DOCTOR OF PHILOSOPHY

Feature selection and classification of non-traditional data : examples from veterinary medicine

Hoare, Zoe

Award date:
2007

Awarding institution:
Bangor University

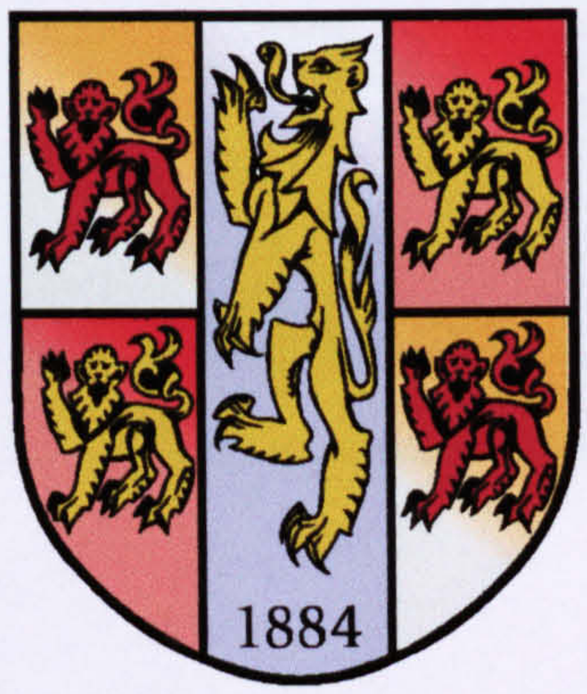
FEATURE SELECTION AND CLASSIFICATION OF NON-TRADITIONAL DATA.

EXAMPLES FROM VETERINARY MEDICINE

Zoë Susannah Jane Hoare

School of Informatics, University of Wales, Bangor.

• PRIFYSGOL CYMRU •
UNIVERSITY OF WALES
BANGOR



Thesis submitted to the University of Wales

in Candidature for the degree of Doctor of Philosophy

Abstract

Early diagnosis of notifiable diseases in the veterinary domain is important with regard to agriculture, the health sector and the economy. With no diagnostic test in the live animal for either BSE or Scrapie many cases may be mis-diagnosed.

Traditionally, data for pattern recognition is stored as recorded cases of interest either labelled with their outcome (suitable for supervised classification) or unlabelled. Each case is described by a collection of symptoms, recorded as present / absent. These are called "binary features". In the case of medical data, the amount of cases recorded in this way may be limited for many reasons. To overcome this lack of data expert-estimated probability tables have been proposed as a substitute. These "non-traditional" tables contain the estimated percentage frequencies of clinical symptoms in various diseases. The construction of the tables assumed that the clinical signs (features) were independent given the diseases (classes).

Given the "non-traditional" data, various feature selection techniques were applied and compared in this study in order to select a reduced subset of features (symptoms). The potential, limitations and stability of Sequential Forward Selection (SFS) in particular, were investigated.

Decision trees and Naïve Bayes classifier models were applied for the diagnosis task. The apparent success and stability of Naïve Bayes in the medical domain led to an in-depth investigation of the effects of this type of data and its inherent assumptions on the model. Naïve Bayes is known to be optimal in the case of independent features, which is the condition assumed by the estimated probability tables in the "non-traditional" data. Various proposed adaptations to the Naïve Bayes model were investigated with regard to their optimality when the independence assumption is violated. Finally, the performance of Naïve Bayes with regard to traditionally stored medical data with binary features was assessed. Naïve Bayes and its adaptations performed well with the traditional data. Since the effect of assuming independence when it is not true is minimal, using the "non-traditional" data with the Naïve Bayes classifier can be a practical solution for veterinary diagnosis.

Acknowledgements

Firstly, I would like to thank my supervisor, Ludmila Kuncheva. Without her inspiration, guidance and overall faith in me this work would never have come to be. I also wish to thank Chris Whitaker who has been there to guide me through all things statistical and Dylan who was a great source of discussion and information. On the veterinary side of my work I wish to thank Peter Cockcroft and Victor del rio Vilas.

I also need to thank my family and friends who have given me the support to do this work over the past few years. To Box, who has had more random conversations about mathematics and mad cow disease than he cares to remember, thank you. To my Mum and Dad who are always there for me. Finally, to Lesley and Vicky who'll listen when I need them to and then distract me when I needed to be.

Thank you all.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 What is Pattern Recognition?

Informally, the pattern recognition process can be likened to a game of twenty questions or Animal, Vegetable, Mineral. More formally pattern recognition is an area of mathematics based on identifying and classifying objects. The area has undergone much development since the 1960's.

An object, $\mathbf{x}$ can be described by n different characteristics. In pattern recognition literature the descriptive characteristics of an object are termed features or variables. Therefore $\mathbf{x} = [x_1, \ldots, x_n]^T$ where each x_j is an individual feature.

The classification methods of pattern recognition can take one of two forms, unsupervised or supervised. Within unsupervised pattern recognition clustering techniques are applied in order to find natural groupings within a given set of objects. Supervised pattern recognition assumes that the class (group) of each object is known during training. Let $\Omega = \{\omega_1, \ldots, \omega_c\}$ be a set of mutually exclusive classes. Supervised classification builds a classifier, D using the labelled objects in the training set.

$$D : \Re^n \rightarrow \Omega. \tag{1.1}$$

A discriminant function, $g_i(\mathbf{x})$, is used to denote the support for an object $\mathbf{x}$ in class ω_i given by D. The discriminant function with the maximal value determines the class assigned to the object $\mathbf{x}$. The classifier training can be governed by making the least error or minimising a cost function on a training data set.

The output of a stable classifier will not be greatly affected by small alterations in the training data. Classifiers for which the alteration of even a single training point can lead to radically different overall decisions are termed unstable. The required stability of a classifier model may affect the choices made during the process of classifier construction.

The posterior probability, which is the probability of class ω_i given the a particular object $\mathbf{x}$, can be calculated using Bayes formula,

$$P(\omega_i|\mathbf{x}) = \frac{P(\omega_i)p(\mathbf{x}|\omega_i)}{p(\mathbf{x})}, \tag{1.2}$$

where $P(\omega_i)$ is the prior probability of class ω_i, (the frequency of class ω_i amongst the N training set objects), $p(\mathbf{x}|\omega_i)$ is the class-conditional probability mass function, (the probability of $\mathbf{x}$ occurring in class ω_i) and $p(\mathbf{x})$ is the unconditional probability of $\mathbf{x}$. The minimal possible error is guaranteed by selecting the class with the maximal posterior probability.

Given an infinite training set the estimates used in equation 1.2 will be accurate. This will mean that the posterior probability estimates given by the Bayes formula will be

accurate. By selecting the class with the maximal posterior probability estimate the minimal possible error is guaranteed. Thus, in this case the Bayes formula will provide the optimal method of classification. However, with only finite training sets the estimates become less reliable. Part of applying pattern recognition methods to real world situations is to find the classifier that best fits the problem. There is no optimal classifier for every situation.

1.2 Where does Veterinary Science fit in?

The joining of two seemingly unrelated areas of science can be intriguing and challenging. For the improvement in knowledge in all areas of science, knowledge from various areas need to be combined and applied. Using classifiers and pattern recognition in veterinary science is not a new idea [21,22,50,100,136,140]. The data collected by domain experts may not be typical of that used by pattern recognition methods. The data collection process may lead to inherent assumptions. There is a need to know what effects and differences could be endured by accepting any inherent assumptions. What are the best classifiers and processes to use with untypical possibly assumption-bound data.

1.3 What are BSE and Scrapie?

BSE and Scrapie are both forms of transmissible spongiform encephalopathies (TSE), that are fatal neurodegenerative diseases with no known cure. TSE's can be characterised by a long incubation period, a clinical course of between two to six months and a lack of immune response despite the disease not actively suppressing the immune system. This section gives an overview of the two forms of TSE looked at in this study.

1.3.1 Scrapie

Scrapie was first diagnosed in Great Britain in 1732 [89]. It is a non-febrile fatal chronic disease of sheep and goats. There are around 15 different strains of Scrapie currently identified. The long incubation period is followed by the clinical onset in sheep between 2 and 5 years old. The early signs of the disease are transient and unspecific. They include weight loss and subtle behaviour changes.

Affected sheep may disassociate themselves from their flock and appear to be hyper-alert to anything differing from the normal routine. When left alone the sheep may stand with a vacant gaze and lowered head. Scrapie may cause a sheep to appear inactive and lazy in comparison to other sheep in the flock.

Weight loss can occur despite a normal appetite. Mastication and bolus regurgitation may be reduced, which can result in emaciation. Early on in the clinical presentation of the disease there may be abnormal drinking patterns, such as taking water little and often. This can be matched by abnormal "little and often" urination. Increased teeth grinding (Bruxism) and lip licking may also be witnessed.

Licking may also be attributed to the increased grooming noted in affected sheep. Most cases of Scrapie demonstrate pruritic symptoms, such as rubbing against objects, scratching, licking and self-nibbling. Intense rubbing may induce discolouration or loss of wool and eventually hyper-pigmentation or lesions of the skin. Discolouration of fleece around the mouth can be due to excessive flow of saliva (Ptyalism) as a result of swallowing difficulties.

The "scratch test" is performed by rubbing a sheep on its back. A positive result would be seen as a "nibble reflex" shown each time the test is repeated. The "nibble reflex" can be described as licking and smacking of the lips giving the sheep an expression of "satisfaction". In more advanced cases touching the back can be enough to elicit a nibble reflex. Spontaneous nibbling reflexes may also be seen in advanced cases. A positive result to the scratch test on its own is not indicative of Scrapie and must be considered in conjunction with other clinical symptoms.

Infected sheep can present a wide-based stance or stand with a crouching position of the hindquarters. Gait abnormalities (Ataxia) usually occur as the disease progresses, initially the gait appears stiff. Bunny hopping of the hind limbs when the sheep is made to run or a high stepping gait (Hypermetria) may be seen. These gait deficits will eventually lead to difficulty in getting up and recumbency, (lying with hind legs stretched out behind).

Head tremors can progress into whole body tremors. These can be exaggerated by excitement or handling. This can eventually lead to collapse and seizures in more advanced cases. Ocular abnormalities are rarer in Scrapie but can occur as Nystagmus or visual impairment.

Biochemical markers for Scrapie have been found in blood neurotransmitters, hormones, metabolites and immunoglobulin. These have not resulted in a ante-mortem test for the presence of the disease as yet. There is also the possibility of using electroencephalographic tests (EEG) to aid in diagnosis [131].

Various diseases can be considered as differential diagnoses these include Mange, Bacterial dermatitis, Pregnancy toxaemia, hepatic encephalopathy, various toxins and parasite migration.

1.3.2 BSE

Bovine Spongiform Encephalopathy (BSE) was first described in Great Britain in 1986. It is an afebrile neurological disease that results in fatality. BSE remains a sporadic disease and as such the awareness of the clinical signs is low. The early signs are unspecific in nature and so many cases get presented when the case is acute [28]. BSE naturally progresses slowly with a long incubation period of $2\frac{1}{2}$ to 8 years, [115]. The usual onset of the disease is in cattle between 4 and 6 years of age. The clinical signs of BSE are variable from day to day but progressive over time. The risk of missing early signs can be reduced if the animals with early non-specific signs are challenged in such a way as to exaggerate the suspected signs. In stressed cows the signs tend to become more apparent. Weight loss or loss of condition can be the first apparent signs but these can be attributed to many other diseases and events.

Some of the more common signs are changes in behaviour and movement. Affected cows may become apprehensive of herd mates, human visitors to the herd, changes to its environment and being restrained. Some cases present an intense stare, known as "the BSE look" which can affect the ocular structures. Affected cows may become unpredictable kicking out during handling or milking. This change in behaviour has led many cows to be sent to slaughter as they become more difficult to handle. This in turn led to many cases going undiagnosed.

Hyperaesthesia, a hyper reactivity to certain environmental cues or external stimuli may be induced by touching the head or neck, thus inducing an exaggerated reaction. Unaffected cattle will normally react the first few times but then become used to the stimulus. However, repeated stimulation in extremely affected cases can result in collapse or seizure. Touching the top of the tail may induce a calming effect. Other changes in behaviour can be demonstrated by head bobbing, body tremors, excessive nose licking, bruxism, nose wrinkling or yawning. Self inflicted skin lesions (Pruritus) can also occur but are less common than in cases of Scrapie. During the periods of apprehension/ excitement the heart rate of the cow may remain low despite its apparent agitated state.

In early cases of BSE the change in movement may present as a refusal to walk or run and any movement appearing "stiff". In more advanced cases this progresses to a wide based stance, low head carriage, and standing with the rear limbs placed under the abdomen. There could also be difficulty in getting up from lying with a period of sitting like a dog in the process. Ataxia is normally first noticed in the hindquarters. As the disease progresses this becomes more severe resulting in stumbling, slipping and falling. Knuckling of fetlocks may also be seen but is rarer. These progressive gait deficits may lead to recumbency. DEFRA states that unless another cause for the recumbency can be clearly established the case should be considered as BSE [28].

BSE cases may also have a lower milk yield, another cause of early slaughter before confirmed diagnosis. There is the possibility that there will be a decreased reaction to Xylazine, a drug that normally induces sedation in unaffected cattle.

There are certain differential diagnoses that have to be considered when these clinical signs are present. These can be extraneural or neural. The extraneural diseases include Ectoparasitism, Slow wasting diseases, Ocular disease, Ovarian cysts. The neural diseases include Hypomagnesaemia, Rabies, Lead poisoning, Listeriosis and Hepatic encephalopathy.

A normal temperature will normally be recorded. Respiration rates may be slightly increased. There are no reported changes in the haematology, biochemistry or cytology of the blood, urine or cerebrospinal fluid. It has been noted that there may be changes in energy metabolism and levels of lactic acid or amino acids in the blood. As with Scrapie, abnormal EEG recordings can be reported.

There is as yet no specific ante-mortem test. Diagnosis is confirmed by testing for detection of the active agent in brain tissue post mortem.

1.3.3 The spread and effects of BSE and Scrapie.

The true incidence of Scrapie in Great Britain is unknown but a survey suggested that around $\frac{1}{3}$ of flocks have been affected [115]. It became a notifiable disease in Great Britain in 1991. It is believed that Scrapie was introduced to the USA in 1947. Between 1947 and 1992 657 flocks were affected in 39 states.

The prion protein PrP^{Sc} is the infective agent associated with Scrapie. It is thought that the incubation period may be determined by genetics of the host. Introducing preclinically infected sheep to the flock is thought to spread the disease. The placenta, foetal fluids, intestine and nasal mucous membranes are known to harbour the disease. Ingestion of infected materials can induce the spread of the disease. Hay mites have also been shown to harbour the infective agent. The agent is very resistant to heat and UV. Infected brain homogenates buried in soil remain infective for three years [115]. Fields that infected sheep have grazed on may also remain infective for three years. The importance of vertical spread from parent to lamb still remains to be determined. If both parents are infected then the risk to the lamb of infection seems to be greater but this may due to horizontal transmission at birth.

Countries in which Scrapie was not enzootic but introduced by import have had some success with eradication schemes. Eradication schemes have not been so successful in countries where Scrapie is enzootic.

The origin of BSE is generally regarded to be transformation from Scrapie. BSE was first reported in Great Britain in 1986 but it may have been a sporadic disease before

this with mis-diagnosis occurring. Prior to this 1986 case the rendering process for the processing of meat and bone meal had undergone a change. Sheep-derived protein was being used in feed for cattle. This mass exposure of the infective Scrapie agent to cattle is thought to have caused the BSE epizootic in the late 1980's and early 1990's in Great Britain.

In 1988 BSE was declared a notifiable disease. There was also a statutory ban on the feeding of ruminant-derived protein to ruminants. The annual incidence of BSE peaked in 1992 and has fallen every year since. This has been attributed to feed ban. By 1996, 59.3% of dairy herds in Great Britain had experienced at least one case. 15.3% of beef herds had experienced a case. This difference is mainly due to the differences in feeding practices and life span of the two types of herd.

By 1998 there had been 170,000 cases of BSE in Great Britain. At the peak of the outbreak there were 1000 new cases being submitted each week. Today there are around ten new cases submitted each week. The first case of BSE in the USA was diagnosed in December 2003.

The risk of BSE increases as the size of the herd increases but the horizontal and vertical transmission of the disease is not thought to be of major importance. This is because the within herd incidence of the disease is low. This meant the economic importance to the individual farmer of the outbreak was not great. Compensation payments were made for all cows that were slaughtered. In 1989 certain beef products were banned from entering the human food chain. The national cost in Great Britain was felt by the measures needed for detection, control, compensation, disposal and the loss of the export market. British beef is still banned in about 100 countries including the USA and Australia, [27].

One of the biggest costs was the fear of the link between BSE and a human TSE, variant-Creutzfeldt-Jakob disease, (vCJD). CJD was first discovered in 1920 in Germany. CJD generally has a long incubation period with clinical onset between 50 and 75 years of age. The clinical symptoms include a rapid onset of dementia. The majority of cases are sporadic but there are cases of familial occurrence due to genetic mutation and also a very small percentage due to man to man transmission during surgery. In 1996 there was the discovery of a different form of CJD. This form of CJD differed from the already present strains due to the early onset of the clinical symptoms; in patients under the age of 50. This strain was named vCJD and was attributed to eating beef products before the ban was introduced in 1989. As yet there has been no link between Scrapie and human TSE's.

BSE has been found in other species especially zoo animals that were fed fallen stock. Domestic cats were found to have the disease in 1990 [115]. In 2004 it was reported that BSE had been found in a goat that had been diagnosed with Scrapie 15 years ago. As yet

BSE has not been found in sheep. Sheep that have been experimentally infected with BSE have had a more rapid progression of the disease. The clinical signs have been similar to that of Scrapie but with less frequent occurrences of pruritus and a greater incidence of ataxia.

Other current research into BSE and Scrapie includes looking at the role of the PrP genotype in susceptibility to the disease, studies of epidemiology and into the modes of transmission and diagnosis.

1.4 Cross - referencing: Pattern recognition to veterinary domain

Here the terminology used in pattern recognition literature is paired to that used in the veterinary domain. Table 1.1 shows the parallel between the notions from the two areas and also the notations used throughout this study.

Table 1.1: A cross reference of pattern recognition notation to veterinary terminology.

Pattern Recognition context	**Notation**	**Veterinary context**
Features	x_j	Symptoms / Clinical signs
Classes	$\Omega = \{\omega_1, \ldots, \omega_c\}$	Diseases
Prior Probability	$P(\omega_i)$	Prevalence / Pre-test probability
A feature set	X	The set of all signs
A feature vector	$\mathbf{x}$	The collection of signs for a particular animal
The training set	N	The set of recorded cases
Classification algorithm (Classifier)	$D : \Re^n \rightarrow \Omega$	Diagnostic process Stored template / Standard / Pattern matching
Discriminant function	$g_i(\mathbf{x})$	Diagnostic evidence for a disease
Posterior probability	$P(\omega_i \mid \mathbf{x})$	Probability of a disease explaining a set of signs. Hypothetical probability / Post-test probability
Class conditional probability mass function	$p(\mathbf{x} \mid \omega_i)$	Frequency of occurrence of a set of clinical signs observed within a disease

1.5 BSE and Scrapie data

Traditionally data for supervised classification tasks is held as a set of recorded cases each labelled with their outcome class. These cases are then used to train the classifier, D. The more training data there is available the more accurate the resulting classifier. However, for medical data there are many reasons as to why traditional data sets for diseases are not

Table 1.2: Structure of the non-traditional probability table data.

	ω_1	...	ω_i	...	ω_c
x_1	$P(x_1\|\omega_1)$	...	$P(x_1\|\omega_i)$	...	$P(x_1\|\omega_c)$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
x_j	$P(x_j\|\omega_1)$	...	$P(x_j\|\omega_i)$	...	$P(x_j\|\omega_c)$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
x_n	$P(x_n\|\omega_1)$	...	$P(x_n\|\omega_i)$	...	$P(x_n\|\omega_c)$

plentiful. These reasons include not many cases being recorded for rare diseases, uncertainty about which symptoms to record, data collection not being performed uniformly across the population and recorded cases generally being suspected cases of a particular disease leading to low variability in the symptoms recorded overall in the data.

1.5.1 Non-traditional data - Probability Tables

A method to overcome the problems with traditionally stored data is to use what we will refer to as non-traditional data. For BSE and Scrapie there exists expert-estimated probability tables. An example of the probability table structure is shown in Table 1.2. Entry (j, i) is the estimated probability of symptom (feature) x_j being present given the presence of a certain disease (class) ω_i. Three domain experts were asked to estimate these probabilities. The three estimates were then averaged to give the estimates in the probability tables.

Construction of the probability tables using the experts' estimates led to two assumptions,

1. Assumption of independence. As no consideration was given to groups of features indicating specific diseases it must be assumed that the features are class-conditionally independent.

2. Reliability of the probability estimates. Since there is no way to validate or calibrate the expert estimates, it must be assumed that the probabilities represent the true values or very close estimates thereof.

The BSE probability table contains the conditional probabilities for 242 features given 57 diseases (BSE and 56 alternative diagnoses). The Scrapie probability table contains the conditional probabilities of 285 features given 63 classes (Scrapie and 62 alternative diagnoses). Both tables were provided courtesy of Dr. Peter Cockcroft, Cambridge veterinary School, University of Cambridge, UK.

The classification task has various perspectives here. For example, we may wish to separate the main disease of interest (BSE or Scrapie) from *any* other disease, this will

be called "the two-class problem". This problem is constructed by taking the average probability estimate of the alternative diseases. The prior probabilities are taken to be equivalent for the two classes, i.e., $P(\text{BSE}) = P(\text{Non-BSE}) = 0.5$. The probability of a feature x_j given class BSE can be taken from the probability tables, $P(x_j = 1|\text{BSE})$ is the value taken directly from the table while $P(x_j = 0|\text{BSE}) = 1 - P(x_j = 1|\text{BSE})$. The probability for feature x_j in the second class, Non-BSE, is calculated from the probability tables by taking the average of the remaining alternative diseases, $P(x_j|\text{Non-BSE}) = \frac{1}{c-1}\sum_{\omega_i \neq \text{BSE}} P(x_j|\omega_i)$.

Alternatively, we may seek to separate any of the diseases from the remaining diseases, "the multi-class problem". In this task we would seek to use the probability tables as they are. As there is no other information about prevalence, it must be assumed that the prior probabilities are equal, $P(\omega_i) = \frac{1}{c},\ i = 1, \ldots, c$.

1.5.2 Traditional data - Recorded Cases.

Traditional data sets of recorded cases of BSE and Scrapie have been sourced from DEFRA (Department for Environment, Food and Rural Affairs). These traditional datasets contained recorded cases described by various symptoms. Unfortunately, the symptoms in the traditional data and the ones in the probability table data were not the same.

The BSE data contained 204,354 cases described by 31 features courtesy of Judi Ryan, Veterinary Laboratories Agency. The set was divided into 173,759 BSE positive cases and 30,595 BSE negative cases. This gives estimates of the prior probabilities of 85% class BSE positive and 15% class BSE negative among the cases reported to DEFRA.

The Scrapie data contained 3676 cases described by 41 features courtesy of Dr. Victor Del Rio Vilas, Veterinary Laboratories Agency. This was split into 2987 cases of Scrapie positive and 689 cases of Scrapie negative. The prior probabilities for the two classes are 81% for class Scrapie positive and 19% for class Scrapie negative among the cases reported to DEFRA.

The high imbalance of the prior probabilities for the two classes in both data sets is caused by all the cases being suspected of the disease. Also because of this the probability estimates of the feature frequencies would not be representative of the entire population. The scope of the task is therefore reduced to diagnosis of a disease within a set of suspects rather than an entire population.

1.5.3 Error estimation: sensitivity and specificity.

In medicine there are two special types of probabilities, Sensitivity and Specificity [22]. Sensitivity is defined to be the likelihood of a positive result in the patients known to have the disease. Specificity is defined to be the likelihood of a negative result in patients known to be free of the disease. Sensitivity and specificity are inversely related to one another [23]. High specificity means the test being used rarely gives a positive result for the disease of interest in its absence. However, high specificity comes at the expense of low sensitivity which means that the same test would produce a lot of false negatives. There is a need to decide on the balance between sensitivity and specificity. A high specificity / low sensitivity test means fewer animals will be slaughtered unnecessarily but a high number of cases of BSE and Scrapie will be mis-diagnosed. A high sensitivity / low specificity test could result in an unacceptably high number of animals destroyed due to false positives.

Brenner and Gefeller [14] note that sensitivity and specificity of binary diagnostic tests are dependent on disease prevalence contrary to many practioners' beliefs. As disease prevalence increases sensitivity increases and specificity naturally decreases. For example, a diagnostic test used in a clinical environment among patients suspected to have a certain disease will typically have lower sensitivity and higher specificity when applied as a screening tool in the general population where the disease prevalence would be lower.

Sensitivity and specificity may be expressed in the following way,

$$\text{Sensitivity} = \frac{\text{Detected Positive}}{\text{All Positive}},$$

$$\text{Specificity} = \frac{\text{Detected Negative}}{\text{All Negative}}.$$

For a hypothetical data set with $N = tp + fp + fn + tn$ elements, the table below shows the numbers of Guessed positive cases and Guessed negative cases with respect to the actual true values of positive and negative cases ($tp, fp, fn, tn \in Z^+$).

	True +	True -
Guessed +	tp	fp
Guessed -	fn	tn

where tp is a true positive case, a BSE case diagnosed as BSE, fp is a false positive, a non-BSE case diagnosed as BSE, fn is a false negative, a BSE case diagnosed as non-BSE and tn is a true negative case, a non-BSE case diagnosed as such. Then,

$$\text{Sensitivity} = \frac{tp}{tp + fn}$$

and

$$\text{Specificity} = \frac{fn}{fp+tn}$$

The classification error of the diagnostics test that can be calculated as

$$\begin{aligned} \text{Error} &= \frac{fp+fn}{tp+fp+fn+tn} \\ &= \frac{fp}{tp+fp+fn+tn} + \frac{fn}{tp+fp+fn+tn} \\ &= error_+ + error_- \end{aligned} \tag{1.3}$$

where $error_+$ is the proportion of false positives and $error_-$ is the proportion of false negatives. In our case false positives are cases diagnosed as the disease of interest (BSE or Scrapie) when in fact they are something different. False negatives are cases diagnosed as something else when they are actually BSE or Scrapie.

1.6 The Naïve Bayes Classifier

As the Naïve Bayes classifier is optimal for the case when features are class-conditional independent it is suitable for non-traditional data. The Naïve Bayes classifier is employed throughout this study with a deeper analysis undertaken from Chapter 3 onwards. A brief introduction to the model is given here and will be expanded upon in later sections.

Let $\mathbf{x} = [x_1, \ldots, x_n]^T$ be a vector in the feature space $\Re^n$. We assume that $x_1, \ldots, x_n$ are mutually class-conditionally independent. Then the class conditional probability mass function for class $\omega_i, i = 1, \ldots, c$, becomes

$$P(\mathbf{x}|\omega_i) = \prod_{j=1}^{n} P(x_j|\omega_i).$$

The discriminant functions of the classifier guaranteeing the minimum classification error can be taken to be $g_i(\mathbf{x}) = P(\omega_i)P(\mathbf{x}|\omega_i)$. When there are only two classes only one discriminant function is needed; the ratio of $g_1(\mathbf{x})$ and $g_2(\mathbf{x})$ can be used. If the logarithm of this ratio is positive, then $g_1(\mathbf{x}) > g_2(\mathbf{x})$, and we should label $\mathbf{x}$ in class ω_1. If the logarithm is negative then $g_2(\mathbf{x}) > g_1(\mathbf{x})$, and class ω_2 should be assigned. If $g_1(\mathbf{x}) = g_2(\mathbf{x})$, then the logarithm of the ratio is 0, so any of the two class labels can be assigned.

1.7 Aims of the thesis

- To apply pattern recognition methods to the non-traditional probability table data.
- To select a possible feature set for data collection on BSE and Scrapie.
- To analyse various classifiers and their performance with regard to non-traditional data.
- To analyse the effect of the assumption of class-conditional independence on the Naïve Bayes classifier

The main aim of the thesis is to investigate pattern recognition processes with regard to the non-traditional estimated probability table data. We want to discover what is possible with such data. The traditional recorded case data from DEFRA allows the supplementary investigation of the possibilities available. In general, the investigations focus on the ideas of using the non-traditional data in place of the traditional data.

1.8 Organisation of the thesis

To achieve the aims outlined above the thesis is structured as follows:

- Chapter 2 studies the effects on various feature selection methods by using non-traditional probability table data.
- Chapter 3 looks at the classification of non-traditional probability table data using decision trees. It also looks at the structure of the Naïve Bayes classifier.
- Chapter 4 gives an insight into the theoretical errors made by Naïve Bayes on non-traditional data due to using the independence assumption.
- Chapter 5 provides an empirical analysis of the performance of various classifiers when using data with binary features.
- Chapter 6 gives the summary and conclusions that can be drawn overall from this study. It also indicates the possibility of future work in this area.

Table 1.3 gives an outline of the type of data the investigations are based on in each section.

Figure 1.1 gives an idea of the structure of the thesis and the links between the chapters. The relevant literature is reviewed at the beginning of the relevant section due to the wide variety of literature that has been needed for this study.

Table 1.3: Indication of the type of data used in each investigation

Chapter	Investigation	Data type
2	Feature Selection	Non-traditional probability tables
3	Classification	Non-traditional probability tables
4	Errors made by Naïve Bayes	Non-traditional probability tables
5	Performance of Naïve Bayes	Traditional recorded case

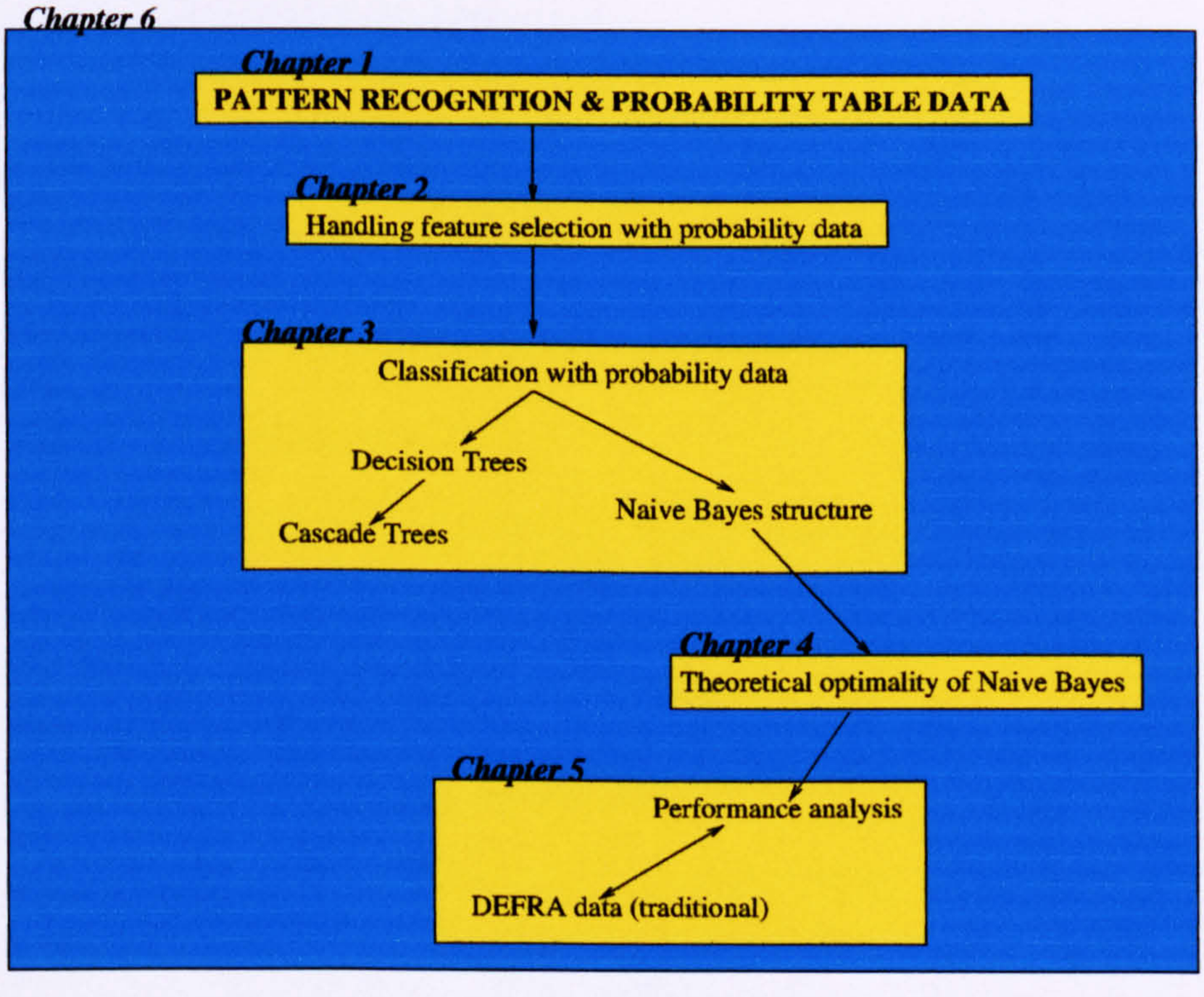


Figure 1.1: Chart indicating the structure of the thesis.

Chapter 2

Feature Selection for probability table data

2.1 What is Feature Selection?

Feature selection is one of the earliest ideas discussed in pattern recognition literature. The aim of feature selection using traditional data is to reduce the dimensionality of the data with a view to reduce the complexity of the problem.

A feature selection algorithm chooses a subset of cardinality d from the original n features, where $d < n$. The methods try to find the most important or relevant d features for each problem.

A study by Dash and Liu [26] separates the feature selection process into 4 steps. First is a generation step, where the next candidate subset is created. Second is an evaluation step, where the generated subset is evaluated using the criterion function.Third is a stopping stage, where the decision is made to either terminate the process with the current subset or continue the generation of feature subsets. Finally, a validation of the chosen subset is made. The stages of the feature selection process, subset generation and evaluation may be broken down as follows,

1. Choose a criterion function, $J(S)$.
2. Pick a subset S of the original set of features X.
3. Build a classifier with the candidate subset S.
4. Calculate $J(S)$.
5. Repeat with various subsets $S \subset X$.
6. Select $S^{\star}$ which minimises $J(S)$.

Generation procedures are used to create the candidate feature subsets. The search methods for the candidate subsets fall into three categories

1. Complete methods - Exhaustive search, Branch and Bound. Every possible candidate subset is generated and checked or eliminated. These methods are the only guaranteed way of finding an optimal subset.
2. Heuristic methods - Sequential search methods. These algorithms use the most appropriate solution found in previous steps to guide the next subset selection. Heuristic methods cannot guarantee to find the optimal set.
3. Random methods - Genetic Algorithms. As the name suggests, the methods generate the candidate subsets randomly. The methods can have an element of guidance but allow "jumps" (mutation) in the logic to search alternative areas of the feature space. The random approach cannot guarantee the discovery of the optimal subset.

Though the complete methods guarantee the desirable result of discovery of the optimal subset, the computational complexity required to implement such methods is usually much too high a price to pay. Heuristic and random methods are not as computationally expensive and thus implemented much more frequently, despite not having the guarantee of optimality.

Evaluation of the subsets is performed by the criterion function. Good evaluation of the subsets will aid in the generation procedure especially in the case of heuristic selection methods. The proposed subset with the "best" score for the chosen criterion function is the subset retained at the end of the process. The type of evaluation function used splits feature selection methods into two approaches, Wrapper or Filter. Wrapper methods use a direct measure of the chosen classifier performance with the candidate subset. The use of the classifier directly assesses the subset for the particular problem. Filter methods use an indirect measure for the evaluation of a candidate subset. This could be a measure of the dependency of the features within the subset or a separability measure of the classes given the candidate features. Dash and Liu [26] describe the criterion function or evaluation function as being one of five types - distance measure, information measure, dependency measure, consistency measures and finally classifier error. The first four measures can be considered as filter feature selection whilst classifier error is wrapper feature selection. Filter methods can be simpler and quicker to implement than wrapper methods. A study by Aha and Bankert, [2] focuses on the differences and effects of wrapper feature selection methods compared to filter feature selection methods. The study offers evidence for and concludes that wrapper methods are generally more successful than filter methods. This is attributed to wrapper methods using a direct measure of the performance of the characteristic of interest, whilst filter methods utilise an indirect measure. Blum and Langley [12] also discuss wrapper and filter methods at length. The computational cost of a wrapper method is outlined as a "major disadvantage" to the methods. Filter methods are highlighted to allow the selection of minimal combinations of features that can discriminate perfectly amongst the classes.

A study by Pudil and Novovicova [114] looked to present some guidelines on the method of feature selection to choose based on the knowledge of the problem needing to be solved. A preliminary flowchart is built indicating the methods of feature selection to choose based on the characteristics of the problem. For example, if the total number of features is greater than 30 sequential feature selection methods are recommended otherwise a branch and bound search is suggested.

Jain and Zongker [62] evaluate different feature selection methods, looking specifically at their advantages and disadvantages for particular problems. The experiments conducted in the study demonstrated the existence of the curse of dimensionality, also

known as Hughes paradox or the peaking phenomenon. For a feature selection algorithm there appears to be an optimal number of features that can be selected. Adding more features causes the classification error to rise. This effect seems counterintuitive. The more information about a problem is used, fewer mistakes should be expected. This effect has been attributed to the fact that traditional data sets are finite in size and, as such, only imperfect estimates of probability distributions may be found.

Kudo and Sklansky [75] also compare feature selection algorithms for classifiers. The study incorporates a comparison of branch and bound methods, sequential algorithms and genetic algorithms on a variety of small, medium and large data sets. In conclusion it is seen that the sequential algorithms can give better results than the other methods for the small and medium sized data sets.

An optimal feature selection method cannot be improved in terms of accuracy but the time complexity leaves a lot to be desired. An improved branch and bound method, (IBAB) proposed by Chen, [19] aims to reduce the search time that the conventional branch and bound method usually requires. Partial paths, which are subpaths of branch and bound paths, are searched for. If a partial path is found such that its criterion function value is less than the current stored best for partial paths then all full paths containing this partial path are ignored. However, by reducing the time taken to perform the branch and bound search optimality is compromised.

The balance between time, complexity and accuracy is difficult to observe. Many problems naturally prevent the use of optimal methods. The improvement of non-optimal methods in terms of accuracy is therefore paramount.

Feature selection literature abounds with novel attempts at new and modified approaches of feature selection. Attempts have been made at removing all noise and irrelevant features [85], new measures of discrimination [65], assessing the relevance of each feature with regard to each class [138], redefining the problem by changing the formulation into a linear programming problem [5], or redefining the concept of classifiability based on the overlap of the patterns of opposite classes [34].

Combination of current feature selection methods to exploit their perceived strengths has also been used. An approach proposed by Murphey and Guo [103], uses a hybrid statistical algorithm. The method has two major steps; the first is to rank all the features based on statistics found from the data sample. The second step selects features based on their rank and their performance with a Bayesian expectation-maximisation classifier. Another hybrid statistical method proposed by Bins and Draper, [10] is based on the strengths of three steps. Step one removes all the irrelevant features (features that do not contribute anything to the task), step two removes all the redundant features (features that duplicate information given by other features) and step three selects the subset of the

required size from the remaining features.

If feature selection is based directly on the classification error and not on an auxiliary measure such as correlation then an algorithm must base its decisions on estimates of the error. For large data sets the obtained error estimates are expected to be good thus reducing this problem. However, for small sample data sets the error estimation can be problematic and so the performance of the error estimator will have an impact on the feature selection algorithm. The choice of the error estimator for feature selection in this small sample situation can make more of a difference than the choice of feature selection algorithm according to a study by Sima *et al.*, [135].

As there can be no one optimal method of feature selection for every problem, the quest is to find the method that applies best to the problem at hand. A new addition to the large feature selection family may be required in response to a particular problem.

2.2 Independent binary features in feature selection

Bressan and Vitria look at the ideas behind the selection and classification of independent features [16]. The study looks at using statistically independent features and their effect on classification.

For n independent binary features the class conditional probability mass function for class $\omega_j, j = 1, \ldots, c$ becomes

$$P(\mathbf{x}|\omega_i) = \prod_{j=1}^{n} P(x_j|\omega_i) \tag{2.1}$$

The discriminant functions of a classifier, $g_i(\mathbf{x}) = P(\omega_i)P(\mathbf{x}|\omega_i)$, guarantees the minimum possible classification error. With only two classes in a problem, only one discriminant function is needed as the ratio of the two discriminant functions may be taken into account. If the logarithm of the ratio is positive then $g_1(\mathbf{x}) > g_2(\mathbf{x})$, and so the object will be classified to class ω_1. A negative logarithm indicates the classification of $\mathbf{x}$ into class ω_2. A logarithm of zero indicates that the discriminant functions have equal value and therefore the case may be classified into either class with equal classification error. Let p_j denote $P(x_j = 1|\omega_1)$ and q_j denote $P(x_j = 1|\omega_2)$. The class conditional p.m.f for feature x_j in class ω_1 will then take the form $P(x_j|\omega_1) = {p_j}^{x_j}(1-p_j)^{(1-x_j)}$. Similarly the class-conditional p.m.f for feature x_j in class ω_2 becomes $P(x_j|\omega_2) = {q_j}^{x_j}(1-q_j)^{(1-x_j)}$ Then a discriminant function for a two class problem with binary features can be formulated as

$$g(\mathbf{x}) = \log \frac{g_1(\mathbf{x})}{g_2(\mathbf{x})}$$

$$
\begin{aligned}
&= \log\left[\left(\frac{P(\omega_1)}{P(\omega_2)}\right)\left(\frac{P(\mathbf{x}|\omega_1)}{P(\mathbf{x}|\omega_2)}\right)\right] \\
&= \log\frac{P(\omega_1)}{P(\omega_2)} + \log\frac{P(\mathbf{x}|\omega_1)}{P(\mathbf{x}|\omega_2)} \\
&= \log\frac{P(\omega_1)}{P(\omega_2)} + \log\left[\prod_{j=1}^{n}\left(\frac{{p_j}^{x_j}}{{q_j}^{x_j}}\right)\left(\frac{(1-p_j)^{(1-x_j)}}{(1-p_j)^{(1-x_j)}}\right)\right] \\
&= \log\frac{P(\omega_1)}{P(\omega_2)} + \sum_{j=1}^{n} x_j \log\left(\frac{p_j}{q_j}\right) + \sum_{j=1}^{n}(1-x_j)\log\left(\frac{1-p_j}{1-q_j}\right) \\
&= \log\frac{P(\omega_1)}{P(\omega_2)} + \sum_{j=1}^{n}\log\left(\frac{1-p_j}{1-q_j}\right) + \sum_{j=1}^{n} x_j \log\left(\frac{p_j(1-q_j)}{q_j(1-p_j)}\right) \qquad (2.2)
\end{aligned}
$$

So

$$g(\mathbf{x}) = W_0 + \sum_{j=1}^{n} W_j x_j \qquad (2.3)$$

where

$$W_0 = \log\frac{P(\omega_1)}{P(\omega_2)} + \sum_{j=1}^{n}\log\left(\frac{1-p_j}{1-q_j}\right)$$

and

$$W_j = \log\left(\frac{p_j(1-q_j)}{q_j(1-p_j)}\right), j = 1, \ldots, n.$$

If there are c classes in the problem then there will be c discriminant functions $g_i(\mathbf{x})$, $i = 1, 2, \ldots, c$, of the form

$$g_i(\mathbf{x}) = \log P(\omega_i)P(\mathbf{x}|\omega_i)$$

with

$$P(\mathbf{x}|\omega_i) = \prod_{j=1}^{n} P(x_j|\omega_i)$$

Then

$$
\begin{aligned}
g_i(\mathbf{x}) &= \log P(\omega_i) + \sum_{j=1}^{n}\log\left[P(x_j = 1|\omega_i)^{x_j}(1 - P(x_j = 1|\omega_i))^{(1-x_j)}\right] \\
&= \log P(\omega_i) + \sum_{j=1}^{n} x_j \log P(x_j = 1|\omega_i) + \sum_{j=1}^{n}(1-x_j)\log(1 - P(x_j|\omega_i)) \\
&= \log P(\omega_i) + \sum_{j=1}^{n}\log(1 - P(x_j = 1|\omega_i)) + \sum_{j=1}^{n} x_j \log\frac{P(x_j = 1|\omega_i)}{1 - P(x_j = 1|\omega_i)} \qquad (2.4)
\end{aligned}
$$

Equation (2.4) can be rewritten as

$$g_i(\mathbf{x}) = \sum_{j=1}^{n} W_{i,j} x_j + W_{0,i} \tag{2.5}$$

where

$$W_{i,j} = \log \frac{P(x_j = 1|\omega_i)}{1 - P(x_j = 1|\omega_i)}$$

and

$$W_{0,i} = \log P(\omega_i) + \sum_{j=1}^{n} \log(1 - P(x_j = 1|\omega_i))$$

In both cases the linear discriminant functions (2.3) and (2.5) guarantee the minimum possible error across the whole feature space $\{0, 1\}^n$.

A study by Elashoff *et al* [39], showed that the best set of d independent binary features did not always consist of the best d single features. This result is expanded upon by Toussaint, [144]. Toussaint gives an example where the best pair of independent binary features need not contain the best single feature.

These two results are compounded by a further result by Cover, [25]. This result shows that two independent copies of the "worst" event may provide better results than that of the individually best event.

Elashoff *et al* [39] indicate that using a stepwise selection rule is always at least as good as choosing the two best single features, leading us to look in the direction of sequential feature selection.

2.3 Sequential feature selection from probability data

Classical feature selection techniques assume the use of a traditional data set. A study by Sima *et al.* [135] demonstrates that sequential feature selection can perform close to optimal when true error is employed as the evaluation criterion. It is well known that knowledge of the true error in reality is impossible. However, larger samples will provide better error estimation and therefore allow better performances. Schulerud and Albregtsen [130] indicate that medical data is often represented with a large number of features but only a small collection of cases, thus affecting the knowledge of the true error.

The choice of the error estimator and the amount of data will greatly affect the chances of success or failure. However, with probability table data the problem of sample size is avoided as we work directly with the estimated probabilities that are provided. The assumption is now that the probabilities provided are correct.

By using non-traditional probability data we already have a model of the probability distribution for the features across the given classes and as such have to accept this as true. The optimal set of features in this case would be the complete set of features and as such feature selection for non-traditional data has a different motivation.

In our case the motivation for feature selection is set by outside factors determined by the domain experts. The non-traditional data contains probability estimates for more than 200 clinical symptoms. It would be unrealistic to collect traditional data over such a large number of features out in the field. Feature selection is employed to reduce the number of features to a "reasonable" figure. "Reasonable" may be defined by considering that the collection of data on each feature costs a given amount. If there is only limited funds allowing data collection across five features then we would want the five most discriminating features for the task.

Sequential forward selection (SFS) and sequential backward selection (SBS) are non-optimal heuristic methods of feature selection. SFS starts with an empty subset. The value of the criterion function is calculated for each individual feature; the feature with the best value is added to the subset. The next step adds each remaining feature temporarily into the subset, the criterion function is calculated for the subset before the feature is removed and the next feature tried. Once all remaining features have been tried the feature with the best value for the criterion function is added to the subset. This procedure is repeated until d features have been selected. Sequential Backward Selection (SBS) starts with all the features in the subset and tries to remove one at a time to find the best subset at each step. This removal is repeated until d features are left. The time complexity of SBS is much greater than SFS, especially with large n and small d, and is therefore not always preferable despite indications that it can be more successful [2, 31].

One of the main drawbacks of the sequential selection processes is that it creates nested sets. Once a feature has been added or removed it is not considered again.

At the first step of SFS the individually best feature is placed into the subset. This choice is never reconsidered, but from the result by Toussaint [144] it is known that this feature may not occur in the best pair of features. In SFS the features are selected purely in conjunction with those features already selected. So even during the first few steps of the algorithm we are running the risk of stepping away from the optimal subset. Thus sequential methods do not guarantee to find the optimal feature subset.

In this thesis the potentials, limitations and stability of SFS for non-traditional data are considered. The following questions need to be answered

1. Is SFS monotone on the number of features? That is, if subset S contains k features then can it be claimed that by adding feature x_{k+1} to the subset the value of the criterion function will be equal to or better than the value of the criterion function

for the subset with k features.

2. Is the error reduction monotone? Is the largest drop in error seen by adding in the first feature, followed by successively smaller error drops as each feature is added in?

3. How reliable are the results? Is the set of features sensitive to small changes in the probability estimates given by the experts?

2.3.1 Is SFS monotone on the number of features?

This question is equivalent to the following question: Does the peaking phenomenon occur also for non-traditional data? For the non-traditional data the finite data is replaced by estimates of the probability distributions of the features, assuming class-conditional independence. The following proposition is accepted as "folklore" within pattern recognition texts and as such holds for non-traditional data as well. A novel proof is given for completeness.

