3	42	235
INO	6	4	2	66	5	124	5	3	13	228
PAN	5	1	2	22	10	7	9	-	16	72
RCO	6	1	1	69	8	5	1	182	14	287
DYS	6	4	1	153	17	14	2	9	247	453
TOTAL	300	35	49	1434	189	220	38	239	404	2908

Table B1-3 GBN Classifier (CADA)

	APP	DIV	PPU	NSAP	CHO	INO	PAN	RCO	DYS	TOTAL
APP	308	2	1	66	1	1	3	2	1	385
DIV	1	45	6	23	3	12	-	1	1	92
PPU	1	-	38	5	5	2	1	-	4	56
NSAP	271	13	1	729	12	26	4	24	20	1100
CHO	1	2	4	24	162	7	13	2	20	235
INO	5	3	3	26	1	179	3	1	7	228
PAN	-	2	3	10	4	4	45	1	3	72
RCO	6	-	-	27	4	1	1	247	1	287
DYS	3	3	9	58	19	5	15	1	340	453
TOTAL	596	70	65	968	211	237	85	279	397	2908

Table B1-4 Doctors (CADA)

	APP	DIV	PPU	NSAP	CHO	INO	PAN	RCO	DYS	TOTAL
APP	199	1	1	172	2	4	2	-	4	385
DIV	1	25	2	31	3	25	-	3	2	92
PPU	1	2	6	9	7	17	1	-	13	56
NSAP	30	11	1	933	16	35	1	33	40	1100
CHO	1	-	4	34	115	27	1	11	42	235
INO	4	13	2	68	6	105	3	5	22	228
PAN	2	1	3	20	9	8	2	-	27	72
RCO	3	4	-	85	7	6	1	178	3	287
DYS	2	3	1	82	22	29	2	4	308	453
TOTAL	243	60	20	1434	187	256	13	234	461	2908

Table B1-5 SCN – 'polytree' Classifier (CADA)

	APP	DIV	PPU	NSAP	CHO	INO	PAN	RCO	DYS	TOTAL
APP	947	9	5	229	5	6	1	5	6	1213
DIV	23	120	5	41	3	23	1	4	2	222
PPU	6	0	113	15	6	1	6	0	3	150
NSAP	539	23	16	1024	76	64	36	77	89	1944
CHO	10	3	10	52	406	23	23	5	23	555
INO	14	6	6	46	2	254	3	3	4	338
PAN	3	2	20	17	26	10	126	2	18	224
RCO	12	2	0	53	7	1	1	301	0	377
DYS	4	1	8	40	21	7	21	0	248	350
TOTAL	1558	166	183	1517	552	389	218	397	393	5373

Table B2-1 Doctors (LEEDS)

	APP	DIV	PPU	NSAP	CHO	INO	PAN	RCO	DYS	TOTAL
APP	800	60	85	150	8	61	20	19	10	1213
DIV	8	133	18	9	3	32	5	11	3	222
PPU	3	5	103	3	7	9	16	-	4	150
NSAP	267	154	87	831	55	155	122	130	143	1944
CHO	3	26	49	13	334	21	72	9	28	555
INO	16	45	29	11	6	185	26	6	14	338
PAN	5	8	48	6	28	37	72	6	14	224
RCO	15	16	21	34	6	10	18	254	3	377
DYS	3	8	14	20	10	17	62	10	206	350
TOTAL	1120	455	454	1077	457	527	413	445	425	5373

Table B2-2 MIM Classifier (LEEDS)

	APP	DIV	PPU	NSAP	CHO	INO	PAN	RCO	DYS	TOTAL
APP	763	18	162	226	3	26	4	6	5	1213
DIV	9	102	18	55	2	27	2	4	3	222
PPU	13	7	87	4	9	7	21	-	2	150
NSAP	240	79	108	1024	46	178	60	109	100	1944
CHO	6	7	46	31	288	35	75	12	55	555
INO	16	24	28	32	3	205	13	8	9	338
PAN	5	2	47	18	26	31	64	6	25	224
RCO	14	12	8	56	3	10	4	267	3	377
DYS	1	3	23	36	11	28	28	-	220	350
TOTAL	1067	254	527	1482	391	547	271	412	422	5373

