DATA MEASURES THAT CHARACTERISE CLASSIFICATION PROBLEMS

CHRISTIAAN MAARTEN VAN DER WALT

DATA MEASURES THAT CHARACTERISE CLASSIFICATION PROBLEMS

by

CHRISTIAAN MAARTEN VAN DER WALT

Submitted in partial fulfilment of the requirements for the degree
Master of Engineering (Electronic Engineering)

in the

Faculty of Engineering, the Built Environment and Information Technology

UNIVERSITY OF PRETORIA

Promoter: Professor E. Barnard

February 2008

SUMMARY

DATA MEASURES THAT CHARACTERISE CLASSIFICATION PROBLEMS

BY

CHRISTIAAN MAARTEN VAN DER WALT

PROMOTER: PROFESSOR E. BARNARD

DEPARTMENT OF ELECTRICAL, ELECTRONIC AND COMPUTER ENGINEERING

MASTER OF ENGINEERING (ELECTRONIC)

We have a wide-range of classifiers today that are employed in numerous applications, from credit scoring to speech-processing, with great technical and commercial success. No classifier, however, exists that will outperform all other classifiers on all classification tasks, and the process of classifier selection is still mainly one of trial and error.

The optimal classifier for a classification task is determined by the characteristics of the data set employed; understanding the relationship between data characteristics and the performance of classifiers is therefore crucial to the process of classifier selection. Empirical and theoretical approaches have been employed in the literature to define this relationship. None of these approaches have, however, been very successful in accurately predicting or explaining classifier performance on real-world data.

We use theoretical properties of classifiers to identify data characteristics that influence classifier performance; these data properties guide us in the development of measures that describe the relationship between data characteristics and classifier performance. We employ these data measures on real-world and artificial data to construct a meta-classification system.

The purpose of this meta-classifier is two-fold: (1) to predict the classification performance of real-world classification tasks, and (2) to explain these predictions in order to gain insight into the properties of real-world data.

We show that these data measures can be employed successfully to predict the classification performance of real-world data sets; these predictions are accurate in some instances but there is still unpredictable behaviour in other instances.

We illustrate that these data measures can give valuable insight into the properties and data structures of real-world data; these insights are extremely valuable for high-dimensional classification problems.

Keywords: artificial data, classification, classification prediction, classifier selection, data characteristics, data measures, data analysis, meta-classification, pattern recognition, supervised learning.

OPSOMMING

DATAMETINGS WAT KLASSIFISERINGSPROBLEME KARAKTERISEER

DEUR

CHRISTIAAN MAARTEN VAN DER WALT

STUDIELEIER: PROFESSOR E. BARNARD

DEPARTEMENT VAN ELEKTRIESE, ELEKTRONIESE EN REKENAAR-INGENIEURSWESE

MEESTER IN INGENIEURSWESE (ELEKTRONIES)

'n Groot verskeidenheid klassifiseerders word vandag geïmplementeer in 'n wye reeks toepassings, van kredietwaardigheid voorspellings tot spraakverwerking, met groot tegniese en kommersiële sukses. Daar bestaan egter geen klassifiseerder wat die beste presteer vir alle klassifiseringsprobleme nie en die proses van klassifiseerderseleksie is hoofsaaklik 'n iteratiewe empiriese proses.

Die optimale klassifiseerder vir 'n klassifiseringsprobleem word bepaal deur die eienskappe van die probleem. 'n Goeie begrip vir die verwantskap tussen dataeienskappe en klassifiseerdergedrag word dus vereis om die optimale klassifiseerder te selekteer. Verskeie empiriese en teoretiese benaderings is al gevolg in die literatuur om hierdie verwantskap te beskryf. Geen van hierdie benaderings het egter daarin geslaag om klassifiseringfouttempo's van regtewêreld data akkuraat te voorspel of te verduidelik nie.