Proposition 1 *The theoretical error of the Bayes classifier does not increase when the feature set is augmented.*

Proof. We show that this holds for the case of independent binary features and two classes. Suppose that k features have already been selected, creating a feature space F with 2^k elements. Let $\mathbf{x}^{(k)}$ be the feature vector with the k features selected so far. Denote

$$\begin{aligned} a &= P(\omega_1)P(\mathbf{x}^{(k)}|\omega_1) \\ b &= P(\omega_2)P(\mathbf{x}^{(k)}|\omega_2) \\ c &= P(x_{k+1}=1|\omega_1) \\ d &= P(x_{k+1}=1|\omega_2) \end{aligned}$$

The error for $\mathbf{x}^{(k)}$ is $e(\mathbf{x}^{(k)}) = \min\{a, b\}$. By adding in a new feature, x_{k+1}, every $\mathbf{x}^{(k)} \in F$ is replaced by two new elements, $[\mathbf{x}^{(k)}, 0]^T$ and $[\mathbf{x}^{(k)}, 1]^T$, thereby doubling the number of elements in F. The error for $\mathbf{x}^{(k+1)}$ is

$$e(\mathbf{x}^{(k+1)}) = \min\{ac, bd\} + \min\{a(1-c), b(1-d)\}. \tag{2.6}$$

The reduction of the error for $\mathbf{x}^{(k)}$ is

$$\begin{aligned} \Delta(x_{k+1}) &= e(\mathbf{x}^{(k)}) - e(\mathbf{x}^{(k+1)}) && (2.7) \\ &= \min\{a, b\} - (\min\{ac, bd\} + \min\{a(1-c), b(1-d)\}) && (2.8) \end{aligned}$$

Using the representation $\min\{f,g\} = \frac{1}{2}(f+g-|f-g|)$, the following expression is arrived at

$$\Delta(x_{k+1}) = \frac{1}{2}\left(|\underbrace{(a-b)}_{A} - \underbrace{(ac-bd)}_{B}| - |\underbrace{a-b}_{A}| + |\underbrace{ac-bd}_{B}| \right) \tag{2.9}$$

Noticing that $|A| = |A-B+B| \leq |A-B| + |B|$, hence $|A-B| \geq |A| - |B|$, then

$$e(\mathbf{x}^{(k)}) - e([\mathbf{x}^{(k)}, x_{k+1}]) = \Delta(x_{k+1}) \geq 0. \tag{2.10}$$

This holds for every $\mathbf{x}^{(k)} \in F$. ■

For the case of the non-traditional data SFS is indeed monotone on the number of features.

Duin *et al* [37] derive the conditions under which the addition of a new feature does not decrease the error. A threshold of the probability ratio of class ω_1 to class ω_2, $\alpha(\mathbf{x}^{(k)}) = \frac{b}{a}$ is used. If $\frac{c}{d}$ and $\frac{1-c}{1-d}$ are simultaneously bigger or smaller than α then there will be no improvement by using the feature. Duin *et al* illustrate this result by plotting the equations $\frac{c}{d} = \alpha$ and $\frac{1-c}{1-d} = \alpha$ as lines in the plane (c,d). These lines define the region of no improvement. A new feature, x_{k+1} is represented by a point on the plane (c,d). Feature $x^{(k+1)}$ will not contribute to the error if and only if (c,d) falls in the no-improvements region for all elements of the current feature space F. The contribution of a particular $\mathbf{x}^{(k+1)}$ to the error is not only a function of (c,d) but involves a and b too. A feature therefore cannot be considered in isolation. Subsequent features to be added to the subset have to be estimated with respect to the previously selected features.

2.3.2 Is the sequence of error reductions monotone?

It is reasonable to expect that the largest drop in error is at the beginning of the search followed by smaller drops in error. The example below shows that the error reduction does not occur monotonically as expected.

Let the prior probabilities of a two class problem be $P(\omega_1) = 0.35$ and $P(\omega_2) = 0.65$. Consider two feature x_1 and x_2 such that $P(x_1 = 1|\omega_1) = 0.2$, $P(x_1 = 1|\omega_2) = 0.7$, $P(x_2 = 1|\omega_1) = 0.9$ and $P(x_2 = 1|\omega_2) = 0.4$. If nothing but the prior probabilities are used then the Bayes error takes the value $0.35 = \min\{0.35, 0.65\}$. Now consider what would happen to the error if one of the features was selected.

The error for using only feature x_j becomes

$$e(x_j) = \sum_{x_j} \min_i \{P(\omega_i)P(x_j|\omega_i)\}, \tag{2.11}$$

Table 2.1: Example of the error reductions in SFS

	$P(x_1 = 1\|\omega_i)$	$P(x_1 = 0\|\omega_i)$
$P(\omega_1)$	$0.35 \times 0.2 = 0.070$	$0.35 \times 0.8 = 0.280$
$P(\omega_2)$	$0.65 \times 0.7 = 0.455$	$0.65 \times 0.3 = 0.195$
	$P(x_2 = 1\|\omega_i)$	$P(x_2 = 0\|\omega_i)$
$P(\omega_1)$	$0.35 \times 0.9 = 0.315$	$0.35 \times 0.1 = 0.0.035$
$P(\omega_2)$	$0.65 \times 0.4 = 0.260$	$0.65 \times 0.6 = 0.390$

where the summation is over $x_j = 0$ and $x_j = 1$. Using equation 2.11 and the terms from Table 2.1 the error that would be incurred by using either x_1 or x_2 in conjunction with the prior probabilities can be calculated as follows

$$\begin{aligned} e(x_1) &= \sum_{x_1} \min_i \{P(\omega_i)P(x_1|\omega_i)\} \\ &= \min\{0.070, 0.455\} + \min\{0.280, 0.195\} \\ &= 0.070 + 0.195 \\ e(x_1) &= 0.265 \end{aligned} \tag{2.12}$$

$$\begin{aligned} e(x_2) &= \sum_{x_2} \min_i \{P(\omega_i)P(x_2|\omega_i)\} \\ &= \min\{0.315, 0.260\} + \min\{0.035, 0.390\} \\ &= 0.260 + 0.035 \\ e(x_2) &= 0.295 \end{aligned} \tag{2.13}$$

As $e(x_1) < e(x_2)$ feature x_1 would be selected to add to the subset. The reduction in error is $0.35 - 0.265 = 0.085$. The next step would be to add x_2 to the subset.

$$\begin{aligned} P(\omega_1)P(x_1 = 1|\omega_1)P(x_2 = 1|\omega_1) &= 0.070 \times 0.9 = 0.063 \\ P(\omega_2)P(x_1 = 1|\omega_2)P(x_2 = 1|\omega_2) &= 0.455 \times 0.4 = 0.182 \\ P(\omega_1)P(x_1 = 1|\omega_1)P(x_2 = 0|\omega_1) &= 0.070 \times 0.1 = 0.007 \\ P(\omega_2)P(x_1 = 1|\omega_2)P(x_2 = 0|\omega_2) &= 0.455 \times 0.6 = 0.273 \\ P(\omega_1)P(x_1 = 0|\omega_1)P(x_2 = 1|\omega_1) &= 0.280 \times 0.9 = 0.252 \\ P(\omega_2)P(x_1 = 0|\omega_2)P(x_2 = 1|\omega_2) &= 0.195 \times 0.4 = 0.078 \\ P(\omega_1)P(x_1 = 0|\omega_1)P(x_2 = 0|\omega_1) &= 0.280 \times 0.1 = 0.028 \\ P(\omega_2)P(x_1 = 0|\omega_2)P(x_2 = 0|\omega_2) &= 0.195 \times 0.6 = 0.117 \end{aligned} \tag{2.14}$$

Using the set of probabilities calculated from equations 2.14 the error created by using both features is

$$\begin{aligned} e(x_1,x_2) &= \sum_{x_j} \min_i \{P(\omega_i)P(x_j|\omega_i)\} \qquad (2.15)\\ &= \min\{P(x_1=1,x_2=1|\omega_i)\}+\min\{P(x_1=1,x_2=0|\omega_i)\}\\ &\quad +\min\{P(x_1=0,x_2=1|\omega_i)\}+\min\{P(x_1=0,x_2=0|\omega_i)\}\\ &= \min\{0.063,0.182\}+\min\{0.007,0.273\}\\ &\quad +\min\{0.252,0.078\}+\min\{0.028,0.117\}\\ &= 0.063+0.007+0.078+0.028\\ e(x_1,x_2) &= 0.176 \end{aligned}$$

The reduction in the error this time is $0.265-0.176=0.089$. The second error step is larger than the first one. This indicates that the sequence of error reductions is not monotonic.

Further to this, the following example shows that even if there are no features that will reduce the Bayes error, by randomly selecting one of them, the following feature can make a further error reduction. Consider $P(\omega_1)=0.4$ and $P(\omega_2)=0.6$. Let feature x_1 have the probabilities for being present (0.3, 0.42) for class ω_1 and class ω_2 respectively. Similarly let the probabilities for feature x_2 be (0.6, 0.73). The Bayes error using only the prior probabilities would be 0.4. Adding in either of the features will bring no reduction in the error.

$$e(x_1) = 0.12+0.28=0.4 \qquad (2.16)$$

$$e(x_2) = 0.24+0.16=0.4 \qquad (2.17)$$

However if feature x_1 is selected anyway then by adding feature x_2 an error reduction of 0.01804 is obtained.

$$e([x_1,x_2]) = 0.072+0.048+0.168+0.09396=0.3816 \qquad (2.18)$$

This is curious as this means a seemingly redundant feature can be added that enables further reduction in the error by subsequent features. The explanation is that the features are considered in conjunction with others. The importance of individual features changes as more features enter the subset. This effect indicates that the size of the error reduction should not be considered as a stopping criterion for the feature selection procedure.

2.3.3 The SFS procedure with probability data.

The theoretical Bayes error for a problem with non-traditional data is monotonic, adding in more features will not increase the error. As we have a model of the probability distribution given by the non-traditional probability tables we know that the optimal set of features is the entire set, (assuming that the probability estimates are correct). Feature selection in this case will not suffer from the peaking effect. It seems that we could keep selecting features until we reach a plateau of error reduction, if the computational ability of the processor allows. However, this is not the case as we have shown that the sequence of error reductions is not monotonic, a plateau of error may be reached but we cannot guarantee that another feature would not reduce the error further without checking first. That is, we may select k features in succession that do not cause any reduction in the error while feature $(k+1)$ could cause a drop in the error.

The standard procedure of selecting one feature at a time was used for the simulations. Recall that $a = P(\omega_1)P(\mathbf{x}^{(k)}|\omega_1)$ and $b = P(\omega_2)P(\mathbf{x}^{(k)}|\omega_2)$, then the fastest way to calculate the error is to maintain a list with the elements of the current feature space with the corresponding a and b values. The list starts with just one element containing the prior probabilities. To check a new feature x_{k+1}, the list is expanded by creating two elements in the place of each single element of the feature space. The new elements have the parameters (ac, bd) and $a(1-c), b(1-d)$. This implementation is fast but space-consuming as the list contains 2^k elements and needs to store two values for each.

The criterion function used for the evaluation of each subset at each stage is the error rate of the Naïve Bayes classifier, which under the assumption of class-conditional independence is the true Bayes classifier.

2.3.4 Combination of the remaining classes

Finding features important for determining BSE versus Non-BSE or Scrapie versus Non-Scrapie can be seen as one task. However, with this task there is a need to combine all "Non-disease" into a single class. This combination will create the reduced "two-class" problem.

Consider a problem with c classes, $\Omega = \{\omega_i\}, i = 1, \ldots, c$. Suppose that class c is the class of interest and comes with prior probability $\frac{1}{c}$. Let all the remaining classes $\omega_1, \ldots, \omega_{c-1}$ have equal prior probabilities, $P(\omega_i) = \frac{1}{c}$. The original multi-class problem can be reduced to a two-class problem, $\Omega = \{\omega^{(1)}, \omega^{(c)}\}$ by combining $c-1$ classes into a single class $\omega^{(1)}$. Using set theory, the following proposition holds.

Proposition 2 *The probability of the variable x having the value 1 given class* $\omega^{(1)}$ *is*

$$P(x=1|\omega^{(1)}) = \frac{1}{(c-1)}\left(\sum_{i=1}^{c-1} P(x=1|\omega_i)\right).$$

Proof. Suppose that $A, B_1, \ldots, B_{c-1}$ are events so that $B_i \cap B_j = \emptyset \; \forall \; i, j, \; i \neq j$. Thus events $(A \cap B_i)$ and $(A \cap B_j)$ are mutually exclusive. From set theory,

$$A \cap (B_1 \cup B_2 \cup \ldots \cup B_{c-1}) = (A \cap B_1) \cup (A \cap B_2) \cup \ldots \cup (A \cap B_{c-1}) \qquad (2.19)$$

Taking the probabilities of (2.19),

$$P\Big(A \cap (B_1 \cup \ldots \cup B_{c-1})\Big) = P(A \cap B_1) + \ldots + P(A \cap B_{c-1}) \qquad (2.20)$$

Taking the combination of the classes $\omega^{(1)} = \omega_1 \cup \omega_2 \cup \ldots \cup \omega_{c-1}$, and the set (event) A as $x = 1$, the probability of $x = 1$ given class $\omega^{(1)}$, using equation (2.20) becomes

$$\begin{aligned} P(x=1|\omega^{(1)}) &= \frac{P(x=1,\omega^{(1)})}{P(\omega^{(1)})} \\ &= \frac{P(x=1,\omega_1) + P(x=1,\omega_2) + \ldots + P(x=1,\omega_{c-1})}{P(\omega^{(1)})}. \end{aligned} \qquad (2.21)$$

Looking at each term in (2.21) separately,

$$P(x=1,\omega_i) = P(x=1|\omega_i)P(\omega_i).$$

Equation (2.21) becomes,

$$P(x=1|\omega^{(1)}) = \frac{P(\omega_1)P(x=1|\omega_1) + \ldots + P(\omega_{c-1})P(x=1|\omega_{c-1})}{P(\omega^{(1)})}. \qquad (2.22)$$

The prior probability of ω_i, $P(\omega_i) = \frac{1}{c}$ (all classes are equiprobable). $P(\omega^{(1)}) = 1 - \frac{1}{c} = \frac{c-1}{c}$. Substitute in (2.22) to get

$$\begin{aligned} P(x=1|\omega^{(1)}) &= \frac{\frac{1}{c}P(x=1|\omega_1) + \ldots + \frac{1}{c}P(x=1|\omega_{c-1})}{\frac{c-1}{c}} \\ P(x=1|\omega^{(1)}) &= \frac{1}{c-1}\sum_{i=1}^{c-1} P(x=1|\omega_i) \end{aligned} \qquad (2.23)$$

Equation (2.23) is the average of the $c-1$ conditional probabilities. ■

The average value of all the remaining class conditional probabilities can be taken to create the second class for the two-class problem.

Table 2.2: The 15 Scrapie signs selected by SFS and the cumulative error, sensitivity and specificity (in %)

#	Feature	Error	Sensitivity	Specificity
1	Hyperaesthia	9.87	86.7	93.6
2	Weight Loss	7.47	86.7	98.4
3	Pruritus	2.93	97.3	96.8
21	Increased respiratory rate	2.63	97.3	97.4
4	Abnormal behaviour	2.21	99.3	96.3
5	Underweight	1.37	98.3	98.9
9	Tremor	1.18	99.3	98.3
22	Sudden death	1.02	99.3	98.6
6	Dysmetria	0.92	99.3	98.9
7	Ataxia	0.67	99.3	99.3
8	Grinding Teeth	0.55	99.5	99.4
10	Trembling	0.45	99.6	99.5
11	Alopecia	0.36	99.6	99.7
12	Seizures or syncope	0.32	99.7	99.7
13	Rumen hypomotility	0.29	99.7	99.7

2.3.5 Application to Scrapie and BSE probability tables

Using the reduced two class version of the probability data for Scrapie and BSE with equal prior probabilities, SFS was applied to select the "best" 15 features for each data set that would discriminate best between the disease of interest and any other disease. The subset size of 15 was chosen in consultation with the domain experts as a feasible number of features for possible collection of "traditional" recorded case data.

Scrapie

The 15 signs selected by SFS for Scrapie are given in Table 2.2. The first column of the table gives the importance rank of the individual feature. Feature numbered 1 (Hyperaesthia) in this column has the greatest absolute difference between $P(x_1 = 1|\text{Scrapie})$ and $P(x_1 = 1|\text{Non} - \text{Scrapie})$ among all 285 features. The features are shown in the table in the order they were selected by SFS into the subset. The subset of 15 features selected by SFS has a calculated error of 0.29%. 13 of the individually best features are contained within the subset of 15 and they are selected in an order similar to their importance rank. By choosing the individually best 15 features based solely on their importance rank an error of 0.33% is obtained. This is slightly higher than the set selected through SFS. However this difference is too small to claim that SFS has selected a better subset.

Table 2.3: The 15 BSE signs selected by SFS and the cumulative error, sensitivity and specificity (in %)

#	Feature	Error	Sensitivity	Specificity
1	Gait uncoordinated	17.17	78.0	87.66
7	Aggression	12.62	88.78	85.98
4	Feed intake <50% normal	8.52	88.78	94.18
9	Teeth Grinding	6.76	93.6	92.88
11	Hypo-responsive to external stimuli	5.62	93.6	95.16
13	Posture of paired limbs abnormal	4.92	95.52	94.64
19	Dyspnoea, unspecified	4.32	95.52	95.82
20	Convulsions, unspecified	3.87	95.52	96.74
14	Back arched	3.39	96.86	96.36
21	Heart rate >100 bpm	3.01	96.86	97.12
23	Restlessness	2.74	96.86	97.66
2	Gait stumbling	2.33	99.38	95.98
3	Hyper-responsive to external stimuli	1.61	98.16	98.62
5	Tremor	1.35	99.40	97.90
6	Body weight less than normal	1.14	98.76	98.96

BSE

From Table 2.3 the variation of the order of the signs from the individually best order is more substantial than that for Scrapie. Still 11 of the 15 individually best features have been selected among the 15 best. The inclusion of features not in the subset of individually best 15 indicates their relative importance when considered in conjunction with other features. Unfortunately, it is not possible to access any information about the potential dependencies in the real data.

Our experiments showed that the predicted reduction in error using features selected by SFS was quite substantial. For using just 12 features selected by SFS for Scrapie the error rate has been pushed down to less than half a percent. Using the 15 features selected for BSE by SFS reduces the error rate to less than two percent. Although SFS does not guarantee finding an optimal subset it does provide a good practical solution in this case, within the limits of the two assumptions.

2.3.6 How reliable are the results?

The adequacy of the SFS with respect to the validity of the assumptions needs to be considered. The proposed feature selection cannot be easily tested with regards to the independence assumed. This is because there is no indication of what the real-life dependencies between the features may be. Therefore imposing a model of dependencies between the features will not give us any insight into the effects on feature selection

related to the real-life problem.

The entries in the probability tables were constructed as the average estimate of three domain experts. The original estimates produced by each expert would provide a valuable insight into the variance of the averaged estimate. This would give an indication of the variance of the estimates in the probability table. Ultimately, we would be able to run an analysis using each individual experts estimates to assess the agreement between the selected feature subsets. Unfortunately, we do not have access to these individual estimates only the averaged value provided in the probability table. Therefore, we have no information on the variation of the estimates, i.e. the disagreement / agreement of the experts with regard to these probability estimates.

The selection of a subset using SFS can be tested on perturbed values of the estimated frequencies though. This will give us an insight into how robust SFS is with regard to the averaged experts estimates that are used in the non-traditional data set.

Consider normal distributions for each frequency with mean equal to the expert estimate and standard deviations, $\sigma = \{0.1, 0.2, 0.3\}$. SFS was run 1000 times to select 10 features on each of the perturbed frequencies to find out how similar the selected subsets were. The procedure ran as follows,

- For each value of σ repeat 1000 times:-

    1. For every $P(x_j|\omega_i)$ in the probability table apply a random adjustment within a normal distribution such that the mean of the distribution is $P(x_j|\omega_i)$ with standard deviation equal to σ
    2. Select 10 features using SFS from this perturbed probability table
    3. Store the 10 features

For each variation of σ we have 1000 subsets of 10 features selected by SFS. To evaluate the similarity of the subsets the features were ranked. A feature receives rank 10 if it is selected first, reducing to 1 for the feature selected last. All features not selected are given a rank of zero. The sum of the ranks of each feature across the 1000 subsets gives an idea of that feature's importance. A rank of 10,000 would indicate that the feature has been selected first in every single subset. If the features total rank is below 1000 then it is not present in all 1000 sets. A ranked list can be compiled for each value of σ. A high match between these lists would indicate a robust feature selection procedure. Features that are worth a second look or those that may not be too reliable may be brought to our attention. These may be features that were not originally selected by the first run of SFS but appear high on the ranked list or features that were selected by the original SFS selection that do not appear in the ranked list.

Table 2.4: The subsets using the perturbed probability estimates (a) Scrapie data (b) BSE data

$\sigma = 0.1$		$\sigma = 0.2$		$\sigma = 0.3$	
#	Score	#	Score	#	Score
1	5734	1	3025	1	1236
2	5301	2	2561	3	1160
3	4479	3	2509	2	1063
5	3642	5	2093	4	1000
4	3562	4	2048	5	902
6	2066	8	1628	7	796
10	1985	7	1605	10	779
7	1894	6	1454	8	769
9	1892	9	1354	6	766
8	1853	10	1244	9	709
11	1511	11	1084	11	655
12	828	12	730	143	569
13	752	13	718	12	454
15	621	14	454	16	409
17	621	143	441	13	402

(a)

$\sigma = 0.1$		$\sigma = 0.2$		$\sigma = 0.3$	
#	Score	#	Score	#	Score
1	4333	1	2485	1	1111
4	3893	3	1930	3	1032
2	3458	2	1911	2	885
3	3227	5	1827	5	770
7	2689	4	1663	4	759
5	2047	6	1577	7	748
6	2041	7	1264	8	607
9	1737	8	1256	6	569
8	1440	10	754	10	544
10	1111	9	663	9	544
14	959	12	650	15	443
13	880	13	513	13	397
15	868	14	494	26	377
16	755	11	442	63	373
18	737	15	430	17	372

(b)

Tables 2.4(a) and (b) give the top 15 features according to their rank scores for Scrapie and BSE respectively.

The sensitivity analysis revealed that the top 10 ranked features for all three lists were the top ten individual features in a slightly different order, for all three values of σ and both Scrapie and BSE data sets.

Note that features 21 and 22 are present in the original Scrapie SFS selection (Table 2.2) but absent from all three of the perturbed sets, Table 2.4(a). This means that "Increased respiratory rate" (21) and "Sudden death" (22) are not as reliable as first thought. In fact, these two features do not appear in the first 20 ranked features for any of the perturbed subsets. These two features are replaced by Abnormal proprioceptive positioning (15) and Tetraparesis (17) for $\sigma = 0.1$, Constipation (14) and Abdominal distension (143) for $\sigma = 0.2$, and Abdominal distension (143) and Excitement (16) for $\sigma = 0.3$. This result suggests that these six features, replacing those removed, warrant addition into any proposed subset for data collection.

Table 2.4(b) gives the features selected for perturbing the probability estimates of the BSE data. The variation in features selected is slightly greater than for the Scrapie data.

There are four features from the original subset that are not selected at all in the three perturbed subsets. These features may not be as reliable as first indicated by the SFS selection. The features are Convulsions, unspecified (20), Heart rate >100bpm (21), Restlessness (23) and Dyspnoea, unspecified (19). These are the four features in the original

SFS subset that do not occur in the top 15 individually best features (i.e. importance rank greater than 15). Hypo-responsive to external stimuli (11) is also absent when $\sigma = 0.1$ and $\sigma = 0.3$. For $\sigma = 0.3$, Back arched (14) is omitted.

There are three features that appear as replacements in all three perturbed subsets. They are Gait abnormal all 4 legs (8), Licking, rubbing or chewing of self (10) and Frenzy (15). These are joined by various features for the varying values of σ. When $\sigma = 0.1$ the added features include Body weight very low (16) and Rising difficulty (18). When $\sigma = 0.2$ the added feature is Gait abnormal, unspecified (12). In the final perturbed subset when $\sigma = 0.3$ there are three added features. These are Milk yield less than normal (26), Joint(s) fetlocks knuckling (63) and Gait falls easily (17).

These nine added features all need to be taken into consideration when proposing a subset of features for data collection. The variation of the feature lists also indicate to the domain experts what happens if the probability estimates given in the non-traditional data vary. For example, if the non-traditional data is precise, which we have to assume is the case, then we would obtain the feature sets given in Tables 2.2 and 2.3. However, if the true estimates vary from those given by a standard deviation of σ (assuming the estimates vary along a normal distribution) the subsets achieved would be those given in Table 2.4.

2.4 Comparison of feature selection methods

This section studies a comparison of five feature selection methods on the multiclass probability data. The feature sets are evaluated by the classification error of the Naïve Bayes classifier. The complexity of a feature selection method is measured by the number of calculations of the classification error needed to select d out of n features. The classification task for the multi-class data is to discriminate between all the featured diseases. Features can either indicate the presence of a disease or the definite absence of a disease.

2.4.1 Single Best (SB)

Select the best d features from the original n based on their individual criterion scores. This method does not guarantee the optimal subset but is quick and sometimes surprisingly efficient. The classification error for each feature is calculated once, the list of errors is then ordered and the best d features are selected to make up the subset. Thus SB needs only n evaluations to select the subset regardless of the value of d.

2.4.2 Genetic Algorithm (GA)

Genetic algorithms follow the ideas of biological evolution, using natural selection, mating and mutation. The process evolves a population of m chromosomes. A chromosome is an individual feature subset.

The GA used in this study goes through the following steps,

1. (a) Generate a random population P_0 of m chromosomes
   (b) Evaluate P_0 using a fitness function
2. Take all of P_i to be the mating set
3. Pick two parent chromosomes with replacement from P_i. Perform crossover: pick a point inside the chromosome and switch the end parts of the two chromosomes to create two new children
4. Put the two children chromosomes into the offspring set, O.
5. Continue crossover until O contains m children chromosomes.
6. Mutate the offspring set. Randomly add or remove a feature with a prespecified mutation probability, $p_m = 0.2$
7. Evaluate the chromosomes in O using the fitness function
8. Form the next population, P_{i+1} by selecting the best m chromosomes from $P_i \cup O$
9. Repeat from step 2 for a pre-selected number of generations, $T = 50$.

The fitness function is the classification accuracy of the NB classifier using the particular chromosome. To ensure that the selection does not run towards selecting all the available features a cost parameter, α, was used. The fitness function, f, limits the amount of features selected.

$$f = \text{Accuracy} - \alpha \times |\text{chromosome}|. \tag{2.24}$$

Genetic algorithms have a number of parameters that are preselected :- probability of crossover, p_c, the probability of mutation, p_m, the cost parameter, α, and the number of generations, T. For the implementation here $p_c = 0.5$, $p_m = 0.2$, $\alpha = 0.05$ and $T = 50$. The method needs $Tm + m$ evaluations of the NB accuracy.

Many improvements to GA in the literature are aimed at the fitness and cost functions, [18, 84, 134]. A comparative study by Ferri *et al* [44] suggested that as more features are added into a problem the performance of a GA worsens.

Table 2.5: The class-pairs method for feature selection (Ji and Bang (2000) [65])

	C_{ij}	
x_k	$\dots \quad P_{ij}(x_k) \quad \dots$	T_k
	E_{ij}	

2.4.3 Class-Pairs (CP)

The class-pairs method proposed by Ji and Bang [65] is aimed at reducing the number of features in a multiclass problem. The main concept of the method is to select a single feature that best discriminates between each pair of classes. A table is constructed (Table 2.5), where C_{ij} is the class pair (ω_i, ω_j), x_k is the k-th feature and $P_{ij}(x_k)$ is the discriminatory power of feature k for class pair (ω_i, ω_j). This is calculated as the probability of correct classification between class ω_i and ω_j for feature x_k.

E_{ij} denotes the relative ease of classifying the pair C_{ij} and T_k is the relative discriminatory power of feature x_k, where

$$E_{ij} = \sum_k P_{ij}(x_k), \tag{2.25}$$

$$T_k = \sum_{ij} P_{ij}(x_k) \tag{2.26}$$

The selection procedure starts with an empty set. The smallest value of E_{ij} identifies the class pair, $C'_{i,j}$ (column in Table 2.5) that is hardest to discriminate. For this class pair the feature with the highest discriminatory value $P_{ij}(x_k)$ is added to the subset, if it has not already been selected. If more than one feature has the highest $P_{ij}(x_k)$ for the chosen column, then the one with the highest T_k value is added. Column C_{ij} is removed from the table and the process continues with the next hardest pair to discriminate. The method stops when all class pairs have been covered.

The maximum number of features the algorithm will select is $\min\{(c(c-1))/2, n\}$. Ji and Bang claim that the number of features selected will be much less than this. If $d > \min\{(c(c-1))/2, n\}$ then the method will select as many features as it allows for. If $d < \min\{(c(c-1))/2, n\}$ then the method can be restricted to select only d features although this may mean that not all class pairs will be covered.

The complexity of the method measured by the number of evaluations of $P_{ij}(k)$ is $\frac{c(c-1)n}{2}$. This reflects the preparation phase of setting up Table 2.5, and does not take into account the subsequent selection procedure.

2.4.4 Feature pairs (FP)

Feature-pairs (FP) method starts with an empty set. All pairs of features are evaluated and the best pair added to the subset. While the desired number of features d is not reached add the features from the next best pair that are not already among the selected features. If a state is reached where there are $d-1$ features selected then either select both of the features from the next pair and create a subset with $d+1$ features or randomly select one member of the pair of features to make d features in the subset. The complexity of the Feature-pairs method is $\frac{n(n-1)}{2}$.

2.4.5 Small-scale simulation

For the small scale simulation the total number of features n was limited to 20.

- For c=3 to 10
  - Generate a probability matrix of size $20 \times c$
  - Select a subset of features using each of the feature selection methods
  - Generate 100 "traditional" binary style data points for each of the c classes
  - Label the data points using the NB classifier a with each of the subsets

The error is estimated as the percentage mismatch between the generated label and the true class label.

Preselected d

Feature selection methods may either select a predefined d number of features or automatically define the size of the subset. In this first simulation d was pre-defined. 50 random $20 \times c$ matrices for each pair of classes and subset size, (c, d), where $c = 3, 4, \ldots, 10$, $d = 2, 3, \ldots, 10$ were generated, i.e. the above procedure was replicated 50 times for each d.

In the results in Table 2.6(a) it can be seen that the algorithm with the lowest error rates for all c is the feature-pairs method (FP). The performance of class-pairs (CP) is gradually worsening due to the effect of only being able to select two features and not covering all the classes as the algorithm is designed to. As d and c are increased CP begins to perform more in line with the other algorithms. When d is increased it can be seen that SFS provides the best solution, Table 2.6(b)

For CP with a small number of classes, the subset may contain fewer than d features. This is due to the maximum number of features that algorithm can possibly select is $\frac{c(c-1)}{2}$. So if $d > \frac{c(c-1)}{2}$, then the subset will contain fewer than d features. The CP algorithm may also naturally select less than $\max\{\frac{c(c-1)}{2}, n\}$ features by design. This

Table 2.6: Classification error (in %) with $c = 3, \ldots, 10$ classes for (a) $d = 2$ features and (b) $d = 10$ features for the four feature selection methods

c	CP	SB	FP	SFS
3	18.41	22.49	17.09	17.98
4	37.80	36.84	29.63	31.40
5	49.07	48.00	40.55	43.45
6	55.97	54.15	48.69	50.47
7	62.97	60.83	55.04	56.93
8	67.40	64.23	60.20	61.76
9	70.05	67.87	64.06	64.66
10	72.38	70.84	67.17	67.97

(a)

c	CP	SB	FP	SFS
3	15.20	5.06	5.17	4.95
4	15.65	7.72	7.67	7.01
5	16.23	10.90	11.53	10.05
6	15.52	13.62	13.72	12.38
7	16.86	16.89	16.87	15.59
8	19.38	19.56	19.56	18.06
9	22.57	23.01	22.83	20.84
10	24.84	23.98	23.89	22.42

(b)

means that the algorithm can naturally select less than d features. There is no user control over this effect. To compare CP fairly to other the algorithms the method has to be allowed to run fully, utilising the covering concept that it exploits. This is done by the simulations in the next section.

No preselection of d

The size of the subset was not pre-defined. The number, n of features to select from was kept at 20. The simulation procedure was replicated 50 times allowing CP to select as many features as it required. The FP and SB algorithms were set to run to select the same amount of features as CP. SFS was not used for computational time reasons.

Table 2.7 gives the comparison of CP to SB and FP. The second column of the table denotes the average number of features selected by CP over the 50 runs for each c. For all the values of c, CP gives a lower error rate than either SB or FP. These results indicate that when CP is not restricted to select a pre-determined d features then the method can perform well. With a large number of classes though there is the risk of selecting a large number of features. The method would not work for a two-class problem, as the maximum number of features it would select in this instance would be 1, the best feature to discriminate between class 1 and class 2.

Experiments were also carried out with the GA. The subset with the best value for the fitness function after the $T = 50$ generations was taken to be the selected one. SFS, FP and CP were run each time to select the same number of features. Here SFS was used as the Genetic Algorithm seemed to select fewer features than CP.

The results for the comparison to GA are given in Table 2.8. Again the second column gives the average number of features selected for each subset in the 50 runs. Whilst the performance of GA, SB and FP are fairly in line with one another, SFS is achieving a

Table 2.7: Comparison of the classification error (in %) of CP, FP and SB

c	Features d	CP	SB	FP
3	2.66	14.93	18.66	15.02
4	4.60	16.28	17.56	17.52
5	6.44	15.36	17.25	16.66
6	8.24	16.42	17.32	17.48
7	11.04	14.92	15.96	16.48
8	12.26	14.90	15.50	15.32
9	13.66	15.01	15.40	15.34
10	14.80	14.86	15.00	15.34

Table 2.8: Comparison of the classification error (in %) of GA, FP, SFS, and SB

c	Features d	GA	SB	FP	SFS
3	3.52	14.87	15.73	12.94	12.43
4	4.52	19.03	19.26	18.30	16.19
5	5.10	22.50	22.16	21.14	18.96
6	5.18	27.61	28.19	27.29	25.10
7	5.80	30.60	31.90	30.94	28.52
8	6.04	32.38	34.44	32.46	30.43
9	6.02	36.74	37.77	36.51	34.93
10	6.54	35.93	38.03	36.89	34.87

lower error rate for each value of c. This again indicates the practicality of using SFS to select a subset for this type of problem

2.4.6 Larger-scale simulation

The simulation was run so that 50 matrices were generated for each of the values of d. The number of classes, c was set at 50. SFS and GA were excluded from the large scale simulation due to their larger computational times. The total number of features was increased from $n = 20$ to $n = 100$. The number of selected features d was varied from 5 to 50 in increments of 5.

The results in Table 2.9 show that FP produces consistently lower error rates than CP and SB. However, the performance of CP begins to fall more in line with the performance of FP and SB as the number of features selected is increased. This may be due to the method exploiting more of the potential of its covering concept.

Table 2.9: Classification errors (in %) of CP, SB and FP for the case where $c = 50$.

Features d	CP	SB	FP
5	81.79	81.73	77.43
10	60.18	59.63	55.30
15	40.09	39.50	36.33
20	25.17	24.53	22.98
25	14.81	14.73	13.32
30	8.70	8.73	7.84
35	4.78	4.77	4.26
40	2.77	2.66	2.50
45	1.50	1.56	1.42
50	0.79	0.84	0.78

2.4.7 Application to BSE and Scrapie probability tables

All the methods were applied to the multiclass versions of the BSE and Scrapie probability data. The Scrapie problem having 62 classes and 285 features. The BSE problem having 57 classes and 242 features. The task for this data is to find the "best" subset of size d using the non-traditional probability table data as it is.

Scrapie

The first trial ran SFS, CP, FP and SB to select 12 features. The second trial allowed CP to run to select as many features as it needed to take advantage of its covering concept. FP and SB were set to match the total number of features ultimately selected by CP. Finally, GA was allowed to select the number of features that it deemed fit. FP, CP and SB were set to select the same number of features for comparison.

Table 2.10 gives the results of each of the methods and the number of features they selected. The table shows that once again SFS performs well, if a small number of features are to be selected. The only disadvantage is the time needed to run the algorithm.

The results also show that if CP is allowed to run all the way, then the subset selected performs well. In this case the subset selected by CP outperforms the subset of the same size selected by SB or FP. Admittedly the difference in performance of the FP and CP subsets is so small that it cannot be said that the subset selected by CP is any better than that selected by FP.

For GA to try and select a better subset the parameters were adjusted to $p_c = 0.8$, $p_m = 0.01$ and $\alpha = 0.005$. Varying the parameters seemed to have little effect on the success of the method as the subset selected by the GA did not perform as well as any of the other subsets selected by the other methods. The GA was allowed to select the best performing subset of the size it wished. In comparison to the GA subset it was shown

Table 2.10: Error rates (in %) for the feature selection methods using the multiclass probability tables for Scrapie

Method (d)	Error
SFS (12)	59.75
SB (12)	76.35
CP (12)	69.30
FP (12)	66.10
CP (77)	6.25
SB (77)	9.92
FP (77)	6.49
GA (67)	16.44
CP (67)	7.41
SB (67)	13.76
FP (67)	7.22
All (242)	2.52

that the other methods could find a better subset of the same size. This was surprising to some extent in the light of previous successes reported in the literature [134]. The main advantage of GA is that they can discover good subsets of dependent features by chance. In the experiments, the simpler selection methods may have performed better as the features were treated as independent in all experiments. This meant the advantage of the GA was left unexploited. Experiments performed in a study by Ferri *et al* [44] also suggested that the performance of GA decreases as the number of features increased. It is suggested that this is due to the fact that the region of feature space being searched increases quickly as the number of features increases.

BSE

SFS, CP, FP and SB were run to select ten features. CP was then run to select as many features as it wished. FP and SB were run to select the same amount of features. The error rates achieved by the selected subsets are given in Table 2.4.7. The error rates associated with using all the available features are also given.

The best performing method when selecting a small set of features is again SFS. When CP is allowed to run fully then the method does well once again, selecting a subset of features that can outperform the sets selected by FP and SB.

As it happened the GA also selected 58 features. The resulting error rates for the 58 features selected by the GA are worse than the errors for the 58 features selected by the three other methods.

As expected none of the subsets get close to the error rates when using all the features for either Scrapie or BSE. However, with over 200 signs in each set the vet is not likely

Table 2.11: Error rates (in %) for the feature selection using the multiclass probability tables for BSE

Method (d)	Error
SFS (10)	42.58
SB (10)	64.32
CP (10)	58.65
FP (10)	54.82
CP (58)	1.72
SB (58)	3.09
FP (58)	2.56
GA (58)	9.04
All (242)	0.49

to record all of the signs that are present or absent in every sheep or cow. Checking a smaller selection of signs would seem a much more feasible task. Selecting such a set by SFS would be sensible. The main difference between CP and FP is whether or not d is pre-determined. CP performs well but may select too many features, as to exploit the benefit of its covering concept d is not pre-determined

2.5 Chapter summary

This chapter has investigated feature selection techniques being applied to the "non-traditional" probability table data.

The potentials, limitations and stability of SFS for the non-traditional data were investigated resulting in the following conclusions:-

- SFS is monotone on the number of features.
- The sequence of error reductions produced by SFS is not monotone.

The SFS procedure in relation to non-traditional probability table data was explained and applied to the BSE and Scrapie tables. A set of 15 "important" features for each disease was found. By "important" features we mean features that are best to discriminate between the disease of interest (BSE or Scrapie) and the combined remainder of differential diagnoses that were provided.

The stability of this SFS procedure with regard to the reliability of the experts' estimates was tested. The majority of the selected features remained undisturbed by this simulation. However, it was indicated that a few of the selected features may not be as reliable as first thought. A few features that had not been included in the selected subset

were also brought to our attention. These features would also need to be considered in any future work. For the two class task SFS provides a good practical solution.

Finally, a collection of feature selection methods were applied to the multi-class version of the BSE and Scrapie probability tables. This time to find features that would be helpful to distinguish between all types of disease.

In this situation SFS still performed well for a small subset. However, SFS is restricted by its time complexity. For a larger subset selection CP and FP provided reasonable and practical alternatives with a much lower time complexity.

Smaller subsets of features selected from the probability tables may be considered as an aid in the collection of "traditional" data. Data collection based on smaller subsets of the more discriminating features is a more feasible task. In the absence of "traditional" data collected uniformly across the population, the next question would be how to handle probability table data with regard to designing classifier models for the classification of new cases.

Chapter 3

Classification for probability table data

There is a need to be able to handle the situation where only limited "traditional" data exists but "non-traditional" data is readily available. Given the classifier requirements of simplicity, efficiency and comprehensibility the next step is to consider the standard classifier models and their processes with regard to the "non-traditional" data.

There are two types of learning associated with classification problems, supervised and unsupervised. Supervised learning or discrimination takes a set of example cases with class labels. These labelled cases are used to design an automatic process of classifying future data - a classifier. Unsupervised learning is applied to a set of example cases that are not labelled. Clustering techniques can be used to find natural clusters or groupings within the data.

The BSE and Scrapie probability data presents a supervised learning problem. However, in place of the set of labelled cases there are subjectively estimated probabilities $P(x_j = 1|\omega_i)$ where x_j is the jth binary feature, ω_i is the ith class and value 1 of x_j means that x_j is present. The nature of the problem dictates that any classifier used needs to be easy to understand. If the domain experts understand the decision making process of the classifier they are far more likely to trust any decision it makes [133].

Decision trees and Naïve Bayes are simple classifiers that have been effective for many types of problems. They both allow a domain expert to follow the decision making process.

3.1 Decision Tree Classifiers

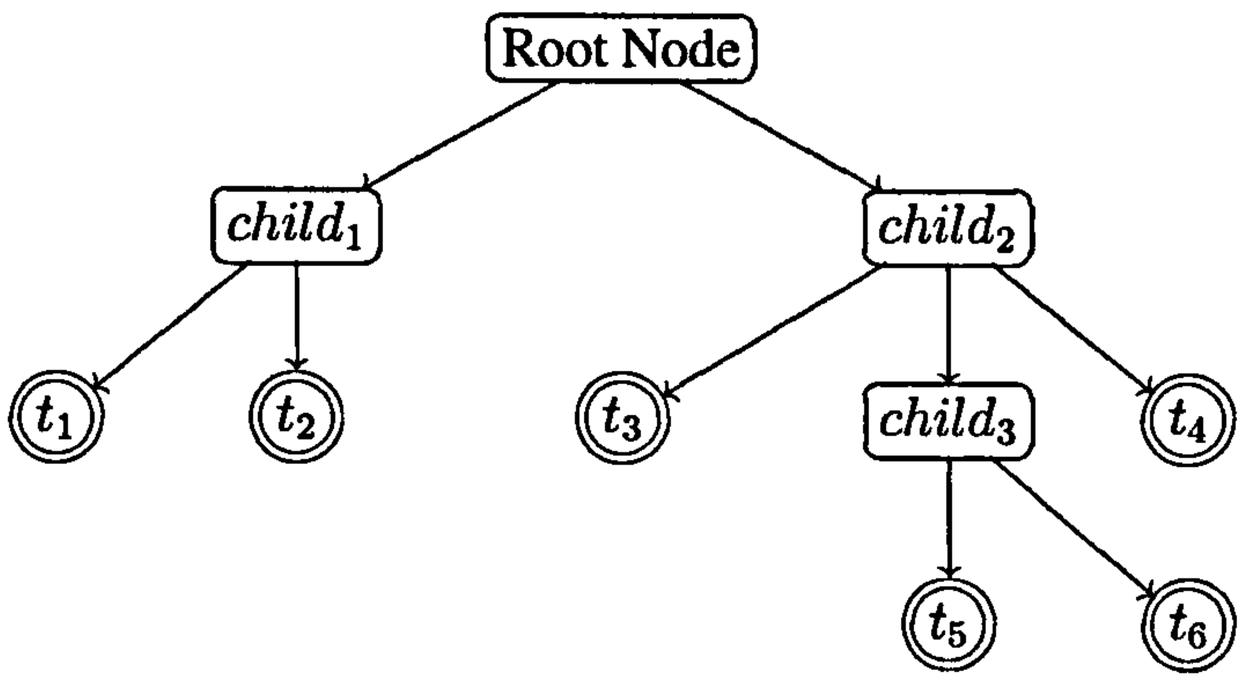


Figure 3.1: A typical decision tree classifier.

A decision tree is a directed acyclic graph. A decision tree, such as the one depicted in Figure 3.1, is made up of a single non-terminal root node, non-terminal child nodes, terminal child nodes and branches (arrows) describing the paths in the tree. Terminal nodes (or leaves) are marked with a class label. To classify a new case, a path through the

tree to a terminal node is created by following decisions made at each non-terminal node. Upon reaching a terminal node the case will be given the class label marking that node.

The process of decision tree design is summarised by Safavian and Landgrebe into four main objectives, [122] : 1) classify correctly as much as possible of the training sample, 2) generalise beyond the training sample, 3) be easy to update when more training sample becomes available and 4) have simple structure.

Descriptions of decision tree construction are given in depth by [36, 77, 108, 150]. One of the most widely used methods of construction is the top-down approach. The top down design starts at the root node and proceeds by splitting the training data down until terminal nodes are created. Webb, [150] divides top-down construction into three phases: 1) finding a splitting rule, 2) deciding on the terminal nodes and 3) assigning class labels to the terminal nodes.

The splitting rule or criterion dictates the structure of the whole tree. The criterion is used to partition the data at each level of the tree. The choice of the criterion involves how many partitions the data can be split into at each level of the tree and what type of features the criterion will accept. The overall complexity and comprehensibility of the tree depends on the selection of the criterion. Most criteria in top down decision tree design depend upon some greedy heuristic. These heuristics are not easily understood by the non-expert user. Berzal *et al* [9] support easy to understand splitting rules. The authors develop understandable rules based on the probability of the most common class in each subtree. A subtree is a portion of a tree that can be viewed as a complete tree in itself. The subtree corresponding to the root node is the entire tree while a subtree corresponding to any other node is called a proper subtree.

The classical decision tree model uses a single feature at each non-terminal node to determine the split. Another decision tree design uses a subset of features as a discriminant function at each non-terminal node to determine the split. This "feature-based" tree loses some of the simplicity of the standard tree. "Feature-based" decision trees entail the use of a selection method to establish the features to use at the non-terminal nodes. The feature selection problem at a non-terminal node is to find the subset that gives the best possible partition of the data reaching this node, rather than reducing the dimensionality of the initial problem. The number of features that should be chosen to represent each split is crucial. The more features used at a node implies the better discrimination achieved. However, the use of more features at a node increases the complexity and the computational running time of the tree. Studies by Brown *et al* [17] and Sethi and Yoo [132] have indicated that the performance of the trees is improved by the use of multi-feature splits rather than single feature splits at a node. Brown *et al* note the lack of available mechanisms for creating the multi-feature splits.

A study by Mingers in 1989 [102] claimed that random feature selection methods provide results as good as using orthodox feature selection methods. Empirical results due to Liu and White [92] show that orthodox methods do in fact outperform the random attempts. Safavian and Landgrebe [122] argue that an advantage of the decision tree structure is the occurrence of different subsets of features at non-terminal nodes.

Phase two of construction involves determining which nodes will become terminal nodes. This involves considering the purity of the candidate nodes, the overall size of the tree and whether pruning methods are to be used. The design of a decision tree aims to achieve terminal nodes that are as pure as possible, i.e. all data arriving at a terminal node will have the same class label. However, achieving pure nodes may result in unnecessarily large trees with only a small number of example cases reaching terminal nodes. Safavian and Landgrebe [122] indicate that smaller trees are less sensitive to statistical irregularities in the training data and can therefore be preferable to larger trees. Many decision tree methods now adopt pruning methods to avoid the effects of overfitting the training data [42, 43, 102, 154]. Pruning is a method of "cutting back" a tree to reduce its size while maintaining as much accuracy as possible.

The final phase of assigning class labels to the terminal nodes could be considered the simplest phase of construction. The majority of construction methods assign the class label to the terminal node that will minimise any misclassification cost, e.g. the label of the majority class amongst the labelled examples at the terminal node.

There are many proposed alterations to decision tree design in the literature [55, 77, 97, 109]. Lim *et al* [88] give an empirical comparison of decision trees and other classification methods. The study suggests that the error rates of the majority of classifiers were not significantly different but their computational times varied widely. This implies that the choices made in designing a decision tree can be influenced by desirable resultant properties such as comprehensibility and complexity.

Multi-class problems may be decomposed into multiple two class problems in a variety of ways. Masulli and Valentini [96] and Tax and Duin [141] both consider using two-class classifiers for multi-class classification tasks. The two class tasks may be either created as pitting one class against another thus requiring a total of $\frac{c(c-1)}{2}$ classifiers for a c class problem or use a class-modular approach requiring only c classifiers.

The class modular approach for a c class problem is to create c two-class classifiers whose individual problems are class ω_i versus class $\omega^{(1)}$, where $\omega_i,\ i = 1 \ldots c$, is the class of interest for the problem and $\omega^{(1)}$ is the remaining $c - 1$ classes combined.

Binary tree classifiers are a special type of decision tree classifier where every non-terminal node has exactly two child nodes. It is reasonable to consider binary trees as any node with multiple answers can be equivalently represented with binary nodes. The sim-

plicity of a binary tree is attractive. Berzal *et al* [9] indicate that using a binary structure makes a tree larger with fewer leaves. The larger the tree the longer the implied training time and any subsequent classification will be. It has been shown that to construct a truly optimal binary decision tree would be an NP complete problem [61].