Table B2-3 NB Classifier (LEEDS)

	APP	DIV	PPU	NSAP	CHO	INO	PAN	RCO	DYS	TOTAL
APP	567	20	159	391	21	23	6	9	17	1213
DIV	13	39	12	92	6	43	5	5	7	222
PPU	25	2	71	15	13	7	8	1	8	150
NSAP	145	29	77	1246	52	126	22	78	169	1944
CHO	7	6	34	102	284	27	26	13	56	555
INO	19	10	16	80	10	169	11	5	18	338
PAN	15	5	31	50	34	18	28	3	40	224
RCO	10	4	26	125	6	12	8	167	19	377
DYS	4	2	14	97	9	18	18	9	179	350
TOTAL	805	117	440	2198	435	443	132	290	513	5373

Table B2-4 GBN Classifier (LEEDS)

	APP	DIV	PPU	NSAP	CHO	INO	PAN	RCO	DYS	TOTAL
APP	477	7	8	680	2	28	2	3	6	1213
DIV	5	68	6	81	2	43	-	11	6	222
PPU	11	2	24	43	12	28	13	2	15	150
NSAP	123	32	9	1373	41	141	14	79	132	1944
CHO	4	3	14	74	327	46	9	13	65	555
INO	11	18	11	112	6	143	6	8	23	338
PAN	3	-	17	50	29	50	11	5	59	224
RCO	7	4	3	156	2	15	4	183	3	377
DYS	3	2	4	85	11	26	8	7	204	350
TOTAL	644	136	96	2654	432	520	67	311	513	5373

Table B2-5 SCN – 'polytree' Classifier (LEEDS)

APPENDIX C – LEEDS/CADA Statistical tables

Final Diagnosis	# Cases	% Cases	MIM Predictive Value %	Doc Predictive Value %	NB Predictive Value %	GBN Predictive Value %	Poly Predictive Value %	MIM Likelihood Ratio	Doc Likelihood Ratio	NB Likelihood Ratio	GBN Likelihood Ratio	Poly Likelihood Ratio
APP	1213	22.58	65.95	78.07	62.86	46.74	39.28	7.36	8.72	6.83	4.98	4.75
DIV	222	4.13	59.91	54.05	45.72	17.57	30.63	16.15	36.90	17.05	9.57	16.88
PPU	150	2.79	68.67	75.33	58	47.33	16	23.74	86.61	12.71	10.05	10.53
NSAP	1944	36.18	42.77	52.67	52.67	64.12	70.63	2.98	2.83	2.92	2.58	2.46
CHO	555	10.33	60.09	73.15	51.89	51.17	58.83	16.22	23.80	13.78	11.88	16.36
INO	338	6.29	54.59	75.15	60.65	50	42.31	11.04	38.74	13.57	11.11	6.84
PAN	224	4.17	32.14	56.25	28.57	12.5	4.91	5.67	30.40	7.52	5.63	4.09
RCO	377	7.02	67.37	79.84	70.69	44.30	48.54	22.81	49.64	29.23	13.91	15.38
DYS	350	6.51	58.86	70.86	62.86	51.14	58.29	16.70	30.81	19.85	9.95	13.27

Table C1-1 Statistical Table Predictive Values and Likelihood Ratios (LEEDS)