Ons maak gebruik van die teoretiese eienskappe van klassifiseerders om dataeienskappe te identifiseer wat klassifiseerder fouttempo beïnvloed; hierdie dataeinskappe word gebruik om ons te lei in die ontwikkeling van datametings wat die verwantskap tussen dataeienskappe en

klassifiseerderfouttempo beskryf. Ons implementeer hierdie metings op regtewêreld sowel as kunsmatige data om sodoende 'n meta-klassifiseerder te skep. Die meta-klassifiseerder dien twee doeleindes: (1) dit voorspel die klassifiseerder fouttempo's van regtewêreld datastelle, en (2) verduidelik klassifisering voorspellings om sodoende meer insig te verkry in die data eienskappe van regtewêreld data.

Ons illustreer dat datametings suksesvol geïmplementeer kan word om klassifiseerderfouttempo's te voorspel vir regtewêreld data. Hierdie voorspellings is akkuraat in sekere gevalle, maar daar is steeds onvoorspelbare gedrag in ander gevalle.

Ons demonstreer hoe hierdie dataeienskappe gebruik kan word om insig te verkry in die eienskappe en strukture in regtewêreld data. Hierdie insigte is uiters waardevol vir hoëdimensionele klassifiseringsprobleme.

Sleutelwoorde: data-analisering, data-eienskappe, data-metings, klassifiseerderseleksie, klassifisering, kunsmatige data, leer met toesighouding, meta-klassifisering, patroonherkenning, voorspelling van klassifiseringfouttempo's.

Trust in the Lord with all your heart and lean not on your own understanding; in all your ways acknowledge Him, and He will make your paths straight

Proverbs 3:5 NIV

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

1NN	Nearest-Neighbour
DT	Decision Tree
Gauss	Gaussian
GMM	Gaussian Mixture Model
GMMd	Gaussian Mixture Model (diagonal covariance)
GMMf	Gaussian Mixture Model (full covariance)
kNN	k-Nearest-Neighbour
ML	Machine Learning
MLP	Multilayer Perceptron
MST	Minimum Spanning Tree
MVN	Multivariate Normality
NB	Naïve Bayes
NFL	No-Free-Lunch
OTS	Off-Training-Set
PAC	Probability Approximately Correct
PCA	Principal Component Analysis
PR	Pattern Recognition
SD	Standard Deviation
SVM	Support Vector Machine
UCI	University of California, Irvine
VC	Vapnik Chervonenkis
WPS	Wrapped Progressive Sampling

CHAPTER ONE

INTRODUCTION

1.1 INTRODUCTION

The quest to optimise the performance of trainable classifiers has a long and varied history. Soon after the design of the earliest parametric and linear classifiers, researchers found refinements (such as polynomial classifiers and the nearest-neighbour (1NN) rule) that produced more accurate classification on comparable data sets. Hence, the quest for "the most accurate" classifier was initiated, and several generations of candidates for that title have been proposed: kernel functions, neural networks, support vector machines, etc.

In some ways, this activity has been extremely productive – we today have a wide range of classifiers that are employed in numerous applications [1], from credit scoring to speech processing, with great technical and commercial success. However, from another perspective, this entire enterprise can be considered a dismal failure: we still do not have a single classifier that can reliably outperform all others on a given data set [2, 3, 4] and the process of classifier selection is still largely one of trial and error.

This apparent contradiction would not be surprising in the context of purely parametric classifiers, since the accuracy of a particular parametric classifier on a given data set will

clearly depend on the relationship between the classifier and the data. The concept of a single best parametric classifier is clearly not useful, and a trial-and-error process will generally be required to find the parametric form that best describes a given data set (although statistical tests may be employed to guide that search). In the realm of non-parametric classifiers, however, there is less awareness of the need to harmonise the characteristics of data and classifiers.