A modification to binary trees studied by Lee and Oh [86] in order to decrease the overall computation time is a class-modular approach to the splitting criterion. Lee and Oh propose that at each node of the tree the class group is partitioned into two distinct subgroups. The choice of the partition is optimised by a GA procedure. The class-modular approach is adopted for implementing classification at a node. At a node the c class-modular two-class classifiers classify a test case, $\mathbf{x}$. A class is identified according to a majority voting from the c classifiers.

Cascade classifiers are a special subset of the binary decision tree classifiers which can be integrated with the class-modular design approach.

3.2 Cascade Decision Tree Classifiers

Definition. A cascade classifier is a binary decision tree such that at each level there is at least one terminal node. Two terminal nodes mark the end of the tree.

For a c class problem a cascade tree will need to have $c - 1$ non-terminal nodes. At each non-terminal node a decision is made between a single class and the combination of the remaining classes, the class-modular approach. The design of a cascade tree involves deciding which single class to separate to the terminal node at each level. Once a class is chosen for a terminal node, this class is removed from all further calculations in the tree.

3.2.1 Application to non-traditional probability data

The cascade decision tree can be considered for probability table data due mainly to the simplicity of the tree design. The design of the cascade tree allows domain experts to follow the logic of the decision making processes. The model lends itself to the case with a large number of classes. In the probability tables there are 57 classes to distinguish between for the BSE set and 63 classes for the Scrapie set. Using a single feature to determine the split between a single class and the remaining classes would be of interest to the veterinary domain due to the ease of interpretation. Due to the binary nature of the features, once a feature has been used to distinguish between a class and the combined remainder then this feature can not be used to determine another split further down the tree. The presence or absence of a particular feature appearing at a node will have already been determined and so will not have a different value later on down the tree.

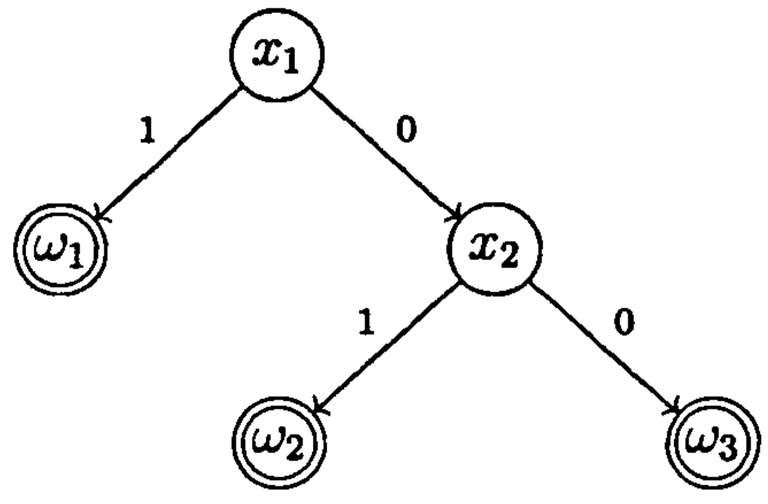


Figure 3.2: A simple cascade classifier ($c = 3$ classes, $n = 2$ features).

There are two major design aspects of this type of tree classifier to be considered. 1) A two class problem needs to be created at each non-terminal where one class is set apart and the remainder of classes are combined into a single class. 2) A feature or feature subset needs to be selected to determine the split at each non-terminal node.

The two class problem at each node may be created in the same way as the creation of the two class problem in Proposition 2, Section 2.3.4. The average value of all remaining class-conditional probabilities can be taken to create the second class. The next section looks at ways the feature selection for the non-terminal nodes may be considered.

3.2.2 Error Calculation

Consider the example given in Figure 3.2. This is a single-feature-split cascade tree with three classes. Without loss of generality accuracy may be considered in place of error as $P(correct) + P(error) = 1$. Equation (3.1) gives the probability of correct classification for the whole tree.

$$\begin{aligned} P(\text{correct}) = \quad & P(\omega_1)P(x_1 = 1|\omega_1) + P(\omega_2)P(x_1 = 0, x_2 = 1|\omega_2) \\ & + P(\omega_3)P(x_1 = 0, x_2 = 0|\omega_3). \end{aligned} \tag{3.1}$$

This can be extended to the general case with c classes. The features and the classes can be relabelled so that they appear in increasing order and all the children to the left take $x_k = 1$.

$$\begin{aligned} P(\text{correct}) &= P(\omega_1)P(x_1 = 1|\omega_1) + P(\omega_2)P(x_1 = 0, x_2 = 1|\omega_2) + \ldots \\ &+ P(\omega_{c-1})P(x_1 = 0, \ldots, x_{c-1} = 1|\omega_{c-1}) \\ &+ P(\omega_c)P(x_1 = 0, \ldots, x_{c-1} = 0|\omega_c). \end{aligned} \tag{3.2}$$

A dependency issue arises from equations (3.1) and (3.2). Equation (3.2) takes dependence into account by using $P(x_1 = 0, \ldots, x_i = 1|\omega_i)$.

A much simpler form of (3.2) is obtained by assuming independence of the selected features. The joint probability is

$$P(x_1 = 0, \ldots, x_i = 1|\omega_i) = P(x_1 = 0|\omega_i) \ldots \ldots P(x_{i-1} = 0|\omega_i)P(x_i = 1|\omega_i). \quad (3.3)$$

During the design of the tree either method of calculation can be chosen. Equation (3.2) will be restrictive when handling larger cascades. Equation (3.3) loses information on the dependence between the features and this may be unacceptable.

3.2.3 Methods of design for probability table data

The choice of the splitting function influences the whole design of the tree. To design a cascade tree with one feature at each split five functions were tried. These included an independent misclassification cost (IND), a dependent misclassification cost (DEP), a Gini impurity function (GINI), an Information gain function (INFO) and a random function (RAND).

The data available to train these cascades is the estimated probability matrices. Each entry in the matrix is the probability that feature x_j is present given class ω_i, $P(x_j = 1|\omega_i)$. At initialisation of each of the trees the classes are assumed to be equiprobable thus all having a prior probability of $P(\omega_i) = \frac{1}{c}$. All the cascades are generated using top-down approaches.

The two cascades using the misclassification error criterion, (IND) and (DEP), use the class modular approach to choosing a single class to separate off at each level of the tree. The main difference between the two types of cascade with misclassification functions is the way the error is calculated.

Cascade IND. This cascade uses the class-modular approach by separating a single class ω_i from all remaining classes, $\omega^{(1)}$. The feature probabilities for class $\omega^{(1)}$ are calculated as the mean of all class conditional feature probabilities for the classes that have not already been separated off as a leaf of the tree. The tree is designed using a top-down procedure. At each node the class to separate off and the feature to label the node are chosen by calculating the error for every feature-class pair, (x_j, ω_i),

$$e(x_j, \omega_i) = \min \Big[P(\omega_i)(1 - P(x_j = 1|\omega_i)) + P(\omega^{(1)})P(x_j = 1|\omega^{(1)}), \\ P(\omega_i)P(x_j = 1|\omega_i) + P(\omega^{(1)})(1 - P(x_j = 1|\omega^{(1)})) \Big] \quad (3.4)$$

where $P(\omega_i)$ is the prior probability of class i. The first term in Equation 3.4 is equivalent to calculating the probability of error for feature x_j having the value 0 to separate off

class ω_i. The second term is the probability of using feature x_j with value 1 to separate off the class.

The feature-class pair giving the lowest error at each stage are selected and are then removed from all further calculations.

Cascade DEP. The cascade utilises the class modular approach selecting a single feature at each non-terminal node to separate a class. The difference to the independent cascade is the calculation of the error for each feature-class pair is influenced by the features already selected. At each non-terminal node the error is calculated for each available feature-class pair (x_j, ω_i),

$$\begin{aligned} e(x_j, \omega_i) \;=\; \min \Big[& P(\omega_i)(1 - P(x_j = 1|\omega_i))t_i + P(\omega^{(1)})P(x_j = 1|\omega^{(1)})t_{(1)}, \\ & P(\omega_i)P(x_j = 1|\omega_i)t_i + P(\omega^{(1)})(1 - P(x_j = 1|\omega^{(1)})t_{(1)} \Big] \end{aligned} \tag{3.5}$$

where t_i and $t_{(1)}$ contain the probability knowledge about the previously selected features given class ω_i and $\omega^{(1)}$ respectively. The feature-class pair minimising the error at each stage label the level of the tree. The feature-class conditional probabilities are stored in the knowledge using the t parameters.

Cascades using impurity functions. An impurity function for a decision tree measures how pure a sample of the training examples are at a given node. A "pure node" would indicate that all examples at the node originate from the same class. A completely impure node would mean that all examples at that node are from different classes. The impurity functions measure the scale of impurity between these two extremes. In practice a decision tree with pure nodes would give good results across the training data but may not be adaptive enough to cope with any new test data. A trade-off between purity and overfitting is looked for. These two cascade designs use the Gini criterion and the Information Gain criterion respectively. As we have binary features then we are looking for the feature that creates the best improvement in the respective impurity measure. With continuous-valued features we would also have to look for the best threshold level to split a particular feature on.

The Gini impurity index measures the error rate committed if a class label was drawn randomly from the distribution of labels present at the current node. That is the measure looks the purity of the predicted "child" nodes created by using a particular feature. The best assessed feature will be selected to create the next level of the tree. The Information Gain criterion is an estimate measure of the amount of useful information that is gained about $\mathbf{x}$ by using feature x_j. By looking at the possible predicted child nodes of the tree created by using feature, x_j the measure assesses its effectiveness. The processes described here enable the calculations of the impurities given non-traditional probability

table data.

Both trees update their stored knowledge of the preceding feature-value pairs within the construction of the tree. At initialisation the only knowledge held is the prior probabilities of the classes and so Pr is initialised as a vector of size c containing the prior probabilities of each class. In other words $Pr(i) = P(\omega_i)$ at initialisation. For each split the impurity functions are calculated for each available feature across all available classes,

Cascade Gini (GINI).

$$\text{Impurity}_{\text{Gini}}(x_j, \omega_i) = P_{j1}\left(1 - \sum_{i=1}^{c}(P_jleft(i))^2\right) + P_{j0}\left(1 - \sum_{i=1}^{c}(P_jright(i)^2)\right) \tag{3.6}$$

where c is the number of remaining classes.

Cascade Info (INFO). Using the negative in equation (3.7) allows the feature and class selected to be those minimising the function at each stage.

$$\begin{aligned}\text{Impurity}_{\text{Info}}(x_j, \omega_i) \;=\; & -P_{j1}\left(\sum_{i=1}^{c}(P_jleft(i)\log P_jleft(i))\right) \\ & -P_{j0}\left(\sum_{i=1}^{c}(P_jright(i)\log P_jright(i))\right)\end{aligned} \tag{3.7}$$

where vectors P_jleft and P_jright are constantly updated after each selection,

$$P_jleft = \frac{Pr(i)P(x_j = 1|\omega_i)}{P_{j1}} \tag{3.8}$$

$$P_jright = \frac{Pr(i)(1 - P(x_j = 1|\omega_i))}{P_{j0}} \tag{3.9}$$

where $Pr(i)$ is the prior knowledge for the i^{th} class, initially all entries are $P(\omega_i)$. P_{j1} and P_{j0} are values calculated from this knowledge vector $Pr(i)$ where

$$P_{j1} = \sum_{i=1}^{c} P(x_j = 1|\omega_i)Pr(i) \tag{3.10}$$

$$P_{j0} = 1 - P_{j1} \tag{3.11}$$

The feature-class pair minimising the impurity function at each non-terminal node are selected to label that level of the tree. Once the feature and class have been selected Pr is updated by replacing it with either P_jleft when the selected feature has a value of 1 or P_jright when the selected feature has a value of 0.

Cascade Random (RAND). Both the classes and features were selected randomly at

the non-terminal nodes of the tree.

3.2.4 Simulated data

The aim of the simulation was to establish the ability of the various cascade designs on different types of probability data. This would give an indication of splitting functions that were robust or successful with particular types of the probability data.

Random probability matrices with a varying number of classes and number of features were generated. The number of classes, c, ranged from three to fifteen. The number of features, f, ranged from $c-1$ to 20. For each of the pairs (c, f),

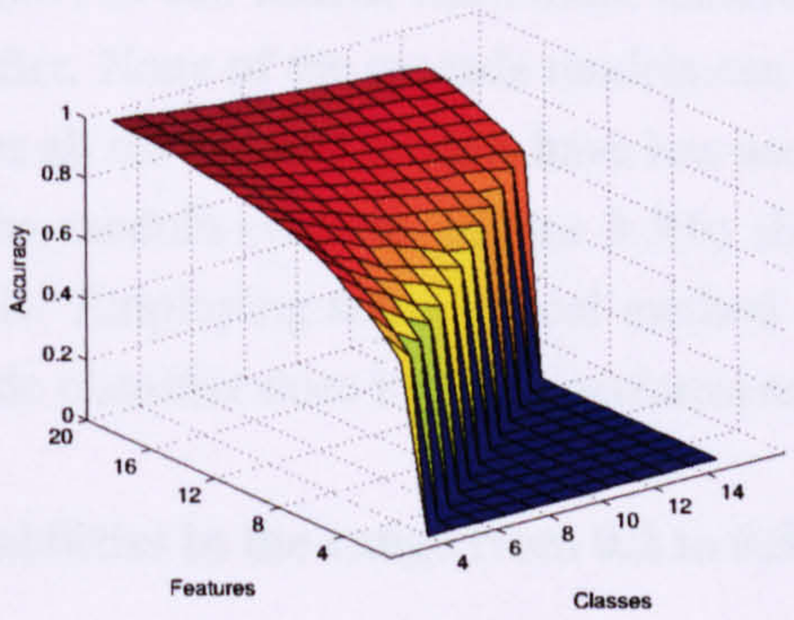
- Repeat 100 times,
  - Generate a random probability matrix A with c classes and f features.
  - Build the five cascade classifiers using A, by the described procedures.
  - Generate 100 binary test vectors for each class, 1 to c.
  - Classify each of these vectors using each cascade classifier.
  - Store the accuracy as the percentage match between the generated labels and the true class labels

The accuracy of the Naïve Bayes classifier employing all features was calculated in each trial as a comparison. The accuracy given in the results for each cascade is the average of the 100 trials. There were four experimental variants for the generation of the probability matrix, A. For the first experiment probabilities in A ranged from 0 to 1. The second experiment used probabilities generated in the range from 0.2 to 0.8. The third simulated experiment used probabilities generated in the probability tails, 0 to 0.2 and 0.8 to 1. Finally, probabilities were generated with a skewed distribution. 90% of the probabilities were generated to lie in the range 0 to 0.1 with the remaining 10% spread across the range 0.1 to 1.

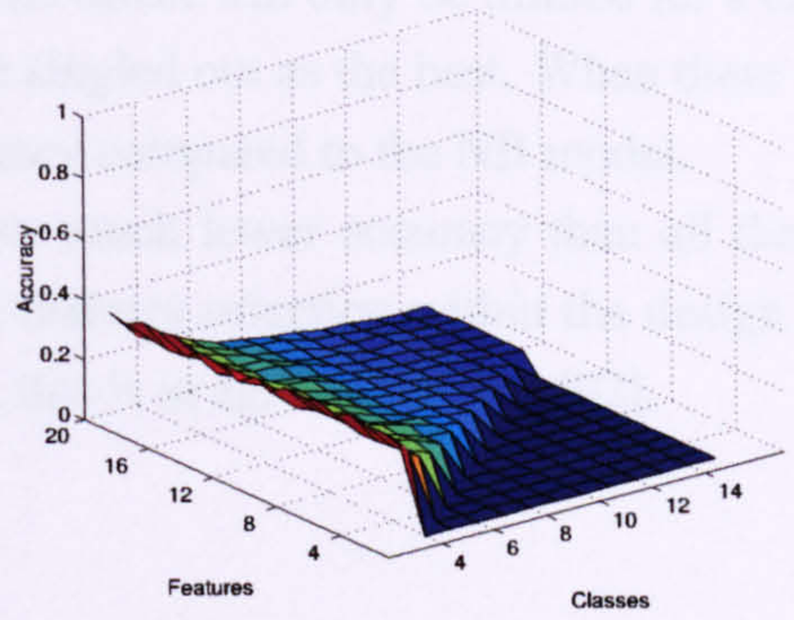
Probabilities in the range 0 to 1

Figures 3.3 (a) to (f) display the results of each of the classifiers across the range of classes and features.

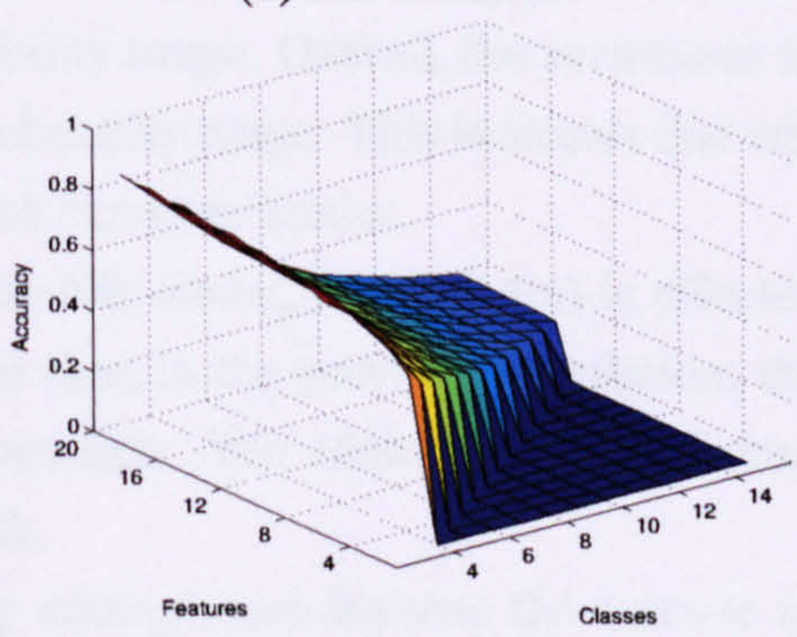
Figure 3.3(a) shows the accuracies achieved by using the NB model taking all available features into account simultaneously. As expected, as the number of features is increased the accuracy also increase (trend in the *Features*-axis). This is due to the increase in the information available from the extra features. It is also evident that as more classes are added into the task the accuracy begins to drop (trend in the *Classes*-axis).



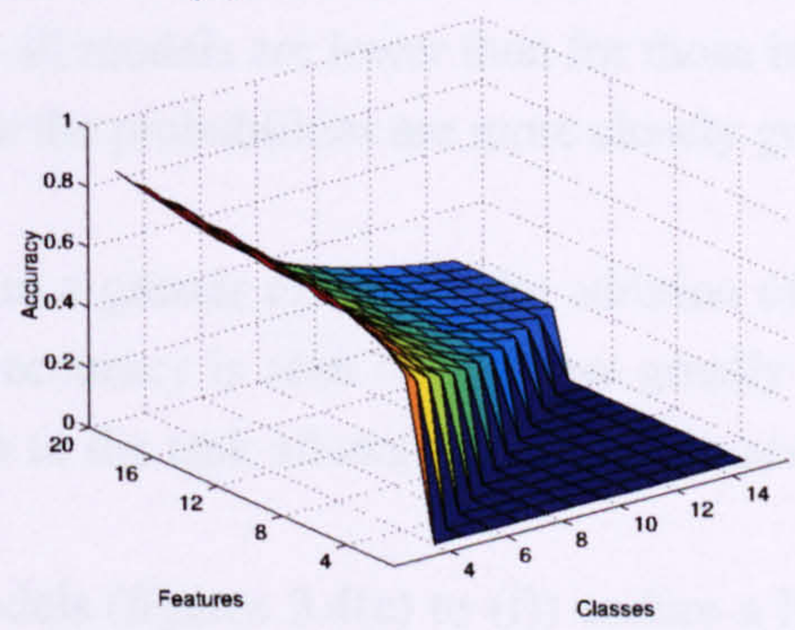
(a) All features



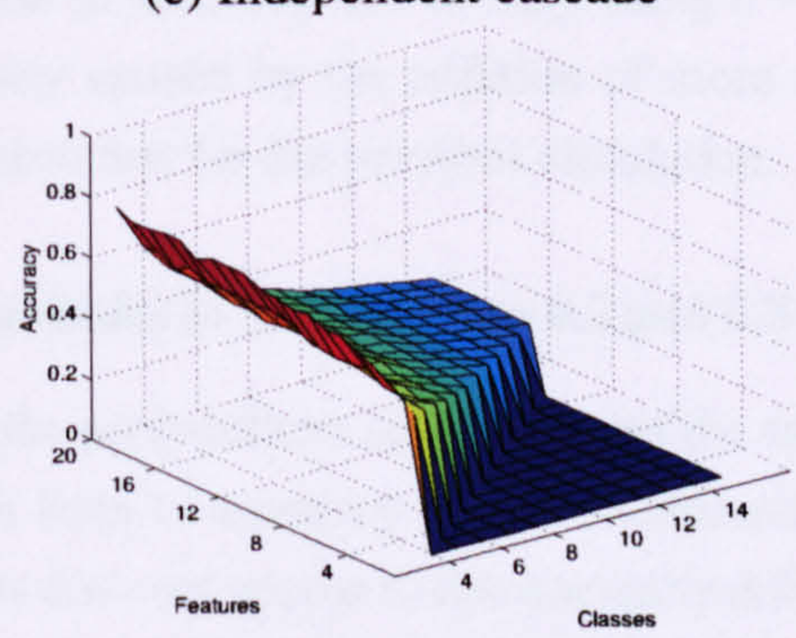
(b) Random cascade



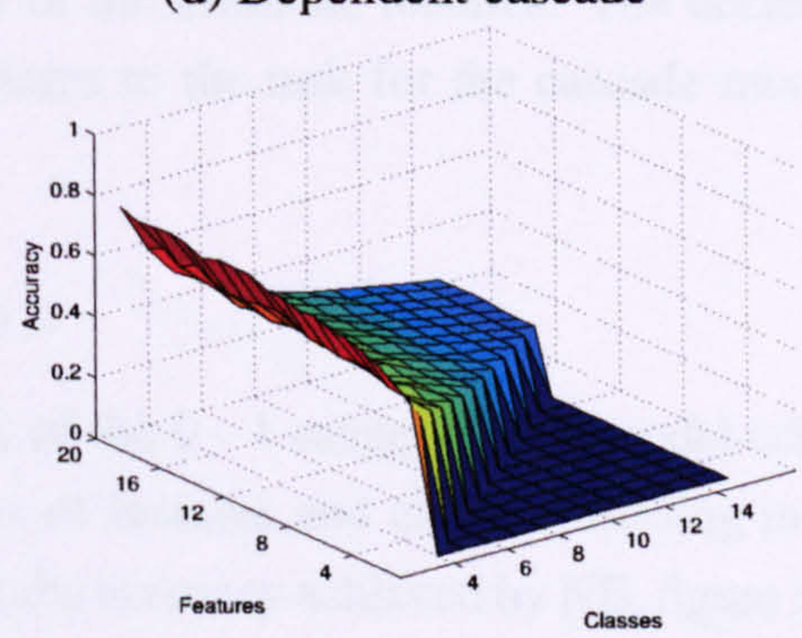
(c) Independent cascade



(d) Dependent cascade



(e) Gini cascade



(f) Info Gain cascade

Figure 3.3: Results in the 0 - 1 probability range.

Figures 3.3(c) to (f) present similar results to one another. The trend in the *Classes*-axis showing a decrease in accuracy as more classes are added is more evident than for the NB model, Fig 3.3(a). The increase in accuracy achieved by adding in more features is less obvious than for the NB model. A cascade model will select at most $c-1$ features, so while NB can benefit from more features the effect will only be limited for a cascade classifier. None of the cascade models can be singled out as the best. When there are 15 classes all the cascade models have low accuracy compared to the NB model.

The random cascade, Figure 3.3(b) shows much lower accuracy than all the other models. Employing some logical method of features selection within the design of the cascade classifier does improve performance, this is in agreement with [92].

Probabilities in the range from 0.2 to 0.8

Figures 3.4(a) to (f) display the accuracies achieved by the classifiers in the 0.2 - 0.8 probability range. Overall, the accuracies for all models are lower than for those in the 0 - 1 probability range. This indicates that when the probabilities are more closely grouped the task becomes harder.

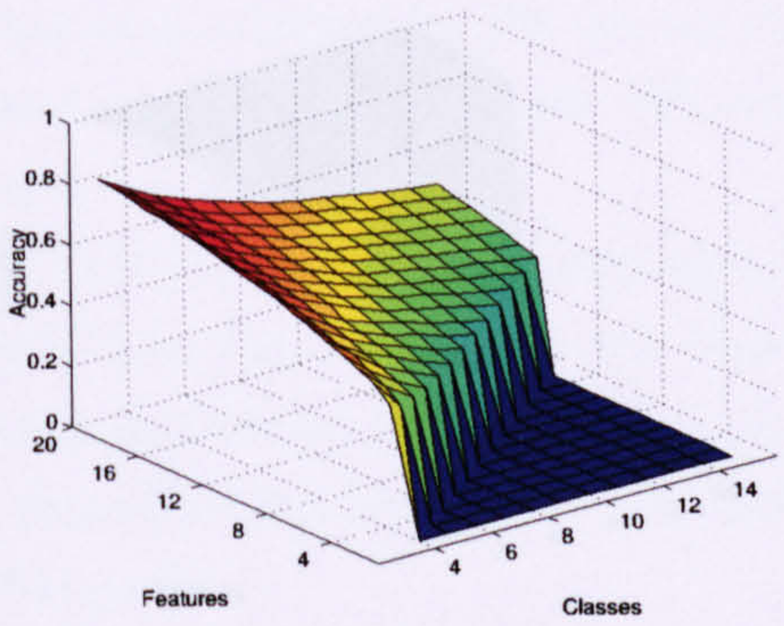
The NB model, figure 3.4(a) is affected to a greater extent by the addition of more classes than in the previous simulation; the accuracy is seen to decrease greatly in the *Features*-axis. The addition of more features to the task allows an increase in accuracy for NB.

By adding more features the cascade models (figures 3.4(c) to (f)) endure a limited increase in accuracy due to only using $c-1$ of the available features. The decrease in accuracy caused by the addition of more classes to the task for the cascade models is similar to that for the previous simulation.

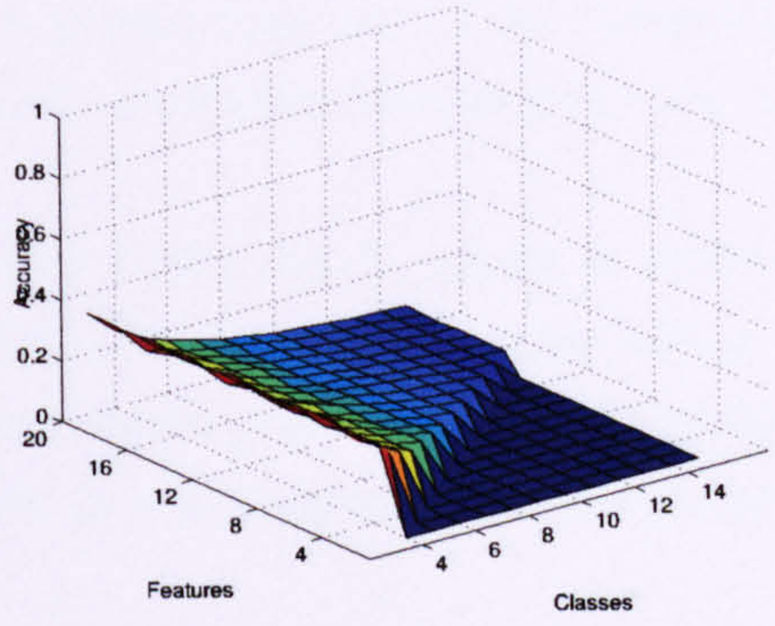
Probabilities in the tails, 0 to 0.2 and 0.8 to 1.

With the probabilities separated into the tails of the 0 - 1 range the NB model achieves a high level of accuracy for all combinations of features and classes. Adding in more classes does not appear to detrimentally affect the accuracy achieved by NB, figure 3.5(a). Again in comparison to all classifier models the random cascade performs poorly across all pairings of features and classes.

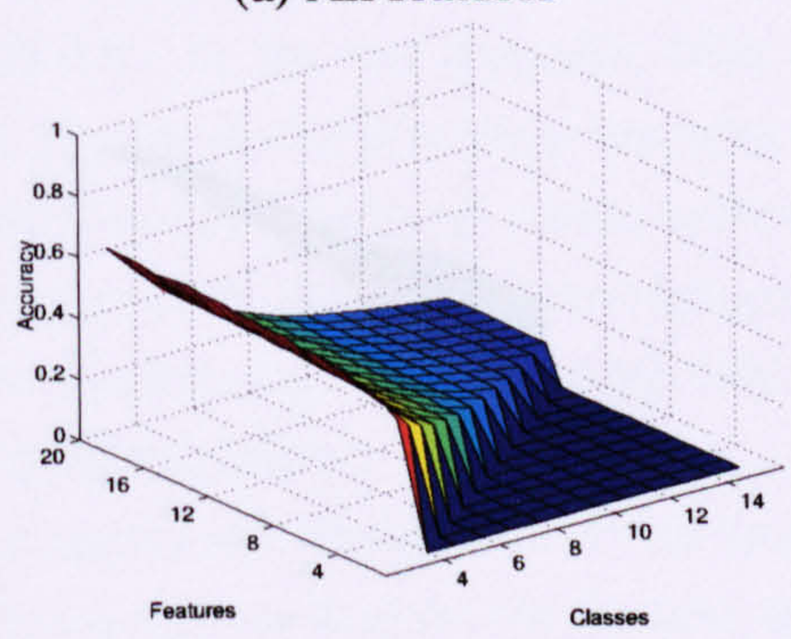
The independent and dependent cascades, figures 3.5(c) and (d) achieve overall higher accuracies than the Gini and Info gain designed cascades, figures 3.5 (e) and (f). However, adding more classes to the task causes a large drop in accuracy for all four cascade models. With 15 classes in the task the accuracy achieved by the cascade models is much lower than that achieved by the NB model.



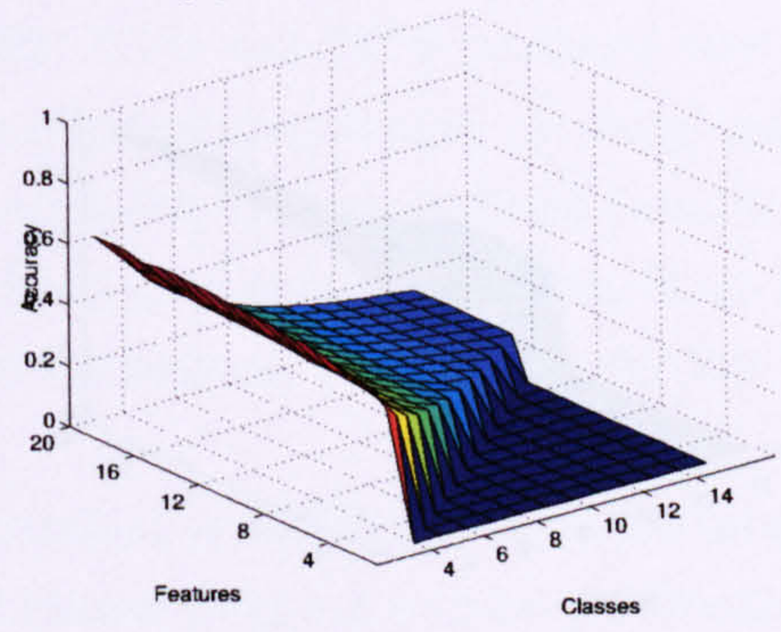
(a) All features



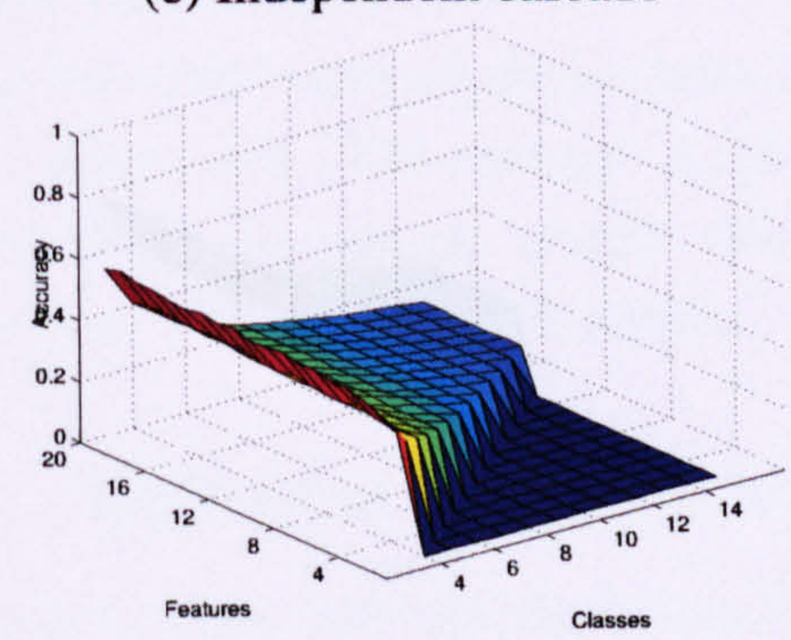
(b) Random cascade



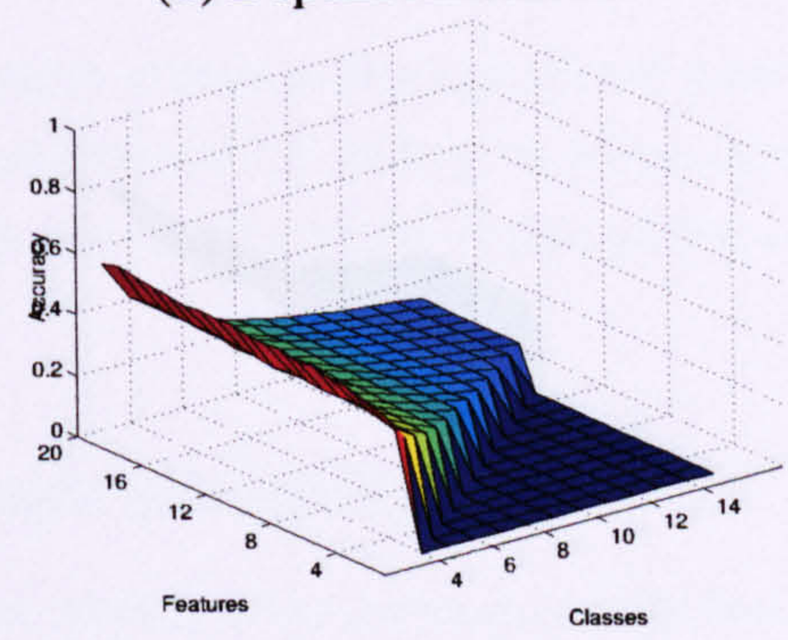
(c) Independent cascade



(d) Dependent cascade

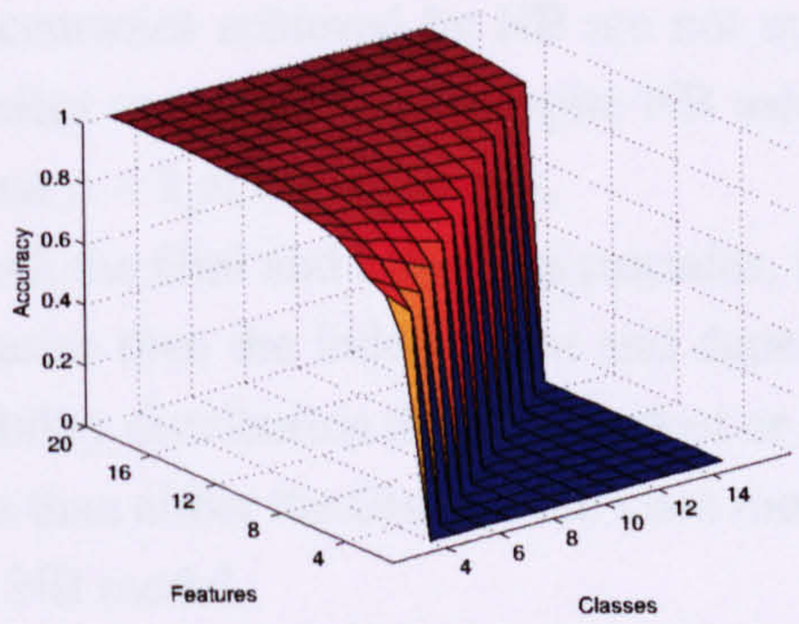


(e) Gini cascade

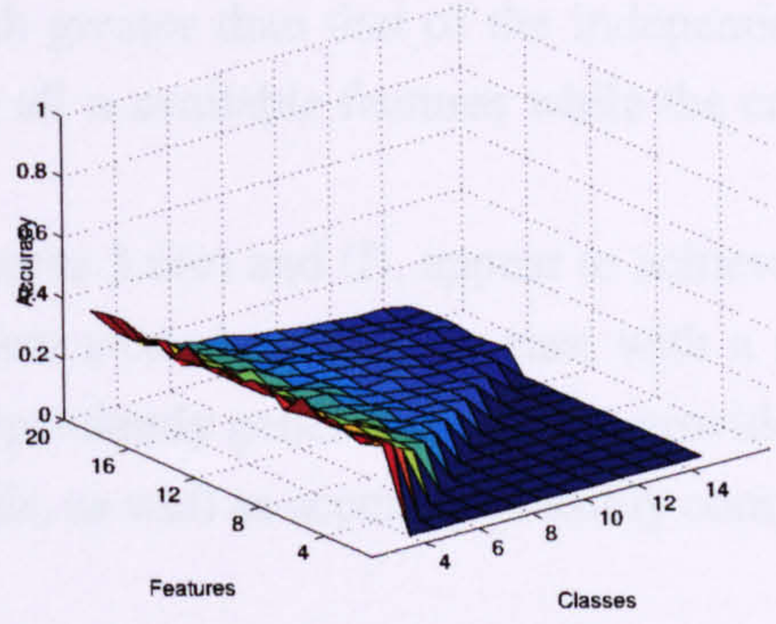


(f) Info Gain cascade

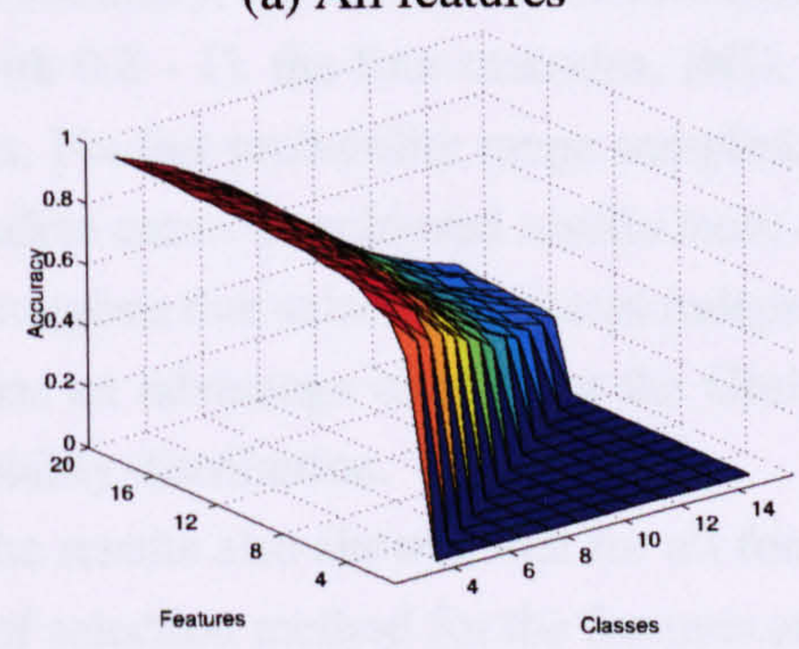
Figure 3.4: Results in the 0.2 - 0.8 probability range.



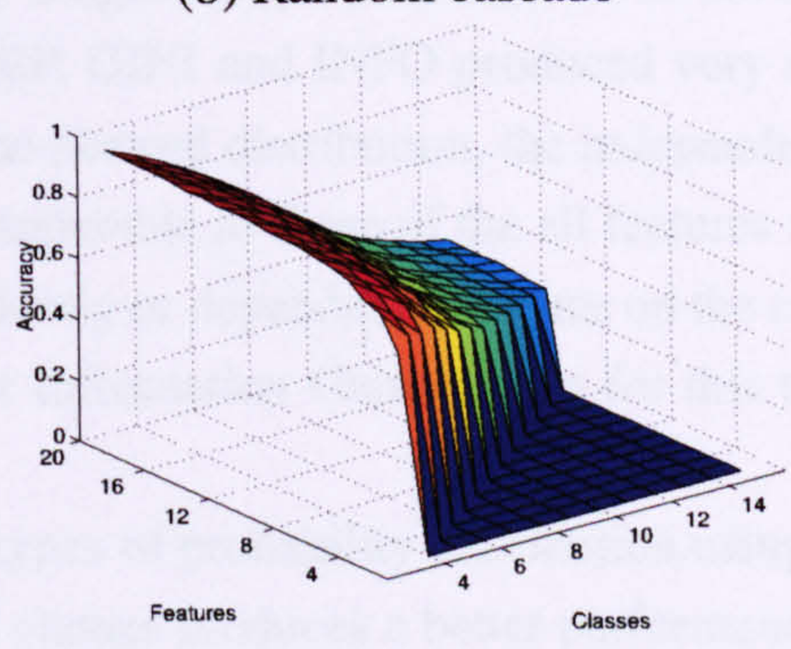
(a) All features



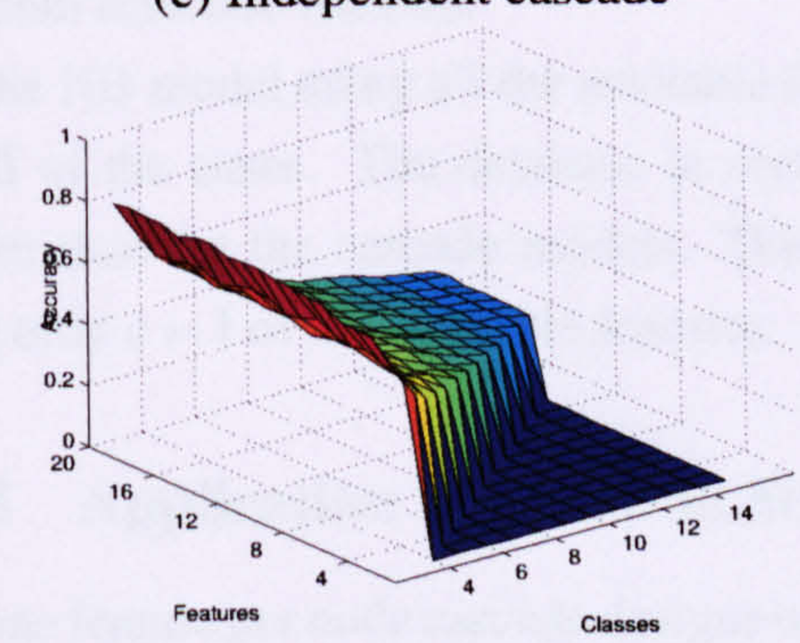
(b) Random cascade



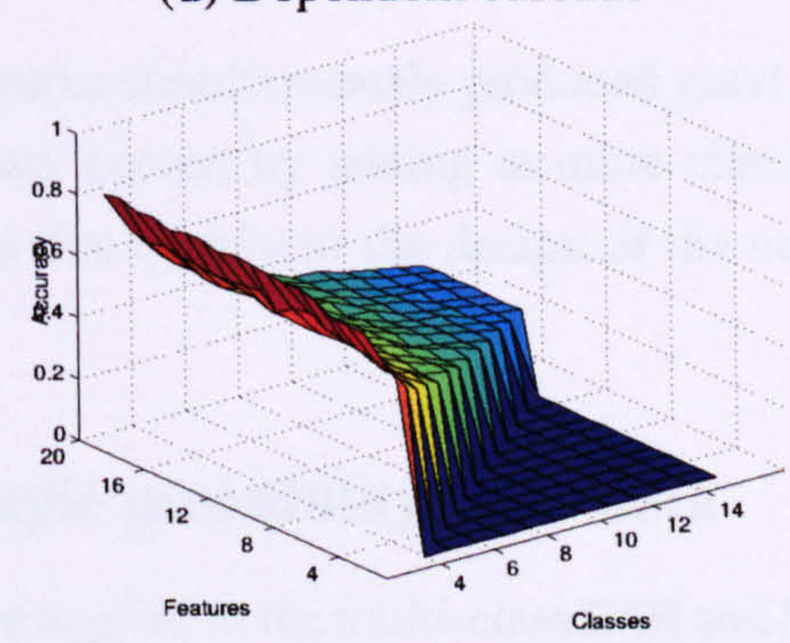
(c) Independent cascade



(d) Dependent cascade



(e) Gini cascade



(f) Info Gain cascade

Figure 3.5: Results in the 0 - 0.2 and 0.8 - 1 probability tails.

Skewed probabilities

Having the probabilities in skewed distributions causes a reduction in the accuracies achieved by NB using all the available features, Figure 3.6(a). Figures 3.6(c) and (d) for the independent and dependent cascade appear similar to the NB model, Figure 3.6(a). The accuracies achieved by NB are not much greater than that of the independent and dependent cascades, this is despite NB using all n available features while the cascades only use $c-1$ of the n features.

Both the Gini and Info Gain cascades, figures 3.6(e) and (f), appear to achieve lower accuracies then the independent and dependent cascades. For the case with a skewed probability distribution the independent or dependently generated cascade provide better results than either the Gini or Info Gain models, as well as accuracy possibly comparable to the NB model.

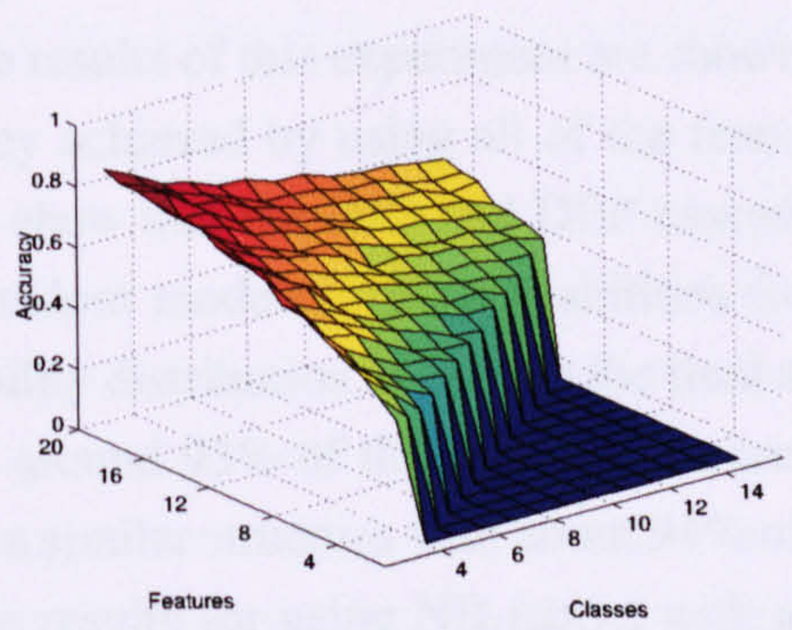
In summary, for the first three probability ranges sampled (0 to 1, 0.2 to 0.8 and 0 - 0.2 with 0.8 - 1), the four cascades, IND, DEP, GINI and INFO produced very similar results. For last probability range sampled, the skewed distribution, the independent and dependent cascades achieved results more comparable to those of the all features model. This suggests that selecting features independently or dependently for use on the cascade tree has an advantage over using the Gini or Information Gain criteria for this type of probability distribution.

The results also showed that for all four types of probability distribution using some type of selection method for the features and classes produces a better performance than a random selection method.

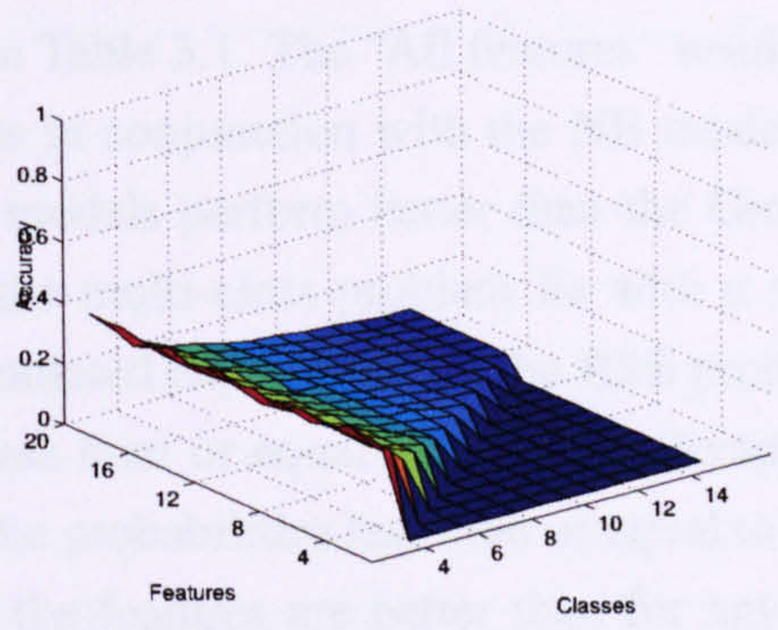
The NB model using all the available features simultaneously produced good results for all of the cases. The decrease in accuracy caused by adding in more classes was smaller than for the cascade models. This is due mainly to the design of the cascades using only $c-1$ of the available features.