Final Diagnosis	# Cases	% Cases	MIM Sensitivity	MIM Specificity	Doc Sensitivity	Doc Specificity	NB Sensitivity	NB Specificity	GBN Sensitivity	GBN Specificity	Poly Sensitivity	Poly Specificity
APP	1213	22.57584	0.714924	0.902915	0.607831	0.930275	0.71462	0.895379	0.704786	0.858597	0.739907	0.844259
DIV	222	4.13177	0.292308	0.981903	0.722892	0.980411	0.401186	0.976465	0.333333	0.965183	0.49635	0.970588
PPU	150	2.791736	0.226872	0.990445	0.617486	0.992871	0.165242	0.987001	0.160998	0.983982	0.251309	0.976125
NSAP	1944	36.1809	0.772052	0.741038	0.675016	0.761411	0.690958	0.763557	0.567494	0.780419	0.51704	0.78988
CHO	555	10.32942	0.730559	0.954948	0.735507	0.969094	0.738462	0.946418	0.652124	0.945114	0.756663	0.953759
INO	338	6.290713	0.349763	0.968321	0.652956	0.983146	0.374088	0.972435	0.38106	0.965717	0.275	0.959819
PAN	224	4.168993	0.173913	0.969349	0.577982	0.980989	0.235727	0.968637	0.210526	0.962595	0.164179	0.959857
RCO	377	7.016564	0.569507	0.975036	0.758186	0.984727	0.648058	0.977827	0.574871	0.958682	0.589372	0.961679
DYS	350	6.514052	0.485849	0.970903	0.631043	0.979518	0.521327	0.973743	0.349951	0.964826	0.398438	0.969965

Table C1-2 Statistical Tables Sensitivity and Specificity values (LEEDS)

Final Diagnosis	# Cases	% Cases	MIM Predictive Value %	Doc Predictive Value %	NB Predictive Value %	GBN Predictive Value %	Poly Predictive Value %	MIM Likelihood Ratio	Doc Likelihood Ratio	NB Likelihood Ratio	GBN Likelihood Ratio	Poly Likelihood Ratio
APP	385	13.23934	77.46753	79.87013	76.23377	59.15584	51.68831	19.61169	15.35922	23.42432	12.61426	11.68112
DIV	92	3.163686	68.75	48.91304	53.26087	15.48913	26.90217	30.77732	38.39599	28.84931	14.83017	17.46914
PPU	56	1.925722	76.78571	67.85714	66.07143	57.14286	10.71429	63.60548	92.33675	67.53665	79.44444	18.00779
NSAP	1100	37.82669	62.38636	66.27273	80.13636	82.15909	84.81818	4.454493	3.918286	7.051729	4.735178	5.728856
CHO	235	8.081155	62.34043	68.93617	59.14894	53.19149	48.82979	19.22883	28.07833	20.03444	16.32491	13.92506
INO	228	7.84044	56.90789	76.86404	64.47368	54.16667	45.83333	14.84922	37.87677	18.86092	14.52894	8.746889
PAN	72	2.475928	42.01389	62.5	31.94444	12.15278	2.777778	12.97948	55.52114	11.74519	9.856505	6.362637
RCO	287	9.869326	81.96864	84.32056	81.44599	63.41463	62.10801	33.9154	51.69732	41.23578	19.31273	18.70861
DYS	453	15.57772	66.39073	75.05519	75.99338	54.41501	67.88079	12.65998	18.98923	17.03267	7.409252	11.28338

Table C2-1 Statistical Table Predictive Values and Likelihood Ratios (CADA)

Final Diagnosis	# Cases	% Cases	MIM Sensitivity	MIM Specificity	Doc Sensitivity	Doc Specificity	NB Sensitivity	NB Specificity	GBN Sensitivity	GBN Specificity	Poly Sensitivity	Poly Specificity
APP	385	13.23934	0.687608	0.964939	0.515075	0.9697	0.837973	0.964226	0.760434	0.939716	0.815574	0.93018
DIV	92	3.163686	0.326031	0.989407	0.636042	0.983199	0.443439	0.984629	0.401408	0.972933	0.4125	0.976387
PPU	56	1.925722	0.29913	0.995297	0.584615	0.993632	0.453988	0.993278	0.666667	0.991608	0.311688	0.982691
NSAP	1100	37.82669	0.869772	0.804743	0.750579	0.799432	0.832979	0.881876	0.63034	0.866881	0.649835	0.886568
CHO	235	8.081155	0.635575	0.966947	0.760563	0.974926	0.709184	0.964602	0.660502	0.95954	0.615282	0.955815
INO	228	7.84044	0.546316	0.963209	0.747335	0.980115	0.575906	0.969466	0.564571	0.961142	0.407407	0.953423
PAN	72	2.475928	0.196748	0.984842	0.530973	0.990344	0.205817	0.982477	0.217391	0.977944	0.153846	0.97582
RCO	287	9.869326	0.701425	0.979318	0.838095	0.98282	0.814719	0.980242	0.759916	0.960652	0.760939	0.959327
DYS	453	15.57772	0.766242	0.939475	0.746023	0.95392	0.755348	0.955653	0.610905	0.917548	0.6703	0.940594