Several empirical studies have shown that the choice of optimal classifier does in fact depend on the data set employed [5, 2], and some guidelines on classifier selection have been proposed [4]. These guidelines do not, however, provide much insight into the specific characteristics of the data that will determine the preference of classifier; several theoretical approaches have also been employed to predict the performance of classifiers in an *a priori* fashion [6, 7, 8]; we will show in the next sections that these approaches fall short of a comprehensive solution to the task of classifier selection.

A significant amount of insight into the theoretical properties of classifiers and of data will be required to describe the relationship between data characteristics and classifier performance fully; we will search for such insight by (1) identifying data properties that influence classification performance and (2) measuring these properties from data.

1.2 OVERVIEW

The purpose of this study is to investigate the relationship between data characteristics and classifier performance; we will develop data measures to define this relationship and will use these data measures to develop a meta-classification system that will make classifier performance predictions. We will use the meta-classification system to construct a framework to analyse the properties of real-world data and explain classification predictions.

The outline of this thesis is as follows:

- We identify data properties that influence classifier performance in Chapter 2.
- We propose data measures to quantify these properties in Chapter 3.

- We validate the efficacy of these data measures in Chapter 4.
- We use these measures to construct a meta-classification system in Chapter 5 and we show how these measures can be used to explain the classification predictions of the meta-classifier.
- We conclude by describing some of the implications of our findings in Chapter 6.

1.3 BACKGROUND

Various strategies have been employed to describe the relationship between classifiers and the problems they try to solve; these approaches are summarised as follows:

- Empirical studies have been performed to compare the performance of classifiers on different real-world data sets [2, 5] and to predict the domain of competence of classifiers [4, 9]. A heuristic meta-learning search method has been proposed by [10] to find the optimal parameter settings of classifiers and to estimate the generalisation performance of these classifiers.
- Data measures to characterise the difficulty of classification problems were studied by [11]; their focus was on the geometrical complexity of the decision boundaries between classes.
- A theoretical framework was developed in [6, 7] to predict and compare the generalization performance of classifiers.
- Statistical learning theories, such as that of Vapnik and Chervonenkis (VC) [8], have been used to place bounds on the generalisation error rates of data sets.

We will discuss each of these approaches in some detail in this section; we will also note the limitations of these approaches.

1.3.1 EMPIRICAL STUDIES

Several comparative studies have been conducted to determine features in data that predict classification performance. Tax and Duin [5] considered a one-class classification problem;

19 classifiers and 101 real-world data sets were used. They defined two features to characterise data sets, namely the effective sample size and the class overlap. They found that the most significant variable that characterises a data set well is the effective sample size (the ratio between the number of observations and variables in a data set). Although this is a useful insight, it clearly is a limited view of the variability that may be present in data sets – by themselves, these two measures give only limited insight into the way that various classifiers will perform on a particular data set.

Brazdil *et al.* [4] performed a comparative study based on the results of the StatLog Project [2]. The StatLog project compared 22 classifiers on more than 20 different real-world data sets. The aim of [4] was to obtain a set of rules to predict classification performance of data sets. Statistical and information theoretic measures were used to extract features from data sets; these measures were used together with the classification results of the StatLog project to construct an expert system, named the Application Assistant, to predict the classification performance of various classifiers on a particular data set. The C4.5 decision tree algorithm [12] was used to construct rules from the given data. The classification results were considered one at a time by the C4.5 algorithm, until a final set of rules had been constructed. All the rules had a confidence measure to indicate their usefulness. The rules that were generated by the expert system were not very meaningful owing to a lack of training data – it is easy to find counterexamples to the conclusions reached in [4]. An example of such a rule is

$$\text{Discrim-Appl}_{\downarrow} 8 \text{ N} <= 1000 \; 0 \; 0.247$$

This rule states that the linear discriminative classifier will perform well with a confidence or information score of 0.247 if the number of samples in the data set is less or equal to 1000. All the rules with an information score of more than 0.2 are considered useful. This is clearly not a rule that will hold in general, since only the size of the data set is considered; several other relevant measures that are relevant to linear separability, including the dimensionality and number of classes in the data set, are ignored.