3.2.5 Application to BSE and Scrapie probability table data

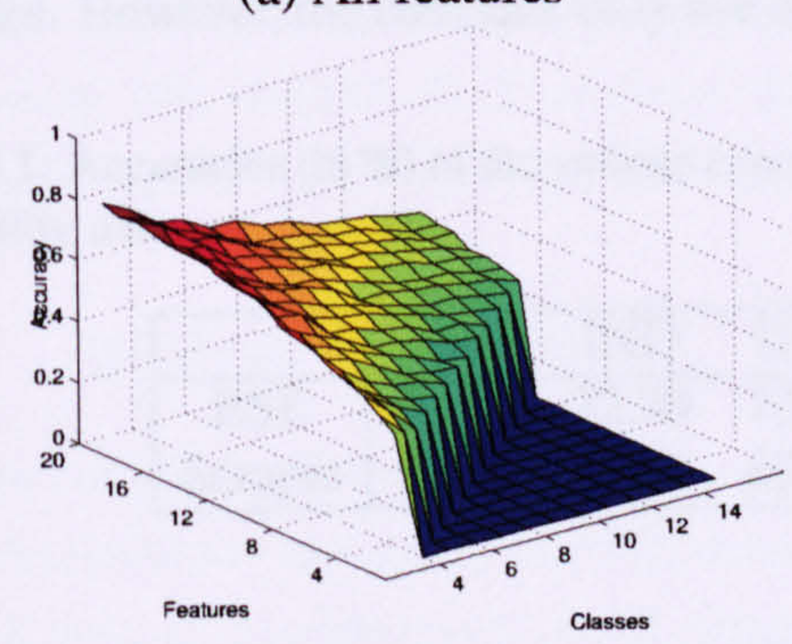
The one feature per node cascade designs were applied to the multi-class BSE and Scrapie non-traditional probability data. The BSE probability data contains 57 classes and 242 features. The Scrapie probability data contains 63 classes and 285 features. The experiment was conducted as for the simulated data but with the BSE and Scrapie matrices replacing the generated matrices. Using the probability tables a cascade decision trees were constructed for the two problems. Binary vectors representative of the probability distributions given for class were generated as the test cases to give the accuracy of the procedures.



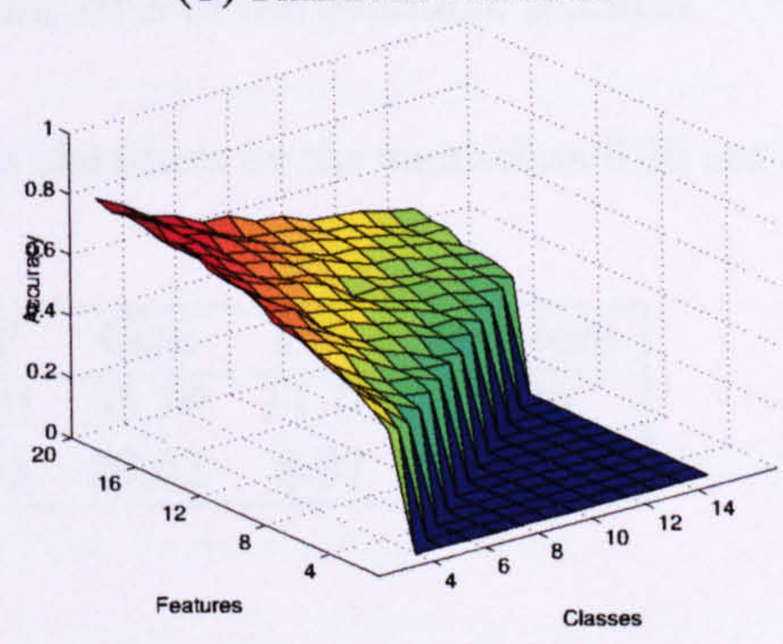
(a) All features



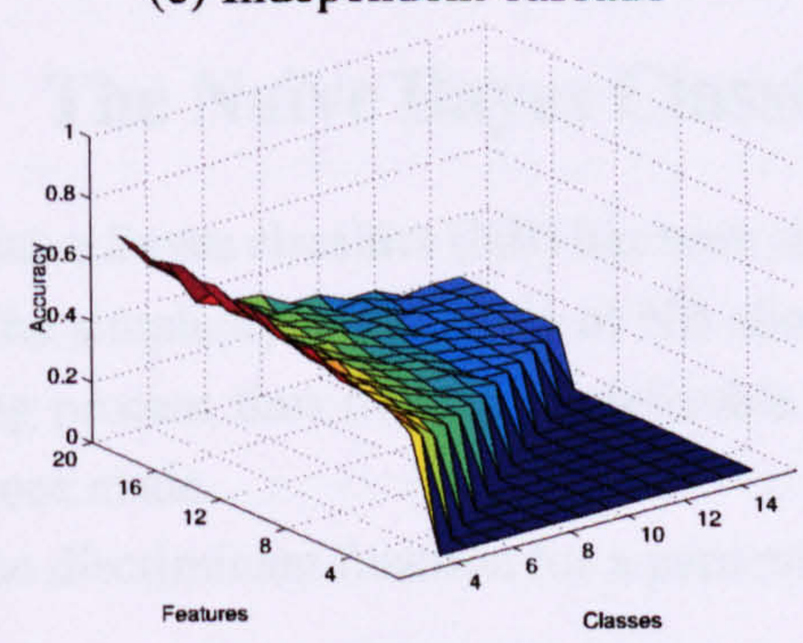
(b) Random cascade



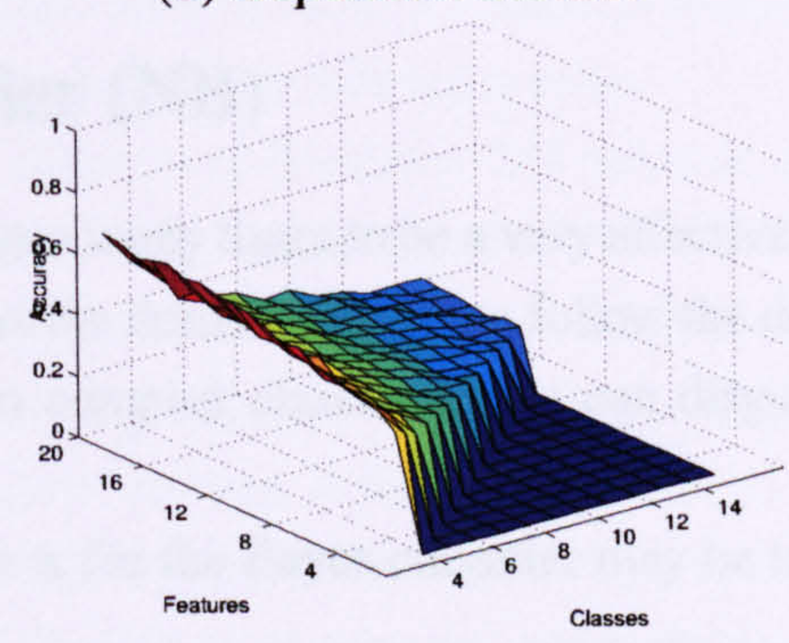
(c) Independent cascade



(d) Dependent cascade



(e) Gini cascade



(f) Info Gain cascade

Figure 3.6: Results in the skewed probability distribution.

- Build the five cascade classifiers using the BSE / Scrapie matrix
- Generate 100 binary test vectors for each class, 1 to c
- Classify each of these vectors using each of the cascade classifiers

The results of this experiment are shown in Table 3.1. The "All features" result is the accuracy achieved by using all of the features in conjunction with the NB model. The results show that the IND and DEP cascade models perform better than the Gini, Info and Random models. The probabilities for the multi-class problem lie with a skewed probability distribution as seen in the final simulated experiment. In the BSE probability matrix around 93% of the probabilities are less than or equal to 0.1. The Scrapie data shows a similar structure with about 94% of the probabilities less than or equal to 0.1.

The results for using NB model with all the features are better than for any of the cascades. However, the cascades only use about 20% of the available features.

Table 3.1: Accuracies (in %) of the various cascade classifiers for the multi-class BSE and Scrapie probability data.

	All	IND	DEP	Gini	Info	Random
BSE	99.30	72.98	73.00	33.79	11.70	1.75
Scrapie	97.35	63.21	63.51	29.62	8.37	1.59

3.3 The Naïve Bayes Classifier (NB)

The Naïve Bayes classifier (NB) has been shown many times to be a very effective classifier. The simplicity and stability of NB allows the domain experts to follow the decision making process thus making it preferable to complex classifiers that can disguise the decisions made.

The discriminant function for a particular $\mathbf{x}$ for the Bayes classifier may be taken to be

$$g_i(\mathbf{x}) = P(\omega_i)p(\mathbf{x}|\omega_i),\ i = 1 \ldots c \tag{3.12}$$

where $P(\omega_i)$ is the prior probability of class ω_i and $p(\mathbf{x}|\omega_i)$ is the class conditional pdf of $\mathbf{x}$. The class, ω_i that gives the maximal value to the discriminant function will be assigned to $\mathbf{x}$.

If the n features of a problem are considered to be class-conditionally independent,

the class-conditional pdf may be calculated as a product of n individual pdfs,

$$p(\mathbf{x}|\omega_i) = \prod_{j=1}^{n} p(x_j|\omega_i),\ i = 1 \ldots c \tag{3.13}$$

By using this function in the discriminant the classifier becomes the Na¨ive Bayes model,

$$g_i(\mathbf{x}) = P(\omega_i) \prod_{j=1}^{n} p(x_j|\omega_i),\ i = 1 \ldots c. \tag{3.14}$$

Under the condition of class-conditionally independent features the classifier of choice would be the NB. In the probability table data the probabilities for the presence of a feature (symptom) given a particular class (disease) were considered independently. In other words, no other features were considered when the probability was calculated making the features class-conditionally independent of one another. In this case it is sensible to consider NB, despite the knowledge that the assumption of the probabilities being independent of one another is likely, in reality, false.

3.3.1 Applications of NB

NB has been applied widely across classification tasks. The simplicity, efficiency, ease of implementation and interpretability have all led to its widespread use.

The ease of interpretation of the model is especially evident in applications to the medical domain [50, 143]. Kononenko [73] states the preference of medical specialists to use NB as their model of choice due to its logical decision making process.

A review by Lewis [87] states that, over 40 years of literature, NB methods account for most applications of supervised learning to information retrieval. The success of NB in this domain is surprising as the assumptions of NB almost never hold for textual domains. Lewis notes three ways in which adapting NB has been attempted, 1) relax the independence assumption - this has not had great success in the text domain, 2) modify the feature set to make the assumption true - results so far have shown that it is hard to correlate the impact of any modifications, 3) reason as to why the assumptions are not needed - such as for the two class case allowing a weaker "dependence" assumption.

McCallum and Nigam [98], Bennett [8] and Forman and Cohen [45] all agree that NB is amongst the top competitors for classification tasks in text domains. Forman and Cohen go on to show that NB tends to be insensitive to the distribution of prior probabilities in the training data. This is good for text domains where the class of interest may only have a few examples compared to the class of non-interest. This characteristic has also been put to good use for web searches, [124] and image analysis [66].

As NB does not guarantee accurate probability estimates it seems an unlikely candidate to use in ranking tasks. However, studies have used the model successfully in this domain [159,163]. Whilst it may be true that accurate probability estimation of NB would improve the performance for ranking, the converse is not true. Improving performance for ranking does not necessarily improve the accuracy of the probability estimation.

3.3.2 A Meta-analysis of NB adaptations

The possible inaccuracy of the probability estimates of NB leading to imprecision of the entire model has given rise to an abundance of proposed adaptations to the model in the literature. Adaptations to NB could compromise the models elegance and computational simplicity [54].

Meta-analysis is a technique for comparing various studies to try and draw out a consensus of opinion amongst the research. The technique is mainly used in social sciences, biology and psychology [24, 60, 90]. The review of scientific studies can be done in an unscientific way using traditional literature review methods. Meta-analysis is a more scientific method of reviewing various related studies [155]. As the analysis is structured any bias imposed by the authors own personal views is avoided. The interpretations of the studies findings are effectively translated in relation to one another avoiding possible misinterpretations. Studies that have used meta-analysis in pattern recognition have covered classification algorithms [137], face recognition algorithms [112] and clustering algorithms [63]. These studies have carried out a quantitative analysis of the performance of the group of algorithms. The meta-analytic procedure consists of a series of five steps.

1. Create an encoding scheme to apply to the raw data (Published studies). This involves formulating the question that needs to be answered by the analysis and a method of how to code characteristics from the data.

2. Select a sample of studies. The choices of why the selections were made should be explained here.

3. Encode each sample study with the scheme created at step 1. This involves taking the studies and transforming them into data that can be analysed

4. Analyse the encoded data by statistical techniques (adapted where necessary). The chosen statistics will produce a summary of the studies under review

5. Output any patterns or structures found. This will be the answers to the questions, and possibly the discovery of other patterns.

In place of a traditional survey of the various adaptations made to NB a meta-analytic study can be formulated. This will give a more structured view of the alterations that have been tried.

3.3.3 Formulating an encoding scheme

To create an encoding scheme the questions that the analysis is aiming to answer need to be formulated. The questions for this analysis are

1. Which methods are structurally similar?

2. What relationships, if any, are there between the adaptations?

3. What techniques have been used to try and optimise upon the original NB model?

To be able to answer these questions important characteristics of the adapted algorithms need to be identified. These characteristics should be related to the structural differences of the model compared to the original NB. The 19 chosen characteristics are listed in Table 3.2. All characteristics are binary questions with "Yes" encoded as 1 and "No" encoded as 0. Each method can then be described by a binary vector of length 19 relating to the characteristics. For those characteristics that are not self-explanatory a small explanation is included here.

Characteristic 2, "Was the NB formula (equation 3.14) adapted?" indicates whether a variation of the original NB formula is used, (for example, equation 3.15 of Bayesian networks).

Characteristic 7, "Were eager learning methods used?" Machine learning distinguishes between two types of supervised learning, eager and lazy. *Eager learning* takes the labelled training data and trains a classifier model on it. *Lazy learning*, on the other hand, stores all the training data until the time of classification. Eager learning takes less storage space than lazy learning as only the classification model need be stored rather than the whole training set. New cases can be easily added to the lazy model without a need for re-training. Labour intensity may be the deciding aspect on which method to use, with eager learning being intensive at training and lazy learning being intensive at the time of classification. A special issue of *Artificial Intelligence Review* contains a comprehensive set of studies that review and investigate the lazy/eager learning distinction. Aha provides a concise overview of the area of lazy learning in the introductory editorial for this special issue [1].

Characteristic 8,"Were Bayesian networks used?". Bayesian networks originate from work done by Pearl in 1988 [111]. A Bayesian network is an annotated directed acyclic

Table 3.2: The 19 binary features used in the description of the NB adapted methods.

1	Was the data discretised initially
2	Was the NB formula (equation 3.14) adapted
3	Was there any feature selection prior to using NB
4	Were ensembles of classifiers used
5	Were 1^{st} order dependencies considered
6	Were dependencies greater than 1^{st} order considered
7	Were eager learning methods used
8	Were Bayesian networks used
9	Were Decision Trees used
10	Was Clustering used
11	Were Fuzzy classification methods used
12	Was Sequential Forward Selection used
13	Was Sequential Backward Selection used
14	Were Genetic algorithms used
15	Was Boosting used
16	Was Feature extraction used
17	Did the method update the probability table produced by NB
18	Was the method tested on a wide range of data sets (10+)
19	Was the method using any type of randomness in its calculations

graph, G, where vertices represent the features and the edges represent direct dependencies between the two vertices (features). By direct dependencies we mean that if there is a directed edge from x_i to x_j it may be read as feature x_i "causes" x_j. The Bayesian network defines a unique joint probability distribution over the set of features given by

$$p(\mathbf{x}) = \prod_{i=1}^{n} p(x_i|\text{Parents of } x_i \text{ in } G). \tag{3.15}$$

When all the nodes have the same parent corresponding to a class, the pdf is conditioned by this class. For example, the pdf in Figure 3.7(a) models the case of conditionally independent features, as assumed by NB

$$p(\mathbf{x}|\omega_1) = p(x_1|\omega_1)p(x_2|\omega_1)p(x_3|\omega_1). \tag{3.16}$$

In this case each feature node has the class node as its only parent. The general Bayesian network in Figure 3.7(b) corresponds to the following class-conditional pdf,

$$p(\mathbf{x}|\omega_1) = p(x_1|\omega_1)p(x_2|x_1,\omega_1)p(x_3|x_1,\omega_1). \tag{3.17}$$

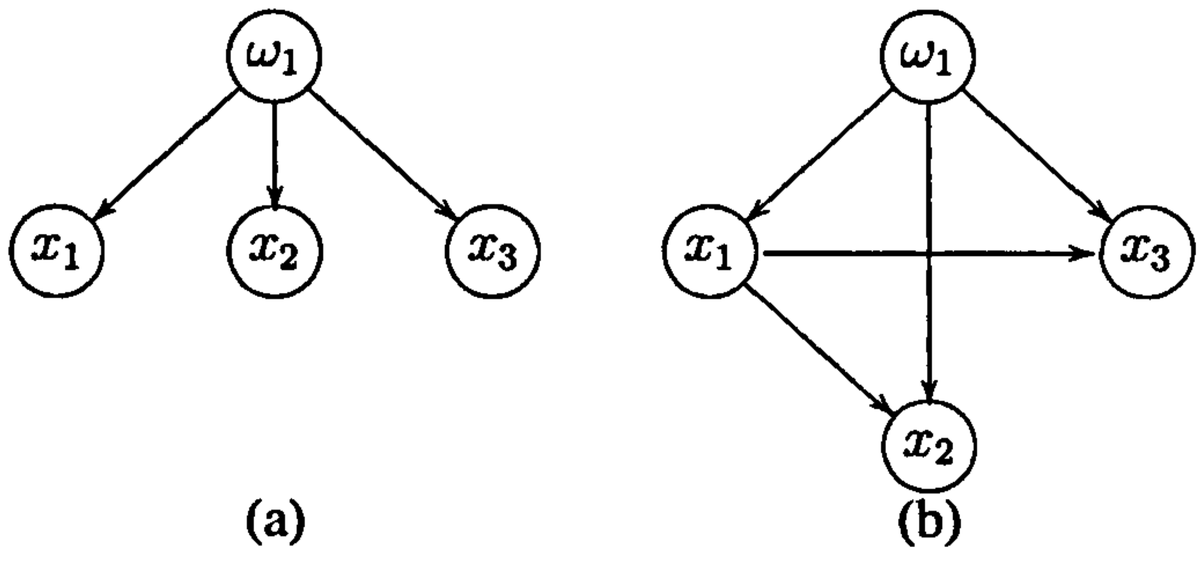


Figure 3.7: A Bayesian network for (a) conditionally independent features, (b) a general model.

A Bayesian network encodes that "each feature is independent of its non-descendants in the graph given its parents". These independencies are used to reduce the number of parameters needed to characterise a probability distribution. The idea is to induce the best network that models the probability distribution given by the training data. This is generally done in practice using a heuristic search to find the best candidate over the space of networks. This process relies on a scoring function that assess the merits of each candidate network.

The maximum number of edges from one feature to other features is called the level of dependency of the network. Figure 3.7(a) shows a zero dependency network meaning that no dependencies between the features are taken into account. The network in Figure 3.7(b) has a dependency level of one, as both x_2 and x_3 depend on x_1. If we were to add another edge in Figure 3.7(b) directed from x_2 to x_3 then we would achieve a network of level two dependency, (x_3 would be "caused" by x_1 and x_2).

Characteristic 15, "Was boosting used?" Boosting is a machine learning technique that focuses the learning of a model on the cases that are hard to classify. The first polynomial time boosting algorithm was proposed by Schapire in 1990 [127]. This was improved upon by Freund in 1995 to make it more efficient [46]. This improvement was optimal in many cases but had practical drawbacks. Adaboost proposed by Freund and Schapire [47] solved many of the practical difficulties of boosting and is widely used today. An introduction to boosting algorithms is given by Schapire in [128]. The boosting procedure successively classifies weighted versions of the training data. The data is reweighted after each classifier is built. The reweighting depends upon how successful the previous classification was. This ensures that the hard to classify cases are brought to the attention of the classifier. Direct application of boosting does not generally improve the performance of NB due to the fact that NB classifiers are very stable.

Characteristic 17, " Did the method update the probability table produced by NB?" NB naturally produces an estimate of $p(\mathbf{x}|\omega_i)$, some of the methods adjust this estimate once it has been calculated. This is different to characteristic 2 due to the NB formula

(equation 3.14) remaining untouched. An adaptation to the NB formula affects all estimates uniformly. Allowing a method to update the estimates afterward allows selective updating; not every estimate will necessarily be updated.

With the encoding scheme in place, the selection of the adaptations to include in the analysis must start.

3.3.4 Selection of sample studies

A total of 37 studies were selected from a variety of peer-reviewed journals and conference proceedings. The studies were selected retrospectively. A Google search using the keywords "Improve Naive Bayes" produced the majority of the more popular, well cited studies. The references of these studies were checked to find other studies they cited. In turn the references of these papers were also checked. This was continued until the 37 studies were found. This gave coverage within the bounds of these 37 studies that are all linked by the Google search and their references. The aim was to give coverage of the selected studies looking at the adaptations applied to the general NB model. Any domain-specific adaptation was excluded as these methods may have different aims and problems to solve with regard to NB. A comparison study by Lewis [87] demonstrates this by concentrating on the particular issues that arise in applying NB to textual data.

The methods are listed in Table 3.3 in an order which will aid their explanation. The earliest reference in this table is Kononenko's Semi-Naïve Bayesian classifier, 1991, [72]. As the papers have been published in peer-reviewed journals and conference proceedings this does bias the study toward those models that have had success (although in some cases only marginal) over NB. This biasing is justified as these models are likely to be tried on new data sets as a "straw man". The rest of this section comprises of a brief description of each of the NB adapted methods.

1 Naïve Bayes, (NB). One of earliest references of NB is by Minsky and Papert in *Perceptrons*, Chapter 12. The reference used in the table is for the one of the most widely used reference of NB, by Duda and Hart, [36] (reference is the second edition of this book). The "original" Naïve Bayes model we refer to uses discriminant function in the form of equation 3.14, to model the distribution given by the training data.

2 Decision Tree Naïve Bayes hybrid, (NBTree), Kohavi, 1996 [69]. A standard decision tree is grown with a NB deployed at the leaves, creating a penultimate layer of the tree. These classifier leaves output a class label when a new case is submitted to them, acting as the final decision of the tree. The NB at the classifier leaves is grown using the training data that arrives at the leaf after being processed by the tree. The design is an attempt to approximate whether the generalisation accuracy for a NB at each leaf is higher than a single NB at the current node.

Table 3.3: The selected adapted NB methods.

Number	Name (Abbreviation used in this study)	Year	Ref.
1	Original Naïve Bayes (NB)	1969	[36]
2	Decision Tree NB Hybrid (NBTree)	1996	[69]
3	Iterative Bayes (IB)	2003	[49]
4	Ensemble Feature Selection NB (EFSNB)	2002	[146]
5	Sequential Forward Selection NB (SFSNB)	2003	[145]
6	Sequential Backward Selection NB (SBSNB)	2003	[145]
7	Genetic Algorithm NB (GANB)	2003	[145]
8	Tree augmented Bayesian Network (TAN)	1997	[48]
9	Probability Dependence Tree 2 (PDT2)	1999	[67]
10	SFS then NB (SFS to Bayes)	1994	[82]
11	Sequential Forward Selection & Joining (SFSJ)	1996	[110]
12	Sequential Backward Selection & Joining (SBSJ)	1996	[110]
13	K-Dependence Bayesian Network (KdepBN)	1996	[123]
14	Aggregating One Dependence Estimators (AODE)	2005	[151]
15	Lazy Bayesian Rules (LBR)	2000	[165]
16	Conditional Independence Tree (CITree)	2004	[162]
17	Selective Neighbourhood NB (SNNB)	2002	[156]
18	Independent Component Analysis NB (ICABayes)	2002	[15]
19	Improved NB Classification (INBC)	2001	[91]
20	Lazy version of TAN (LazyTAN)	2002	[149]
21	Interval Estimation NB (IENB)	2003	[120]
22	Random TAN (RTAN)	2004	[93]
23	Adapted Boosting for NB (ActiveBoost)	2004	[148]
24	Adjusted Probability NB Classifier (APNBC)	1998	[152]
25	Homologous NB (HNB)	2002	[59]
26	Fuzzy NB (FNB)	2002	[139]
27	Interpretable Boosted NB (IBNB)	1998	[117]
28	Large Bayes (LB)	1999	[99]
29	Semi-NB (SNB)	1991	[72]
30	Kernel-based & Joining NB (KJNB)	2004	[29]
31	NB Committees (NBC)	1998	[164]
32	Boosted NB (BNB)	2002	[30]
33	Clustered NB (CNB)	2003	[147]
34	Neuro-Fuzzy NB (NFNB)	1999	[105]
35	Minimum Description Length Principle in NB (MNB)	2000	[68]
36	Selective Bayesian Classifier (SBC)	2002	[116]
37	Boosted Levelled NB Trees (BLNBT)	1999	[142]
38	Extended Bayes (EB)	2004	[121]

3 Iterative Bayes, (IB), Gama, 2003 [49]. IB begins with a contingency table built by the standard NB, an iterative procedure then updates these tables by cycling through all the training examples.

4 Ensemble Feature Selection Naïve Bayes, (EFSNB), Tsymbal and Puuronen, 2002 [146]. An ensemble of NB classifiers is created. Each NB is trained on a randomly sampled subset of the original set of features. Various methods of ensemble integration are considered.

5, 6, 7 Sequential Forward Selection Naïve Bayes, (SFSNB), Sequential Backward Selection Naïve Bayes (SBSNB), Genetic Algorithm Naïve Bayes, (GANB), Tsymbal *et al*, 2003 [145]. In each of these methods an ensemble of NB classifiers is created. Each NB is trained on a subset of the original features. The selection of the feature subsets is sequential forward selection, sequential backward selection and genetic algorithm respectively. SFS and SBS are iteratively applied to obtain the base classifiers. GA operations of mutation and crossover are iteratively applied to provide the subsets of features for the base classifiers. Again, various methods of ensemble integration are considered.

8 Tree Augmented Bayesian Network, (TAN), Friedman *et al*, 1997 [48]. A network is grown in which the class variable has no parents. Each feature has as parents the class variable and at most one other feature, making it a one-dependency network. The tree structured Bayesian network is constructed by the Chow and Liu procedure that finds the maximal weighted spanning tree in a graph. This is done using the conditional mutual information between the two features given the class variable.

9 Probability Dependence Tree 2, (PDT2), Keogh and Pazzani, 1999 [67]. The network is initialised to NB, Figure 3.7(a). Each node is considered for Superparent in turn by extending edges to every node without a parent, other than the class node. The node that increases accuracy the most is chosen as Superparent. The accuracy of extending an edge from Superparent to each available node is calculated. (An available node has only the class node as a parent). The edge giving the best accuracy is kept and the search for the next Superparent is started. The process stops when no significant gains in accuracy are made by introducing any more Superparents.

10 Sequential Forward Selection and Naïve Bayes, (SFS to Bayes), Langley and Sage, 1994 [82]. A subset of the original features is chosen by sequential forward selection. A NB is then trained using only these selected features.

11, 12 Sequential Forward Selection and Joining, (SFSJ), Sequential Backward selection and Joining, (SBSJ), Pazzani, 1996, [110]. Subsets of features are selected for the NB classifier. At each step of the feature selection, one of the features may be added to (SFS), or removed from (SBS) the subset, or a feature may be joined to another one already present in the subset. The joining operation creates a new compound feature

to replace the original features. For example, consider feature set $\{x_1, x_2, x_3\}$ and another feature set where x_1 and x_2 are joined as a single new feature, $x_{1,2}$, to replace the pair. In the first case, the approximation of the class-conditional pmf for class ω_k is $P(\mathbf{x}|\omega_k) = P(x_1|\omega_k)P(x_2|\omega_k)P(x_3|\omega_k)$, while in the second case, this approximation is $P(\mathbf{x}|\omega_k) = P([x_1, x_2]^T|\omega_k)P(x_3|\omega_k)$. As more features are joined the probability estimates of the compound feature becomes less reliable than that of the individual features and this must be taken into account. More than two features may be joined by successive applications of the joining operation. Each step is only taken if the improvement in accuracy made by the change exceeds a pre-defined threshold.

13 K-Dependence Bayesian Network, (KdepBN), Sahami, 1996 [123]. The space of k dependencies is searched for the most appropriate Bayesian network for the problem. The value of k is decided by the user.

14 Aggregating One Dependence Estimators, (AODE), Webb, Boughton and Wang, 2005 [151]. The class-conditional pdf, $P(\mathbf{x}|\omega_k)$ is approximated as the average of n "mini"-pdfs, one for each feature. Each such "mini"-pdf is calculated from a one dependence Bayesian network, where the respective feature is the parent of all nodes (the other parent being the class label). In other words, a "mini"-pdf for feature x_i given class ω_k is $\prod_{j=1}^{n} P(x_j|\omega_k, x_i)$. The term for x_i is only taken in the summation for approximating $P(\mathbf{x}|\omega_k)$ if the number of objects in the training set with value x_i exceeds a predefined threshold.

15 Lazy Bayesian Rules, (LBR), Zheng and Webb, 2000 [165]. For each test case LBR generates an appropriate rule with a conjunction of feature-value pairs as its antecedent and a local NB as its consequent. The local NB is built using the subset of training cases that satisfy the antecedent of the rule. This NB is then used to classify the test case.

16 Conditional Independence Tree, (CITree), Zhang and Su, 2004 [162]. CITree represents a joint distribution over all features explicitly defining the conditional dependencies among them. In growing the tree the feature, given which all the other features has the maximum conditional independence, should be selected at each step. In other words, the feature with the greatest influence on all other features should be selected.

17 Selective Neighbourhood Naïve Bayes, (SNNB), Xie *et al*, 2002, [156]. SNNB is a lazy method proposed for discrete features. The Hamming distance from the submitted test case, $\mathbf{x}$, to all training examples is calculated and stored. Let n be the number of features and therefore the maximum possible Hamming distance. A NB classifier is trained for $k = 1, \ldots, n$ using the examples within distance k from $\mathbf{x}$. The accuracy of each such NB is estimated using leave-one-out. The most accurate NB is selected to label the test case $\mathbf{x}$.

18 Independent Component Analysis Naïve Bayes, (ICABayes), Bressan and Vitrià, 2002 [15]. The independent component analysis (ICA) of an n-dimensional random vector is the linear transform which minimises the statistical dependence between its components. The class-conditional pdf in the new transformed space is the true product of the marginals. Feature selection is applied in the transformed space by keeping the first d components and discarding the rest. This amounts to a feature extraction with respect to the original space.

19 Improved Naïve Bayes Classification, (INBC), Liu *et al*, 2001 [91]. A genetic algorithm is used for feature selection prior to applying NB. The method also uses past classified test data without verified labels by introducing it into the training set.

20 Lazy version of Tree Augmented Network, (LazyTAN), Wang and Webb, 2002, [149]. This is the lazy variant of the Superparent algorithm PDT2, [67]. The network is grown based on the specific values of the test instance **x**. Each feature only has two values, equal or not equal to the test case. This reflects the specific dependencies between the feature values of the current test case not the joint probability distribution for all of the features.

21 Interval Estimation Naïve Bayes, (IENB), Robles *et al*, 2003 [120]. The model calculates confidence intervals for the NB point estimations of $P(x_i|\omega_j)$. Combinations of values from each interval are found by an heuristic search. Each combination is evaluated using a devised measure of predictive accuracy. The combination of values with highest predictive accuracy is selected.

22 Random Tree Augmented Network, (RTAN), Ma and Shi, 2004 [93]. An ensemble of TAN classifiers is grown. Each TAN is trained on a bootstrap sample of the training data. The ensemble is integrated using the majority voting method.

23 Adapted Boosting for Naïve Bayes, (ActiveBoost), Wang *et al*, 2004 [148]. A new test case is labelled and then added into the training set. The updated data set is then used to train another NB. The new cases in the training set are down-weighted accordingly to reflect the lower confidence in their label.

24 Adjusted Probability Naïve Bayes Classifier, (APNBC), Webb and Pazzani, 1998, [152]. The method uses the probability distributions produced by NB. The method attempts to identify linear adjustments to apply to the class probabilities. These linear adjustments are not to make the estimate more accurate but to push the rank in the right direction. An adjustment factor is associated with each class. The inferred probability for a class is multiplied by the corresponding factor. Adjustments are tuned so as to maximise the resubstitution accuracy.

25 Homologous Naïve Bayes, (HNB), Huang and Hsu, 2002 [59]. The model takes advantage of the knowledge that multiple cases submitted for labelling come from the

same unknown class. Such problems occur in speaker verification where there are many examples known to be from one speaker but it is not known which speaker in particular.

26 Fuzzy Naïve Bayes, (FNB), Störr, 2002 [139]. Each feature value of **x** is accompanied by a degree of membership in the interval $[0, 1]$. The case, **x**, also belongs to each class with a degree of membership in the interval $[0, 1]$. The NB formula has been adjusted accordingly to accept this representation but reverts back to NB if the extreme degrees of 0 and 1 are used throughout.

27 Interpretable Boosted Naïve Bayes, (IBNB), Ridgeway *et al*, 1998 [117]. The method aims to improve NB by boosting yet still have an end product that is interpretable by the user.

28 Large Bayes, (LB), Meretakis and Wüthrich, 1999 [99]. The concept of itemset is introduced as a feature subset where the features have particular values. For example, an itemset of $\{x_1, x_2, x_3, x_4\}$ (binary) is $\{x_2 = 0, x_4 = 1\}$. Most frequent itemsets are stored, along with information about their contribution towards class labels (class support). When a new case is submitted, itemsets within the case are identified that correspond to the stored ones. The class supports for these itemsets are used to compute the probability that the case belongs to a particular class. The model reduces to NB when the itemsets are all of size one.

29 Semi-Naïve Bayes, (SNB), Kononenko, 1991 [72]. SNB partitions the features into groups using statistical tests of independence. The model takes the form

$$P(\omega_i, \mathbf{x}) = P(\omega_i)P(\mathcal{A}_1|\omega_i)\dots P(\mathcal{A}_k|\omega_i), \tag{3.18}$$

where $\mathcal{A}_k$ are the disjoint groups of features. It is taken that x_i is conditionally independent of x_j if and only if they are in different groups.

30 Kernel-based and Joining Naïve Bayes, (KJNB), Denton and Perrizo, 2004 [29]. Features are joined if they are highly correlated. To store the training data effectively, a structure called a P-Tree is used.

31 Naïve Bayes Committees, (NBC), Zheng, 1998 [164]. A set of NB classifiers is generated in sequential trials. NB_Base is generated as founder of the committee using all of the features. To generate the next classifier, NB_1, a subset of features, F, is randomly generated. The subset is generated to contain approximately half the number of original features. If the error produced by NB_1 is less than NB_Base then NB_1 is added to the overall set of classifiers else it is discarded. If NB_1 is added to the set of classifiers then the probability of selecting the features in F is increased. If NB_1 was discarded then the probability of selecting the features in F is decreased. A new subset of features, F', is randomly generated taking into account the adjustments in each features probability of selection. This new subset of features F' is used to generate NB_2. The subset of features

at each stage contains approximately half the available features. This heuristic search creates a set of NB classifiers based on different subsets of features. This set of classifiers is then used as a committee (ensemble).

32 Boosted Naïve Bayes, (BNB), Diao *et al*, 2002 [30]. Boosting strategy is applied to NB. The training samples are selected by the bootstrap method.

33 Clustered Naïve Bayes, (CNB), Vilalta and Rish, 2003 [147]. The examples from each class are clustered. The training data is then relabelled using the cluster labels. Suppose that each class is clustered into 3 clusters. If there were c original clusters, there will be $3 \times c$ new class labels. A NB is trained on the data with the new labels. When a new case is classified it is assigned to one of the new labels by NB. As there is one-to-one correspondence between the new (overproduced) and the original class labels, the corresponding original label is recovered for **x**. This method aims to avoid the problem of classes spread out over the feature space.

34 Neuro-Fuzzy Naïve Bayes, (NFNB), Nürnberger, *et al* 1999 [105]. Neuro-fuzzy classification systems derive fuzzy classifiers from data using neural-network inspired learning. The method maps NB to a neuro-fuzzy classifier.

35 MDL Principle in Naïve Bayes, (MNB), Kleiner and Sharp, 2000 [68]. This method starts with a Bayesian network representing class-conditional independence, as the one in Figure 3.7(a). Dependencies are subsequently modelled by adding directed edges to the network. Minimum Description Length (MDL) score is used in seeking a trade-off between the mutual information gain and increase in the network complexity due to adding an edge.

36 Selective Bayesian Classifier, (SBC), Ratanamahatana and Gunopulos, 2002 [116]. SBC runs the decision tree algorithm, C4.5 on 10% of the training set. The features on the first three levels of the decision tree are selected. This is repeated five times on a different 10% selections of the training data. The feature set is the union of all of the features selected from the five runs of C4.5. This union set is then used to train NB.

37 Boosted Levelled Naïve Bayes Trees, (BLNBT), Ting and Zheng, 1999 [142]. A standard decision tree of user defined depth is grown as in the NBTree [69] method. The NB at the leaf is trained using all cases that fall at that leaf. Features that appear on the path leading to the leaf are not used by NB. Boosting is then applied to improve the performance of the tree structure. It is anticipated that the NB classifiers have become unstable due to the introduction of the tree structure.

38 Extended Bayes, (EB), Rosell and Hellerstein, 2004 [121]. EB finds sets of dependent features using an information gain measure. These features are joined and stored as a new feature set, F'. A new case is labelled once by NB on the original feature set and once by NB on F'. This produces two predictions for the test case. If the two labels

Table 3.4: The data matrix of the 38 selected studies encoded by the 19 characteristic features.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1 NB	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0
2 NBTree	1	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	1	0
3 IB	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0
4 EFSNB	1	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0
5 SFSNB	1	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0
6 SBSNB	1	0	1	1	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0
7 GANB	1	0	1	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0
8 TAN	1	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	1	0
9 PDT2	1	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0
10 SFS to Bayes	1	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0
11 SFSJ	1	1	1	0	1	1	1	0	0	0	0	1	0	0	0	0	0	1	0
12 SBSJ	1	1	1	0	1	1	1	0	0	0	0	0	1	0	0	0	0	1	0
13 KdepBN	1	1	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
14 AODE	1	1	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	1	0
15 LBR	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0
16 CTTree	1	0	1	0	1	1	1	0	1	0	0	0	0	0	0	0	0	1	0
17 SNNB	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
18 ICABayes	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
19 INBC	1	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0
20 LazyTAN	1	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0
21 IENB	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0
22 RTAN	1	1	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	1	1
23 ActiveBoost	1	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0
24 APNBC	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0
25 HNB	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
26 FNB	1	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0
27 IBNB	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0
28 LB	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0
29 SNB	1	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
30 KJNB	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
31 NBC	1	0	1	1	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0
32 BNB	1	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0
33 CNB	1	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	1	0
34 NFNB	1	1	0	0	0	0	1	0	1	0	1	0	0	0	0	0	1	0	0
35 MNB	1	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
36 SBC	1	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0
37 BLNBT	1	0	1	0	0	0	1	0	1	0	0	0	0	0	1	0	0	1	0
38 EB	1	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0

match then the case is labeled with the prediction. If the two labels differ then a further analysis is carried out to arrive at a class label.

3.3.5 Encoding of the selected studies

The encoding of the studies involved applying the 19 characteristic questions to each method. Thus, each method is expressed as a binary vector of length 19. Together these vectors create a 38×19 binary data matrix, Table 3.4. A 38×38 matrix of the pairwise Hamming distances between methods can then be created. This matrix of distances is hard to visualise and so the techniques of multi-dimensional scaling can be used.

3.3.6 Analysis of the studies - Multi-Dimensional Scaling

Multi-Dimensional Scaling methods reduce high dimensional feature spaces into lower dimensional ones [94, 129]. Usually two or three dimensions are derived so that the data can be easily visualised. The idea is to map the data to the lower-dimensional space aiming to preserve all interpoint distances.

Principal Components Analysis (PCA)

Principal Components Analysis (PCA) is a method used to reduce the dimensionality of data. New variables are derived as linear combinations of the original variables. The components are ranked by importance so that the first component "explains" most of the variability of the data. The plane spanned by the first two principal components is used to plot the 38 data points, Figure 3.9(a).

Sammon mapping

Sammon mapping was proposed in 1969 by Sammon [126]. Sammon mapping starts by projecting the k objects (the 38 methods) onto 2 random dimensions. This mapping is then refined by using a stress function so as to preserve as much as possible of the original distances between the objects. The stress function is a gauge of the error of the current mapping. The stress is calculated using the distance between two points, d_{ij}, in the m-dimensional space, (in our case $m = 19$), and the distance between the two corresponding points, δ_{ij} in the 2-dimensional space.

$$\text{Stress Function} = \frac{1}{\sum_{i=1}^{k-1}\sum_{j=i+1}^{k}\delta_{ij}}\sum_{i=1}^{k-1}\sum_{j=i+1}^{k}\frac{(\delta_{ij}-d_{ij})^2}{\delta_{ij}} \tag{3.19}$$

The Sammon mapping implementation used for this study was a Matlab code taken from the SOM Toolbox [3]. The input to the routine was the pairwise distance matrix calculated from the 19 features, producing the Sammon mapping shown in Figure 3.9(b). To terminate the algorithm the limit on the stress function was chosen to be 0.65 (multiple runs of the algorithm indicated that this was the minimum achievable stress).

Self-Organising Maps

Self-Organising maps (SOM) are a neural network method of reducing high-dimensional data into two dimensions for easy visualisation, [71]. The 2d-map is typically structured as a rectangular or hexagonal grid. The goal of the learning in the SOM is to get different parts (neurons) of the 2d-map to respond similarly to certain input patterns.

Let M be the total number of neurons, where M is defined by the user. Each neuron i on the 2d grid has an associated vector $\mathbf{s}_i = [s_{i1}, s_{i2}, \ldots, s_{im}]$, where m is the dimension of the original data and $i = 1, \ldots, M$. A similarity measure between $\mathbf{s}_i$ and the input, $\mathbf{x}$ is calculated. The similarity measure identifies the best matching neuron (BMU) and as such the input $\mathbf{x}$ will be associated with that particular neuron. As attachment of the inputs is made using a similarity measure then similar inputs will be attached to (activate) the same neurons. A SOM is initialised using 3 steps.

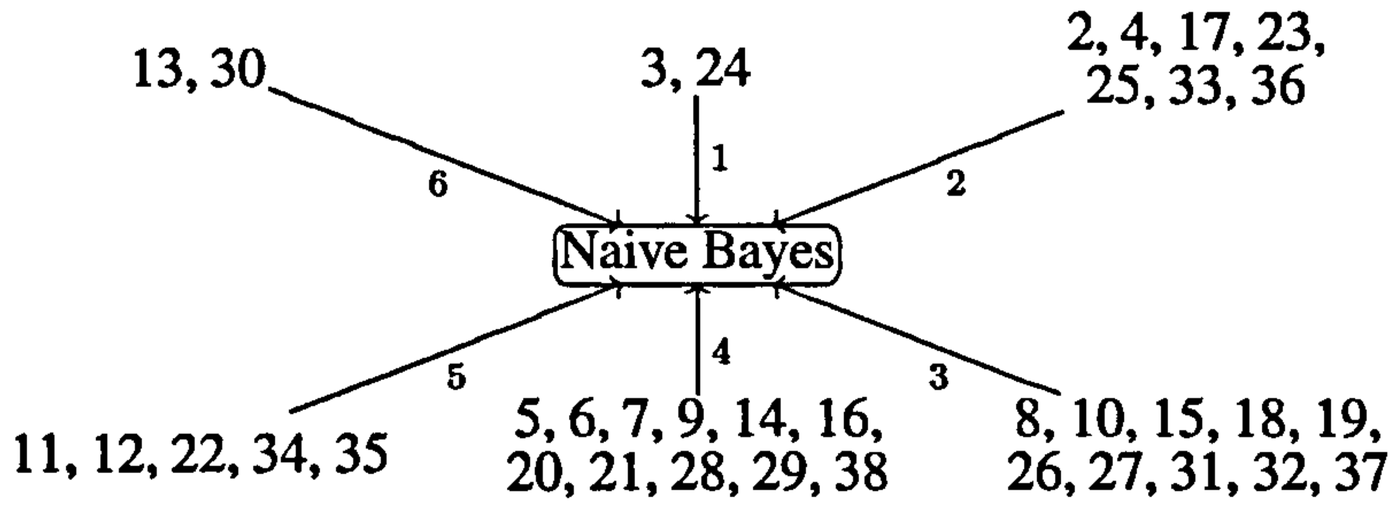


Figure 3.8: The Hamming distance of each adapted method from the original NB model.

1. Initialisation of $\mathbf{s}_i$, $i = 1, \ldots, M$.

2. Training. In each step one case $\mathbf{x}$ from the input data is selected and a similarity measure is calculated between it and each $\mathbf{s}_i$. The best matching unit (BMU) is found.

3. Updating. The vectors corresponding to BMU and its neighbourhood are updated.

The implementation of the SOM algorithm used for this study was taken from the SOM Toolbox. It was used with all the default settings, linear initialisation and batch training. In linear initialisation the values of $\mathbf{s}$ are ordered and taken from the linear subspace spanned by the two principal eigenvectors of the input data. Batch training means that updating takes place after the whole training data has been processed. To do this, at step 2, all updates of BMU for the training data are stored and then the resultant updates are applied at step 3. The resultant mapping shows the inputs on the neuron they activated, Figure 3.12.

3.3.7 Landscapes of the NB methods

Figure 3.8 shows the Hamming distance of each adapted method from the original NB method (**1**) according to the 19 features. All 37 adaptations are within a distance of 6 from NB. To move further away may compromise the simplicity that NB has. Two methods only differ from NB by 1 feature, **3** IB [49] and **24** APNBC [152]. Both methods differ from NB by feature 17 - "Did the method update the probability table produced by NB?" Both methods allow NB to run untouched with small updates of $p(\mathbf{x}|\omega_i)$. Methods **13** KdepBN [123] and **30** KJNB [29] rate as the furthest placed methods from NB. Both methods consider high order feature dependencies. The techniques used to capture the dependencies complicate the process so some of the NB simplicity is sacrificed.