Table C2-2 Statistical Tables Sensitivity and Specificity values (CADA)

APPENDIX D – AAP Network structures

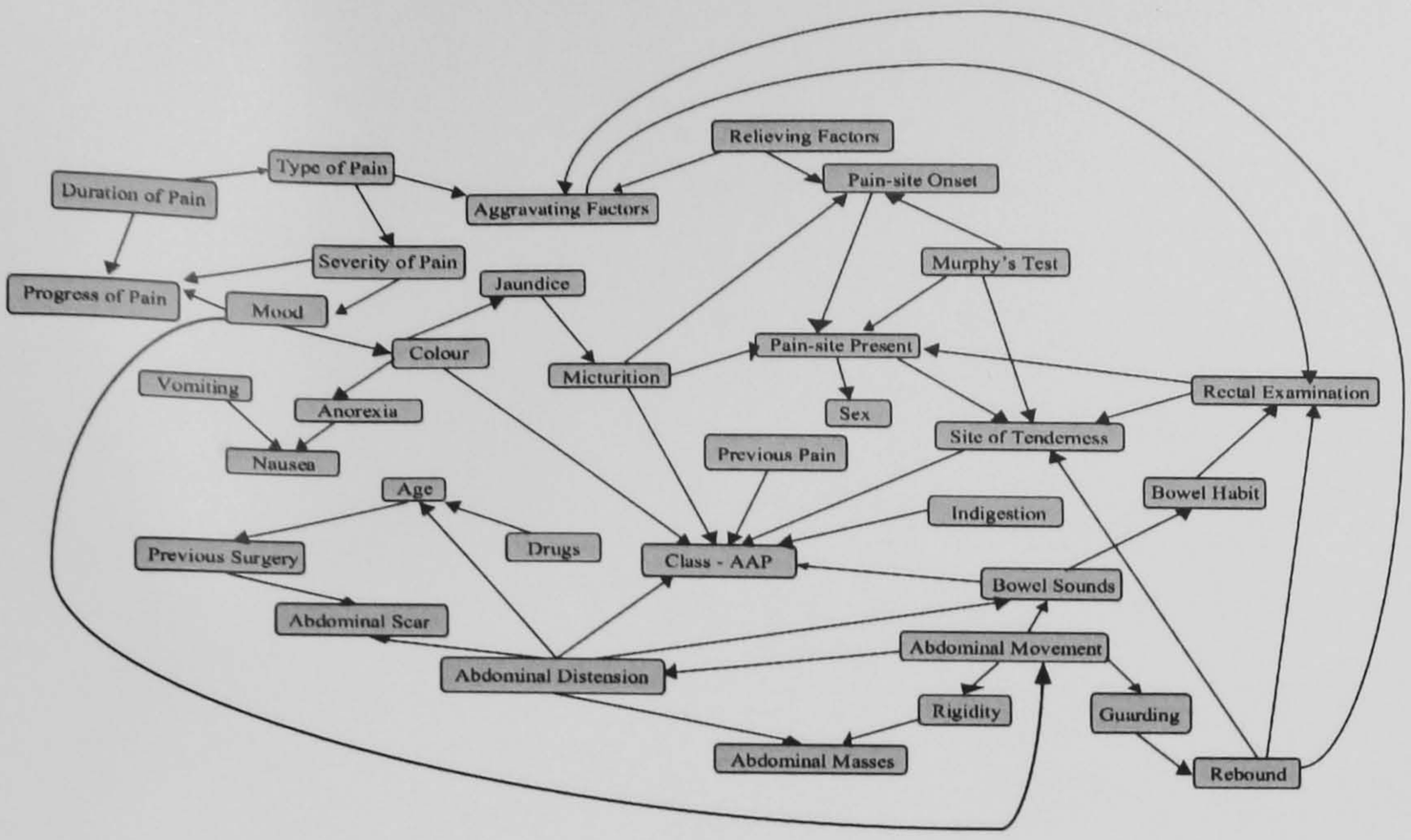