A meta-learning ranking algorithm based on the work of Brazdil and Soares [13, 14] has been developed and included into the Weka [15] machine learning package. This algorithm

offers advice on classifier selection; the accuracy, training and testing times are considered in the ranking of classifiers. The meta-classifier is trained with benchmark data sets; the classification error rates and times of these benchmark data sets are calculated and several data measures are performed to characterise these data sets. New data sets are characterised by calculating their data measures and performance rankings are predicted by finding the most similar benchmark data set. The data measures used to characterise data sets are statistical and information theoretic measures used in [2, 16].

Landmarking is a different type of approach used to define the domains of competency of classifiers (opposed to the traditional approach of calculating data measures and classifying or clustering these data measures). The performance of simple classifiers (landmarkers) are used to generalise the domain of competency to more complex classifiers. The selection criteria of landmarkers are computational complexity within reasonable bounds and biases that are reasonably different.

Pfahringer *et al.* [9] investigated and compared a landmarking meta-classification approach to a meta-classification approach where information theoretic data measures were used; the information theoretic measures of the Statlog project [2] were employed. Four landmarkers were used to construct a landmarking meta-classifier; all of these were decision tree classifiers with minimal node complexity. The classification performances of the 1NN, Naïve Bayes (NB), C5.0 with boosting, neural network, rules learning and decision tree classifiers were predicted using these landmarkers. 320 two-class artificial data sets were generated to construct the meta-classification data set; these artificial sets all had between 5 and 12 boolean attributes. The following meta-classifiers were employed to classify the meta-data: C5.0 trees, C5.0 rules trees, boosted C5.0 trees, RIPPER (a two-rule inducer), LTREE (an oblique decision tree inducer), linear discriminant, 1NN and NB classifiers. The predictions made by the landmarking and information theoretic meta-classifiers were compared.

It was not conclusively shown in [9] that one of these approaches is significantly better than the other; this suggests that additional data measures will be required to describe the relationship between data characteristics and classifier performance fully.

van den Bosch [10] proposed a wrapped progressive sampling (WPS) algorithm to find the optimal parameters settings for various classifiers and to estimate the generalisation performance of these classifiers; an implementation of this algorithm is available at [10].

WPS is a heuristic search method used to optimise parameters; progressive sampling is used to decrease the number of classifier setting combinations with increasing amounts of training data. Classifier wrapping is used to partition training data (generated by progressive sampling) into internal training and test sets; 10-fold cross-validation is used to estimate the training set generalisation performance for a specific set of parameters. These performance estimates on the various progressive training sets are used to determine the optimal parameter settings as well as the optimal number of training samples that must be used to obtain optimal generalisation performance.

Classification error rates of classifiers trained with default classifier settings were compared to classifiers trained with WPS. Ten real-world data sets obtained form the UCI Machine Learning repository [17] and five classifiers were used in this comparison. It was shown that two of the fifty classification results were significantly worse and seventeen of the fifty classification results were significantly better when employing the WPS procedure.

The estimates of the WPS generalisation performances of classifiers can be used to select the optimal classifier for a given classification task; these estimates fail, however, to provide information on these classification performances. This information is required to gain insight into data properties and how these properties influence classifier performance.

These empirical studies have shown that understanding the relationship between data characteristics and classifier performance is crucial; this relationship is not, however, fully described by any of these approaches.

1.3.2 CHARACTERISING THE COMPLEXITY OF CLASSIFICATION PROBLEMS

Ho and Basu [11] studied 12 data measures to characterise the complexity of classification problems; their focus was mainly on the geometrical complexity of classification problems. The measures under study were grouped into the following categories: (1) measures of overlap of individual features, (2) measures of separability of classes and (3) measures of geometry, topology and density manifolds.