Whilst Figure 3.8 gives the distance of the adaptations from the original NB model, it provides little insight about the relationships between the individual methods. The

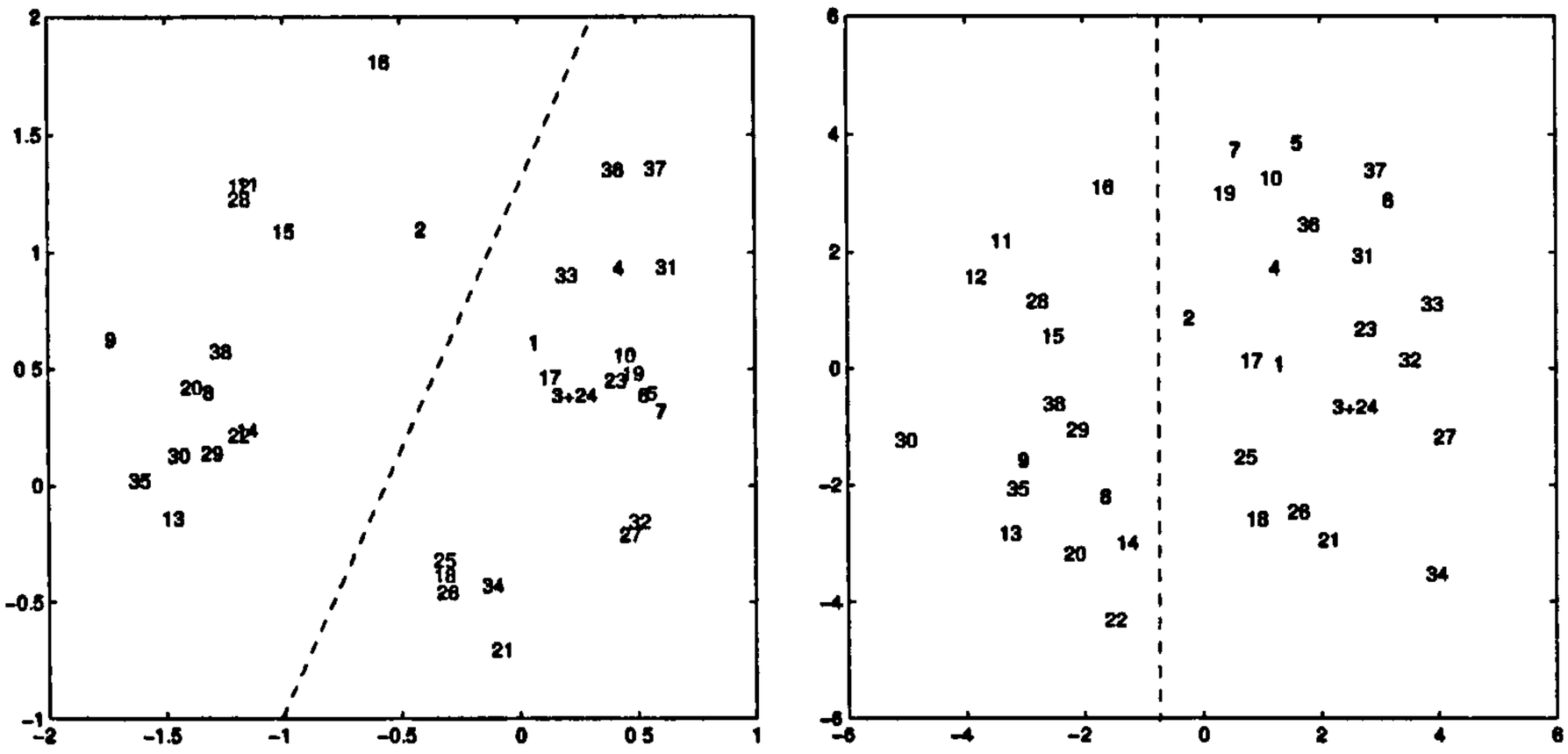
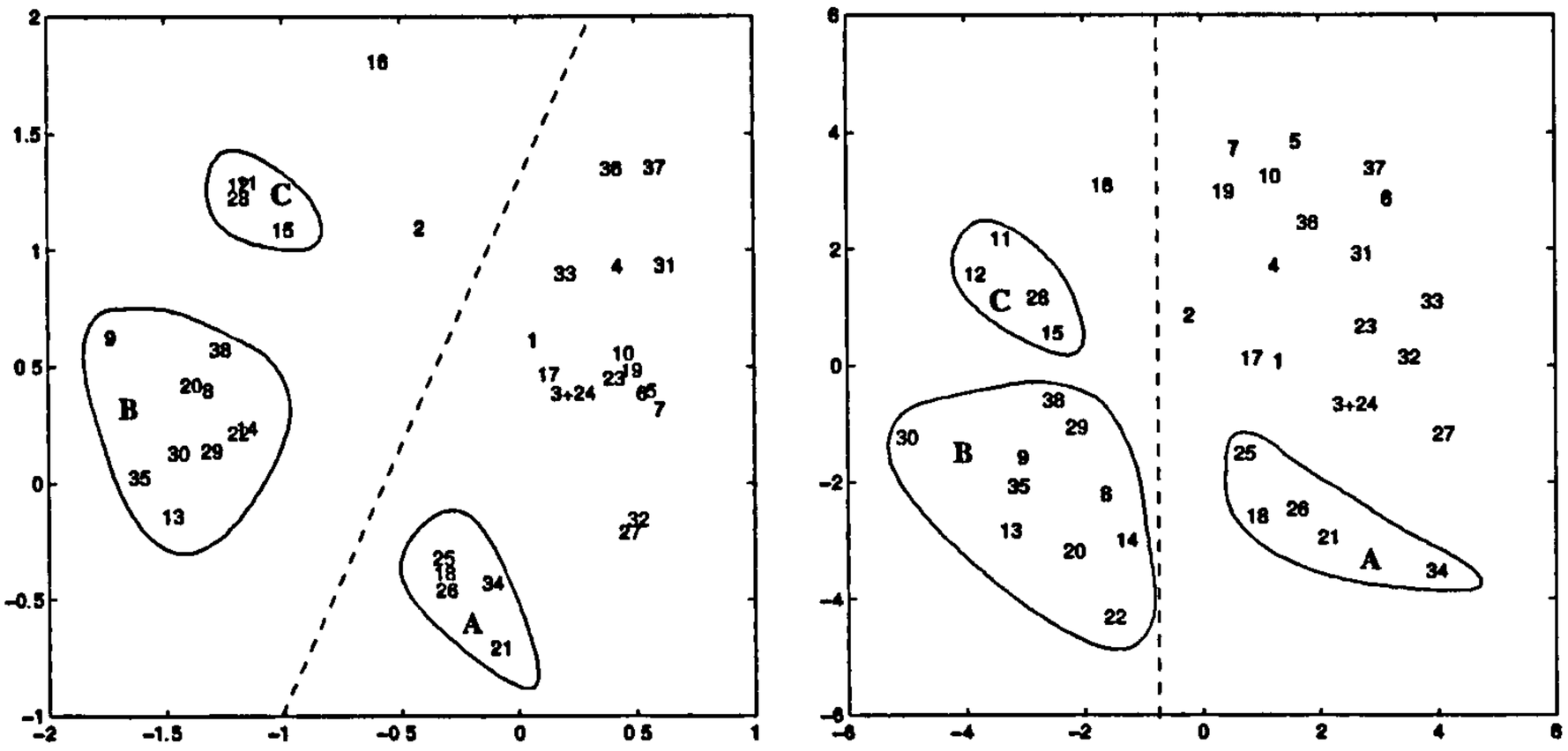


Figure 3.9: Landscape representations of the 38 NB adapted methods (a) PCA representation (b) Sammon mapping.

Sammon mapping and PCA representation shown in Figure 3.9 begin to show the relationships between the methods. They are designed to give a visual insight rather than an accurate diagram. IB and APNBC are represented by 3 + 24 as these characteristic features do not separate the two methods. The dashed line on both diagrams divides the methods into two groups. Method **2** NBTree [69] is the only method to "swap" groups between the two representations. The group on the right-hand side of both representations contain methods that are not looking for explicit dependencies but rather features that work well together (feature selection) and natural tendencies of the data (clustering and focusing of resources on the hard to distinguish data areas (boosting)). The methods on the left of the visualisations look for feature dependencies more explicitly. The methods in this group try to model dependencies, combine dependent features or use feature dependencies as guidance for the model construction.

Further attempts to cluster the methods result in the groups depicted in Figure 3.10. The methods contained in each group are listed in Table 3.5. Cluster A, shown on the right-hand side of both diagrams, consists of 5 methods. All the methods in this cluster adjust the feature space aiming to add more information to the data on which NB is trained.

The ten methods contained in Cluster B are all the methods that specifically use Bayesian networks. The group contains the two methods placed furthest away from NB, **13** KdepBN and **30** KJNB. The consideration of the feature dependencies in the explicit way of Bayesian networks complicates the methods thus distancing them from the original NB model in terms of simplicity and possibly efficiency.



(a) (b)

Figure 3.10: Landscape representations of the 38 NB adapted methods (a) PCA representation (b) Sammon mapping, with respective clusterings.

The final coherent cluster, C, seen in both representations contains four methods that all join features. These methods aim to uphold the independence assumption by combining dependent features into single "new" features. These methods differ from the group of Bayesian networks as they create "new" features from dependent ones rather than model the dependency within a network.

There are no clear clusters in the remaining part of Figure 3.10. However, it is possible to detect regions as shown in Figure 3.11. This illustration depicts clusters A, B and C as well as two more regions, using their relative geometric location on the plots. Tree structures can be seen to branch across the top of both Sammon and PCA representations. These methods (**2** NBTree [69], **16** CITree [162], **36** SBC [116], **37** BLNBT [142]) can be seen as providing a link between joining and selecting features. Tree methods *select* a feature or group of features for each split in the tree. The features on a path of a tree can then be considered to be *joined*. The difference between tree structures and feature selection methods cannot be clearly stated. Methods in the feature selection area try to find features that work well together. Tree structures methods also try to find complementary features to label a path. The "space transformation" group, cluster A, contains feature extraction methods which are the natural "partner" to feature selection methods and so the areas sit naturally adjacent to one another in the visual representation

The consistency of these regions is also demonstrated by the SOM representation in Figure 3.12. Each rectangle corresponds to a neuron. All the neurons are drawn with uniform size. The tone of neuron indicates the size; the darker the tone of the neuron the larger it is. The methods are displayed on the neuron they activated when submitted

Table 3.5: The adapted methods in the three clusters shown on Figure 3.10.

Cluster	Number	Method and Reference
A	18	Independent Component Analysis NB (ICABayes) [15]
A	21	Interval Estimation NB (IENB) [120]
A	25	Homologous NB (HNB) [59]
A	26	Fuzzy NB (FNB) [139]
A	34	Neuro-Fuzzy NB (NFNB) [105]
B	8	Tree Augmented Bayesian Network (TAN) [48]
B	9	Probability Dependence Tree 2 (PDT2) [67]
B	13	K-Dependence Bayesian Network (KdepBN) [123]
B	14	Aggregating One-Dependence Estimators (AODE) [151]
B	20	Lazy version of TAN (LazyTAN) [149]
B	22	Random TAN (RTAN) [93]
B	29	Semi-NB (SNB) [72]
B	30	Kernel-based & Joining NB (KJNB) [29]
B	35	MDL principle in NB (MNB) [68]
B	38	Extended Bayes (EB) [121]
C	11	Sequential Forward selection & Joining (SFSJ) [110]
C	12	Sequential Backward Selection & Joining (SBSJ) [110]
C	15	Lazy Bayesian Rules (LBR) [165]
C	28	Large Bayes (LB) [99]

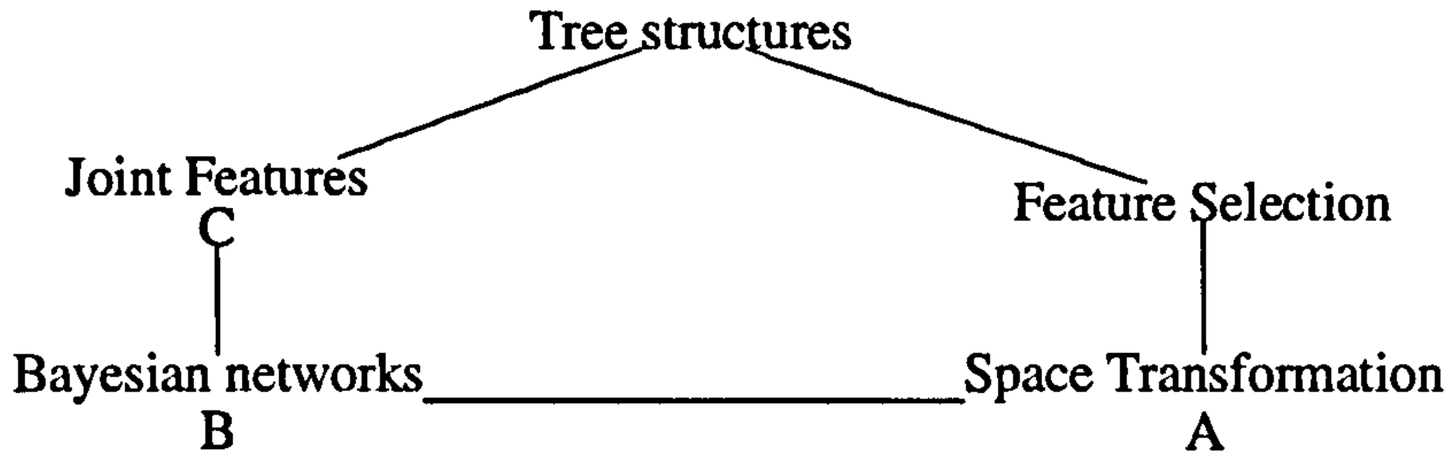


Figure 3.11: Breakdown of the landscape areas of adaptations to NB from the representations.

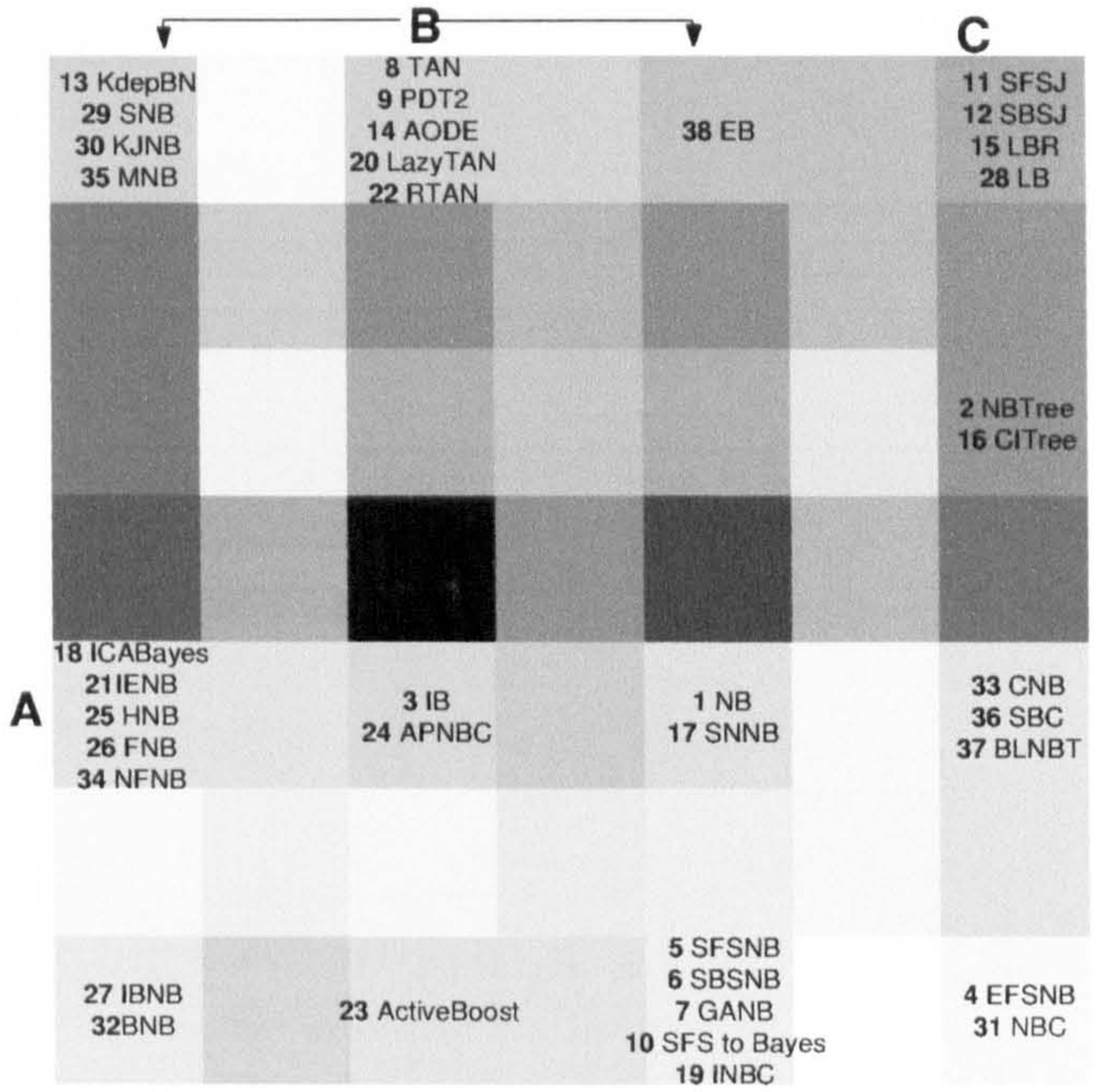


Figure 3.12: SOM of the 38 NB methods.

to the trained SOM. The darker horizontal line of neurons in the middle defines two groups. Again methods **2** and **16** appear close to this border. The SOM confirms the intact appearance of clusters A and C. Cluster B has not been invaded by other methods but rather split into three smaller neighbouring clusters. The regions depicted in Figure 3.11 can still be traced in the SOM.

The time span of the adaptation publications can be separated into two eras - early (1991 - 1997) and recent (1998 - 2005). The years of publication for each of the methods are shown on Figure 3.13. The earlier methods appear mainly in the Bayesian network area of the landscape, cluster B. Many of the earlier methods considered using knowledge of the feature dependencies as the key to improvement. The only method on the "feature space" side (lower half of the SOM) from the early methods is **10** SFS to Bayes by Langley and Sage [82] in 1994.

Cluster A depicts the latest area, taking the adaptations to a more complex level. It appears that there are recent studies in all areas. This means that, over the 15 years of research, there is no agreed single way of improving upon the NB model.

To study the effect that an adaptation could have on NB (**1**) each feature in turn was added or removed from the NB descriptor vector. The "new" method was then submitted to the SOM in order to see which neuron of the map it activated. The results are given in Figure 3.14. The arrows depict where the method would move to and are labelled by the

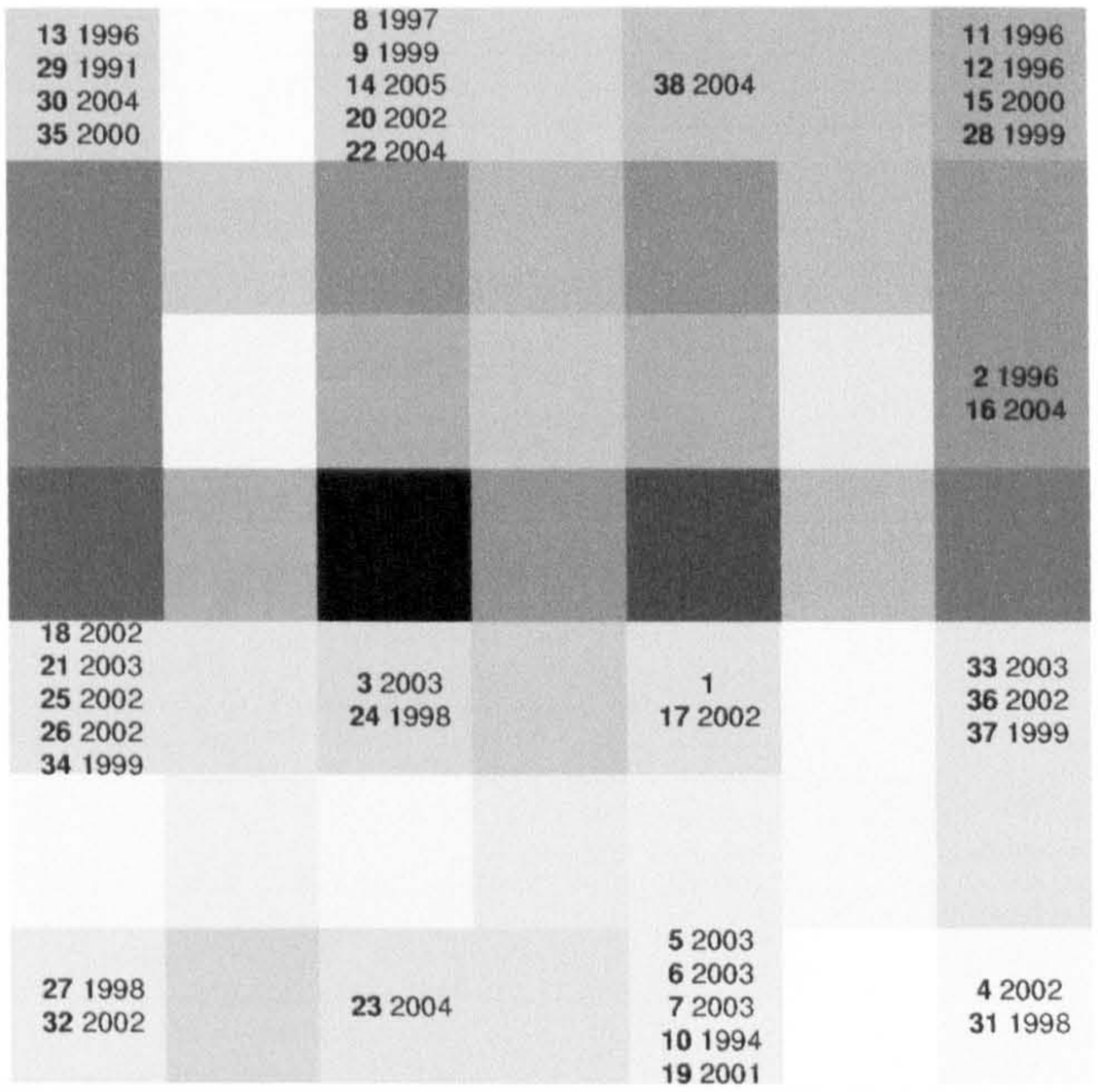


Figure 3.13: SOM of the 38 NB methods depicted by the relevant year of publication.

feature that would cause the move. For example, if either feature 5 or 6 (considerations of feature dependencies) were added to NB the model makes a move towards cluster B - the Bayesian network cluster. NB also moves towards the Bayesian network cluster if the NB formula is adapted (feature 2) or Bayesian Networks are directly incorporated (feature 8).

Updating the probability table (feature 17) or using fuzzy classification methods (feature 11) moved NB towards cluster A - Space transformation. Using decision trees (feature 9) moved NB towards the decision tree area of the SOM. However the unexpected move was made by adding feature 3, using feature selection prior to using NB. Adding this feature in also moved NB towards the decision tree area of the SOM. This is unexpected as the methods using feature selection appear in a neuron directly below NB in the SOM (**5** SFSNB, **6** SBSNB, **7** GANB, **10** SFS to Bayes, **19** INBC). By adding in feature selection we would expect to move to this group. The reason behind this may be due to feature 18 (testing of the method on a wide range of data), according to the original studies these "feature selection" methods were tried on less than 10 datasets, whereas NB has been tested on many types of data.

By conducting the meta-analysis we have gained answers to the analysis questions:-

1. Which methods are structurally similar?

   The landscapes are built using the selected structural features of the methods. The distance between the methods placed on the landscapes indicates the similarities

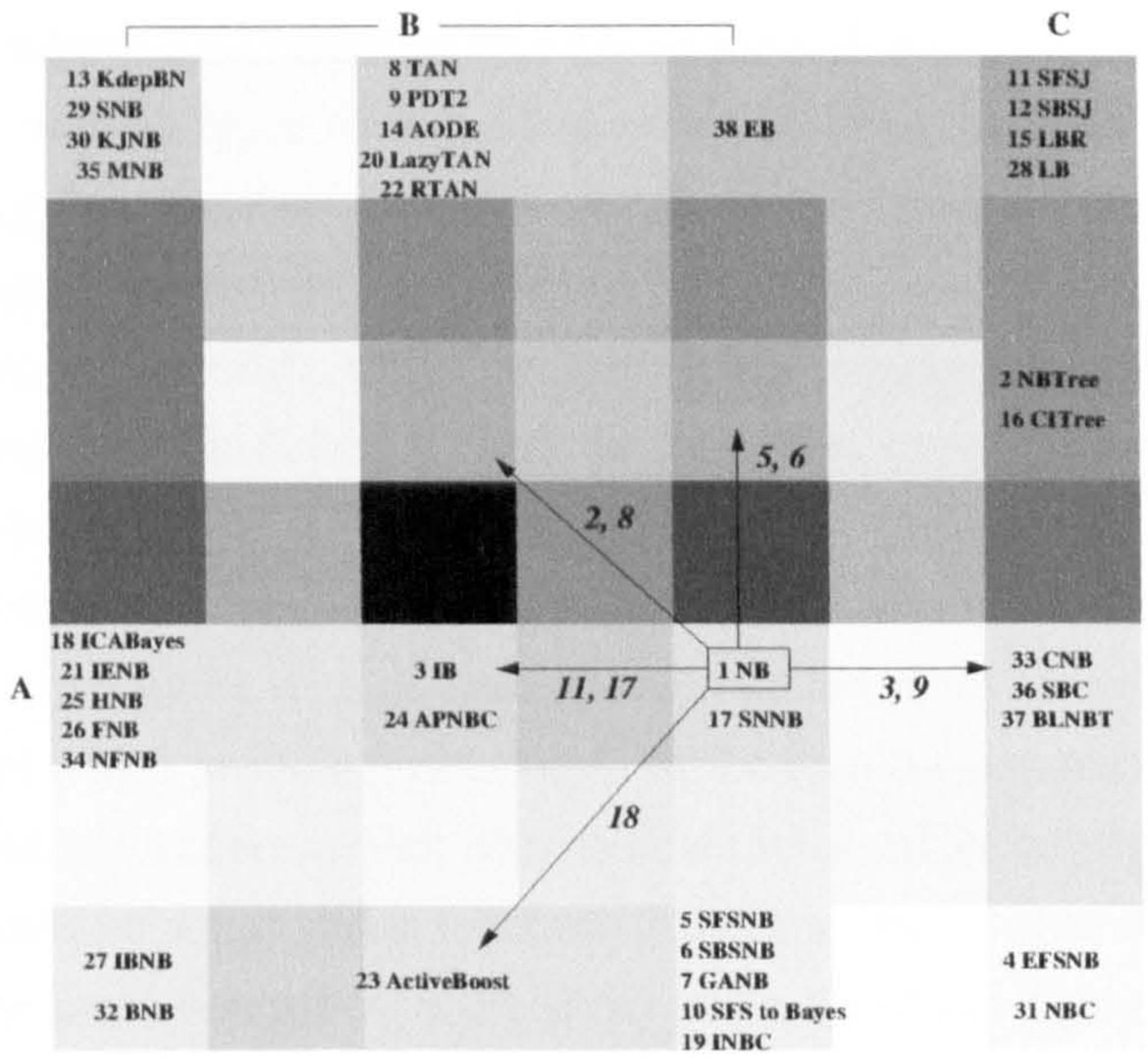


Figure 3.14: The effects of adjusting the characteristic structural features of NB by one.

between them when described by these characteristics.

2. What relationships, if any, are there between the adaptations?
   The relationships between the methods can be expanded upon by the adjustments carried out in Figure 3.14. The addition or removal of structural features from particular methods may lead to a closer relationship with nearby methods.

3. What techniques have been used to try and optimise upon the original NB model?
   The techniques used to adapt NB in the selected studies have been separated into the five general areas shown by Figure 3.11. Many adaptations have been kept simple in order to enjoy the benefits of NB, with fewer adaptations ranging into the perceived complex areas such as space transformation.

The landscapes presented in this study show where any new variant of NB can be fitted. These landscapes provide the nearest neighbours of the variants which would be good to use in comparison studies as "straw men".

Both the PCA and SOM visualisations, (Figures 3.9 and 3.12) suggest that there is space for new variations of the methods. The largest gap which can be seen in all the representations lies between "Bayesian networks" , cluster B and "Space Transformation", cluster A. This would suggest the formation of a hybrid method combining feature extraction and Bayesian networks. The popularity of hybrid methods to combine the good aspects of various models could possibly be exploited to achieve success here.

The landscapes have also provided us with an idea of what variants to NB would be suitable to try on various types of data. For example, if we have no knowledge of the feature relationships in our data then the models in the "Joint Features" or "Bayesian networks" areas would not be used to their full potential. However, the "Tree structures" and "Feature selection" models while not giving any insight into feature relationships would expose features in the data that work well together.

On the other hand, if we had data that required that the feature space and features remain interpretable to the domain experts then "Joint features" and "Feature space transformation" models would be of little use. The remaining areas all result in easily interpretable classifiers which provide good insight of the data for domain experts.

From the SOM landscape we can also indicate what effect adjusting the describing features would have on a particular method. Adapting the structural features within a method will allow more control over the exact make-up of the methods and their "distance" from NB. We are cautious to draw any deeper conclusions from the landscapes as they would only be applicable to the features we have selected for this study. While the analysis is subjective as to the features selected it still produces an interesting insight into the methods selected.

While an empirical evaluation would generate concrete results about the types of data a particular method is good for, the study would still be limited to the data used. By providing the landscapes this study has grouped the variants into "bitesize" areas of adaptations. This provides an uncomplicated view of the structural changes that have been applied, lessening the need to trawl through many piles of literature to find similar NB variants.

It is worth noting that across the 38 methods studied and those considered in the application of NB none were adapted to handle probability table data.

3.4 Chapter Summary

The requirements of a classifier set out by the veterinary domain specify a need for simplicity and comprehensibility. Domain experts like to be able to follow the logic of any decisions made by a classifier.

This chapter began by applying a cascade tree model to probability table data for solving the multi-class problem. This model was seen to cope well with the skewed probability data, as is held in the BSE and Scrapie tables.

Naïve Bayes has also attracted attention in the medical domain partly due to its simplicity and efficiency. As NB is known to be optimal when the features are class-conditionally independent it made sense to apply it to the probability table data where

one of the assumptions was that the features were class-conditionally independent. The second part of this chapter was dedicated to a survey of the adaptations applied to NB in attempts to improve upon the original model. The survey was conducted via a meta-analysis of the structural differences of the models resulting in landscapes of the relationships between the various models. From the literature surveyed no NB model had been specifically adapted to cope with probability table data.

The adaptations to improve NB invariably led to an interest in the apparent optimality or good performance of the model even when the assumption of class-conditionally independent features is clearly violated.

Chapter 4

The optimality of NB for binary features

Adapting NB to improve its performance led to an abundance of literature and a large number of new models that have achieved varying degrees of success. However, without any alterations NB has been seen to be successful for many types of data over the years. Studies have emerged that begin to piece together the puzzle as to why NB can give such good performance even in cases when the class-conditional independence of features clearly does not hold.

In 1992 Langley *et al* carried out an analysis of Bayesian classifiers [81]. The study concluded that it should not be assumed that simple algorithms like NB would not perform as well as more complex methods. The study also noted that no studies had previously examined the extent to which violation of the conditional independence assumption of NB affected performance.

Through two studies in 1996 and 1997 Domingos and Pazzani showed that the independence assumption is a sufficient but not a necessary condition for the optimality of NB [32, 33]. This was an insight as to why NB could perform well even when the independence assumption was violated. They also demonstrated that NB can not recognise all linearly separable functions. NB is unable to recognise the 8-of-25 concept, where the concept is true if at least 8 of the 25 Boolean variables are true.

Langley and Sage [83] analysed NB behaviour on large training sets with large numbers of features. The study looks at the effects of irrelevant features and noise in the training data. These characteristics did not have the sizable effect expected on the performance of NB.

Zhang *et al* [161] shows how the sampling of the training (or testing) sets affect NB. Some classifiers, such as Perceptrons, can represent any linearly separable function no matter how the training and testing sets are sampled. This is shown not to be true for NB. A linearly separable function that can be recognised by NB by sampling uniformly may become unrecognisable to NB when sampling of the training set is not performed uniformly.

In 2001 Rish *et al* attempted to find characteristics of data that affected the performance of NB [118, 119]. It has been seen that the strength of feature dependencies within the data is not a good predictor of the performance of NB [33]. The characteristic that actually appeared as a better predictor of the performance was the entropy of the class-conditional marginal distributions, $P(x_j|\omega_i)$. The monotone increase of NB error is related to the monotone increase of the entropy of the class conditional marginal distributions.

Hand and Yu present a review of NB and an argument as to why the model should not be ignored [54]. Suggestions as to why NB performs so well are that:- 1) as NB requires the estimation of fewer parameters compared to other models, the estimates of the

pdfs have lower variance, 2) feature selection used in previous studies may have falsely favoured NB as highly correlated features may have been eliminated and 3) the estimation of the probabilities given by NB need not be accurate for the correct classification to be given.

Zhang and Ling [160] suggest that it is the distribution of the feature dependencies and not the dependencies themselves that are important to the performance of NB. By using the conditional mutual information, $I(x_i, x_j|\omega_k)$ of the features x_i and x_j across the two class problem they begin to explain this effect,

$$I(x_i, x_j|\omega_k) = \sum_{x_i,x_j} P(x_i, x_j, \omega_1) \ln \frac{P(x_i|x_j, \omega_1)}{P(x_i|\omega_1)} + P(x_i, x_j, \omega_2) \ln \frac{P(x_i|x_j, \omega_2)}{P(x_i|\omega_2)} \quad (4.1)$$

When $\frac{P(x_i|x_j,\omega_1)}{P(x_i|\omega_1)} > 1$ and $\frac{P(x_i|x_j,\omega_2)}{P(x_i|\omega_2)} < 1$ the dependence between x_i and x_j in both class ω_1 and ω_2 support classifying x into class ω_1. The information association between x_i and x_j should be the sum of them, but in equation 4.1 they cancel each other out. When the evidence supports classification into different classes (i.e. $\frac{P(x_i|x_j,\omega_1)}{P(x_i|\omega_1)} > 1$ and $\frac{P(x_i|x_j,\omega_2)}{P(x_i|\omega_2)} > 1$) they should cancel each other out but equation 4.1 reflects the opposite occurring. From this information Zhang and Ling construct a classifier based on the dependence distribution of the features.

In 2004 Zhang [158] further explains this effect by developing a dependence distribution factor. When this factor has a value of one, NB will give the same classification as a model that accounts for all dependencies between the features. It is shown that this distribution factor takes the value one in three cases, 1) when no dependence exists between the features, 2) the dependence of each feature is the same in both classes and 3) opposite influences of dependencies act to cancel each other out. The support some dependencies give for classifying x into class ω_1 is cancelled out by the support that other dependencies give for classifying x into class ω_2.

These studies from 1992 until the present day are all looking to answer the question - "When can the NB classifier be optimal?" There have been various attempts to improve upon the list of necessary and sufficient conditions for the optimality of NB but this list still remains incomplete. The advantages of NB in its learning speed, classification speed, storage space and incrementality all generate the interest in when the model will be optimal.

4.1 Errors of NB

Consider the two feature, two class problem outlined in Table 4.1. The two features are binary-valued, the features may be present, having the value 1, or absent having the

value 0. The entries in the table are the class-conditional probabilities for the respective combination of signs. For example, $a = P(x_1 = 0, x_2 = 0|\omega_1)$. The Bayes error for this problem is

$$\begin{aligned} E_B &= \min\{P(\omega_1)a, P(\omega_2)e\} + \min\{P(\omega_1)b, P(\omega_2)f\} \\ &\quad + \min\{P(\omega_1)c, P(\omega_2)g\} + \min\{P(\omega_1)d, P(\omega_2)h\} \end{aligned} \tag{4.2}$$

where $P(\omega_i)$ is the prior probability of class ω_i

Table 4.1: The dependent distribution of two binary features in a two class problem.

ω_1	$x_1 = 0$	$x_1 = 1$
$x_2 = 0$	a	b
$x_2 = 1$	c	d

ω_2	$x_1 = 0$	$x_1 = 1$
$x_2 = 0$	e	f
$x_2 = 1$	g	h

Table 4.1 is termed the "dependent" distribution because x_1 and x_2 are not assumed to be independent given either of the two classes. If x_1 and x_2 are considered to be independent their joint distribution will be as shown in Table 4.2. The independent distribution can be calculated from the dependent distribution but there is no way of recovering the dependent distribution from the independent distribution.

Assuming that Table 4.2 gives the true distribution the estimate of the Bayes error will be

$$\begin{aligned} E_{IND} &= \min\{P(\omega_1)A, P(\omega_2)E\} + \min\{P(\omega_1)B, P(\omega_2)F\} \\ &\quad + \min\{P(\omega_1)C, P(\omega_2)G\} + \min\{P(\omega_1)D, P(\omega_2)H\} \end{aligned} \tag{4.3}$$

Table 4.2: The independent distribution of two binary features in a two class problem.

ω_1	$x_1 = 0$	$x_1 = 1$
$x_2 = 0$	$A = (a+b)(a+c)$	$B = (a+b)(b+d)$
$x_2 = 1$	$C = (a+c)(c+d)$	$D = (b+d)(c+d)$

ω_2	$x_1 = 0$	$x_1 = 1$
$x_2 = 0$	$E = (e+f)(e+g)$	$F = (e+f)(f+h)$
$x_2 = 1$	$G = (e+g)(g+h)$	$H = (f+h)(g+h)$

Denote by E_{NB} the error made by the NB classifier (assuming Table 4.2) while the true distribution is the one in Table 4.1. There is no easy way of expressing E_{NB} because it will depend on whether or not the Bayes classifier and NB make the same decisions.

For example, let $P(\omega_1) = P(\omega_2) = \frac{1}{2}$, then if ($a > e$ and $A > E$) or ($a < e$ and $A < E$), E_{NB} will have $\min\{a, e\}$ as the first error term in the brackets in Equation 4.3. If the opposite holds, E_{NB} will have $\max\{a, e\}$ as the first error term.

The best way to see the difference between E_B, E_{NB} and E_{IND} is with an example. Consider the problem outlined in Tables 4.3 and 4.4. Let $P(\omega_1) = P(\omega_2) = \frac{1}{2}$. Let $\mathbf{x} = [0, 0]^T$ be the case submitted for classification.

Table 4.3: An example dependent distribution.

ω_1	$x_1 = 0$	$x_1 = 1$
$x_2 = 0$	0.4	0.2
$x_2 = 1$	0.1	0.3

ω_2	$x_1 = 0$	$x_1 = 1$
$x_2 = 0$	0.3	0.1
$x_2 = 1$	0.5	0.1

Table 4.4: The related independent distribution calculated from Table 4.3.

ω_1	$x_1 = 0$	$x_1 = 1$
$x_2 = 0$	0.3	0.3
$x_2 = 1$	0.2	0.2

ω_2	$x_1 = 0$	$x_1 = 1$
$x_2 = 0$	0.32	0.08
$x_2 = 1$	0.48	0.12

In the dependent distribution $\mathbf{x} = [0, 0]^T$ would be classified as class ω_1 with the error of this decision being $P_B(\text{error}, \mathbf{x} = [0, 0]^T) = \min\{0.4, 0.3\} \times \frac{1}{2} = 0.15$.

However, looking at the independent distribution modelled from the dependent distribution $\mathbf{x} = [0, 0]^T$ would be classified as class ω_2. According to the independent distribution $P_{IND}(\text{error}, \mathbf{x} = [0, 0]^T) = \min\{0.3, 0.32\} \times \frac{1}{2} = 0.15$. However, there is a mistake according to the true distribution, i.e., $P_{NB}(\text{error}, \mathbf{x} = [0, 0]^T) = 0.4 \times \frac{1}{2} = 0.2$, so E_{NB} is larger than E_B.

Table 4.5 represents the form of the non-traditional probability tables for the two class problem. The tables depict the probability of each symptom, (feature x_j) being present in the case of each disease (class ω_i). Class ω_1 can be thought of as the class of interest, BSE or Scrapie, and class ω_2 can be thought of as the combined class of all other diseases. Table 4.5 can be used to construct the joint distribution of (x_1, x_2), assuming class-conditional independence. This will result in Table 4.2. In other words, Tables 4.2 and 4.5 are equivalent, and they represent the information accessible in the "non-traditional" data sets.

With the non-traditional data on BSE and Scrapie the only error that can be calculated is E_{IND}, the error related to Table 4.2. This chapter looks at the relationships between the three errors, E_B, E_{NB} and E_{IND} where

- E_B is the Bayes error committed on a given "true" probability distribution.

Table 4.5: The form of the probability data in the relevant notation.

	ω_1	ω_2
$P(x_1 = 1)$	$b+d$	$f+h$
$P(x_2 = 1)$	$c+d$	$g+h$

- E_{IND} is the Naive Bayes error committed on the assumed "independent" probability distribution calculated from the original distribution.

- E_{NB} is the Naive Bayes error made using the "independent" distribution to make classification and relating this choice back to the "true" probability distribution.

4.2 Optimality of NB

For binary features NB can only learn linearly separable functions [36]. For functions such as the exclusive Or problem (XOR) NB will always be suboptimal, although it has been shown that boosting can help NB solve the "XOR+noise" problem but not the basic XOR problem [40]. NB classifier performance may still be close to optimality even for these problems. If E_{NB} or E_{IND} are equivalent to E_B then the NB classifier is optimal.

If the binary values "0" and "1" are treated as numbers then the covariance between two binary features can be calculated separately for each class. The mean for x_1 given class ω_1 is $\mu_1 = 0 \times (a+b) + 1 \times (c+d) = c+d$. The mean for x_2 is $\mu_2 = (b+d)$. The covariance is the expectation of $(x_1 - \mu_1)(x_2 - \mu_2)$ summed across the four values and weighted by the respective probability.

$$\begin{aligned} Cov(x_1, x_2|\omega_1) &= a((0-(c+d))(0-(b+d))) + b((0-(c+d))(1-(b+d))) \\ &\quad +c((1-(c+d))(0-(b+d))) + d((1-(c+d))(1-(b+d))) \\ &= ad - bc \end{aligned} \tag{4.4}$$

Kuncheva [78] shows that NB will be optimal for a two-class two feature problem where $Cov(x_1, x_2|\omega_1) = Cov(x_1, x_2|\omega_2)$. This only holds when the prior probabilities of the two classes are equal. It is shown that for 3 features if all pairwise covariances are equal across the three classes then this no longer holds.

4.3 Empirical bounds

With the non-traditional data the only error that can be calculated is E_{IND}. The question is how E_{IND} is related to E_B, the Bayes error or E_{NB}, NB error for the actual dependent

distribution of features? The following experiments were devised in order to gain an understanding of the relationship between E_{NB} and E_{IND}.

4.3.1 Simulated data

10,000 pairs of random matrices of dependent features, as in Table 4.1, were generated. The independent distributions, (Table 4.2) were then calculated from these. The measure of dependence of the features was taken to be Yules Q statistic. Yules Q statistic varies from -1 to 1. A value of -1 implies the two features always take the opposite value (negatively dependent) and a value of 1 indicates the features always take the same value (positively dependent). A Q-value of zero means that the features are independent. Q_1 expresses the level of dependency in class ω_1 and is calculated as

$$Q_1 = \frac{ad - bc}{ad + bc} \tag{4.5}$$

Restrictive simulation

The distribution of class ω_2 was restricted so that the two features were independent given class ω_2, meaning that $Q_2 = \frac{eh-fg}{eh+fg} = 0$. The classes are also assumed to be equiprobable, $P(\omega_1) = P(\omega_2) = \frac{1}{2}$.

The 10,000 points, $(Q_1, E_{IND} - E_{NB})$ are plotted in Figure 4.1. The figure shows that the difference between E_{NB} and E_{IND} can be positive or negative. It is not clear as to when the error is over or underestimated. When Q_1 is zero so is the difference between the two errors. This is because the features are independent in both classes. All generated matrices become equivalent to those in Table 4.1. The NB classifier is known to be optimal in this situation. As the level of dependency reaches ± 1 the maximal difference is increased. However, the difference between the two errors can still be zero and so the level of dependency does not really indicate the size of the difference. This is in agreement with previous studies that the level of dependency of the features does not influence the error incurred [33, 119]. The shape of the plotted differences is symmetrical and pronounced. The symmetry is due to the encoding of the features being arbitrary, the values of zero and one for the features can be interchanged. The pronounced shape indicates the possibility of a bound being found.

Two *empirical* bounds were found that fit onto the differences depicted in Figure 4.1. Both bounds are depicted in Figure 4.1. Enclosing 95% of the points shown is bound 1

$$B_1 = \pm \frac{Q_1 + {Q_1}^3}{20} \tag{4.6}$$

The second bound actually encompasses 100% of the 10,000 error differences shown in

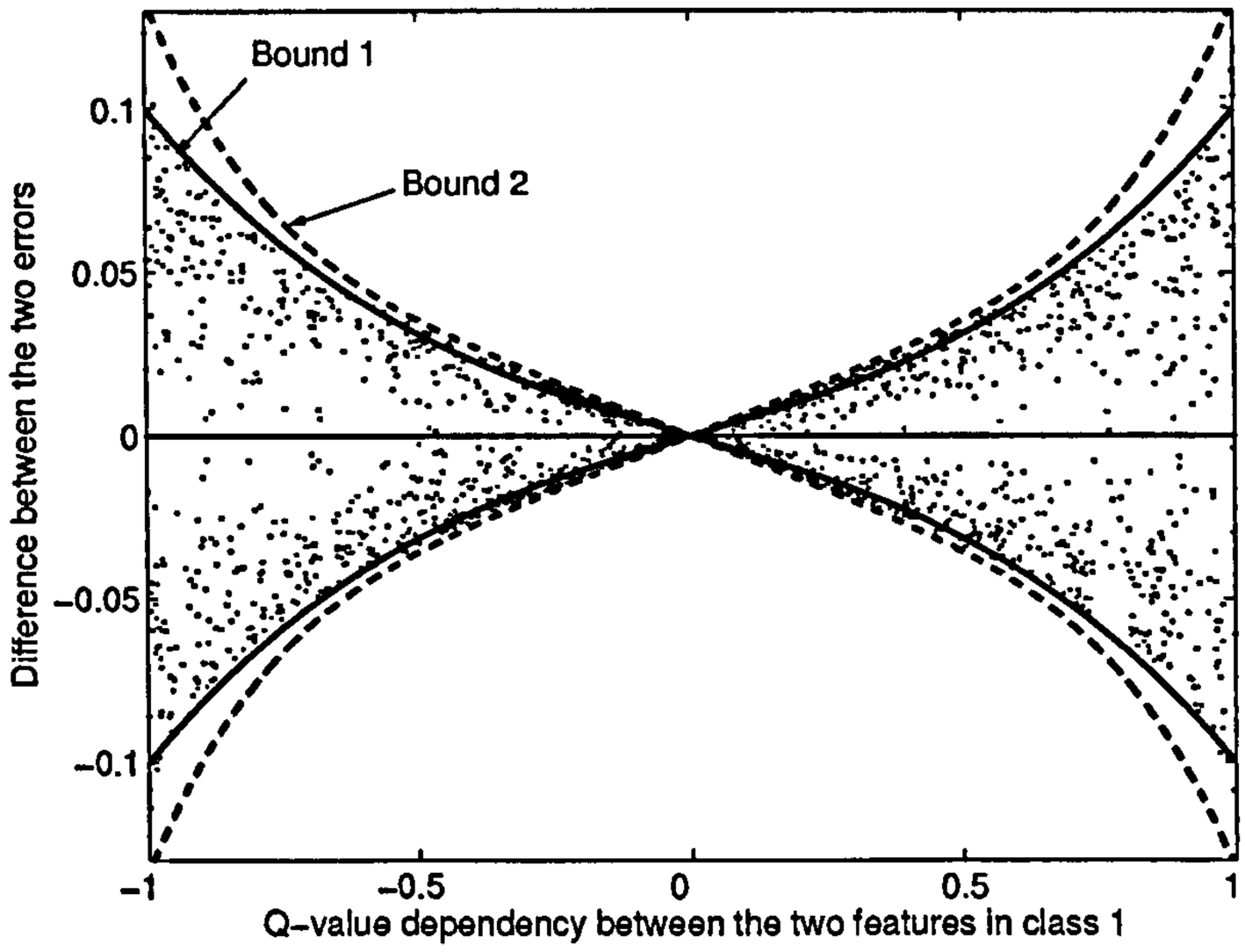


Figure 4.1: Scatterplot of the 10,000 randomly generated data points $(Q_1, E_{IND} - E_{NB})$.

Figure 4.1,

$$B_2 = \pm \frac{Q_1 + {Q_1}^5}{15} \tag{4.7}$$

These bounds indicate that if the two features are independent in one class and the value of the dependency in the other class is known then E_{IND} is likely to be within $\pm B_2$ of the error committed on the "true dependent" distribution.

4.3.2 Real data - traditional recorded case data

By the way the non-traditional probability tables are constructed we have no knowledge of any underlying dependencies between the features and therefore cannot obtain the value of Q. However, using traditional case data will give an indication of the possible distribution of error differences. Four traditional data sets of recorded cases were taken from the UCI data repository [11], SPECT, Wine, Thyroid and Glass.

- SPECT. Separate cardiac SPECT images into normal or abnormal classes based on 22 binary features. The set contains 267 cases.
- Wine. The 178 cases are made up from the chemical analysis of wines grown in the same region of Italy but derived from three different cultivars. A decision is made between these three classes using 13 continuously valued features.
- Thyroid. The 215 cases described by five continuously valued features are the results of lab tests to predict the condition of a patient's thyroid into one of three

classes.

- Glass. Glass left at crime scenes may be used as evidence if correctly identified. The types of glass are separated into six different classes. The 214 cases are described by nine continuously valued features.

None of the data sets contained any missing values. For this simulation any data that was not binary was converted by using the Gini criterion to split the continuously valued features. If a data set contained more than two classes only the first two classes were used. This reduced Wine, Thyroid and Glass to 130 cases, 185 cases and 146 cases respectively.

The data was converted into 2-feature probability problems. This was done by considering each pair of features from the data and generating the "dependent" distribution (Table 4.1). From this the "independent" distribution (Table 4.2) for the two features could be calculated. The Q-value for the two features in the two classes was also calculated. The feature pairs that had a Q-value of exactly zero in at least one of the two classes were then selected, i.e. the two features were independent in at least one of the two classes.

The 22 features in the SPECT data gave 231 feature pairs to consider. From these 231 pairs 42 had a Q-value of zero in at least one class. The Wine data generated 78 2-feature problems. From this there were 57 pairs that had a Q-value of zero in at least one class. The Thyroid data generated 10 2-feature problems with its 5 features. The 10 generated pair problems gave 7 feature pairs with a Q-value of zero in at least one class. The Glass data contained 9 features which allowed the generation of 36 2-feature problems. Out of the 36 feature pairs there were 15 that had a Q-value of zero in at least one class.

For each of the selected feature pairs the value $(Q, E_{IND} - E_{NB})$ was plotted where Q is the Q-value of the class which was not equal to zero. The results together with bound B_2 are plotted in Figure 4.2. All of the plotted points, $(Q, E_{IND} - E_{NB})$ fall within the bound B_2. The points generated by the real data imply that the maximum difference is not as large as the bound allows. For example, Q varies between -1 and 1 so at the extremes the bound B_2 allows a difference between E_{IND} and E_{NB} of $\pm\frac{2}{15}$. In reality the points generated here suggest that in fact the difference between the two errors will be smaller than the suggested bound.