Fig D-1 General Bayesian Network (GBN) Structure

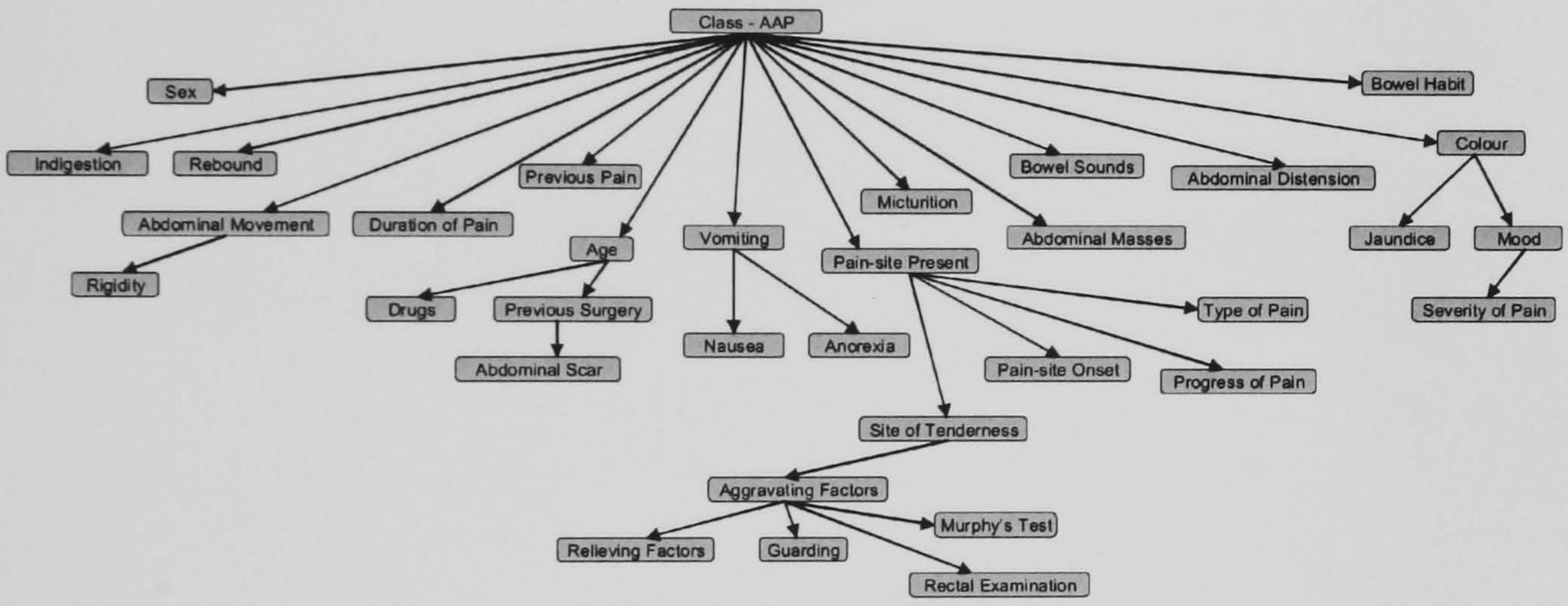


Fig D-2 Mutual