An experiment was designed to compare the measurement values of 14 real-world data sets and 300 randomly labelled uniformly distributed artificial data sets. This experiment showed that real-world problems contain structures significantly different from random-labelled data; it was also shown that the measures of the random artificial data sets differed significantly for different feature dimensionalities – this observation was attributed to the increased sparsity of data in high-dimensional spaces.

A hundred additional Gaussian distributed artificial data sets were generated with varying degrees of separabiliy between the classes. A case study was performed in which the authors: (1) compared the measurement values of a real-world data set to the 100 Gaussian distributed artificial data sets and (2) compared the measurement values of a real-world data set to 14 other real-world data sets obtained from the UCI Machine Learning Repository [17].

Principal component analysis (PCA) dimension reduction was applied to all the measurement values of these data sets in order to investigate the relationships between data. Ho suggests that this reduced measurement feature space may be used to define the domains of competence of various classifiers.

This study gave some insight into the properties of data; the influence of these properties on the performance of classifiers was, however, not investigated.

1.3.3 NO-FREE-LUNCH THEOREMS

The no-free-lunch (NFL) theorems of Wolpert [6, 7] have caused some controversy in the fields of machine learning (ML) and pattern recognition (PR). Various papers have been written to discuss the applicability [7] or inapplicability [18] of the NFL theorems in the context of supervised learning.

Wolpert measures the generalisation performance of a classifier with an off-training-set (OTS) error, which is defined as the error rate of a classifier when the samples in the training and test sets don't overlap.

If the input-output relationship ($x - y$ target function) of a data set is defined as f and the function that is learned by a classier is defined as h, then the OTS error (C) on a training set (d) can be written as a non-Euclidean inner product between the true target function and the target function learned by the classifier [7]. The expected generalisation (OTS) error for a learning algorithm is expressed as

$$E(C|d) = \sum_{h,f} Er(h, f, d)P(h|d)P(f|d), \tag{1.1}$$

where *P(f|d)* is the true target function of the training data d and *P(h|d)* is the target function learned by the classifier on training data d.

This theorem shows that the performance, *E(C|d)*, of a learning algorithm, *P(h|d)*, is determined by how well it is aligned with the actual posterior, *P(f|d)*. The suitability of a learning algorithm for a specific learning task can thus be determined by this equation.

This NFL theorem does not explicitly define any relationship between the true target function of the data *P(f|d)* and the learned target function *P(h|d)*; the NFL theorems are thus still valid even if these two functions are independent. This explains why the NFL theorems have led to counter-intuitive conclusions such as: "unless one can establish *a priori*, before seeing any of the data d, that the f that generated the d is one of the ones for which one's favourite algorithm performs better than other algorithms, one has no assurances that that

learning algorithm performs better than the algorithm of purely random guessing." [7], p.4.

From our perspective, these counter-intuitive consequences of the NFL theorems are a result of their excessive neutrality with respect to the properties of data sets in pattern recognition: for real-world problems, there is always a definite relationship between *P(f|d)* and *P(h|d)*. We return to this matter in the final chapter of this dissertation.

1.3.4 BOUNDS ON GENERALIZATION PERFORMANCE

Computational learning theory is a theoretical approach used to place bounds on the generalisation errors of learning algorithms. Three sets of approaches comprise the main thrust of computational learning theory: statistical physics, the probability approximately correct (PAC) framework and the VC theory [7, 19]. We will use the VC approach as an example to illustrate how theoretical bounds are placed on the generalisation performance of classifiers.

Theoretical boundaries on the generalisation performance of classifiers have been derived by Vapnik [8]; these boundaries are a function of the VC dimension (machine capacity) of a classifier and its classification performance on the training set.