4.4 Theoretical bounds

For the empirical bounds the restrictions on the data were that the bounds only held for the case with two binary features in a two class problem with dependence between the features allowed in only one class. The restrictions on allowing dependency between the

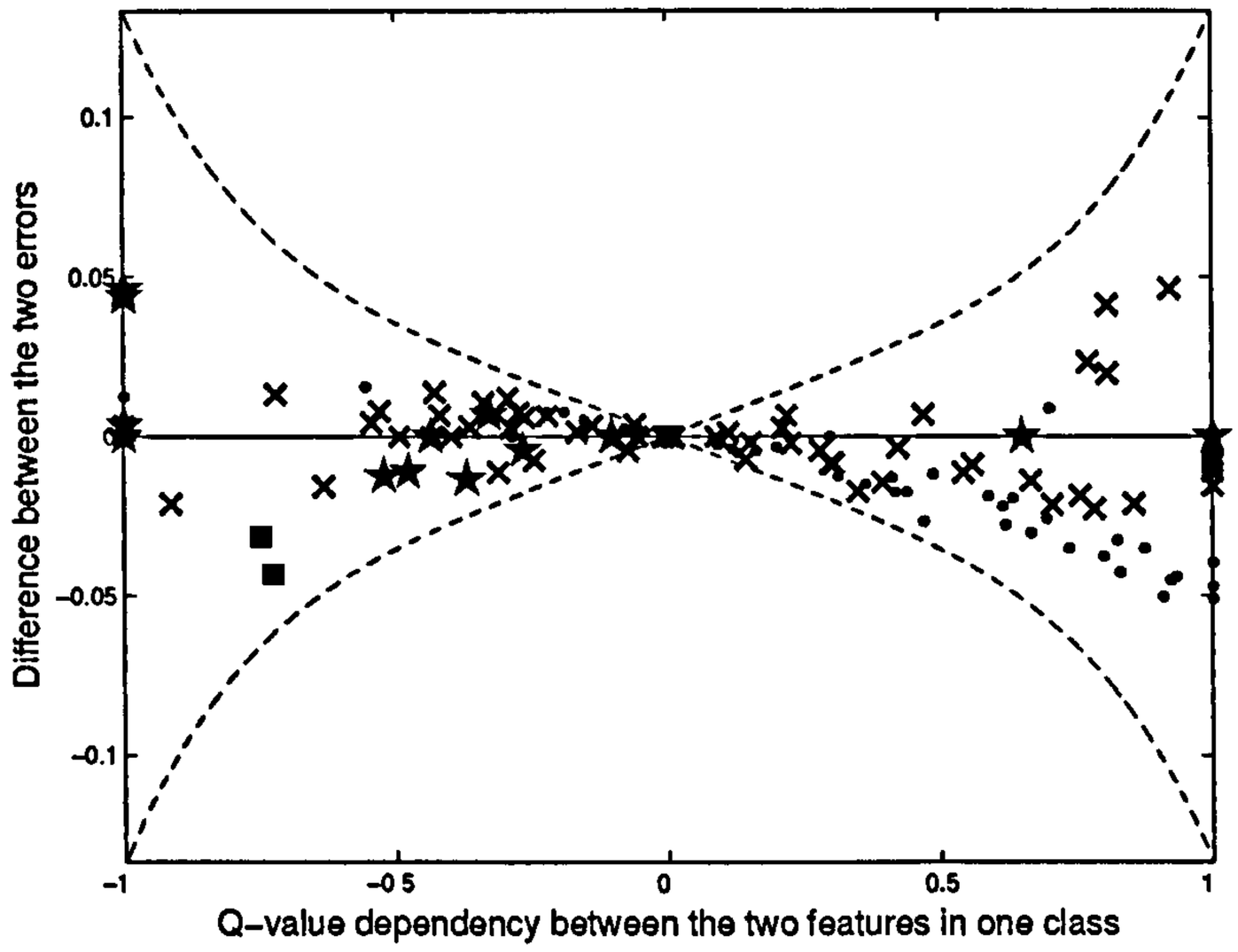


Figure 4.2: Scatterplot of the pairings that fit the requirements. SPECT - •, Wine - ×, Thyroid - ■, Glass - ★

features in only one class and having equal prior probabilities are removed in proposition 3.

Proposition 3 *Let* $\mathbf{x} = [x_1, x_2]^T$ *where* $x_1, x_2 \in \{0, 1\}$ *and let* ω_1 *and* ω_2 *be the classes of interest with* $P(\omega_1) = p$ *and* $P(\omega_2) = (1 - p)$, *then* $E_{IND} - E_{NB}$ *will either take the value 0 or* $\pm(pCov_1 - (1 - p)Cov_2)$, *where* Cov_i *is the covariance between the two binary features given class* ω_i. $Cov_1 = Cov(x_1, x_2|\omega_1) = ad - bc$ *and* $Cov_2 = Cov(x_1, x_2|\omega_2) = eh - fg$.

Proof. There are 4 possible values of (x_1, x_2), and each one can be labelled in one of two classes. Therefore, there are $2^4 = 16$ possible ways of labelling, including the two trivial cases where all four values are labelled in the same class. The proof of this lemma is done by considering all possible labellings.

Consider Table 4.6 rows 1 - 4. Without loss of generality just one of these rows may be considered. Row 1 shows that $[0, 0]^T$ and $[1, 0]^T$ have been assigned to class ω_1 while $[0, 1]^T$ and $[1, 1]^T$ have been assigned to class ω_2. This indicates that $A > E, B > F, C < G$ and $D < H$, so that $E_{IND} = (1 - p)(E + F) + p(C + D)$ and $E_{NB} = (1 - p)(e + f) + p(c + d)$. Expanding out E_{IND} using values from Table 4.2, gives

$$E_{IND} = (1-p)(E+F)+p(C+D)$$

Table 4.6: The possible assignments of the class labels and the resulting differences of ($E_{IND} - E_{NB}$).

	$(0,0)$	$(0,1)$	$(1,0)$	$(1,1)$	Difference
1	ω_1	ω_2	ω_1	ω_2	0
2	ω_2	ω_1	ω_2	ω_1	0
3	ω_1	ω_1	ω_2	ω_2	0
4	ω_2	ω_2	ω_1	ω_1	0
5	ω_1	ω_2	ω_2	ω_2	$p(Cov_1) - (1-p)(Cov_2)$
6	ω_2	ω_2	ω_2	ω_1	$p(Cov_1) - (1-p)(Cov_2)$
7	ω_1	ω_2	ω_1	ω_1	$p(Cov_1) - (1-p)(Cov_2)$
8	ω_1	ω_1	ω_2	ω_1	$p(Cov_1) - (1-p)(Cov_2)$
9	ω_2	ω_1	ω_1	ω_1	$(1-p)(Cov_2) - p(Cov_1)$
10	ω_1	ω_1	ω_1	ω_2	$(1-p)(Cov_2) - p(Cov_1)$
11	ω_2	ω_1	ω_2	ω_2	$(1-p)(Cov_2) - p(Cov_1)$
12	ω_2	ω_2	ω_1	ω_2	$(1-p)(Cov_2) - p(Cov_1)$
13	ω_1	ω_1	ω_1	ω_1	Class of max. prior
14	ω_2	ω_2	ω_2	ω_2	Class of max. prior
15	ω_1	ω_2	ω_2	ω_1	–
16	ω_2	ω_1	ω_1	ω_2	–

$$
\begin{aligned}
&= (1-p)\left((e+f)(e+g)+(e+f)(f+h)\right)+ \\
&\quad p\left((c+d)(c+a)+(d+b)(d+c)\right) \\
&= (1-p)\Big(e(e+f+g+h)+f(e+f+g+h)\Big)+ \\
&\quad p\Big(c(a+b+c+d)+d(a+b+c+d)\Big) \\
&= (1-p)(e+f)+p(c+d)=E_{NB} \qquad (4.8)
\end{aligned}
$$

Consider the assignments given by Table 4.6 rows 5 - 8. Row 5, Table 4.6 indicates that $A > E, B < F, C < G, D < H$, so $E_{IND} = (1-p)E + p(B+C+D)$ and $E_{NB} = (1-p)e + p(b+c+d)$. Expanding E_{IND} to obtain

$$
\begin{aligned}
E_{IND} &= p(B+C+D)+(1-p)E \\
&= p((b+a)(b+d)+(c+d)(c+a)+(d+b)(d+c))+ \\
&\quad (1-p)(e+f)(e+g) \\
&= p\Big(b\underbrace{(a+b+d)}_{1-c}+ad+c(a+b+c+d)+d(a+b+c+d)\Big)+ \\
&\quad (1-p)(e\underbrace{(e+f+g)}_{1-h}+fg)
\end{aligned}
$$

$$\begin{aligned} &= p(b\underbrace{-bc+ad}_{Cov_1}+c+d)+(1-p)(e\underbrace{-eh+fg}_{-Cov_2}) \\ &= p(b+c+d+Cov_1)+(1-p)(e-Cov_2) \\ &= E_{NB}+pCov_1-(1-p)Cov_2 \end{aligned} \tag{4.9}$$

Assignments in Table 4.6, rows 9 - 12 lead to the final possible value of difference between E_{IND} and E_{NB}. In row 9, Table 4.6, $E_{NB} = pa + (1-p)(f+g+h)$.

$$\begin{aligned} E_{IND} &= (1-p)(F+G+H)+pA \\ &= (1-p)((e+f)(f+h)+(e+g)(g+h)+(f+h)(g+h))+ \\ &\quad p(a+b)(a+c) \\ &= (1-p)\left(f\underbrace{(e+f+h)}_{1-g}+eh+g(e+f+g+h)+h(e+f+g+h)\right)+ \\ &\quad p(a\underbrace{(a+b+c)}_{1-d}+bc) \\ &= (1-p)(f\underbrace{+eh-fg}_{Cov_2}+g+h)+p(a\underbrace{-ad+bc}_{-Cov_1}) \\ &= (1-p)(f+g+h+Cov_2)+p(a-Cov_1) \\ &= E_{NB}+(1-p)Cov_2-pCov_1 \end{aligned} \tag{4.10}$$

Consider the assignments in rows 13 - 16 of Table 4.6. In the case of rows 13 and 14 of Table 4.6, row 13 shows that $A > E, B > F, C > G$ and $D > H$. Leading to

$$p(A+B+C+D) > (1-p)(E+F+G+H) \tag{4.11}$$

$$p > (1-p) \tag{4.12}$$

When the features are giving no information then the cases are assigned to the class with the largest prior probability. In the case of row 13 this is class ω_1. For row 14 this becomes class ω_2. For the case when $p = (1-p)$, that is when $P(\omega_1) = P(\omega_2) = \frac{1}{2}$ equation (4.12) becomes a contradiction. This assignment of classes can not be achieved with the NB classifier, in the case with equal prior probabilities. The assignment of class labels to the cases would become random.

Consider finally rows 15 and 16 of Table 4.6. The assignment in row 15 indicates that $pA > (1-p)E, pB < (1-p)F, pC < (1-p)G$ and $pD > (1-p)H$. As the tables are independent then $pAD = pBC$ and $(1-p)EH = (1-p)FG$,

$$pAD > (1-p)EH \tag{4.13}$$

$$(1-p)FG > pBC \tag{4.14}$$

$$pAD > (1-p)EH \quad = \quad (1-p)FG > pBC \tag{4.15}$$

Equation 4.15 implies that $pAD > pBC$ but this is not true and so this assignment can not be obtained by the NB classifier. This is due to NB being a linear classifier and can therefore not recognise exclusive OR problems. The difference of E_{IND} and E_{NB} will therefore take one of three values.

$$E_{IND} - E_{NB} = 0 \tag{4.16}$$

$$E_{IND} - E_{NB} = pCov_1 - (1-p)Cov_2 \tag{4.17}$$

$$E_{IND} - E_{NB} = (1-p)Cov_2 - pCov_1 \tag{4.18}$$

Thus $E_{IND} - E_{NB}$ will either take the value 0 or $\pm\frac{1}{2}(Cov_1 - Cov_2)$.

■

4.4.1 Empirical analysis

Simulated data

Without access to the values of Table 4.1, it is impossible to calculate the value of the difference from equations 4.17and 4.18. By generating 10,000 matrices of the form of Table 4.1 with random prior probabilities, p, the aim is to give an outline of the differences that can occur.

From the 10,000 pairs of matrices only 1949 gave a difference in E_{NB} and E_{IND}. Figure 4.3 shows the histogram of the non-zero differences. About 55% have a difference between -0.05 and 0.05 while 94% have a difference between -0.1 and 0.1. This simulation implies that even when the error values differ, the majority of the time the difference has an absolute value below 0.1.

Real data - traditional recorded cases.

Traditional recorded case data allows us to calculate the dependent probability distribution (Table 4.1) and the associated independent distribution (Table 4.2). SPECT data taken from the UCI repository [11] has 22 binary features. This gives 231 pairs of features to look at. From these 231 pairs of features, 229 pairs had non-zero differences between E_{NB} and E_{IND}. These 229 differences are plotted in the histogram in Figure 4.4. Of these differences 100% fall within the range -0.1 to 0.1.

The difference in the errors caused by assuming features are class-conditionally independent when they are not, is in reality seen to be small in the majority of cases. We have

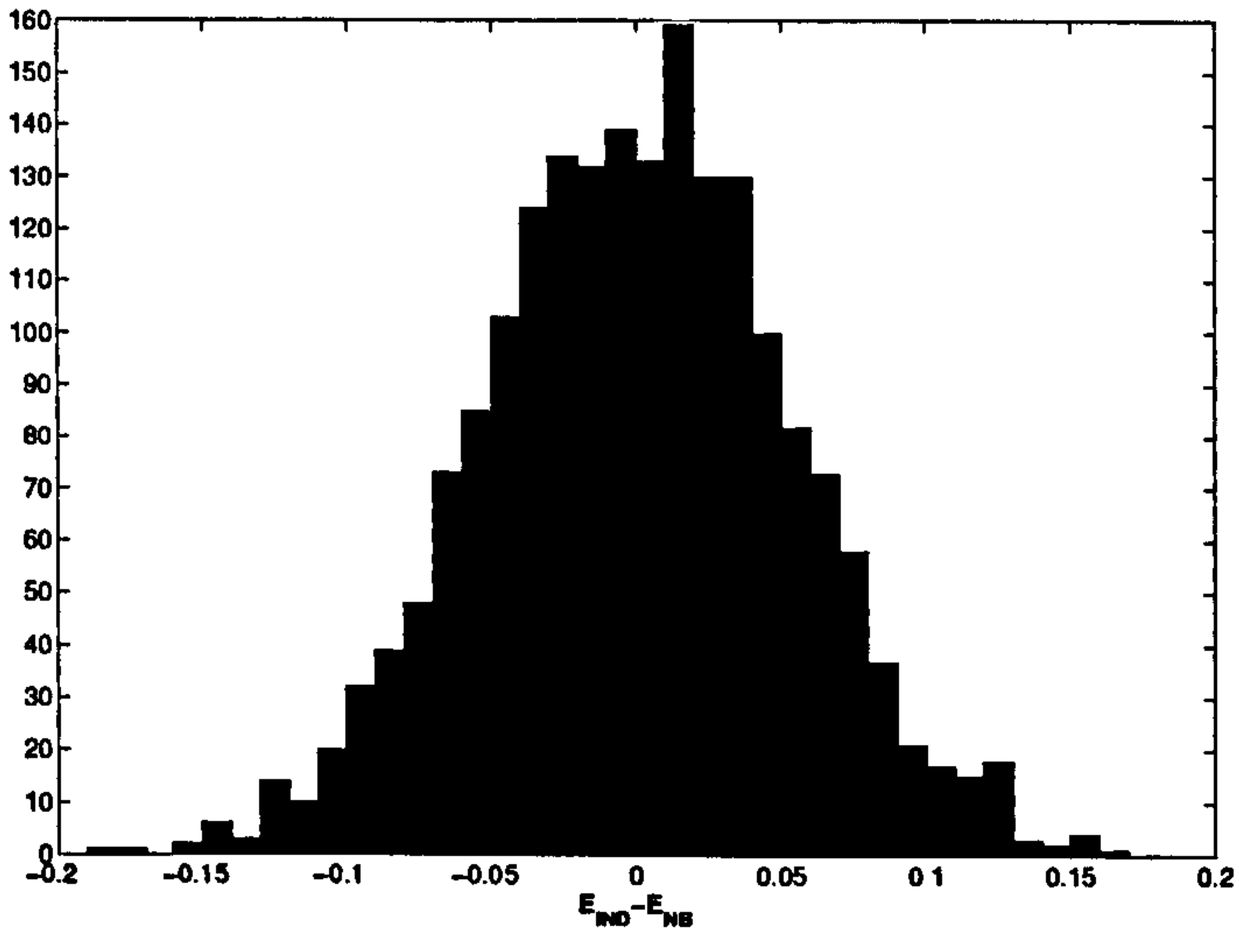


Figure 4.3: Histogram of the values of $E_{IND} - E_{NB} \neq 0$ for the simulated data.

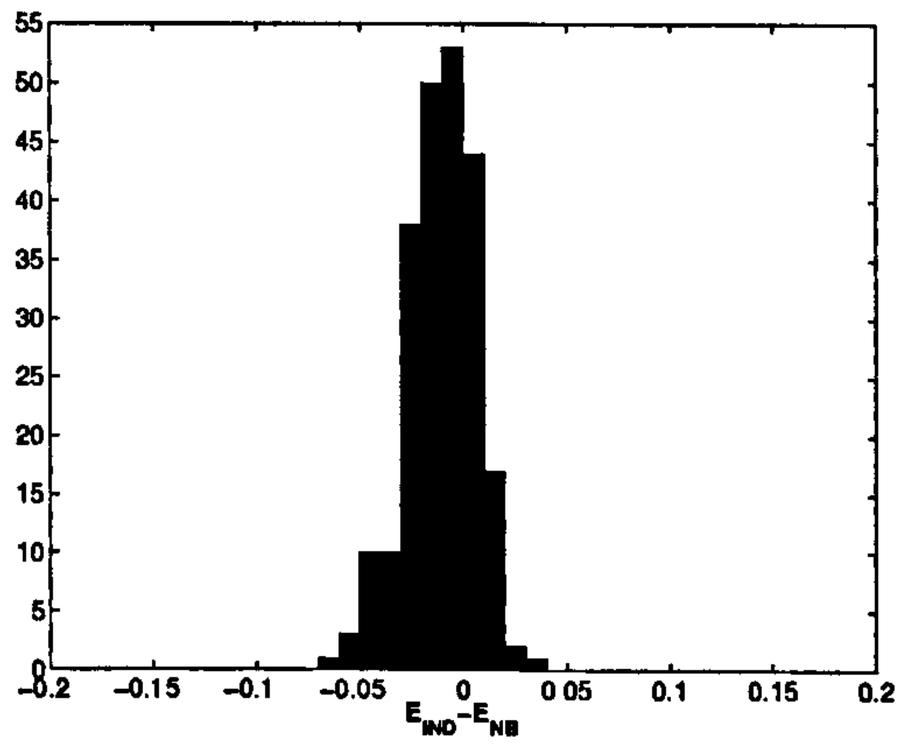
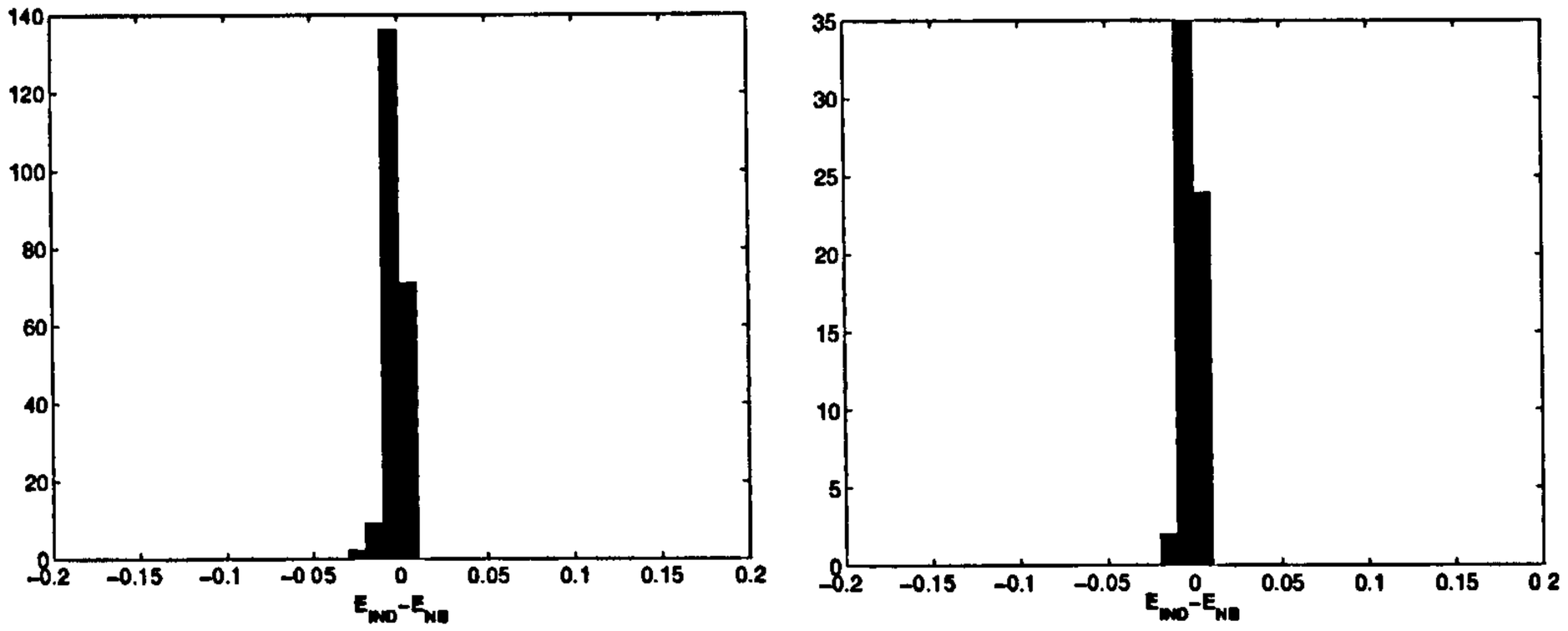


Figure 4.4: Histogram of the values of $E_{IND} - E_{NB} \neq 0$ for SPECT data.



(a) (b)

Figure 4.5: Histogram of the values of $E_{IND} - E_{NB} \neq 0$ for the DEFRA data (a) BSE data (b) Scrapie data.

looked at the difference between the error made by assuming independence and the error made by relating the same choices back to the true distribution.

4.4.2 Application to traditional BSE and Scrapie data

The traditional DEFRA data sets for BSE and Scrapie both give real two-class problems. From this recorded case data we may calculate the dependent and independent distributions required. The BSE data contains 30 features which gives 435 feature pairs. Out of these 435, 218 pairs gave a difference where the difference between E_{IND} and E_{NB} was non-zero. The non-zero differences are plotted in histogram form in Figure 4.5(a). The maximal difference achieved here is 0.0234.

DEFRA provided a reduced version of the Scrapie data which is described by 14 features giving 91 feature pairs. 61 of the feature pairs gave a non-zero difference of E_{IND} and E_{NB}. These are plotted in the histogram in Figure 4.5(b). The maximal difference achieved is 0.0104.

For both sets of data the histograms show that if the difference for the two errors was non-zero then it would fall within the range -0.05 and 0.05.

In summary the difference between the error committed by assuming that the features are conditionally independent when they are not and the error committed when independence is not assumed is in reality minimal in the two feature two class case. This has been supported by the calculation of this difference from "traditionally stored" data.

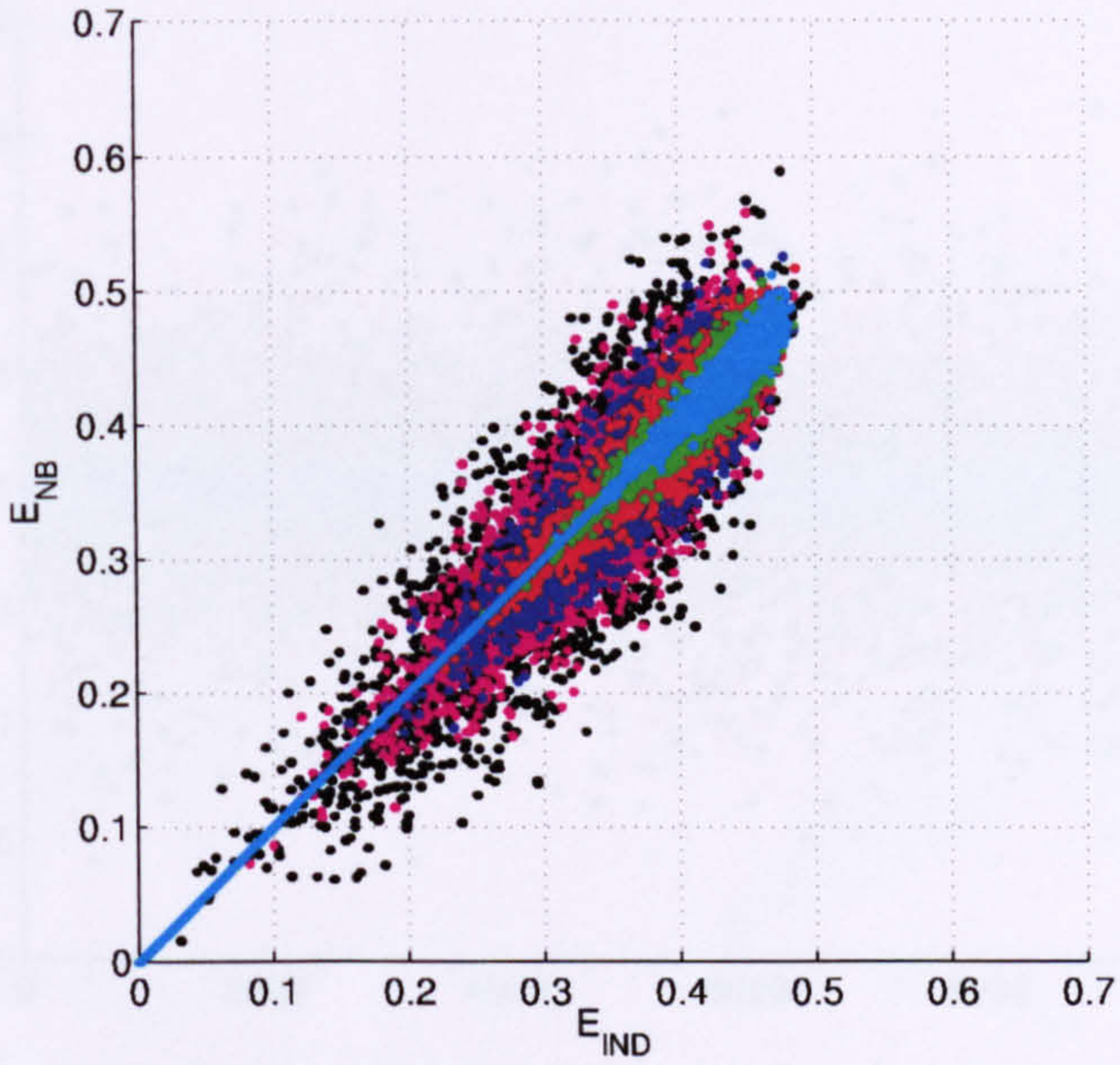


Figure 4.6: Scatterplot of the values of E_{NB} versus E_{IND}, (black $n = 2$, magenta $n = 3$, blue $n = 4$, red $n = 5$, green $n = 6$, cyan $n = 7$)

4.5 Increasing the number of features

Until now only the two feature case has been considered. When more features are added into the problem the structure of the errors becomes more complex. Proving this in the way that Proposition 3 was proved would be inefficient. For 3 features the number of representative binary vectors, **x** is 8. For a two class problem this allows 256 possible combinations of class assignments. A new insight would seem to be needed in order to attempt a proof.

4.5.1 Simulation 1 - Relationships of E_B, E_{NB} and E_{IND}

To study the relationships between the errors as the number of features n increases, random matrices as in Table 4.1 were generated as before allowing the calculation and storage of E_B, E_{NB} and E_{IND}. 10,000 sets of matrices were generated for each n, where $n = 2, \ldots 7$.

The 10,000 points (E_{IND}, E_{NB}) for each n are plotted in Figure 4.6. The diagonal trend of the points indicates a positive relationship between the two types of error. E_{IND} can over and under estimate E_{NB}, demonstrated by the points above and below the diagonal. However, as n increases the cloud of points becomes more concentrated around the diagonal. This indicates that as more features are added to the problem the differences between the error that can be calculated from the assumed "independent distribution",

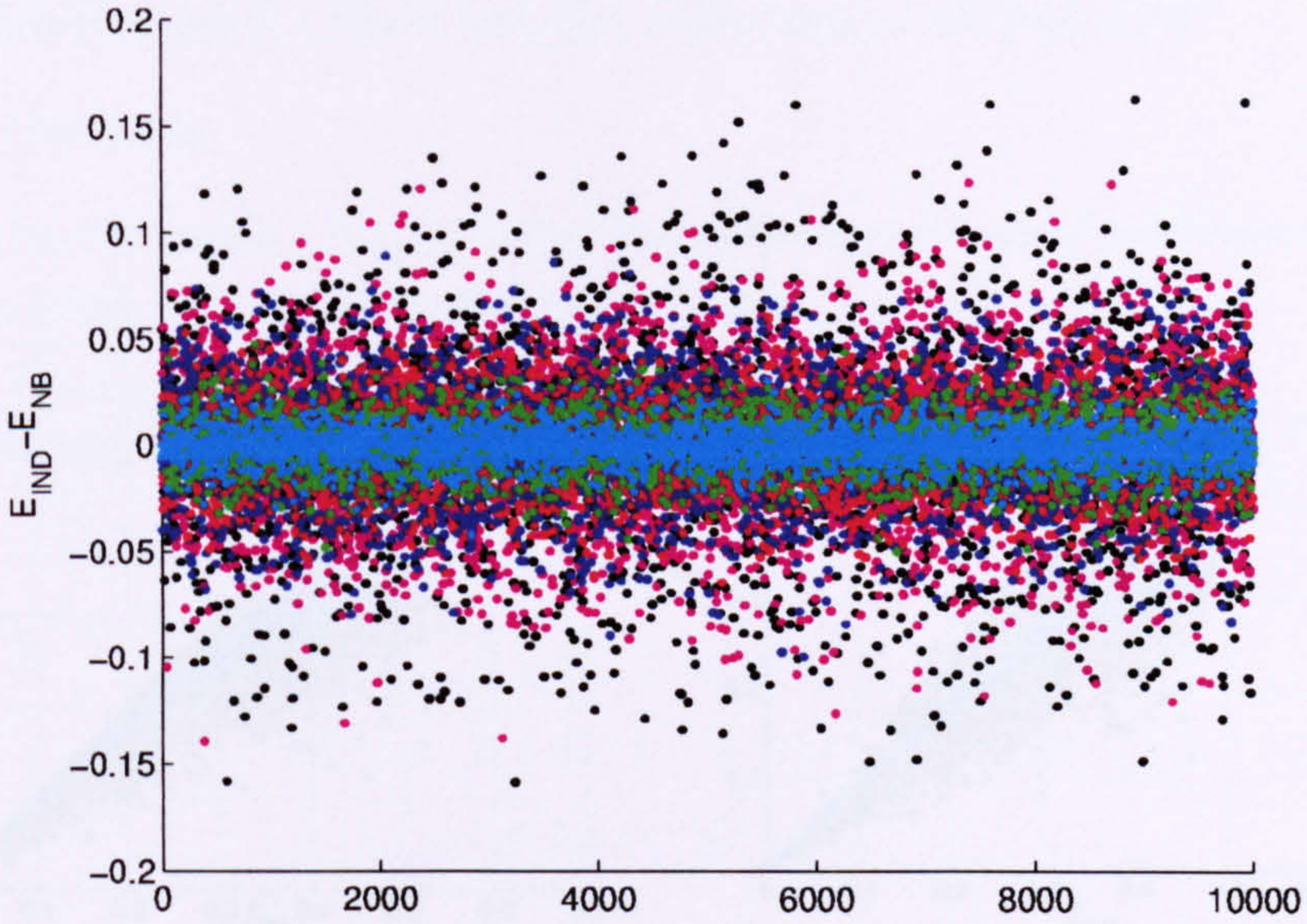


Figure 4.7: Scatterplot of the values of $E_{NB} - E_{IND}$, (black $n = 2$, magenta $n = 3$, blue $n = 4$, red $n = 5$, green $n = 6$, cyan $n = 7$)

E_{IND} becomes closer to the error actually committed by NB on the true "dependent" distribution. This effect is better seen in Figure 4.7 with the vertical spread of $E_{IND} - E_{NB}$ becoming smaller as n increases.

Figure 4.8 (a) and (b) show (E_{NB}, E_B) and (E_{IND}, E_B) plotted respectively. E_B and E_{NB} are both calculated from the true distribution, (Table 4.1 for n=2). E_B is the minimal possible error and therefore $E_{NB} \geq E_B$. This inequality is demonstrated in Figure 4.8(a) where every point lies on or below the diagonal line. In Figure 4.8 it can be seen that E_{IND} can underestimate E_B, i.e. $E_{IND} < E_B$, by the points plotted above the diagonal. In Figure 4.9(b) it can be seen that E_{IND} only underestimates E_B in the cases of two and three features, denoted by the black and magenta points plotted below the zero line. The tendency of E_{NB} and E_{IND} to overestimate as n increases is shown by the arc given to the points that sit below the diagonal in Figures 4.8(a) and (b)

Figures 4.9 (a) and (b) show that by the case of $n = 5$ the size of the maximal difference between either E_{NB} or E_{IND} and E_B does not decrease. E_{NB} and E_{IND} can both match the error of E_B but it is still unclear exactly when this can happen.

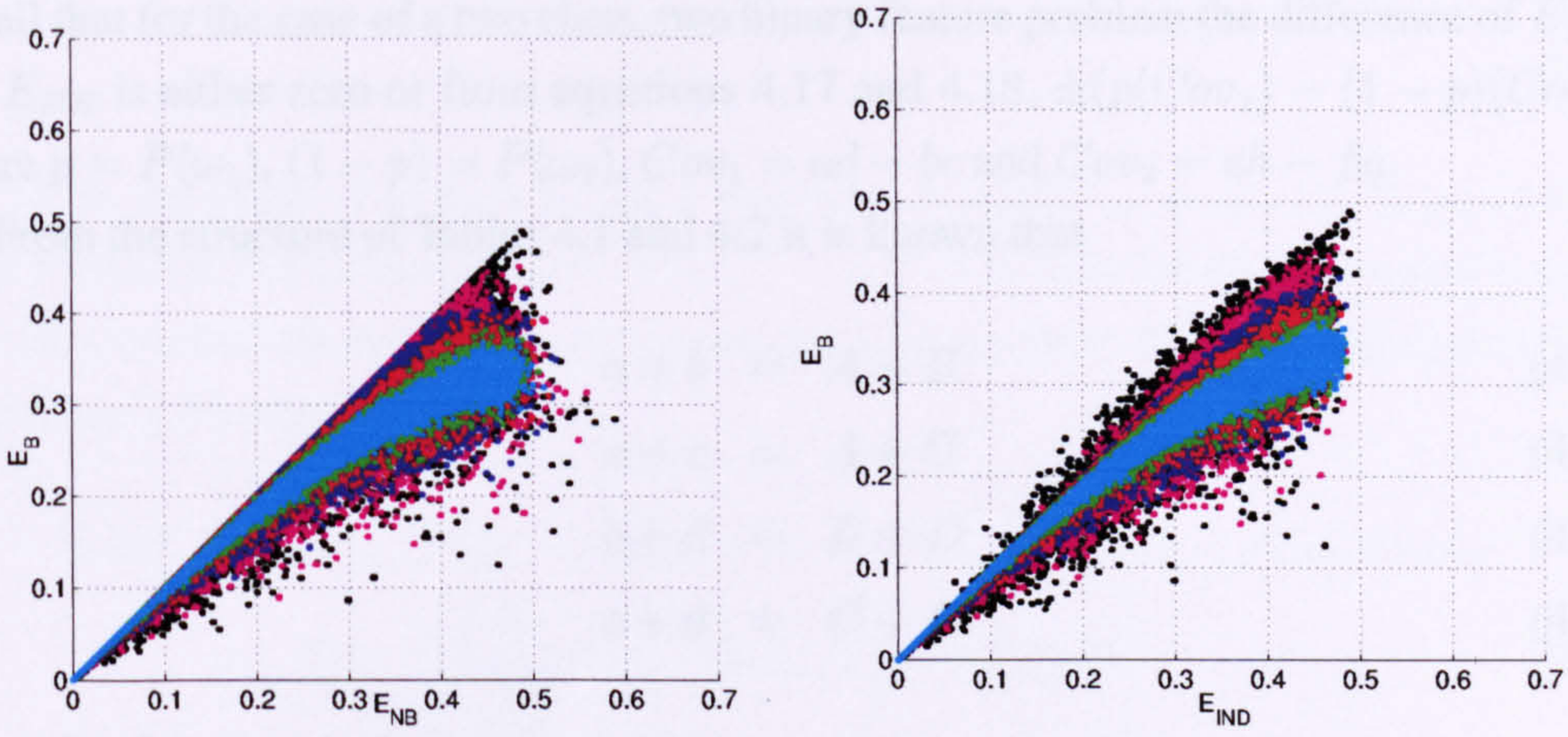


Figure 4.8: Scatterplot of the values of E_{NB} versus E_B, (black $n = 2$, magenta $n = 3$, blue $n = 4$, red $n = 5$, green $n = 6$, cyan $n = 7$)

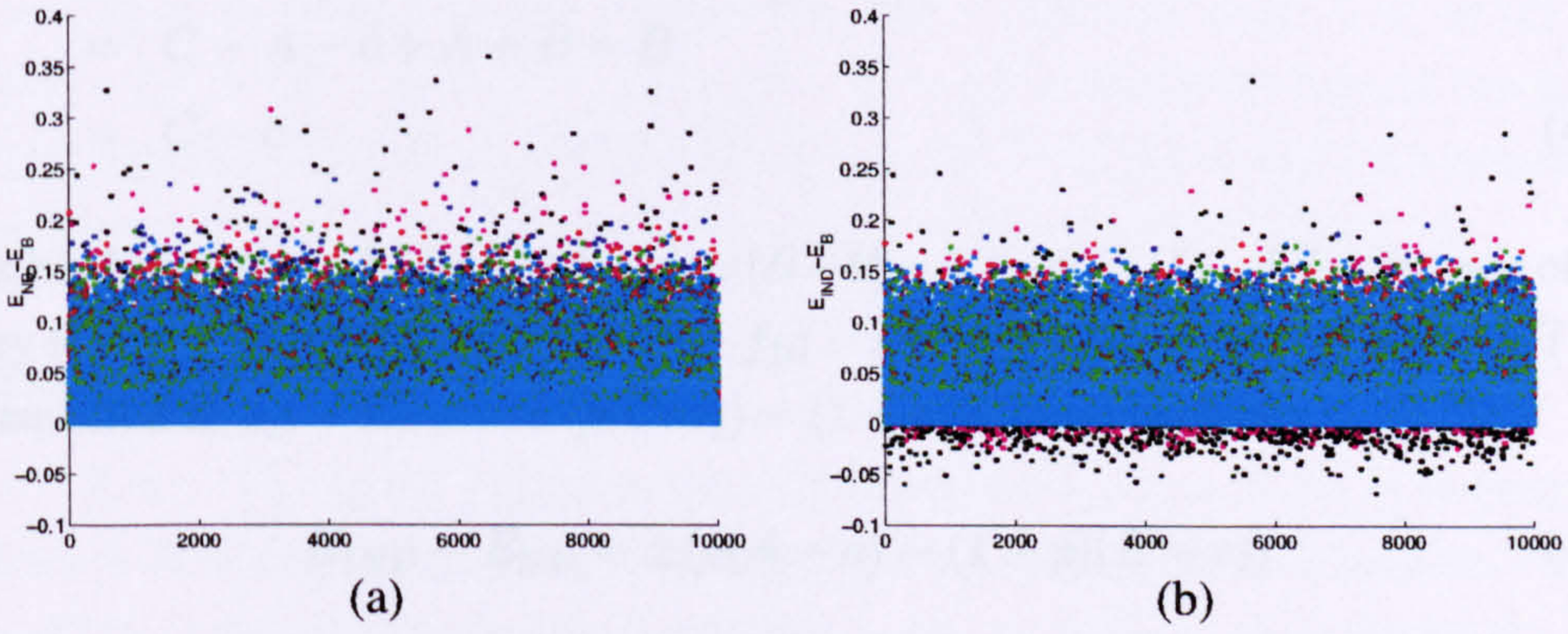


Figure 4.9: Scatterplot of the values of $E_{NB} - E_B$, (black $n = 2$, magenta $n = 3$, blue $n = 4$, red $n = 5$, green $n = 6$, cyan $n = 7$)

4.5.2 Simulation 2 - How are the differences structured?

Case $n = 2$ features.

Recall that for the case of a two class, two binary feature problem the difference of E_{IND} and E_{NB} is either zero or from equations 4.17 and 4.18, $\pm(p(Cov_1) - (1-p)(Cov_2))$ where $p = P(\omega_1)$, $(1-p) = P(\omega_2)$, $Cov_1 = ad - bc$ and $Cov_2 = eh - fg$.

From the structure of Tables 4.1 and 4.2 it is known that

$$a+b = A+B \tag{4.19}$$
$$a+c = A+C \tag{4.20}$$
$$b+d = B+D \tag{4.21}$$
$$c+d = C+D \tag{4.22}$$

Using the above equivalence expressions,

$$\begin{aligned}
Cov_1 &= ad - bc \\
&= C - ac - c^2 - dc - A + a^2 + ab + ac \\
&= C - A - c(C+D) + a(A+B) \\
&= C - A - c(C+D) + (1-b-c-d)(A+B) \\
&= C - A - c(C+D) + (1-c-(B+D))(A+B) \\
&= C - A - c(A+B+C+D) + A + B - C(A+B) - (B+D)(A+B) \\
&= C - A - c + A + B - \underbrace{(b+d)(a+b)}_{B} \\
&= C - A - c + A + B - B \\
&= C - c
\end{aligned} \tag{4.23}$$

It can be shown that $|ad-bc| = |A-a| = |B-b| = |C-c| = |D-d|$ for the case of two binary features. This is also true for $|eh-fg| = |E-e| = |F-f| = |G-g| = |H-h|$ The equation $E_{IND} - E_{NB} = \pm(p(Cov_1) - (1-p)(Cov_2))$ becomes

$$E_{IND} - E_{NB} = \pm(p(A-a) - (1-p)(E-e)) \tag{4.24}$$

Therefore the difference between E_{IND} and E_{NB} will be zero or equation (4.24). Recall that $A = P(x_1 = 0|\omega_1)P(x_2 = 0|\omega_1)$, $a = P(x_1 = 0, x_2 = 0|\omega_1)$, $E = P(x_1 = 0|\omega_2)P(x_2 = 0|\omega_2)$ and $e = P(x_1 = 0, x_2 = 0|\omega_2)$. Then

$$E_{IND} - E_{NB} = \pm(p(A-a) \pm (1-p)(E-e))$$

$$= \pm\Big(p(P(x_1=0|\omega_1)P(x_2=0|\omega_1) - P(x_1=0,x_2=0|\omega_1)) - (1-p)(P(x_1=0|\omega_2)P(x_2=0|\omega_2) - P(x_1=0,x_2=0|\omega_2))\Big) \quad (4.25)$$

Let v_k be the value of feature x_k for a particular $\mathbf{x}$ where $v_k \in \{0,1\}$. A particular $\mathbf{x}$ with n features is represented as $\mathbf{x} = [v_1,\ldots,v_n]^T$. Let $d_1(\mathbf{x})$ be the difference between the class conditional probability of a particular $\mathbf{x}$ using the n features independently and dependently for class ω_1.

$$\begin{aligned} d_1(\mathbf{x}) &= P(\omega_1)P(x_1=v_1|\omega_1)\times\ldots\times P(x_n=v_n|\omega_1) - P(\omega_1)P(\mathbf{x}|\omega_1) \\ &= P(\omega_1)\left[\left(\prod_{k=1}^{n} P(x_k=v_k|\omega_1)\right) - P(\mathbf{x}|\omega_1)\right] \end{aligned} \quad (4.26)$$

Let $d_2(\mathbf{x})$ be the equivalent for the same $\mathbf{x}$ in class ω_2

$$d_2(\mathbf{x}) = P(\omega_2)\left[\left(\prod_{k=1}^{n} P(x_k=v_k|\omega_2)\right) - P(\mathbf{x}|\omega_2)\right] \quad (4.27)$$

For $\mathbf{x} = [x_1=0, x_2=0]^T$, $d_1(\mathbf{x}) = |A-a|$ which in turn is equivalent to $|Cov_1|$. So

$$E_{IND} - E_{NB} = \pm(|d_1(\mathbf{x})| \pm |d_2(\mathbf{x})|) \quad (4.28)$$

To simplify the structure of this difference let $D(\mathbf{x}) = \pm(|d_1(\mathbf{x})| \pm |d_2(\mathbf{x})|)$ then $E_{IND} - E_{NB} = D(\mathbf{x})$. As $|ad-bc| = |A-a| = |B-b| = |C-c| = |D-d|$ and $|eh-fg| = |E-e| = |F-f| = |G-g| = |H-h|$ for the case of two binary features, equation (4.28) is equivalent for any $\mathbf{x} = [x_1 = v1, x_2 = v_2]^T$. An example of the structure of this difference for the case when $n=2$ is given in Figure 4.10.

Simulation $n=3$ features

For $n=3$, $\mathbf{x} = [x_1,x_2,x_3]^T$ where x_i can take the values $v_1,v_2,v_3 \in \{0,1\}$ respectively. Equation 4.28 is no longer equivalent for any given $\mathbf{x}$, i.e. the difference between $P(x_1 = v_1|\omega_1)P(x_2=v_2|\omega_1)P(x_3=v_3|\omega_1)$ and $P(\mathbf{x}=[v_1,v_2,v_3]^T|\omega_1)$ is no longer the same for all $\mathbf{x}$.

Consider the two class problem posed in Figure 4.11 with three binary features. The prior probabilities of the two classes are $P(\omega_1) = 0.5158$ and $P(\omega_2) = 0.4842$. The entries in the "dependent" distribution (Fig 4.11, tables (a) to (d)) are $P(\omega_i)P(\mathbf{x}=[x_1 = v_1, x_2=v_2, x_3=v_3]|\omega_i)$. The entries in the "independent" tables (Fig 4.11, tables (e)

The tables shown include the prior probabilities of the classes, $P(\omega_1) = 0.4513$ and $P(\omega_2) = 0.5487$. Let $\mathbf{x} = [0,0]^T$ and $\mathbf{x}' = [0,1]^T$.

Tables for the "dependent distribution".

ω_1	$x_1 = 0$	$x_1 = 1$
$x_2 = 0$	0.1431	0.1012
$x_2 = 1$	0.0871	0.1199

ω_2	$x_1 = 0$	$x_1 = 1$
$x_2 = 0$	0.0888	0.4532
$x_2 = 1$	0.0009	0.0057

Tables for "independent distribution"

ω_1	$x_1 = 0$	$x_1 = 1$
$x_2 = 0$	0.1246	0.1197
$x_2 = 1$	0.1056	0.1014

ω_2	$x_1 = 0$	$x_1 = 1$
$x_2 = 0$	0.0886	0.4534
$x_2 = 1$	0.0011	0.0055

$$E_{IND} = 0.0886 + 0.0011 + 0.1197 + 0.0055 = 0.2149$$
$$E_{NB} = 0.0888 + 0.0009 + 0.0057 + 0.1012 = 0.1966.$$
$$E_{IND} - E_{NB} = 0.2149 - 0.1966 = 0.0183$$

$$d_1(\mathbf{x}) = 0.1431 - 0.1246 = 0.0185, \quad d_1(\mathbf{x}') = 0.0871 - 0.1056 = -0.0185$$
$$d_2(\mathbf{x}) = 0.0888 - 0.0886 = 0.0002, \quad d_2(\mathbf{x}') = -0.0009 + 0.0011 = 0.0002$$

$$d_1(\mathbf{x}) - d_2(\mathbf{x}) = 0.0185 - 0.0002 = 0.0183 = -d_1(\mathbf{x}') - d_2(\mathbf{x}')$$

If $\mathbf{x}$ is taken to be any $\mathbf{x}$ then it can be seen that $E_{IND} - E_{NB} = \pm|d_1(\mathbf{x})| \pm |d_2(\mathbf{x})|$

Figure 4.10: Example of error differences when $n = 2$

Tables for dependent distribution

Class ω_1.

$x_1 = 0$	$x_2 = 0$	$x_2 = 1$
$x_3 = 0$	0.0862	0.0546
$x_3 = 1$	0.0179	0.0477

(a)

$x_1 = 1$	$x_2 = 0$	$x_2 = 1$
$x_3 = 0$	0.0740	0.0674
$x_3 = 1$	0.0991	0.0690

(b)

Class ω_2

$x_1 = 0$	$x_2 = 0$	$x_2 = 1$
$x_3 = 0$	0.1006	0.0537
$x_3 = 1$	0.0583	0.0281

(c)

$x_1 = 1$	$x_2 = 0$	$x_2 = 1$
$x_3 = 0$	0.0650	0.0396
$x_3 = 1$	0.0874	0.0514

(d)

Tables for independent distribution.