Information Measure (MIM) Structure

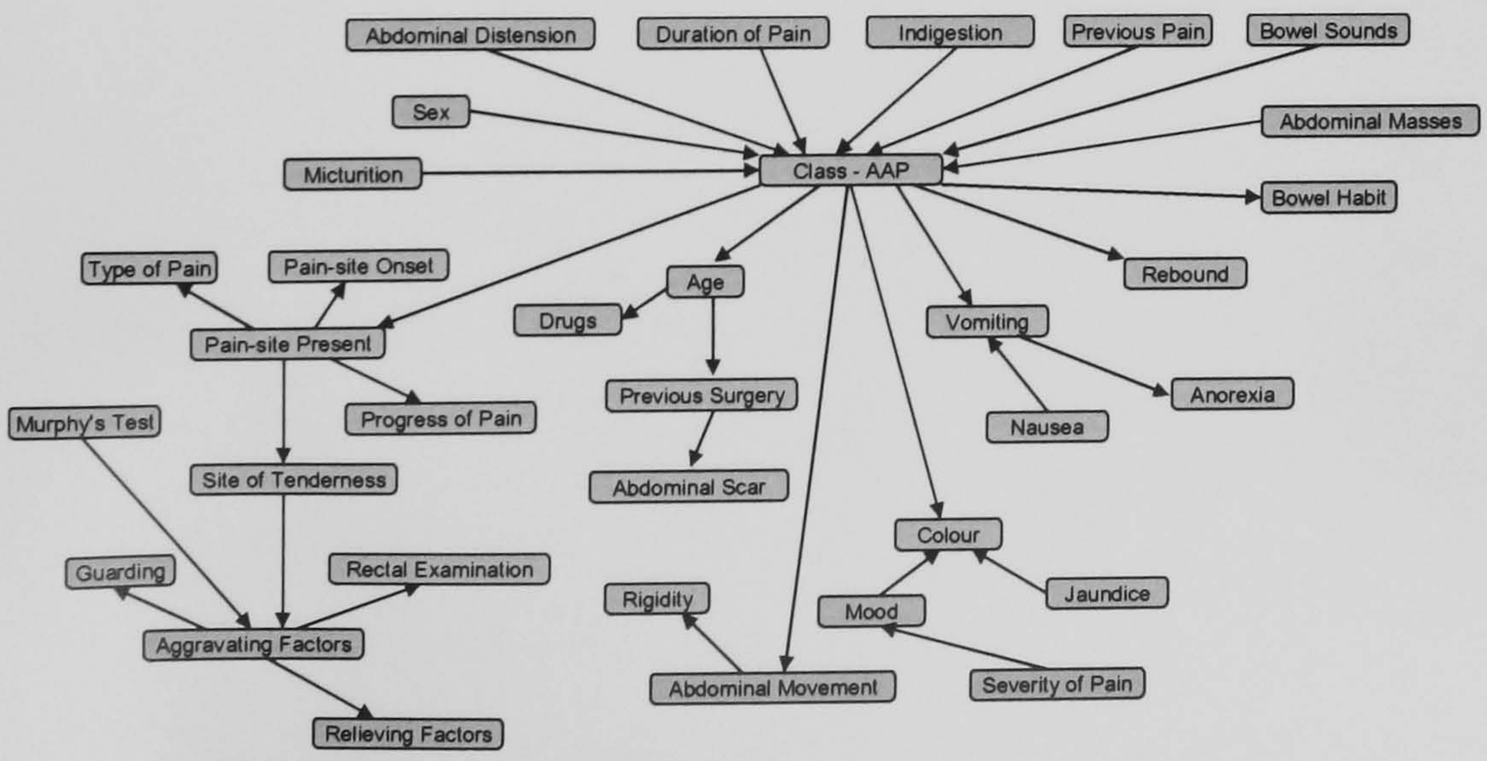


Fig D-3 Singly Connected Network 'polytree' (SCN) Structure