The bound on the generalisation error of a classifier (with parameters α) for a data set with l samples is given by

$$R(\alpha) \leq R_{emp}(\alpha) + \sqrt{\frac{h(\log(2l/h)+1) - \log(\eta - 4)}{l}}, \tag{1.2}$$

where R_{emp} is the classification error rate on the training set, h is the VC dimension of the employed classifier and $1 - n$ is the confidence in the generalisation bound *R(*α*)*.

If the VC dimension of a classifier is infinite (if it can shatter all points in a training set for any value of l) then the generalisation bound *R(*α*)* will become infinite; the use of this bound is thus not informative for classifiers with infinite VC dimensions. Even for finite VC dimension, these bounds have proven to be so weak as to be useless for most real-world

problems [19]. The other two approaches in computational learning theory are similarly impressive from a theoretical perspective, but limited in their practical applicability.

1.4 CONCLUSION

We have discussed the various strategies that have been employed to define the relationship between classifier performance and the problems they try to solve; the NFL theorems and the bound on generalisation performance of classifiers using VC dimensions are very limited in terms of real-world applications. Empirical studies have shown the importance of the relationship between data characteristics and classifier performance; they have, however, failed to describe this relationship in detail.

To address this shortcoming we will identify several data properties that influence classification performance in Chapter 2. We will then propose measures to measure these data properties in Chapter 3 and will evaluate the efficacy of these measures in Chapter 4. Finally, we use these measures to predict the classification performance of real-world data sets in Chapter 5.

CHAPTER TWO

CLASSIFICATION EXPERIMENTS

2.1 INTRODUCTION

In this chapter we perform various classification experiments to investigate the properties of data that influence classification performance. Previous empirical studies have shown that the choice of optimal classifier does in fact depend on the data set employed [2], and some guidelines on classifier selection have been proposed [4]. These guidelines do not, however, provide much insight into the specific characteristics of the data that will determine the preference of a classifier.

To address this shortcoming, we focus on pieces of conventional wisdom which are often repeated in review papers [1] and text books [20] and we investigate the effect of data set complexity on the classification performance of various classifiers. The first wisdom is that discriminative classifiers tend to be more accurate than model-based classifiers at classification tasks (see, e.g. [20], p.77); the second is that k-nearest-neighbour (kNN) classifiers are almost always close to optimal in accuracy, for an appropriate choice of k (e.g. [1], p.17). A common subsidiary to the latter belief is that the best value of k can only be determined empirically. Some other conventional wisdoms are that the support vector

machine (SVM) classifier has exceptionally good generalization performance on all types of classification tasks and that a multilayer perceptron (MLP) classifier with two layers of weights are capable of approximating any continuous functional mapping [21], which gives it superior classification performance on almost all types of classification tasks.

We focus our attention on the following topics:

- Do model-based classifiers substantially outperform discriminative classifiers under any circumstances?
- What attributes of classification data determine the optimal value of k in a kNN classifier?
- Are there specific circumstances that cause the kNN to perform considerably worse than other classifiers?
- Are there any scenarios under which the SVM and MLP classifiers will perform worse than other typical discriminative classifiers?
- What is the effect of data set complexity on the classification performance of classifiers?

We develop a methodology (summarized in Section 2.2) that uses artificial data sets to probe the interaction between classifiers and data set properties. In Section 2.4 we endeavour to answer the five questions posed using this methodology and in Section 2.5 we discuss the implications of our findings.

2.2 METHODS AND DATA

In order to experiment with the relationship between data and classifiers, we have generated several series of artificial data, and experimented with both model-based and discriminative classifiers. We use 10-fold cross-validation to evaluate and compare the performance of the classifiers on the different data sets.