Class ω_1

$x_1 = 0$	$x_2 = 0$	$x_2 = 1$
$x_3 = 0$	0.0607	0.0522
$x_3 = 1$	0.0502	0.0433

(e)

$x_1 = 1$	$x_2 = 0$	$x_2 = 1$
$x_3 = 0$	0.0909	0.0783
$x_3 = 1$	0.0753	0.0648

(f)

Class ω_2

$x_1 = 0$	$x_2 = 0$	$x_2 = 1$
$x_3 = 0$	0.0827	0.0460
$x_3 = 1$	0.0720	0.0400

(g)

$x_1 = 1$	$x_2 = 0$	$x_2 = 1$
$x_3 = 0$	0.0837	0.0465
$x_3 = 1$	0.0728	0.0404

(h)

Figure 4.11: Probability tables for $n = 3$ features example.

to (h)) may be calculated from the "dependent" ones. The entries in the "independent" tables are $P(\omega_i)P(x_1 = v_1|\omega_i)P(x_3 = v_3|\omega_i)P(x_3 = v_3|\omega_i)$. From the tables from Figure 4.11,

$$\begin{aligned} E_{IND} &= 0.0607 + 0.0502 + 0.0460 + 0.0400 + 0.0837 + 0.0465 + 0.0728 + 0.0404 \\ &= 0.4403 \end{aligned} \tag{4.29}$$

$$\begin{aligned} E_{NB} &= 0.0862 + 0.0179 + 0.0537 + 0.0281 + 0.0650 + 0.0396 + 0.0874 + 0.0514 \\ &= 0.4293 \end{aligned} \tag{4.30}$$

Therefore, $E_{IND} - E_{NB} = 0.0109$. Consider $\mathbf{x} = [0,0,0]^T$ and $\mathbf{x}' = [0,0,1]^T$. Then

$$\begin{aligned} d_1(\mathbf{x}) &= P(\omega_1)\left(P(x_1 = 0|\omega_1)P(x_2 = 0|\omega_1)P(x_3 = 0|\omega_1) - P(\mathbf{x}|\omega_1)\right) \\ &= 0.0607 - 0.0862 \\ &= -0.0255 \end{aligned} \tag{4.31}$$

$$\begin{aligned} d_2(\mathbf{x}) &= P(\omega_2)\left(P(x_1 = 0|\omega_2)P(x_2 = 0|\omega_2)P(x_3 = 0|\omega_2) - P(\mathbf{x}|\omega_2)\right) \\ &= 0.0827 - 0.1006 \\ &= -0.0178 \end{aligned} \tag{4.32}$$

$D(\mathbf{x}) = -0.0255 + 0.0178 = -0.0077$. Similarly, $d_1(\mathbf{x}') = -0.0323$ and $d_2(\mathbf{x}') = -0.0137$. Giving $D(\mathbf{x}') = 0.0323 - 0.0137 = 0.0186$ Then

$$\begin{aligned} E_{IND} - E_{NB} &= -D(\mathbf{x}) + D(\mathbf{x}') \\ &= -0.0077 + 0.0186 \\ &= 0.0109 \end{aligned} \tag{4.33}$$

It is suggested that $E_{IND} - E_{NB}$ can take the value zero, $\pm D(\mathbf{x})$ or $\pm D(\mathbf{x}) \pm D(\mathbf{x}')$ where $\mathbf{x}$ and $\mathbf{x}'$ are particular cases of length $n = 3$.

As a theoretical proof of the possible differences at this stage is impractical without new insight, 10,000 sets of probability matrices were randomly generated in the form of Figure 4.11 (a) to (d). From these the probability tables figure 4.11(e) to (h) were calculated. The values of $E_{IND} - E_{NB}$ for each of the 10,000 sets was calculated and stored. The 8 values of $D(\mathbf{x})$ for all $\mathbf{x}$ were also stored.

The 10,000 differences took one of the values,

$$E_{IND} - E_{NB} = 0 \tag{4.34}$$
$$E_{IND} - E_{NB} = \pm D(\mathbf{x}) \tag{4.35}$$
$$E_{IND} - E_{NB} = \pm D(\mathbf{x}) \pm D(\mathbf{x}') \tag{4.36}$$

The differences were separated as follows, 7139 took the value zero. 1528 took the value of equation 4.35 and the remaining 1333 took the value of equation 4.36.

Case $n = 4$ features

The number of possible $\mathbf{x}$ is now increased to 16. Therefore in a two class problem there are 65,536 possible different labellings of the classes.

When 10,000 sets of probability matrices were generated with four features, the number of possible differences increased. The simulated matrices indicated that $E_{IND} - E_{NB}$ could take one of six values.

$$E_{IND} - E_{NB} = 0 \tag{4.37}$$
$$E_{IND} - E_{NB} = D(\mathbf{x}) \tag{4.38}$$
$$E_{IND} - E_{NB} = D(\mathbf{x}) \pm D(\mathbf{x}') \tag{4.39}$$
$$E_{IND} - E_{NB} = D(\mathbf{x}) \pm D(\mathbf{x}') \pm D(\mathbf{x}'') \tag{4.40}$$
$$E_{IND} - E_{NB} = D(\mathbf{x}) \pm D(\mathbf{x}') \pm D(\mathbf{x}'') \pm D(\mathbf{x}''') \tag{4.41}$$
$$E_{IND} - E_{NB} = D(\mathbf{x}) \pm D(\mathbf{x}') \pm D(\mathbf{x}'') \pm D(\mathbf{x}''') \pm D(\mathbf{x}'''') \tag{4.42}$$

Equation 4.42 shows that $E_{IND} - E_{NB}$ can be equivalent to the difference using five of the possible 16 representative $\mathbf{x}$. Figure 4.12 shows that out of the 10,000 generated data points 7044 gave a difference of zero between E_{NB} and E_{IND}. The remaining 2956 points are spread between the differences of equation 4.38 to equation 4.42

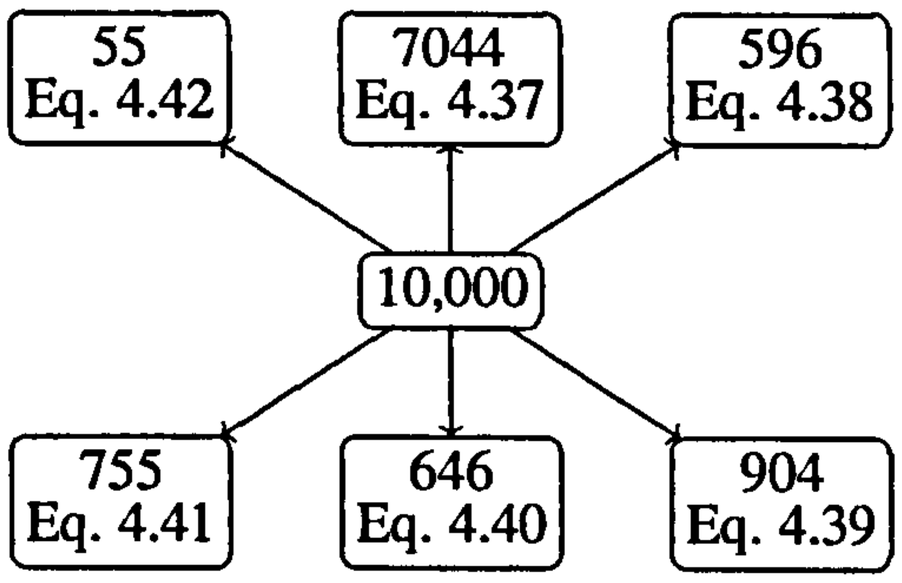


Figure 4.12: The possible errors for $n = 4$ features

However, as there are 65,536 possible class assignments the 10,000 generated probability tables will not cover all possibilities. As such it can not be guaranteed that the differences found are complete. This simulation has given an insight into the structure of the difference between E_{IND}, the error made by assuming conditional independence of the features and E_{NB}, the error relating to the choices made in the independent distribution to the true "dependent" distribution. At this stage no theoretical proof seems likely without a new approach but theses simulations have suggested the possible differences of E_{NB} and E_{IND}.

4.6 Chapter Summary

This chapter investigated the effects of the errors made by NB when class-conditional independence of the features was assumed despite it being untrue. This is precisely the situation given by the assumption made in the probability tables. The NB error calculable from the probability tables, E_{IND} was compared to the error made by NB on the related true distribution that incorporated all feature dependencies, E_{NB}.

Empirical bounds were found for the difference between E_{IND} and E_{NB} in the case of two classes and two features where the features were independent in one class and the level of dependency between the two features (Q measured by Yules Q-statistic) in the second class was known. It was shown that 100% of the differences fell within $\pm B_2$ where

$$\pm B_2 = \frac{Q + Q^5}{15}$$

Removing the restrictions on the levels of dependencies allowed between the features and the requirement for equiprobable classes led to the theoretical result of the actual difference between E_{IND} and E_{NB} for the case of two classes and two features,

$$E_{IND} - E_{NB} = \pm(pCov_1 - (1-p)Cov_2)$$

where Cov_i is the covariance of the two features in class ω_i and p is the prior probability of class ω_1.

The final sections of this chapter began to give an insight into how the difference between E_{IND} and E_{NB} would vary in the two-class case where the number of features increased above two.

The results suggested that assuming class conditional independence when it is not true resulted in a minimal difference between the errors made by NB. Ultimately, NB appears to be a sensible and practical option when using non-traditional probability tables.

Chapter 5

Analysis of Naïve Bayes performance for binary data

5.1 The traditional data

There is an interest in the performance of NB in relation to other classifiers when using medical data with binary features. The two DEFRA data sets are stored with binary features and may be used for such a comparison.

5.1.1 Medical data

An overview of machine learning methods for medical diagnosis is given by Kononenko [74]. The methods can be used to assist specialists in specific diagnostic problems or to train students and non-specialist clinicians in a specialist area. The methods can be used either to predict the outcome of a case or to aid in the diagnosis of a case. An early study applying pattern recognition methods to complex medical data was a study in 1981 by Titterington *et al* [143]. The requirements needed for a learning system to be effective in the medical domain include good performance, an ability to cope with missing or noisy data, a transparency of the diagnostic knowledge, an ability to explain the decisions made and to ultimately reduce the number of tests required to obtain a reliable diagnosis.

Traditional data of recorded labelled cases is not always abundant for the case of medical diagnostics. This can be attributed to many factors, some of which include:-

- In the case of rare diseases there may not be many recorded cases.
- There is uncertainty about which features (symptoms) to collect data on.
- The collection of data may not be performed uniformly across the available population.
- Recorded cases are generally already suspects of the disease in question leading to low variability in the overall data

In 1993 Kononenko [73] noted that despite learning systems being successful with regard to medical diagnostic problems they were not widely accepted. Some of the reasons as to why learning systems are not widely accepted are that 1) the set of symptoms describing the diseases are fixed leaving little room for the accommodation of natural variability in the disease presentation, 2) apparent sensitivity of the models to missing data, 3) generated learning rules containing too few features limiting the ability of explanation, and 4) a subjective resistance by specialists.

In the analysis by Kononenko NB outperforms the specialists; this is taken as an indication of how well NB performs, rather than an indication of inability of the specialists. It is noted that the specialists were prepared to use NB with an incorporated facility of explanation of the model and its decisions. Medical research can often run in parallel with

the classifier design so there is an element of uncertainty in the expert knowledge which may not always be handled well by the classifiers. For example, a study by Marshall *et al* [95] looks to examine the relationship between the dietary intake of a mother and the birth weight of her baby. The collection of the data was run at the same time as designing the classifier (a neural network). The classifier was needed to express the evidence already contained in the evolving study to the medical experts as well as exploring the data as it was gathered for new knowledge. More complex decision making models have been proposed and implemented with varying degrees of success [20, 104, 113, 145, 146].

The veterinary medical domain experiences many of the same problems as the human domain. A study by Geenan *et al* [50] proposed a method with the ability to build classifier models from literature for Classical Swine Fever because of a lack of "traditional" recorded case data.

A survey of pattern recognition methods in veterinary diagnosis was carried out by Cockcroft, [22]. The levels of the use of pattern matching, statistical probabilities and pathophysiological reasoning were investigated. Pattern matching refers to comparing input data (a case) to stored templates for the diseases. The differential diagnosis list is then constructed of the profiles that closely match the input case. Statistical probabilities are computed using the prevalence of the diseases and the frequency of the occurrence of the symptoms observed within those diseases. Pathophysiological reasoning uses the symptoms to identify the abnormalities. A list of differential diagnoses is then constructed using diseases that may explain the symptoms. The results of the study indicated that more experienced clinicians use pattern matching and recalled previous case presentations. The pathophysiological reasoning was used more by the novice clinicians. However, all three methods were used to some extent by the clinicians and students. Many clinicians were ready to accept the pattern recognition methods that explained their decision making process. The formalising of these diagnostic procedures as pattern recognition methods demonstrated that many clinicians already used some form of pattern recognition.

Cockcroft has proposed four pattern-matching models for the diagnosis of BSE [21, 23]. The studies suggest giving a weighting to a diagnosis based on the confidence in the decision. A low confidence score would indicate that the case needs to re-examined for other possible diagnoses. The four pattern matching models were tested on a small data set of 100 recorded cases (50 BSE, 50 Non-BSE) reporting a top accuracy of 72%. Pattern matching models base decisions on feature comparisons which limits their discrimination capabilities.

Medical problems are often recorded with binary features representing the presence and absence of symptoms. Asparoukhov *et al* look specifically at the case of using binary features in such problems in a selection of studies, [5–7]. These studies indicate that the

most well-developed methods for handling binary features are currently statistically based ones. However, the traditional statistical classifiers, which include NB, do not appear to cope well with small sample (sparse) data, being more suited to cases with many features or problems with large sample sizes.

A proposed alternative method to using statistical processes is to transform the binary feature problem into a max/min linear programming problem [5, 7] to minimise the misclassification cost. The results for the method are at least as good as that of the statistical classifiers but at the cost of losing some of the transparency that NB has at the decision-making level. This means that for some cases the linear programming method can outperform NB but the decision-making process is not as clear cut.

5.1.2 Traditional DEFRA data

The two "traditional" data sets obtained from DEFRA for BSE and Scrapie contained recorded cases described by the presence and absence of symptoms (features). After discussion with the domain experts any missing values were taken to mean that the particular symptom was absent. The datasets contained recorded cases that were suspects of BSE or Scrapie. This leads to a concern about the variability of the data. All the recorded cases were sent in as suspects of the disease of interest (BSE or Scrapie). Each case was then diagnosed post-mortem giving it a positive or a negative label. This means that even if the case was eventually labelled negative the presentation of symptoms must have been similar to that of a positive case for it to be submitted as a suspect. The classification task within this data is rather diagnosing BSE/ Scrapie within a set of suspects than diagnosis within the general population. This task is more difficult than the task with a wider variety of symptoms for the negative cases.

The Scrapie dataset contained 3676 cases described by 41 features. Of these, 2987 cases were positive for Scrapie while the remaining 689 cases were Scrapie negative.

The BSE dataset contained 204,354 cases described by 31 features. The cases were split into 173,759 BSE positive and 30,595 BSE negative cases.

5.1.3 UCI data

To analyse the performance of various classifiers when working with binary features a variety of datasets are needed. The most widely used data sets are held in the UCI machine learning repository [11]. The number of these datasets that are described purely by binary features is small, but those containing continuous features may be discretised to represent binary data.

Discretisation

A study by Dougherty, Kohavi and Sahami [35] stated that despite the fact that discretisation has often been applied, up until that point no study had considered how discretisation affected the learning processes. The conclusions of their analysis showed that discretisation of continuous features can significantly improve accuracy. For NB all discretisation methods sampled led to a large increase in accuracy. This seems counterintuitive as discretisation means a loss of information. However, discretisation of a continuous feature may approximate the class distribution of the feature better than assuming an inappropriate Gaussian or normal distribution on continuous features. Kohavi and Sahami [70] noted that discretisation methods that make use of the case labels (supervised discretisation) worked better and so were analysed in their study. Static discretisation methods use one discretisation pass through the data for each feature, independent of the other features. Dynamic discretisation methods search the space of all features simultaneously, therefore capturing interdependencies among the features. The study found that was no significant improvement in using dynamic discretisation over static methods. A comparison of error-based discretisation methods and entropy-based methods was also considered in the study. Error-based methods apply a learner to the discretised data and select intervals that minimise the error on the training data. Entropy-based methods assess the entropy regarding the relationship between the intervals of the feature and the class. While error-based techniques always find the optimal partition to reduce the training set error for each feature individually, entropy-based methods include the feature interactions that are present. As a result of including these interactions the entropy methods may fare better in practice.

Hsu, Huang and Wong [58] looked at the effect of the discretisation of continuous features on the performance of NB. It is suggested that a discretisation method may approximate a distribution more accurately than a NB that assumes the distributions of continuous features to be normal. As NB takes all features into account simultaneously the impact of a wrong discretisation for one feature may easily be absorbed by the others under an error performance measure.

Classification error is made up of three parts, a bias term due to the systematic error of the learning system, a variance term due to random variation in the training data and an irreducible term due to noise in the data. Yang and Webb [157] note that if a feature is discretised with large intervals then it is likely that the intervals will contain the decision boundaries. This in turn will affect the bias term in the error. Smaller intervals can relate to there being a smaller number of cases contained within the interval thus affecting the variance term. With a fixed number of cases there has to be a trade off between the interval size and number of intervals. The study concludes that an optimal universal discretisation

Table 5.1: A summary of the statistics of the UCI datasets used in the comparison study.

Dataset	Features, n	Classes, c	Cases, N
Contraceptive Method Choice (CMC)	9	3	1473
Ecoli (ECOL)	7	5	327
Glass Identification (GLAS)	9	6	214
Haberman's Survival data (HABE)	3	2	306
Iris plant identification (IRIS)	4	3	150
Pima Indians Diabetes (DIAB)	8	2	768
SPECT Heart (SPEC)	22	2	267
Thyroid disease (THY)	5	3	215
Waveform Data Generator (WAVE)	21	3	5000
Wisconsin Diagnostic Breast Cancer (WDBC)	30	2	569
Wine Recognition (WINE)	13	3	178
Yeast (YEST)	8	9	1479

strategy for NB is unobtainable.

The discretisation processes used for the UCI data adopted in this study were static ones. Each feature was examined individually to find the best split. The continuous features were converted to binary using two different criteria.

1. Median. The median value of each feature was found. Any feature value under the median was converted to zero while any feature value over the median value was converted to one.

2. Gini criterion. The gini criterion can be used as an impurity function. The criterion is the error rate that results if a case was randomly drawn and classified given the split. The split that minimises the criterion is chosen.

By using discretisation into binary features the intervals may be large and as such contain the decision boundaries for the features thus affecting the bias term in the error. As this term is due to the systematic error of the classifier all classifiers should be using the same discretised data and will be affected equally.

The UCI datasets selected for the comparison are outlined below. All but one of them (SPECT) required discretisation into binary data first. A summary of their statistics is given in Table 5.1.

Contraceptive Method Choice (CMC)

This dataset is a subset of the 1987 National Indonesia Contraceptive Prevalence Survey. The cases held are married women who were either not pregnant or do not know if they were at the time of interview. The task is to predict the current contraceptive method

choice (no use, long-term methods, or short-term methods) of a woman based on her demographic and socio-economic characteristics. As well as the four continuous features, the dataset contained 4 categorical features, each with four categories. Each categorical feature was converted into four binary features, giving a total of 21 binary features.

Ecoli Database (ECOL)

The problem is to determine the cellular localisation sites of proteins. Classes omL, imL and ims were represented by 5, 2, 2 cases respectively. These classes have been removed giving 327 cases from the original 336 cases.

Glass Identification Database (GLAS)

A task motivated by criminological investigation - glass left at crime scenes can be used as evidence if correctly identified. Class 4 contains no example cases and therefore was removed.

Haberman's Survival Data (HABE)

A dataset of cases from a study conducted between 1958 and 1970 at the University of Chicago Billings Hospital on the survival of patients who had undergone surgery for breast cancer. Classify into two classes, Survived 5 years+ or Died within 5 years.

Iris Plant Database (IRIS)

The task is to predict which of three classes an iris plant belongs to based on structural measurements. One class is linearly separable from the other two. This is the best known database to be found in the pattern recognition literature.

Pima Indians Diabetes Database (DIAB)

The diagnosis of diabetes in patients who are females of at least 21 years of age who are of Pima Indian heritage. This set is a selection of cases take from a larger database.

SPECT Heart Database (SPEC)

The task is to diagnose cardiac Single Proton Emission Computed Tomography (SPECT) images into one of two classes, normal or abnormal.

Thyroid Disease Database (THY)

Results of laboratory tests are used to predict the condition of a patients thyroid into one of three classes, euthyroidism, hypothyroidism or hyperthyroidism. The dataset used is entitled new-thyroid in the UCI directory.

Waveform Data Generator (WAVE)

Each class is generated from a combination of 2 of 3 base waves. The waves are generated with added noise in each feature. These classes of waves are known to be difficult concepts to learn.

Wisconsin Diagnostic Breast Cancer Data (WDBC)

A prediction of whether an image contains benign or malignant tissue. Features are computed from a digitised image of a fine needle aspirate of a breast mass. The features describe the characteristics of the cell nuclei present in the image. The diagnostic dataset is used from the UCI repository.

Wine Recognition Database (WINE)

Chemical analysis results of wines grown in the same region of Italy but derived from three different cultivars. The task is to recognise which cultivars a wine comes from. This data set is described as being good for first testing a new classifier but overall not very challenging.

Yeast Database (YEST)

Predicting the cellular localisation sites of proteins. A comparable problem to the ECOLI data. Class ERL contained only 5 examples and was therefore removed reducing the case number from 1484 to 1479.

5.2 The comparison models

As there can be no single optimal classifier design for every type of data, it has become standard to compare as many different classifiers as possible on the data type. Comparisons of various classifiers abound in the literature since classifiers emerged. NB has been used in many comparison studies as a benchmark against new methods. A study by Holte [56] indicated that very simple classification rules perform well on most commonly

used datasets. It is known that NB can be successful in many situations. In fact on many real world data sets NB gives better test set accuracy than any other known method [41].

Care must be taken when conducting a comparison study as the analysis can easily result in statistically invalid conclusions [125] or misconceptions that can propagate into mistakes in further studies [64]. Salzberg [125] looks at several phenomena that if ignored can invalidate the comparison conducted. These phenomena include tuning of the algorithms after the test data has been used or generalising the datasets held in the UCI repository as a representative population. If a method works well on a particular dataset from the UCI repository it should not be assumed that this is representative of the results achievable on all data of this type.

The possibility of mistakes in a comparison study is highlighted in a study by Jamain and Hand [64]. The study demonstrates how misreporting of the NB method can propagate into a mistake in further studies. The authors unravel a confusion over the apparent disagreement in performance of NB in two particular past studies.

5.2.1 Classifiers

This subsection gives a description of the classifiers that are to be used in the comparison.

Naïve Bayes (NB)

NB considers all features to be class-conditionally independent. Therefore no feature dependencies are taken into account when calculating this model from the data.

1. Training.
   (a) Estimate prior probabilities $P(\omega_i)$ from the training set
   (b) Estimate $P(x_j = 1|\omega_i)$ - construction of the conditional probability table
2. Testing. For each case x,
   (a) Calculate the support for x in each class, $g_i(\mathbf{x}) = P(\omega_i) \prod_{j=1}^{n} p(x_j = v_j|\omega_i)$, where g is a vector whose entries are the level of support for the case x in each class ω_i, and v_j is the value of the j^{th} feature (either 0 or 1).
   (b) Select and output the class with the maximal support as the label for x

Exact Match (EM)

The exact match model stores all the training cases together with their class labels. When a new case is submitted for classifying the model retrieves all cases from the stored training ones that exactly match the input one. From these selected matching cases the class

label that occurs most frequently is used to label the input case. If there are no matching cases stored the input case is labelled by the class with the largest prior probability.

Using the cases that match the feature values exactly allows the exact match classifier to specifically model any feature interactions that are present, i.e. it takes all feature dependencies into account.

1. Training.

    (a) Calculate the prior probabilities of each class from the training set.

2. Testing.

    (a) Select equivalent x from the training set

    (b) The case is labeled with the class that occurs most frequently in the cases selected from the training set.

    (c) If there are no equivalent cases in the training set the case is labelled to the class with the largest prior probability

Exact Match with confidences (EMwc)

The exact match with confidences model works on the same principles as exact match but with a correction when there are no or few representative cases stored in the training data. In place of reverting to labelling the case with the largest prior probability when there are no representative cases the test case is classified using the NB classifier. The NB classifier is also used when the difference in the class estimates is not significant using the Z-statistic,

$$Z = \frac{b_1 - b_2}{\sqrt{\frac{b_1(1-b_1)+b_2(1-b_2)}{r}}}, \tag{5.1}$$

where b_1 is the proportion of matched cases from the most represented class, b_2 is the proportion of matched cases from the second most represented class and r is the total number of matched cases. If this calculated Z is greater than 1.65 the difference between the two proportions is significant at the 95% confidence level and so the best represented class label may be assigned. If the difference is not significant then the case is labelled using NB.

For example, for a two class problem where $\mathbf{x} = [1, 0, 1]^T$ is submitted. In the training set there are 25 cases of $[1, 0, 1]^T$ labelled by ω_1 and 26 $[1, 0, 1]^T$ cases labelled by ω_2. The calculated Z for this would be 0.1981. As this is less than 1.65 the difference is not statistically significant and so the case $\mathbf{x} = [1, 0, 1]^T$ would be classified by the NB model.

1. Training.

   (a) Calculate the prior probabilities of each class from the training set.

2. Testing.

   (a) Select equivalent $\mathbf{x}$ from the training set

   (b) If there are less than 5 equivalent cases in the training set revert to classifying the case with NB.

   (c) Else

   - Calculate the number of equivalent cases in each class from those selected.
   - Calculate the Z value for the two best represented classes using equation 5.1.
   - If $Z > 1.65$ assign the best represented class label to the case
   - Else revert to classifying the case using NB

Probability dependence tree (PDT)

This classifier is a form of Bayesian network where each feature can depend on only the class and at most one other feature. The network approximates the class-conditional pmf of the joint distribution in the form of $n-1$ first order dependencies where n is the total number of features. For a problem with n features the discriminant function is calculated as

$$g_i(\mathbf{x}) = P(\omega_{m_1})P(x_{f(m_1)}|\omega_i)P(x_{m_2}|x_{f(m_2)},\omega_i)\dots P(x_{m_n}|x_{f(m_n)},\omega_i) \tag{5.2}$$

where $m_1,\dots,m_n$ is a permutation of $1,\dots,n$ and $f(m_j) \in \{1,\dots,n\}\setminus\{m_j\}$. Features are conditioned by the class label and at most one other feature.

The dependency between two features is calculated as the pairwise mutual information between them. This dependency network allows a selection of the feature dependencies to be taken into account.

Normal Linear Classifier (NLC)

Calculation of the linear discriminant between the classes of the labelled data assuming normal densities with equal covariance matrices. The joint covariance matrix is the weighted average of the class covariance matrices (weighted by the prior probabilities). This classifier implementation is taken from the Matlab toolbox PRTools [38].

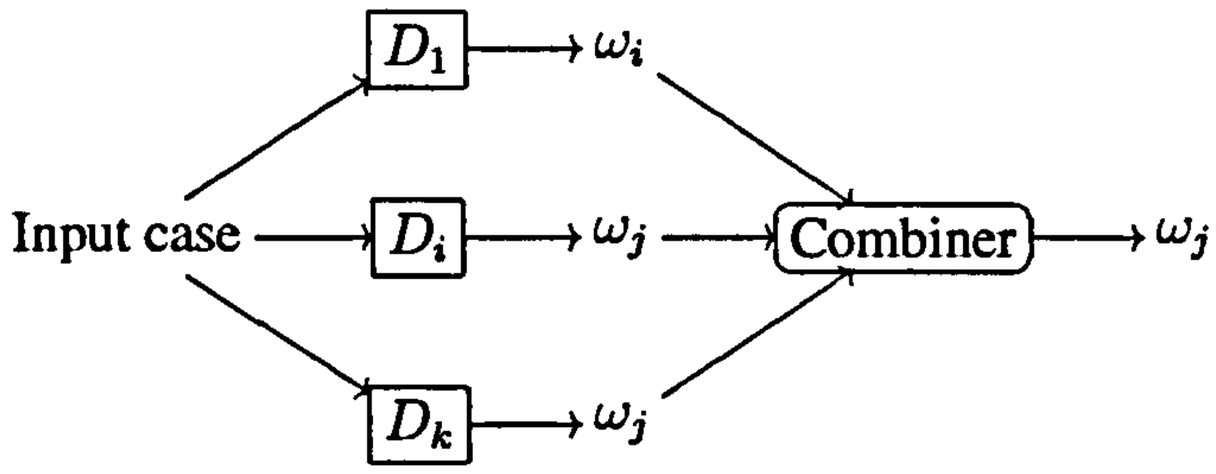


Figure 5.1: A typical classifier ensemble

Logistic Linear Classifier (LOGLC)

Computation of the linear classifier for labeled data by maximising the likelihood criterion using the logistic (sigmoid) function. This implementation is also taken from the Matlab toolbox PRTools [38].

5.2.2 Ensembles

Classifier ensembles follow the logic of "two heads are better than one". An ensemble is a group of classifiers whose outcomes are combined to give an overall decision [51,77]. The performance of an ensemble is rarely worse than the least accurate of its composite members, in many cases better than the average member; and ideally better than the best member.

The ability of an ensemble to improve upon the performance of an individual classifier can be partly attributed to the diversity of the ensemble members. If all the classifiers were identical then there would be no point in combining their results. Diversity may be added to an ensemble by using different classifier models as ensemble members. Perturbing the training data will also increase the diversity of the members. This can be done by bagging (sampling the training set with replacement) or boosting (combining the training set cases with a weighted vote based on the success of previously constructed classifiers) [153]. Another way to increase diversity among the members is to apply feature selection to the original feature set, allowing each classifier member to be trained on a different feature subset. Opitz [107] argues the case for using feature selection with ensembles. The results of the proposed feature selection method created more diversity among the classifiers than either bagging or boosting. GA's produce a population of feature subsets. This "population" of subsets can be used as bases for classifiers to create an ensemble [76].

Figure 5.1 illustrates how an ensemble classifies new input cases. Each of the k classifiers outputs a label for the case. These labels are then combined to give the final decision. The simplest way to combine the results is to use majority voting where the class appearing most frequently among the k classifier outputs is assigned to the case. Kuncheva *et*

al [79] and Oh [106] suggest that the majority vote accuracy of an ensemble will increase if the members are negatively dependent of one another rather than independent. Negative dependency of classifiers mean the tendency of one classifier *not* to follow the other classifiers will be stronger as the number of classifiers increases. Ensembles may also be combined using statistical techniques, belief functions, naïve Bayesian fusion and other schemes [4, 52]. Altincay [4] showed that independence of the base classifiers may not be as important when using some fusion schemes such as Naïve Bayesian fusion (also known as the product rule).

Classifier ensembles have been applied in an attempt to improve NB and Bayesian networks in a variety of ways [93, 123, 151]. NB is seen as a favourable base classifier for an ensemble due to its simplicity and good performance. As NB is a stable classifier it lacks the diversity required to create a good ensemble. Attempts to tackle this lack of diversity between NB models during construction of the ensemble members have included using feature selection [121, 145, 146, 164], boosting [30, 148], estimation of confidence intervals around the point estimations of $p(x_j|\omega_i)$ [120] and adjustments of class probabilities [57]. For a Bayesian network based classifier Kurgan and Cios [80] achieved a significant increase in accuracy at the cost of generating only three to four classifiers in place of a single one.

Ensembles may also be generated to improve performance in a difficult problem area. For example, the recognition rate of a free handwriting recognition system is generally low. Günter and Bunke [53] demonstrate that by selecting well performing feature subsets for the ensemble members an improvement in performance is achieved. A simple classifier ensemble generation procedure can be implemented for comparison to the individual models.

Multiple classifiers

The feature subset is split randomly with replacement to create k subsets, i.e. the subsets are not mutually exclusive. Each of the k subsets are then used to train an individual member of the ensemble. The final decision is made by majority voting of the ensemble members. Ensembles of NB, EM and EMwc have been used with $k = 5$ and $k = 11$.

1. Training
   (a) Randomly select with replacement k subsets of features
   (b) Train k classifiers using each of the subsets of features
2. Testing
   (a) Submit each case to each of the ensemble members to generate a class label

(b) The class with the most votes from the ensemble members is selected as the case label. Ties are broken randomly

Mixed multiple classifiers

The feature subset is split randomly with replacement to create k subsets. When the subset contains less than half the original features use Exact match as the classifier else use NB as the classifier. This was generated with $k = 5$ classifier members and $k = 11$ classifier members. Fusion of the ensemble decision is again made by majority voting.

1. Training

   (a) Randomly select k subsets of features with replacement

   (b) If the subset contains less than half the original number of features then train EM with the subset

   (c) Else train NB with the subset of features

2. Testing

   (a) Submit each case to each of the ensemble members to generate a class label

   (b) The class with the most votes from the ensemble members is selected as the case label. Ties are broken randomly

5.2.3 Experimental setup

As the classes are not represented equally in the datasets, the sampling of the data was stratified. Stratified sampling splits a dataset into training and testing elements ensuring that the two have approximately the same class proportions as the original data set.

- Submit binary data set
- For 100 trials
  - Using stratified sampling, split the data set in 90% for training and 10% for testing.
  - Train each of the 14 classifiers on the 90% training split
    1. Naïve Bayes (NB)
    2. Exact Match (EM)
    3. Exact Match with confidences (EMwc)
    4. Probability Dependence Tree (PDT)

5. Normal Linear Classifier (NLC)
6. Logistic Linear Classifier (LOGLC)
7. Ensemble of 5 Naïve Bayes (5 NB)
8. Ensemble of 11 Naïve Bayes (11 NB)
9. Ensemble of 5 Exact Match (5 EM)
10. Ensemble of 11 Exact Match (11 EM)
11. Ensemble of 5 Exact Match with confidences (5 EMwc)
12. Ensemble of 11 Exact Match with confidences (11 EMwc)
13. Mixed ensemble of 5 Exact Match and Naïve Bayes (5 EMNB)
14. Mixed ensemble of 11 Exact Match and Naïve Bayes (11 EMNB)

- Test the classifiers on the 90% training data split
- Test the classifiers on the 10% testing split of the data
- Store the training and testing errors

- Output the averages of the training and testing errors across the 100 trials.

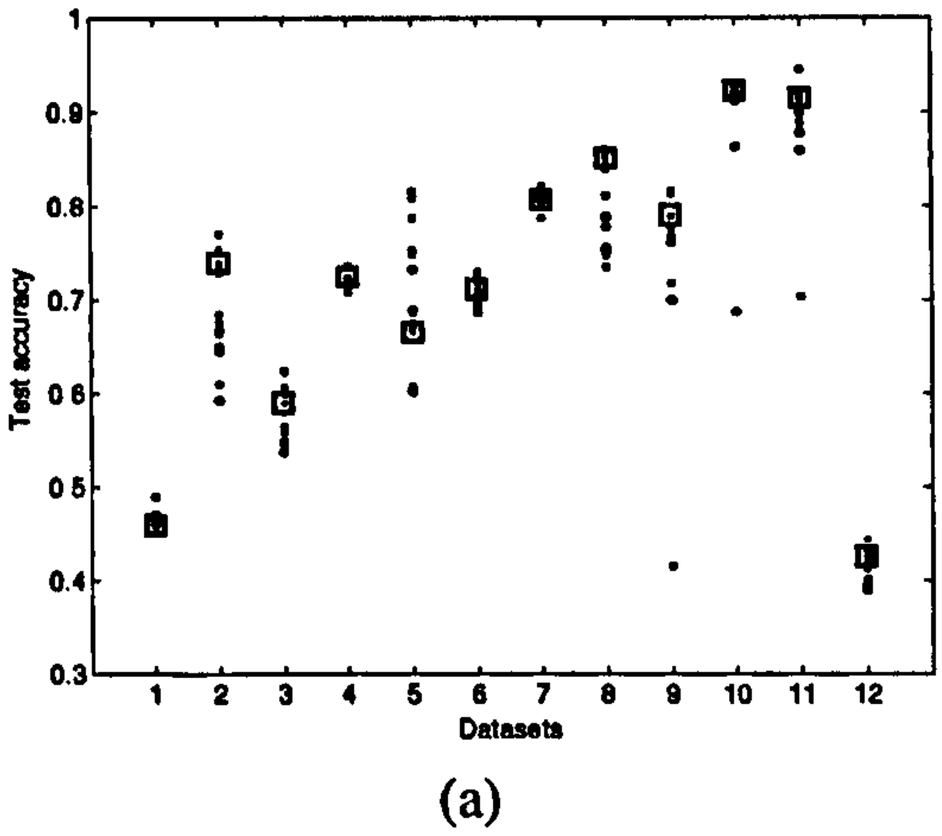
5.3 The comparison results and analysis

5.3.1 Accuracy results

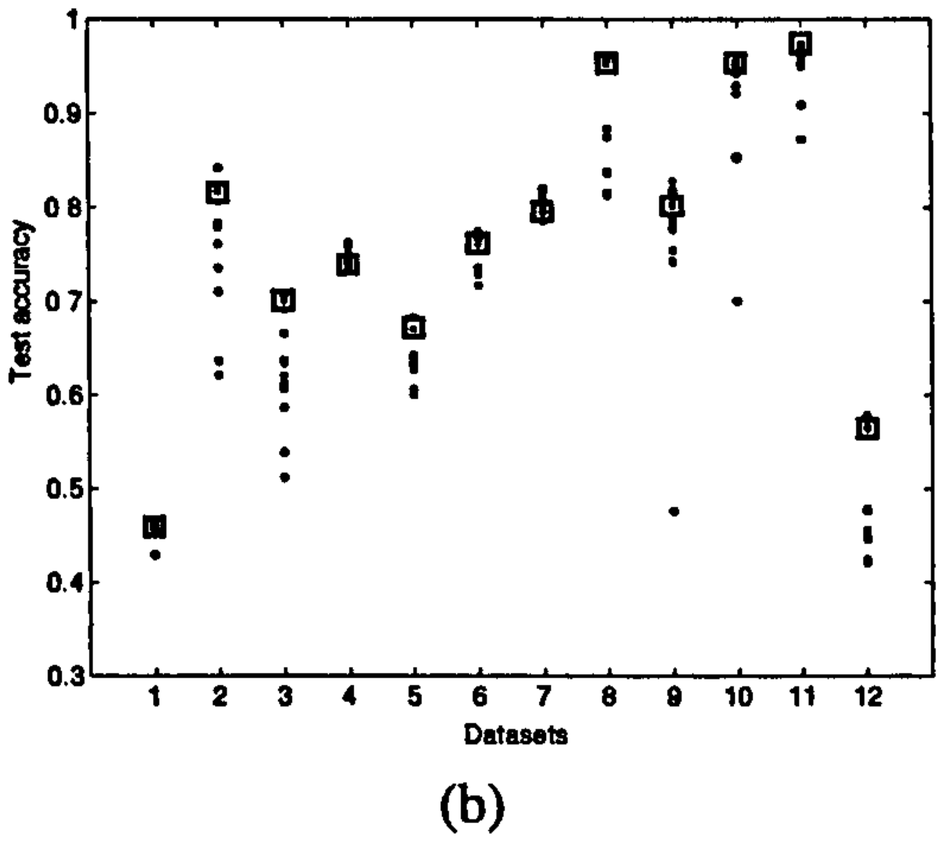
Figure 5.2 shows the spread of the test accuracies achieved by the classifiers on each of the 12 datasets. The squares are centred on the accuracy achieved by the NB classifier. Figure 5.2(a) shows the accuracies achieved when the data was discretised using the median value as the threshold while Figure 5.2(b) shows the results when the split was made using the Gini criterion.

None of the methods appear to perform well for the Contraceptive Method Choice (1) or Yeast (12) datasets with accuracies struggling to reach 60% for both the median split and Gini split versions of the data.

The classifier that has the lower accuracy in Waveform (9) and Wisconsin Diagnostic Breast Cancer (10) for both data split and in the Wine dataset (11) for the median split data is the Exact Match model (EM). This is unexpected especially for the Waveform data as the exact match should perform better with the more training examples that are available. The waveform data has the largest number of training cases available. For the Wine data set (11) exact match appears to achieve lower accuracy when the data has been split using the median threshold but improves to the level of the other classifiers for this data when it is split using the Gini criterion. In fact for the Wine data (11) all



(a)



(b)

Figure 5.2: The test accuracy for the 14 classifiers on each of the 12 datasets (a) Median split data (b) Gini split data.

the classifier models appear to respond better to the data being discretised by the Gini criterion rather then by the median. This is seen by the rise in the spread of the accuracies between Figures 5.2(a) and (b).

NB does not achieve the top performance for any of the given data sets. However, NB appears close to the top performing classifiers for the majority of the data sets. The only dataset the model appears to struggle with is in the median split version of the Iris data (5). Here the poor performance is intriguing as two of the classes are known to be linearly separable, a concept that can be recognised by NB. However, this may be explained by the discretisation as the process results in an adjustment of the feature space thus possibly affecting the linear separability of the classes.

This analysis leads to the question as to whether the performances of the 14 classifier models are significantly different?

5.3.2 Significance of the accuracy results

Each classifier was assigned a rank for each dataset. The classifier achieving the highest mean accuracy was given the highest rank. The ranks for each classifier across the 12 datasets were then summed to give the total rank of each of the classifiers. These total ranks are given in Table 5.2

NB ranks an overall fifth in the data split using the median and an overall of fourth when the data is split using the Gini criterion. The positions of the majority of the classifiers only change at the most by one overall position. However, the probability dependence tree (PDT) moves from third ranked position with the data split using the median down to sixth when the data is split using the Gini criterion. The discretisation process

Table 5.2: Total ranks of the 14 classifiers performance over the 12 discretised data sets.

Classifier	Median		Gini	
	Rank	Overall Position	Rank	Overall Position
NB	107	5	109.5	4
EMwc	108.5	4	127.5	3
PDT	125	3	98	6
NLC	137	2	139	2
LOGLC	141	1	140	1
EM	84	9	72	10
5 NB	41.5	14	37	14
11 NB	56	13	57	13
5 EM	62	12	65	12
11 EM	85	8	87	7
5 EMwc	68.5	10	74	9
11 EMwc	91	6	103	5
5 EMNB	67	11	37.5	11
11 EMNB	86.5	7	82.5	8

used may have an effect on the performances of this classifier.

To find out if the performances of the 14 classifiers on the 12 datasets are different Friedmans two-way ANOVA can be used.

Friedman's two-way ANOVA

Friedmans analysis uses the rank R_{ij} of classifier j in dataset i to calculate a chi-squared statistic,

$$\chi^2 = \frac{12}{nk(k+1)} \left[\sum_{j=1}^{k} \left(\sum_{i=1}^{n} R_{ij} \right)^2 \right] - 3n(k+1) \tag{5.3}$$

where n is the number of datasets, 12, and k is the number of classifiers, 14. If the classifier performances are similar then their ranks would be close to random for the different datasets.

Differences of the classifiers - median split

Take the rankings when the data is split using the median as the discretisation criterion. Comparing all the classifiers the calculated χ^2 value is 57.0405. The degrees of freedom, df for the test are $k - 1 = 13$. The tabulated χ^2 value at the 5% significance level is 22.3620. As calculated $\chi^2 >$ tabulated χ^2 there are significant differences between the classifiers.

There is an interest in the performance of the NB classifier. NB is fifth overall in the rank order. The top five ranked classifiers (LOGLC, NLC, PDT, EMwc and NB) can be compared by calculating the χ^2 statistic. The calculated χ^2 value for these five classifiers is 8.2167. The tabulated χ^2 value is 9.4877, df=4, at the 5% significance level. This result indicates that when the data is split using the median for discretisation the top five performing classifiers are not significantly different from one another at the 5% significance level.

Differences of the classifiers - Gini split

Using the ranks of the 14 classifiers calculated on the Gini discretised data the calculated χ^2 value is 60.2786. At the significance level of 5% the tabulated χ^2 is 22.3620, df=13. As the calculated $\chi^2 >$ tabulated χ^2 there are significant differences between the performances of the 14 classifiers when the data is split using the Gini criterion.

Consider the four classifiers ranked the highest (LOGLC, NLC, EMwc and NB). The calculated χ^2 values is 7.5750. The tabulated χ^2 is 7.8147 at the 5% significance level, df=3. As the calculated χ^2 value is less than the tabulated χ^2 value these four classifiers are not significantly different at the 5% significance level.

Multiple comparisons test for Friedman

As the Friedman two-way ANOVA indicated that there are differences between the classifiers multiple comparison testing will indicate which classifiers are different from which others, [101].

As there are 14 different classifiers there will be $\frac{14(14-1)}{2} = 91$ possible comparisons to make. For the comparison test two classifiers are significantly different at the 5% significance level if

$$\frac{TR_i - TR_j}{SD} > 3.4 \tag{5.4}$$

where $TR_i - TR_j$ is the absolute difference in the total rank of classifier i and classifier j and $SD = \sqrt{12 \times 14 \times \frac{14+1}{6}} = 20.4939$. The value of 3.4 comes from the Z-tables for the two tail significance at the 5% significance level, $1 - \frac{\frac{0.05}{2}}{91} = 0.999725$ giving a Z-table value of 3.4.

Tables 5.3 and 5.4 give the results of the classifiers that are significantly different from one another. In the Gini discretised data, Table 5.4, NB performs significantly better than the ensemble of 5 NB. This result seems counterintuitive as an ensemble is expected to perform no worse than its least accurate member - a single NB. However, this may be explained by the feature sets that the NB models use. For the individual model all the available features are considered simultaneously. This is not the case for the NB that are

Table 5.3: Classifiers that are significantly different using the median discretised data

Classifier	Significantly better than
LOGLC	5 NB, 11 NB, 5 EM, 5 EMNB, 5EMwc
NLC	5 NB, 11 NB, 5 EM, 5 EMNB
PDT	5NB

Table 5.4: Classifiers that are significantly different using the Gini discretised data

Classifier	Significantly better than
LOGLC	5 NB, 11 NB, 5 EM, 5 EMNB
NLC	5 NB, 11 NB, 5 EM, 5 EMNB
EMwc	5 NB, 11 NB
NB	5 NB

members of the ensemble as they get a randomly selected subset of features thus possibly lowering the minimum achievable accuracy of the individual model.

The results of the multiple comparisons of both types of discretised data indicate the significant differences are only minimal, occurring between the "best" ranked and the "worst" ranked classifiers. For the median data only 10 of the possible 91 comparisons proved to be significant. For the Gini data this only increased to 11 significant differences from the possible 91.

This analysis has allowed an insight into the differences between the classifiers but whether the method of discretisation has affected the performance of the classifiers is still unknown.

5.3.3 Testing the main effects and interactions between the classifiers and the discretisation process used

A procedure detailed by Bradley [13] allows a series of Friedman analyses to be performed to test the main effects and interactions in the case where observations are taken under combinations of levels (discretisation processes) involving several different variables (datasets and classifiers). Table 5.5 shows the set-up of the accuracy results to allow the multiple Friedman analyses, where $GA_{i,j}$ is the accuracy of classifier j on dataset i discretised using the Gini criterion and $MA_{i,j}$ is the accuracy of classifier j on dataset i discretised using the median threshold. Only 11 datasets are shown as SPECT did not require any prior discretisation and therefore is not affected by these processes.

Table 5.5: Table for the comparison of the main effects and interactions among several variables

Discretisation	Dataset	Classifier		
		1	...	14
Gini	1	$GA_{1,1}$	...	$GA_{1,14}$
	⋮	⋮	$GA_{i,j}$	⋮
	11	$GA_{11,1}$	...	$GA_{11,14}$
Median	1	$MA_{1,1}$	...	$MA_{1,14}$
	⋮	⋮	$MA_{i,j}$	⋮
	11	$MA_{11,1}$	...	$MA_{11,14}$

Multiple Friedman 1 - Main effect of the classifiers.

To test the main effect of the classifiers Table 5.6 is constructed from Table 5.5. Each entry is the sum of the mean accuracies of classifier j on the gini discretised data and the median discretised data. The ranks of the classifiers on each dataset are then calculated. These ranks then allow the calculation of the χ^2-statistic as in Equation 5.3 to perform the standard Friedman analysis.

The calculated $\chi^2 = 73.9429$. As the tabulated $\chi^2 = 22.3620$, (5% significance level, df=13). The effect of the 14 classifiers are significantly different at the 5% level. This result agrees with the previous Friedman analysis that the classifiers performances are significantly different.

Table 5.6: Table to allow a Friedman test of the main effect of the classifiers

Dataset	Classifier		
	1	...	14
1	$GA_{1,1} + MA_{1,1}$	...	$GA_{1,14} + MA_{1,14}$
⋮	⋮	$GA_{i,j} + MA_{i,j}$	⋮
11	$GA_{11,1} + MA_{11,1}$	...	$GA_{11,14} + MA_{11,14}$

Multiple Friedman 2 - Main effect of the discretisation process.

To study the main effect of the discretisation process Table 5.7 is calculated where each entry is the sum of a row from Table 5.5. The rank for Gini versus median is produced for each datasets. A χ^2-statistic may be calculated from these ranks to give the Friedman analysis.