2.2.1 ARTIFICIAL DATA GENERATION

2.2.1.1 MULTIVARIATE GAUSSIAN DATA

We will generate multivariate Gaussian data by generating independent univariate Gaussian features; we will then rotate and stretch the univariate data with a matrix **A**. The matrix $\mathbf{A}\mathbf{A}^T$ is equivalent to the covariance matrix of the resulting data. We generate n samples per distribution by using d single variable Gaussian distributions of the form

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right\}, \tag{2.1}$$

where μ is the mean and σ^2 is the variance of the distribution; this results in a d-by-n matrix **x.**

We combine the d single dimensional variables into a multivariate Gaussian distribution by using the transformation

$$\mathbf{Y} = \mathbf{A}\mathbf{x} + \mathbf{B}, \tag{2.2}$$

where **B** is the d-dimensional mean of the distribution repeated n times and $\mathbf{A}\mathbf{A}^T$ is the covariance, Σ, of the multivariate distribution. The resulting multivariable distribution may be written as

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{d}{2}}\,|\mathbf{\Sigma}|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\mu)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)\right\}, \tag{2.3}$$

where **x** is a d-component column vector, μ is a $d-$component mean vector, Σ is the d-by-d covariance matrix, (**x** - μ)^T is the transpose of (**x** - μ), Σ^{-1} is the inverse of Σ, and $|\Sigma|$ is the determinant of Σ.

Class-conditional probability density functions for each class in a data set are generated by a weighted mixture of multivariate Gaussian distributions of the form given in (2.3). Data sets are generated for three different experiments explained later in this section.

2.2.1.2 MULTIVARIATE UNIFORM DATA

To generate multivariate uniform data we will employ the same strategy as in Section 2.2.1.1. We will generate univariate uniform features between zero and one of the form

$$p_U(\mathbf{x}) = \begin{cases} \frac{1}{b-a} \; for \; a \leq x \leq b \\ 0 \; for \; x < a \; or \; x > b \end{cases} \tag{2.4}$$

where a is equal to zero and b is equal to unity.

We rotate and stretch these univariate features with an $\mathbf{A}$ matrix similar to the multivariate Gaussian case.

2.2.1.3 MULTIVARIATE GAUSSIAN MIXTURE DATA

We will generate data with a mixture of Gaussians by generating multivariate Gaussian data and then assigning data from different distributions to the same class. The resulting class-conditional probability density functions will be a mixture of Gaussians.

2.2.1.4 MULTIVARIATE CAUCHY DATA

To generate multivariate Cauchy data we generate univariate uncorrelated Cauchy distributed features of the form

$$p_C(\mathbf{x}) = x_0 + \gamma \tan(\pi p_U(\mathbf{x})), \tag{2.5}$$

where x_0 is the location parameter, γ is the scale parameter and $p_u(\boldsymbol{x})$ is the uniform univariate distribution with zero mean; we use a location parameter of zero and scale parameter of one.

We use d single variable Cauchy distributions of the form given in (2.5) to generate n samples per distribution. We introduce rotation and stretch into this data in a similar fashion as the multivariate Gaussian data.

2.2.1.5 CORRELATION OF FEATURES

To introduce correlation into the features of the artificial data we generate a rotation matrix $\mathbf{A}_R$ with the Gram Schmidt orthogonalisation procedure [22]. This procedure is used to ensure that the column vectors of the $\mathbf{A}_R$ matrix are orthogonal to one another and that the features are correlated. We introduce standard deviation into the data by multiplying the rotation matrix $\mathbf{A}_R$ with a stretch matrix $\mathbf{A}_S$. The Gram Schmidt procedure ensures that the eigenvalues of the resulting $\mathbf{A}$ matrix are similar to the diagonal component values of the stretch matrix $\mathbf{A}_S$. We thus ensure that the standard deviations of the correlated data are similar to the standard deviations of the uncorrelated data. We use diagonal $\mathbf{A}$ matrices to generate uncorrelated data; this results in diagonal covariance matrices $\mathbf{A}\mathbf{A}^T$.