The calculated $\chi^2 = 4.4545$. The tabulated $\chi^2 = 3.8415$, df=1 at the 5% significance level. This indicates that the discretisation process does effect the performance of the classifiers on the datasets. However, at the 2.5% significance level $\chi^2 = 5.0239$ the effect of the discretisation performance on the accuracies is no longer significant. The support

Table 5.7: Table to allow a Friedman's test of the main effect of the discretisation process

	Discretisation	
Dataset	Gini	Median
1	$\sum_j GA_{1,j}$	$\sum_j MA_{1,j}$
⋮	⋮	⋮
11	$\sum_j GA_{11,j}$	$\sum_j MA_{11,j}$

for the hypothesis that the effect of the discretisation process is significant is weaker than the hypothesis that the effect of the classifier model, (the effect of the classifier model is still significant at the 0.01% significance level).

Multiple Friedman 3 - Effect of the discretisation process and classifier interaction

The final analysis looks at the effect of the interaction between the classifier models and the discretisation process on the accuracies achieved on the datasets. Table 5.8 is calculated from Table 5.5 by calculating the difference in the accuracies of each classifier model on each dataset. These differences are then used to calculate the χ^2-statistic for the Friedman analysis.

Table 5.8: Table to allow the Friedman test of interaction of the classifier and discretisation process.

	Classifier		
Dataset	1	...	14
1	$GA_{1,1} - MA_{1,1}$	...	$GA_{1,14} - MA_{1,14}$
⋮	⋮	$GA_{i,j} - MA_{i,j}$	⋮
11	$GA_{11,1} - MA_{11,1}$	...	$GA_{11,14} - MA_{11,14}$

In this case $\chi^2 = 17.2519$. As the tabulated $\chi^2 = 22.3620$, df=13 at the 5% significance level, it can be concluded that the effect of the interaction between the discretisation process and the classifier on the accuracies achieved on the datasets is not significant.

Summary of the multiple Friedman analyses

It has been shown that the main effect of the discretisation process on the accuracies achieved on the datasets is not significant at the 2.5% significance level. The effect of the interaction between the discretisation process and the classifiers models is not significant on the accuracies achieved on the dataset. This means that the effect of the classifiers is the main contributing factor to the differences in the accuracies achieved on the datasets.

Table 5.9: Test accuracies (in %) for the 14 classifiers on the DEFRA Scrapie data

Classifier	Average Test Accuracy	Classifier	Average Test Accuracy
1) NB	81.43	8) 11 NB	81.31
2) EMwc	81.45	9) 5 EM	81.19
3) PDT	76.71	10) 11 EM	81.17
4) NLC	81.20	11) 5EMwc	81.29
5) LOGLC	81.53	12) 11 EMwc	81.18
6) EM	79.71	13) 5 EMNB	81.37
7) 5 NB	81.20	14) 11 EMNB	81.32

The Friedman analysis has shown that on average NB performs no worse than any of the top performing models across the datasets used. Despite NB not achieving the best performance for any of the data sets individually it still gave accuracies comparable with those top performing classifiers. This simulation has again indicated the good performance of NB in comparison to more complex classifiers, despite NB not taking any feature dependencies into account.

5.4 Application to traditional BSE and Scrapie data

The two "traditional" DEFRA data sets may be used to make the comparison for the performance of the classifiers.

5.4.1 Scrapie recorded case data

The "traditional" Scrapie data contains 3676 cases described by a total of 41 binary features. Each case is labelled into Scrapie or Non-Scrapie classes. The prior probabilities of the two classes are 81% and 19% respectively. As the classes are not represented equally the sampling of the data was stratified ensuring the 81% Scrapie 19% Non-Scrapie class distribution in both the testing and training sets. The random stratified 90% training 10% testing split was replicated 50 times. The average test accuracies for these 50 runs are given in Table 5.9.

A Friedman analysis of the 14 classifiers over the 50 trials gives $\chi^2 = 187.7960$. The calculated $\chi^2 >$ tabulated χ^2 (22.3620) indicating differences between the performances of the classifiers.

NB ranks in third place for the Scrapie data beaten only by LOGLC and EMwc. By using only the top three classifiers a Friedman analysis shows that there are no significant differences between NB, LOGLC and EMwc (Calculated $\chi^2 = 0.8100$, tabulated $\chi^2 = 5.9915$, df=2, 5% significance level).

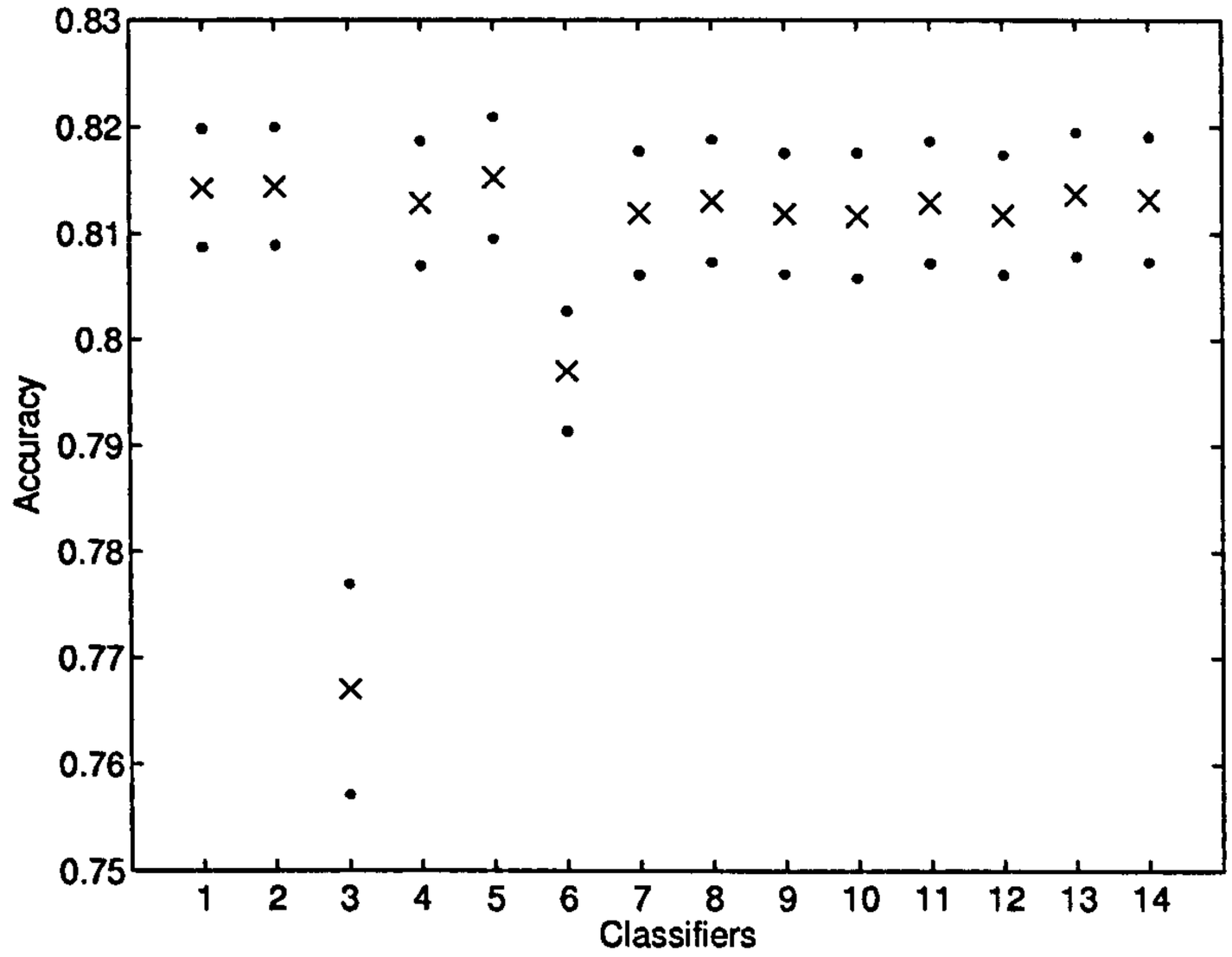


Figure 5.3: Accuracies achieved by the 14 classifiers on the DEFRA Scrapie data, together with their confidence intervals.

Figure 5.3 shows the average test accuracies plotted with their 95% confidence intervals, calculated as

$$\left[A_j - 1.96\frac{\sigma_j}{\sqrt{50}}, A_j + 1.96\frac{\sigma_j}{\sqrt{50}}\right] \tag{5.5}$$

where A_j is the mean accuracy of classifier j and σ_j is the standard deviation of classifier j across the 50 runs.

The performance of two classifiers on a particular dataset, i.e. Scrapie, are said to be significantly different if their respective confidence intervals do not intersect.

From Figure 5.3 it can be seen that the confidence intervals of the majority of the classifiers intersect indicating no significant differences. PDT (3) and EM (6) both give lower accuracies and are as such significantly different to the other classifier models for the Scrapie data.

5.4.2 BSE recorded case data

The "traditional" BSE dataset from DEFRA held 204,354 cases with 173,759 cases of BSE (prevalence of 85%) and 30,595 cases of Non-BSE (prevalence of 15%). Due to the size of this dataset classifier testing was handled as follows

- Split data into 50 sets each of 4000 cases (3400 BSE, 600 Non-BSE).
- For each of the 50 datasets

Table 5.10: Test accuracies (in %) for the 14 classifiers on the DEFRA BSE data

Classifier	Average Test Accuracy	Classifier	Average Test Accuracy
1) NB	81.15	8) 11 NB	83.92
2) EMwc	81.16	9) 5 EM	84.72
3) PDT	78.73	10) 11 EM	84.92
4) NLC	85.25	11) 5EMwc	84.12
5) LOGLC	85.42	12) 11 EMwc	84.30
6) EM	84.76	13) 5 EMNB	84.27
7) 5 NB	83.95	14) 11 EMNB	84.57

  - Split into 90% training and 10% testing used stratified sampling.
  - Train the 14 classifier models on the 90% training.
  - Test the 14 classifier models on the 10% testing data.
  - Store the test accuracies.

- Output the mean of the 50 test accuracies for each of the 14 models.

The mean accuracy of each of the classifiers was calculated as the average across these 50 sets and are shown in Table 5.10.

A Friedman analysis indicates that the 14 classifiers are significantly different (Calculated $\chi^2 = 345.2823 >$ Tabulated $\chi^2 = 22.3620$, df=13, 5% significance level). This time NB ranks 13th over the 14 classifier models. Looking at the mean accuracies plotted with the 95% confidence intervals in Figure 5.4 it can be seen that NB (1) appears lower than the majority of the other classifiers. Here the ensemble models manage to perform well (7) - (14). The top two performing classifiers are NLC (4) and LOGLC (5) appearing significantly better than the other models. These results are surprising with the relative "poor" performance of NB. It is worth noting here that none of the models achieve an accuracy much higher than the 85% prior. This indicates that the classifier models can not perform any better than the clinicians using this data to determine BSE cases within a set of suspects.

The results in Table 5.10 show that NB does not perform as well with the BSE data as all the other data. The LOGLC and NLC perform better than NB. The EM (6) ranks up near the accuracies of these classifiers. In this case the ensembles of classifiers have also performed well especially the ensembles of EM classifiers (9) and (10).

The lack of overlaps in the confidence intervals of the classifiers indicates that the performances of the classifiers are significantly different.

It is worth noting that for both the BSE and Scrapie data the recognition rate of the diseases is around 85% and 81% respectively. This matches the prevalence of the positive

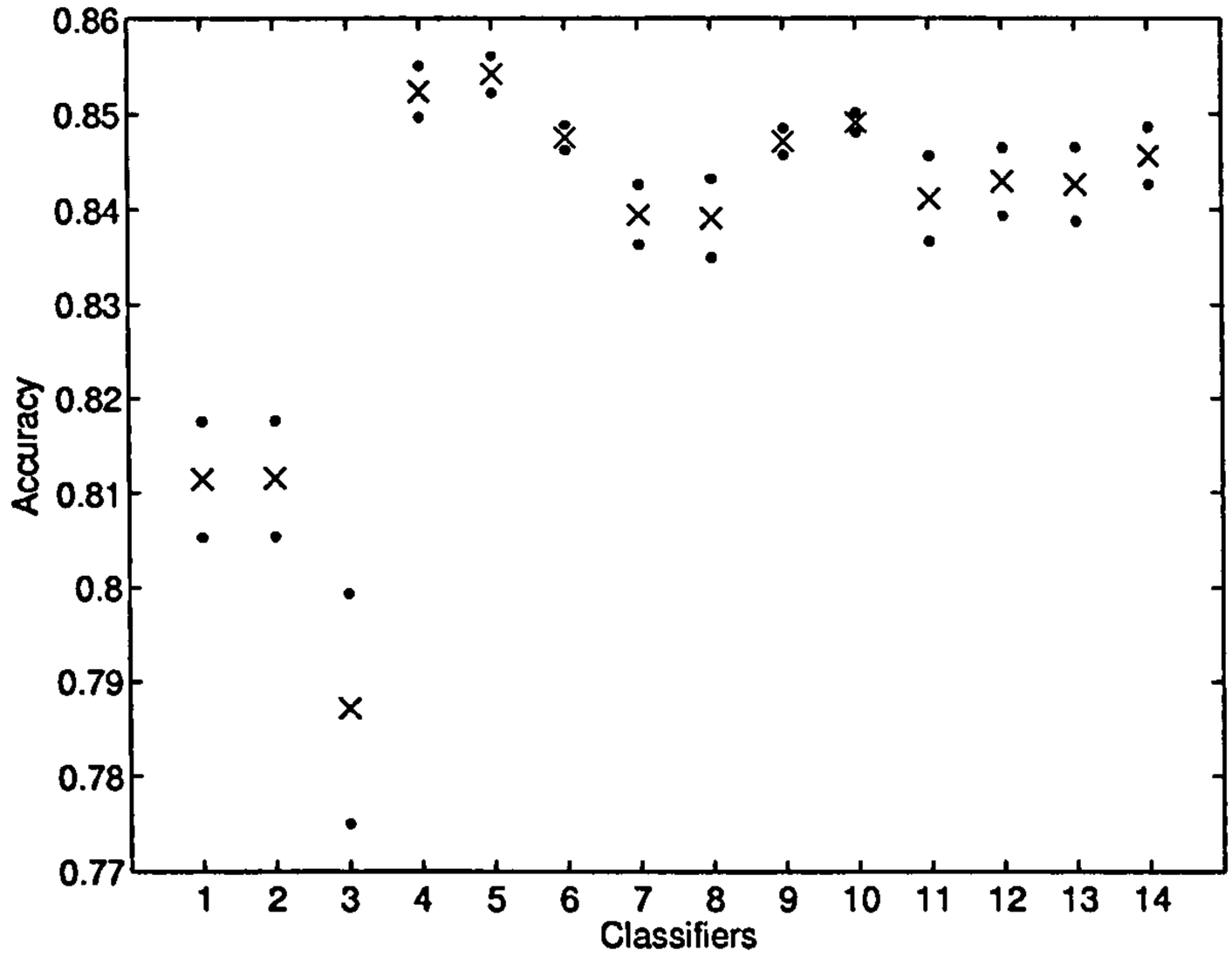


Figure 5.4: Accuracies achieved by the 14 classifiers on the DEFRA BSE data, together with their confidence intervals.

cases in the data. The recognition rate of the classifiers matches the recognition rate of the domain experts within a set of disease suspects.

Classifier models may have more success on the recognition of diseases with data collected across the entire population rather than the narrower set of suspects.

5.5 Chapter Summary

It was shown that the discretisation process or the effect of the interaction between the discretisation and the classifier model on the level of performance was not significant. Therefore, any differences in the accuracies achieved can be mainly attributed to the classifier model.

For the majority of datasets tested NB performed as well as the other "top" performing models (NLC, LOGLC). Due to its performance with the binary data sets NB proved to be a sensible and practical option to consider along with NLC and LOGLC. The explanation ability of the NB model also endears it to practitioners.

For both the BSE and Scrapie data the classifier models were not able to produce accuracies with any great improvement over the prior probabilities. The classifiers can perform no better than the domain experts when classifying within a set of suspects. This again indicates a need for data to be collected across a more general population of cases to fully exploit the advantages of pattern recognition methods.

Chapter 6

Conclusion

Early diagnosis of notifiable diseases like BSE and Scrapie would have an impact on many areas including agriculture, health and the economy. An early diagnosis may prevent transmission to other animals, reduce costs to the farmer and government. There could also be a reduction in the fear of transmission of the disease via British food stock allowing a re-emergence of the export market.

Government agencies like DEFRA collect data on notifiable diseases. However, cases are only reported to such agencies once there is a suspicion of the presence of such a disease. The cases recorded by DEFRA are all therefore suspects of the diseases resulting in low variability across the set of recorded symptoms (features). This makes any classification task from this data harder because the population is reduced to the set of suspects.

As notifiable diseases become rarer within a population (BSE case reporting declining since 1992) there is also a risk of only limited data being available.

The creation of the "non-traditional" expert-estimated probability tables avoids both the above problems. These tables do not come without their own costs though. The class-conditional independence of features is assumed during the expert-estimation phase of the table construction. The estimates represent the conditional probability of a feature within a disease (class). These conditional probabilities may be estimated from a finite traditional data set of recorded cases. As the data set is finite the estimates are known to be imprecise. Accurate probability estimates can only be gained from traditional data if that data is infinite. It is assumed that the experts estimates of the conditional probabilities are reliable.

Domain experts would require that any classifier applied to solve a diagnostic problem would be simple, effective and accurate. Due to the nature of the data it would also be preferable to have a classifier that is stable, meaning that small alterations to the data would not affect the decisions of the model greatly.

The main aim of this study was to investigate the effects and potentials of the the use of the "non-traditional" data in relation to BSE and Scrapie.

6.1 Main investigations and findings.

1. The potentials, limitations and stability of SFS was investigated with regard to selecting a feature subset from the two-class probability table data.

- **Proposition 1** *The theoretical error of the Bayes classifier does not increase when the feature set is augmented.* Proved for the case of independent binary features with two classes.

- **Proposition 2** *The probability of the variable x having the value 1 given class $\omega^{(1)}$ is*

$$P(x=1|\omega^{(1)}) = \frac{1}{c-1}\left(\sum_{i=1}^{c-1} P(x=1|\omega_i)\right).$$

2. Investigation of various feature selection methods for the multi-class probability table data.

3. Investigation of the cascade decision tree classifier performance for classification with multi-class probability data.

4. The implementation of a meta-analysis study of Naïve Bayes and 37 various adaptations of the model to produce landscapes of structural similarity.

5. Investigation into the difference in the errors made by assuming that features are conditionally independent when they are not using the Naïve Bayes classifier, see Table 6.1.

   - **Proposition 3** *Let* $\mathbf{x} = [x_1, x_2]^T$ *where* $x_1, x_2 \in \{0, 1\}$ *and let* ω_1 *and* ω_2 *be the classes of interest with* $P(\omega_1) = p$ *and* $P(\omega_2) = (1-p)$*, then* $E_{IND} - E_{NB}$ *will either take the value 0 or* $\pm(pCov_1 - (1-p)Cov_2)$ *where* Cov_i *is the covariance between the two binary features given class* ω_i*.* $Cov_1 = Cov(x_1, x_2|\omega_1) = ad - bc$ *and* $Cov_2 = Cov(x_1, x_2|\omega_2) = eh - fg$.

6. Comparison investigation of various classifier models on "traditional" data with binary features.

Table 6.1: Links between E_{IND} and E_{NB}

$E_{IND} - E_{NB} =$	Assumptions
$\frac{Q+Q^5}{15}$	- Two binary features - Two equiprobable classes - Features independent in one class
0 or $\pm(pCov_1 - (1-p)Cov_2)$	- Two binary features - Two class task

6.2 Future considerations

There are a number of possibilities for future and continued work on this subject:-

1. The investigations with NB are ongoing. The search for the reasons for the optimality of the model are still not complete. Further to this study investigations into the differences of E_B, E_{NB} and E_{IND} would give further insight into the structures of the errors.

2. Further investigations may lead to a feasible way of extending the theoretical work of the difference between E_{IND} and E_{NB} to include more features. Current simulations have given an insight into the possible structure. Another route of investigation would be to increase the number of classes included in the task.

3. The meta-analytic study of the NB adaptations could lead to many investigations in this area. Studies may be selected by many methods, other than the retrospective process, for inclusion in such a meta-analytic study. For example, by time coverage (uniformly distributed through time), by citation analysis or by model performance to name a few. Formal concept analysis may also be used as another method of visualising natural clusters within the input data in addition to Sammon mapping, PCA analysis and SOM mapping. More meta-analytic studies of this nature would allow a deeper understanding of the produced results allowing stronger recommendations to be made on the basis of such analysis.

6.3 Publications

- Pre-selection of independent binary features: An application to diagnosing Scrapie in sheep. L.I. Kuncheva, C.J. Whitaker, P.D. Cockcroft and Z.S.J. Hoare. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, pages 325 - 332. 2004

- Selection of independent features using probabilities: An example from veterinary medicine. L.I. Kuncheva, Z.S.J Hoare and P.D. Cockcroft. *Journal of Modern Applied Statistical Methods*, Vol.4(2), pages 528 - 537, 2005

- Empirical bounds on error differences when using Naïve Bayes. Z. Hoare. In *Proceedings of the 3rd International Conference on Advances in Pattern Recognition*, pages 28 - 34, 2005

- Naïve Bayes classifier: True and estimated errors for 2-class 2-features case. Z. Hoare. In *Proceedings of the 3rd IEEE Conference on Intelligent Systems*, pages 566 - 570, 2006

- **Submitted.** Landscapes of Naïve Bayes Classifiers. Z. Hoare. *Under second review after a major revision for Pattern Analysis and Applications journal.*

Bibliography

[1] D. Aha. Lazy learning special issue editorial. *Artificial Intelligence Review*, 11:7 – 10, 1997.

[2] D. Aha and R. Bankert. A comparative evaluation of sequential feature selection algorithms. In *Proceedings 5th International Workshop on Artificial Intelligence and Statistics, Fort Landerdale*, 1995.

[3] E. Alhoniemi, J. Himberg, J. Parhankangas, and J. Vesanto. SOM toolbox for Matlab, 1997. http://www.cis.hut.fi/projects/somtoolbox/.

[4] H. Altincay. On Naïve Bayesian fusion of dependent classifiers. *Pattern Recognition Letters*, 26:2463 – 2473, 2005.

[5] O. Asparoukhov and S. Danchev. Discrimination and classification in the presence of binary variables. *Biocybernetics and Biomedical Engineering*, 17:25 – 39, 1997.

[6] O. Asparoukhov and W. Krzanowski. A comparison of discriminant procedures for binary variables. *Computational statistics and Data Analysis*, 38:139 – 160, 2001.

[7] O. Asparoukhov and A. Stam. Mathematical programming formulations for two-group classification with binary variables. *Annals of Operations Research*, 74:89 – 112, 1997.

[8] P. Bennett. Assessing the calibration of Naïve Bayes posterior estimates. Technical Report CMU-CS-00-155, Carnegie Mellon University, Pittsburgh, 2000.

[9] F. Berzal, JC. Cubero, F. Cuenca, and M.J. Martin-Bautista. On the quest for easy to understand splitting rules. *Data and Knowledge Engineering*, 44:31 – 48, 2003.

[10] J. Bins and B. Draper. Feature selection from huge feature sets. In *Proceedings of the 8th IEEE Conference on Computer Vision*, pages 159–165, 2001.

[11] C. L. Blake and C. J. Merz. UCI repository of machine learning databases, 1998. http://www.ics.uci.edu/ mlearn/MLRepository.html.

[12] A. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97:245 – 271, 1997.

[13] J.V. Bradley. *Distribution-Free Statistical Tests*. Prentice Hall, Englewood Cliffs, New Jersey, 1968.

[14] H. Brenner and O. Gefeller. Variation of sensitivity, specificity, likelihood ratios and predictive values with disease prevalence. *Statistics in Medicine*, 16(2):981 – 991, 1997.

[15] M. Bressan and J. Vitria. Improving Naïve Bayes classification using class-conditional ICA. *Lecture Notes in Artificial Intelligence*, 2527:1 – 10, 2002.

[16] M. Bressan and J. Vitria. On selection and classification of independent features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25 (10):1 – 6, 2003.

[17] D. Brown, C. Pittard, and H. Park. Classification trees with optimal multivariate decision nodes. *Pattern Recognition Letters*, 17:699 – 703, 1996.

[18] N. Chaikla and Y. Qi. Genetic algorithms in feature selection. In *Proceedings of the Systems, Man and Cybernetics IEEE SMC Conference*, volume 5, pages 538 – 540, 1999.

[19] X. Chen. An improved branch and bound algorithm for feature selection. *Pattern Recognition Letters*, 24:1925 – 1933, 2003.

[20] C. Christodoulou and C. Pattichis. Medical diagnostic systems using ensembles of neural SOFM classifiers. In *Proceedings 6th International Conference on Electronics, Circuits and Systems*, volume 1, pages 121 – 124, 1999.

[21] P.D. Cockcroft. Pattern matching models for differential diagnosis of BSE. *Veterinary Record*, 144:607 – 610, 1999.

[22] P.D. Cockcroft. A survey of pattern recognition methods in veterinary diagnosis. *J.V.M.E.*, 25(2):21 – 23, 1999.

[23] P.D. Cockcroft. Clinical sign profile likelihood ratios for BSE suspects. *Veterinary Science*, 68:285 – 290, 2000.

[24] H. Cooper and L. V. Hedges, editors. *The Handbook of Research Synthesis*. Russell Sage Foundation, New York, 1994.

[25] T. M. Cover. The best two independent measurements are not the two best. *IEEE Transactions on Systems, Man and Cybernetics.*, SMC-4:116 – 117, 1974.

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

[27] DEFRA. General questions and answers on the European Union export ban affecting beef. http://www.defra.gov.uk/animalh/bse/general/qa/section8.html, 2006.

[28] DEFRA and VLA. Transmissible Spongiform Encephalopathies (TSE) EU community reference library. www.defra.gov.uk/corporate/vla/science/science-tse.htm, 2005.

[29] A. Denton and W. Perrizo. A kernel-based semi-Naïve Bayesian classifier using P-Trees. In *Proceedings of the SIAM International Conference on Data Mining*, 2004.

[30] L. Diao, K. Hu, Y. Lu, and C. Shi. A method to boost Naïve Bayesian classifiers. In *Lecture Notes in Computer Science, Proceedings of the 6th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, pages 115 – 122, 2002.

[31] J. Doak. An evaluation of feature selection methods and their application to computer security. Technical report, Department of Computer Science, University of California, 1992. CSE-92-18.

[32] P. Domingos and M. Pazzani. Beyond independence: Conditions for the optimality of the simple Bayesian classifier. In *Proceedings of the 13th International Conference on Machine Learning*, 1996.

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

[34] M. Dong and R. Kothari. Feature subset selection using a new definition of classifiability. *Pattern Recognition Letters*, 24:1215 – 1225, 2003.

[35] J. Dougherty, R. Kohavi, and M. Sahami. Supervised and unsupervised discretization of continuous features. In *Proceedings of the 12th International Conference on Machine Learning*, 1995.

[36] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification and Scene Analysis, Second Edition*. Wiley-Interscience, New York, 2001.

[37] R.P.W. Duin, C.E. Van Haersma Buma, and L. Roosma. On the evaluation of independent binary features. *IEEE Transactions on Information Theory*, 24(2):248 – 249, 1978.

[38] R.P.W Duin and Delft Pattern Recognition Group. PRTools toolbox for Matlab, 1993. http://www.prtools.org.

[39] J. D. Elashoff, R. M. Elashoff, and G. E. Goldman. On the choice of variables in classification problems with dichotomous variables. *Biometrika*, 54:668 – 670, 1967.

[40] C. Elkan. Boosting and Naïve Bayesian learning. In *International Conference on Knowledge Discovery in Databases*, 1997.

[41] C. Elkan. Naïve Bayesian learning. Technical Report CS97-557, Department of Computer Science, Harvard University, 1997.

[42] F. Esposito, D. Malerba, and G. Semeraro. A comparative analysis of methods for pruning decision trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):476 – 491, 1997.

[43] F. Esposito, D. Malerba, G. Semeraro, and V. Tamma. The effects of pruning methods on the predictive accuracy of induced decision trees. *Applied Stochastic Models In Business And Industry*, 15:277 – 299, 1999.

[44] F. Ferri, P. Pudil, M. Hatef, and J. Kittler. Comparative study of techniques for large scale feature selection. In *Proceedings Pattern Recognition in Practice IV*, 1994.

[45] G. Forman and I. Cohen. Learning from little: comparison of classifiers given little training. In *Knowledge Discovery in Databases: PKDD 2004 Proceedings Lecture Notes in Artificial Intelligence*, volume 3202, pages 161 – 172, 2004.

[46] Y. Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2):256 – 285, 1995.

[47] Y. Freund and R.E. Schapire. A decision theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119 – 139, 1997.

[48] N. Friedman, D. Geiger, and M. Goldschmidt. Bayesian network classifiers. *Machine Learning*, 29(2):131 – 163, 1997.

[49] J. Gama. Iterative Bayes. *Theoretical Computer Science*, 292:417 – 430, 2003.

[50] P.L. Geenen, L.C. van der Gaag, and W.L.A. Loeffen. Building Naïve Bayesian classifiers from literature: a case study in classical swine fever. In *Proceedings of the 16th Belgium-Netherlands Conference on Artificial Intelligence*, pages 227 – 234, 2004.

[51] J. Ghosh. Multiclassifier systems: Back to the future. *Multi-classifier Systems. Lecture Note in Computer Systems*, 2364:1–15, 2002.

[52] G. Giacinto and F. Roli. An approach to the design of multiple classifier systems. *Pattern Recognition Letters*, 22:25–33, 2001.

[53] S. Gu̇nter and H. Bunke. Creation of classifier ensembles for handwritten word recognition using feature selection algorithms. In *Proceedings of the 8th International Workshop, Frontiers in Handwriting Recognition*, pages 183 – 188, 2002.

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

[55] G. Holmes, B. Pfahringer, R. Kirkby, E. Frank, and M. Hall. Multiclass alternating decision trees. *Lecture Notes In Artificial Intelligence*, 2430:161–172, 2002.

[56] R.C. Holte. Very simple classification rules perform well on most data sets. *Machine Learning*, 11:63 – 90, 1993.

[57] S. Hong, J. Hosking, and R. Natarajan. Ensemble modelling through multiplicative adjustment of class probability. In *IEEE International Conference on Data Mining*, pages 621 – 624, 2002.

[58] C. Hsu, H. Huang, and T. Wong. Why discretization works for Naïve-Bayesian classifiers. In *Proceedings of the 17th International Conference on Machine Learning*, pages 399 – 406, 2000.

[59] H. Huang and C. Hsu. Bayesian classification for data from the same unknown class. *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics*, 32:137 – 145, 2002.

[60] J. E. Hunter and F. L. Schmidt. *Methods of meta-analysis.* Sage publications, Newbury Park, London, 1990.

[61] L. Hyafil and R.L. Rivest. Constructing optimal binary decision trees is NP-complete. *Information Processing Letters*, 5(1):15 – 17, 1976.

[62] A. Jain and D. Zongker. Feature selection: evaluation, application and small sample performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19 (2):153 – 158, 1997.

[63] A. K. Jain, A. Topchy, M. H. C. Law, and J. M. Buhmann. Landscape of clustering algorithms. In *Proceedings of the International Conference on Pattern Recognition, ICPR*, pages 260 – 263, 2004.

[64] A. Jamain and D. J. Hand. The Naïve Bayes mystery a classification detective story. *Pattern Recognition Letters*, 2005.

[65] H. Ji and S. Bang. Feature selection for multi-class classification using pairwise class discriminatory measure and covering concept. *Electronics Letters*, 36 (6):524 – 525, 2000.

[66] W. Jin, R. Shi, and T. Chua. A semi-naïve Bayesian method incorporating clustering with pair-wise constraints for auto image annotation. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 336 – 339, 2004.

[67] E. Keogh and M. Pazzani. Learning augmented Bayesian classifiers. A comparison of distribution-based and classification-based approaches. In *Proceedings of the International Workshop on Artificial Intelligence and Statistics*, pages 225 – 230, 1999.

[68] A. Kleiner and B. Sharp. A new algorithm for learning Bayesian classifiers. In *Proceedings of the 3rd IASTED International conference on Artificial Intelligence and Soft Computing*, pages 191 –197, 2000.

[69] R. Kohavi. Scaling up the accuracy of Naïve-Bayes classifiers: a decision-tree hybrid. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, pages 202 – 207, 1996.

[70] R. Kohavi and M. Sahami. Error-based and entropy-based discretization of continuous features. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, pages 114 – 119, 1996.

[71] T. Kohonen. *Self-Organization and Associative Memory*. Springer-Verlag, Berlin, 1989.

[72] I. Kononenko. Semi-Naïve Bayesian classifier. In *Proceedings of the 6th European Working Session on Learning*, pages 206 – 219, 1991.

[73] I. Kononenko. Inductive and Bayesian learning in medical diagnosis. *Applied Artificial Intelligence*, 7:317 – 337, 1993.

[74] I. Kononenko. Machine learning for medical diagnosis: History, state of the art and perspective. *Artificial Intelligence in Medicine*, 23(1):89 – 109, 2001.

[75] M. Kudo and J. Sklansky. Comparison of algorithms that select features for pattern classifiers. *Pattern Recognition*, 33(1):25 – 41, 2000.

[76] L. I. Kuncheva. Genetic algorithm for feature selection for parallel classifiers. *Information Processing Letters*, 46(4):163 – 168, 1993.

[77] L. I. Kuncheva. *Combining Pattern Classifiers. Methods and algorithms*. John Wiley and Sons, Hoboken, New Jersey, 2004.

[78] L. I. Kuncheva. On the optimality of Naïve Bayes with dependent binary features. *Pattern Recognition Letters*, 27:830 – 837, 2006.

[79] L.I. Kuncheva, C.J. Whitaker, C.A. Shipp, and R.P.W. Duin. Is independence good for combining classifiers? In *Proceedings of 15th International Conference on Pattern Recognition*, volume 2, pages 2168 – 2171, 2000.

[80] L. Kurgan and K. J. Cios. Ensemble of classifiers to improve accuracy of the CLIP4 machine learning algorithm. In *Proceedings of SPIE International Symposium on Sensor Fusion: Architectures, Algorithms and Applications*, 2002.

[81] P. Langley, W. Iba, and K. Thompson. An analysis of Bayesian classifiers. In *Proceedings of the 10th National Conference on Artificial Intelligence*, pages 399 – 406, 1992.

[82] P. Langley and S. Sage. Induction of selective Bayesian classifiers. In *Proceedings of the 10th Conference on UAI*, pages 399 – 406, 1994.

[83] P. Langley and S. Sage. Tractable average case analysis of Naïve Bayesian classifiers. In *Proceedings of the 16th International Conference on Machine Learning*, 1999.

[84] P. Lanzi and P. Milano. Fast feature selection with genetic algorithms: A filter approach. In *IEEE International Conference on Evolutionary Computation*, pages 537 – 540, 1997.

[85] G. Lashkia and L. Anthony. Relevant, irredundant feature selection and noisy example elimination. *IEEE Transaction son Systems, Man and Cybernetics - Part B: Cybernetics*, 34:888 – 897, 2004.

[86] J. Lee and I. Oh. Binary classification trees for multi-class classification problems. In *Proceedings of the 7th International Conference on Document Analysis and Recognition (ICDAR 2003)*, page 770, 2003.

[87] D. Lewis. Naïve Bayes (at forty): The independence assumption in information retrival. In *Proceedings of the 10th European Conference on Machine Learning*, pages 4 – 15, 1998.

[88] T. Lim, W. Loh, and Y. Shih. An empirical comparison of decision trees and other classifier methods. Technical Report 979, Department of Statistics, University of Wisconsin, Madison, 1997.

[89] R.D. Linnabary, R.F. Hall, R.B. Wilson, and A.M. Knowles. Scrapie in sheep. *The Compendium*, 13(3):511 – 514, 1991.

[90] M. Lipsey and D. B. Wilson. *Practical Meta-analysis (Applied Social Research Methods)*. Sage Publications, London, 2001.

[91] J.K.N. Liu, B.N.L. Li, and T.S. Dillion. An improved Naïve Bayesian classifier technique coupled with a novel input solution method. *IEEE Transactions on Systems, Man and Cybernetics Part C Applications and reviews*, 31(2):249 – 256, 2001.

[92] W. Liu and A. White. The importance of attribute selection measures in decision tree induction. *Machine Learning*, 15:25 – 41, 1994.

[93] S. Ma and H. Shi. Tree augmented Naïve Bayes ensembles. In *Proceedings of the 3rd International Conference on Machine Learning and Cybernetics*, pages 1497 – 1502, 2004.

[94] B. Manly. *Multivariate Statistical Methods. A Primer*. Chapman and Hall, London, 1986.

[95] A. Marshall, D. Bell, and R. Sterritt. Handling uncertainty in a medical study of dietary intake during pregnancy. *SOFTWARE 2002: Computing in an imperfect world. Lecture Notes in Computer Science*, 2311:206–216, 2002.

[96] F. Masulli and G. Valentini. Comparing decomposition methods for classification. In *Proceedings of the 4th International Conference on Knowledge Based Engineering Systems and Allied Technologies*, 2000.

[97] G. Matinez-Munoz and A. Suarez. Using all data to generate decision tree ensembles. *IEEE Transactions on Systems, Man and Cybernetics - Part C: Applications and Reviews*, 34(4):393 – 397, 2004.

[98] A. McCallum and K. Nigam. A comparison of event models for Naïve Bayes text classification. In *AAAI-98 Workshop on Learning for Text categorization*, 1998.

[99] D. Meretakis and B. Wüthrich. Extending Naïve Bayes classifiers using long itemsets. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 165 – 174, 1999.

[100] R.A. Miller. Medical diagnostic decision support systems: Past, present and future. *Journal of the American Medical Information Association*, 1(1):8 – 27, 1994.

[101] R.G. Miller. *Simultaneous Statistical Inference*. McGraw-Hill, New York, 1966.

[102] J. Mingers. An empirical comparison of pruning methods for decision tree induction. *Machine Learning*, 4(2):227 – 243, 1989.

[103] Y. Murphey and H. Guo. Automatic feature selection - a hybrid statistical approach. In *Proceedings of the 15th International Conference on Pattern Recognition*, pages 382 – 385, 2000.

[104] D. Nikovski. Constructing Bayesian networks for medical diagnosis from incomplete and partially correct statistics. *IEEE Transactions on Knowledge and Data Engineering*, 12(4):509–516, 2000.

[105] A. Nürnberger, C. Borgelt, and A. Klose. Improving Naïve Bayes classifiers using neuro-fuzzy learning. In *Proceedings of the 6th International Conference on Neural Image Processing*, pages 154 – 159, 1999.

[106] S.B. Oh. On the relationship between majority vote accuracy and dependency in multiple classifier systems. *Pattern Recognition Letters*, 24(1-3):359–363, 2003.

[107] D. Opitz. Feature selection for ensembles. In *Proceedings of the 16th International Conference on Artificial Intelligence, AAAI*, pages 379 – 384, 1999.

[108] P. and A.D. Lovie, editors. *New Developments in Statistics for Psychology and the Social Sciences*, volume 2, chapter 4. John Wiley and Sons, BPS books Leicester and Routledge Ltd London, 1991.

[109] Y. Park and J. Sklansky. Automated design of linear tree classifiers. *Pattern Recognition*, 23 (12):1393 – 1412, 1990.

[110] M.J. Pazzani. Searching for dependencies in Bayesian classifiers. In *Learning from Data: AI and Statistics V. Springer-Verlag*, 1996.

[111] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Francisco, 1988.

[112] P. J. Phillips and E. M. Newton. Meta-analysis of face recognition algorithms. In *Proceedings of the 5th IEEE International Conference on Automatic Face and Gesture Recognition, FGR'02*, pages 235 – 241, 2002.

[113] K. Przytula and D. Thompson. Construction of Bayesian networks for diagnostics. In *Aerospace Conference Proceedings*, volume 5, pages 193 – 200, 2000.

[114] P. Pudil and J. Novovicova. Novel methods for subset selection with respect to problem knowledge. *IEEE Intelligent Systems*, pages 66 – 74, 1998.

[115] O.M. Radostits, C.C. Gay, D.C. Blood, and K.W. Hinchcliff. *Veterinary Medicine. A Textbook of the Diseases of Cattle, Sheep, Pigs, Goats and Horses.* Harcourt Publishers Ltd, London, 2000.

[116] C. Ratanamahatana and D. Gunopulos. Scaling up the Naïve Bayesian classifier: Using decision trees for feature selection. In *Proceedings of the Workshop on Data Cleaning and Pre-processing, ICDM'02*, 2002.

[117] G. Ridgeway, D. Madigan, T. Richardson, and J. O'Kane. Interpretable boosted Naïve Bayes classification. In *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining*, pages 101 – 104, 1998.

[118] I. Rish. An empirical study of the Naïve Bayes classifier. In *Proceedings of the International Joint Conference on Artificial Intelligence, Workshop on "Empirical Methods in AI"*, 2001.

[119] I. Rish, J. Hellerstein, and J. Thathachar. An analysis of data characteristics that affect Naïve Bayes performance. Technical Report RC21993, IBM TJ Watson Research Center, 2001.

[120] V. Robles, P. Larranaga, E. Menasalvas, M. S. Perez, and V. Herves. Improvement of Naïve Bayes collaborative filtering using interval estimation. In *Proceedings of the IEEE WIC International Conference on Web Intelligence*, pages 168 – 174, 2003.

[121] B. Rosell and L. Hellerstein. Naïve Bayes with higher order attributes. In *Lecture Notes in Computer Science, Proceedings of the 17th Conference of the Canadian Society for Computational Studies of Intelligence*, pages 105 – 119, 2004.

[122] S. Safavian and D. Landgrebe. A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man and Cybernetics*, 21(3):660 – 674, 1991.

[123] M. Sahami. Learning limited dependence Bayesian classifiers. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, pages 334 – 338, 1996.

[124] S. M. Salvatierra. Using unlabelled data to improve classification in the Naïve Bayes approach: Application to web searches. Technical report, Facultad de Ciencias Economicas y Empresariales, Universidad de Navarra, 2002.

[125] S. L. Salzberg. On comparing classifiers: Pitfalls to avoid and a recommended approach. *Data Mining and Knowledge Discovery*, 1:317 – 328, 1997.

[126] J. W. Sammon. A non-linear mapping for data structure analysis. *IEEE Transactions on Computers*, 18:401 – 405, 1969.

[127] R.E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

[128] R.E. Schapire. A brief introduction to boosting. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, pages 1401 – 1406, 1999.

[129] S. Schiffman, M. L. Reynolds, and F. W. Young. *Introduction to Multidimensional Scaling*. Academic Press, Inc., London, 1981.

[130] H. Schulerud and F. Albregtsen. Many are called but few are chosen. Feature selection and error estimation in high dimensional spaces. *Computer Methods and Programs in Biomedicine*, 73:91 – 99, 2004.

[131] P.R. Scott and C.J. Henshaw. Diagnosis of Scrapie - increasing the accuracy of the provisional ante-mortem examination. *State Veterinary Journal*, 4(2):4 – 6, 1994.

[132] I. Sethi and J. Yoo. Design of mulitcatergory multifeature split decision trees using perceptron learning. *Pattern Recognition*, 27:939 – 947, 1994.

[133] E.H. Shortliffe, B.G. Buchanan, and E.A. Feigenbaum. Knowledge engineering for medical decision making: A review of computer-based clinical decision aids. *Proceedings of the IEEE*, 67(9):1027 – 1224, 1979.

[134] W. Siedlecki and J. Sklansky. A note on genetic algorithms for large-sale feature selection. *Pattern Recognition Letters*, 10:335 – 347, 1989.

[135] C. Sima, S. Attoor, U. Brag-Neto, J. Lowey, E. Suh, and E.R. Dougherty. Impact of error estimation on feature selection. *Pattern Recognition*, 38:2472 – 2482, 2005.

[136] R.D. Smith and M. Williams. Applications of informatics in veterinary medicine. *Bulletin of the Medical Library Association*, 88(1):49 – 51, 2000.

[137] S. Y. Sohn. Meta-analysis of classification algorithms for pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1137 – 1144, 1999.

[138] J. M. Sotoca, J. S. Sanchez, and F. Pla. Attribute relevance in multiclass datasets using the Naïve Bayes rule. In *Proceedings of the 17th International Conference on Pattern Recognition*, volume 3, pages 426 – 429, 2004.

[139] H. P. Störr. A compact fuzzy extension of the Naïve Bayesian classification algorithm. In *Proceedings of InTech VJFuzzy'2002*, pages 172 – 177, 2002.

[140] R. B. Talbot. Veterinary medical informatics. *Journal of American Veterinary Medicine*, 199:2 – 7, 1991.

[141] D. Tax and R. Duin. Using two-class classifiers for multiclass classification. In *Proceedings of the 16th International Conference on Pattern Recognition*, volume 2, pages 124–127, 2002.

[142] K. Ting and Z. Zheng. Improving the performance of boosting for Naïve Bayesian classification. In *Proceedings of the 3rd Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 296 – 305, 1999.

[143] D. M. Titterington, G. D. Murray, L. S. Murray, D. J. Spiegelhalter, A. M. Skene, J. D. F. Habbema, and G. J. Gelpke. Comparison of discrimination techniques applied to a complex data set of head injured patients. *Journal of the Royal Statistical Society. Series A (General)*, 144:145 – 175, 1981.

[144] G. Toussaint. Note on the optimal selection of independent binary-valued features for pattern recognition. *IEEE Transactions on Information Theory*, 618, 1971.

[145] A. Tsymbal, P. Cunnigham, M. Pechenizkiy, and S. Puuronen. Search strategies for ensemble feature selection in medical diagnostics. Technical report, Trinity College Dublin, Ireland, 2003.

[146] A. Tsymbal and S. Puuronen. Ensemble feature selection with the simple Bayesian classification in medical diagnostics. In *Proceedings of the 15th IEEE Symposium on Computer Based Medical Systems*, pages 225 – 230, 2002.

[147] R. Vilalta and I. Rish. A decomposition of classes via clustering to explain and improve Naïve Bayes. In *Proceedings of the 14th European Conference on Machine Learning*, 2003.

[148] L. Wang, S. Yuan, and H. Li. Boosting Naïve Bayes by active learning. In *Proceedings of the 3rd International Conference on Machine Learning and Cybernetics*, pages 1383 – 1386, 2004.

[149] Z. Wang and G. I. Webb. Comparison of lazy Bayesian rule and tree-augmented Bayesian learning. In *Proceedings of IEEE International Conference on Data Mining*, pages 490 – 497, 2002.

[150] Andrew Webb. *Statistical Pattern Recognition*. Newnes, Oxford, 1999.

[151] G. Webb, J. Boughton, and Z. Wang. Not so Naïve Bayes: aggregating one dependence estimators. *Machine Learning*, 58(1):5 – 24, 2005.

[152] G. Webb and M.J. Pazzani. Adjusted probability Naïve Bayesian induction. In *Proceedings of the 11th Australian conference on Artificial Intelligence*, pages 285 – 295, 1998.

[153] T. Windeatt. Vote counting measures for ensemble classifiers. *Pattern Recognition*, 36(12):2743–2756, 2003.

[154] T. Windeatt and G. Ardeshir. Tree pruning methods for output coded ensembles. In *Proceedings of the 16th International Conference on Pattern Recognition*, volume 2, pages 92–95, 2002.

[155] F. M. Wolf. *Meta-Analysis: Quantitative methods for research synthesis*. Sage University Paper no. 59. Series on Quantitative Applications in the Social Sciences, London Sage publications, 1986.

[156] Z. Xie, W. Hsu, Z. Liu, and M. Lee. SNNB: A selective neighbourhood based Naïve Bayes for lazy learning. In *Proceedings of Advances in Knowledge Discovery and Data Mining. PAKDD*, pages 104 – 114, 2002.

[157] Y. Yang and G. Webb. On why discretisation works for Naïve Bayes classifiers. In *Australian conference on Artificial Intelligence*, pages 440 – 452, 2003.

[158] H. Zhang. The optimality of Naïve Bayes. In *Proceedings of the 17th International FLAIRS conference*, Miami Beach Florida, 17 - 19 May, 2004.

[159] H. Zhang, L. Jiang, and J. Su. Augmenting Naïve Bayes for ranking. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 1025 – 1032, 2005.

[160] H. Zhang and C. X. Ling. A fundamental issue of Naïve Bayes. In *Proceedings of the Canadian Conference on Artificial Intelligence*, pages 591 – 595, 2003.

[161] H. Zhang, C. X. Ling, and Z. Zhao. The learnability of Naïve Bayes. In *Proceedings of the 13th Biennial Conference of the Canadian Society for Computational Studies of Intelligence*, pages 432 – 440, 2000.

[162] H. Zhang and J. Su. Conditional independence trees. *Lecture Notes in Computer Science*, 3201:513 – 524, 2004.

[163] H. Zhang and J. Su. Naïve Bayesian classifiers for ranking. *Lecture Notes in Computer Science*, 3201:501 – 512, 2004.

[164] Z. Zheng. Naïve Bayesian classifier committees. In *Proceedings of the European Conference on Machine Learning*, pages 196 – 207, 1998.

[165] Z. Zheng and G. I. Webb. Lazy learning of Bayesian rules. *Machine Learning*, 41(1):53 – 84, 2000.