2.2.1.6 STANDARD DEVIATION

In our experiments we will use two types of covariance matrices; we will use covariance matrices with: (1) feature variances that vary and (2) equal feature variances. Standard deviation (SD) is introduced into the uncorrelated data by multiplying the uncorrelated features with a diagonal stretch matrix ($\mathbf{A}_S$) that has values equal to the desired SD values. The diagonal components of this $\mathbf{A}$ matrix are equal in the case where we use similar SDs for all features. To create features with varying SDs we generate random diagonal components for the $\mathbf{A}$ matrix. These random values are between zero and the specified maximum SD of the features.

SD is introduced into the correlated data by multiplying the $\mathbf{A}_R$ matrix generated by the Gram Schmidt procedure with a diagonal stretch matrix $\mathbf{A}_S$. We use the same stretch matrix $\mathbf{A}_S$ for both correlated and uncorrelated data.

We should note that the SD of a Cauchy distribution is theoretically undefined; for the purposes of this study we will refer to the values of the stretch matrix $\mathbf{A}_S$ as the SDs of the Cauchy distributed data.

2.2.1.7 NOISE

Two forms of noise are relevant in classification problems: input noise affects the class-conditional density functions, and can be adjusted by changing **A** and **B** in (2.2). Output noise is simulated by changing the class labels of the observations in the original data set. In our experiments below, we sometimes need to quantify the extent of the noise. For input noise, this is best achieved through a measure of class overlap (e.g. the Bayes error rate for a given problem [20]); for output noise, the percentage noise is measured by the percentage of class labels that have been changed.

2.2.1.8 CLASSIFIERS

Two model-based and five discriminative classifiers are used in this study. The model-based classifiers are the NB [23] and Gaussian (Gauss) classifiers. The discriminative classifiers are the Gaussian mixture model (GMM), decision tree (DT) [12], kNN [24], MLP and SVM [25] classifiers.

The NB, DT, kNN, MLP and SVM classifiers are implementations of the machine learning package Weka [15] and the Gaussian and GMM classifiers are Matlab implementations available at [26] and [27].

The kNN classifier uses a LinearNN nearest neighbour search algorithm with an Euclidean distance metric; we determine the optimal k value by performing 10-fold cross-validation.

Two variations of the GMM classifier are used: a full covariance GMM classifier (GMMf) and diagonal covariance GMM classifier (GMMd). The GMMf classifier takes correlation between variables in a mixture into account when determining the probability density functions of each mixture. The GMMd classifier assumes that the variables in a mixture are independent. The expectation-maximisation algorithm is used to find the weights, mean values and covariance matrices of the mixtures. The number of mixtures per class must be specified to the GMM classifiers (we use iterative methods to find suitable values).

We use a k-means clustering algorithm to initialise the weights of the mixtures in the GMM

classifier; the prior probabilities of the mixtures are initialised as the proportions of the samples in a cluster belonging to each class; the covariance matrix is initialised as the sample covariance of the points associated with each mixture.

The SVM uses C-Support Vector classification where a regularisation parameter (C) is introduced to incorporate cost due to non-separability for linearly non-separable data; we use a radial basis function kernel. For each experiment the optimal cost parameter (C) and kernel width parameter (g) are determined by performing 10-fold cross-validation; g values in the range [10^{-8},10^{6}] and C values in the range [10^{-8},10^{4}] are considered. The Golden Ratio search [28] is used to search through the C and g dimensions to find the optimal error rate for the SVM classifier.

A single hidden-layer back-propagation MLP is used for which the optimal number of nodes in the hidden layer is determined by 10-fold cross-validation - we search through the range of two to ten hidden nodes.

2.3 EXPERIMENTAL DESIGN

The five research questions introduced in Section 2.1 are studied through the design of targeted data sets. All the experiments are repeated ten times on ten different data sets (with the same properties) to reduce the effect of variability in the results.

2.3.1 EXPERIMENT 1

Experiment 1 uses artificial data sets with Gaussian distributed classes to illustrate where the Gaussian classifier and the NB classifier outperform discriminative classifiers.

The method of data generation explained in Section 2.2.1 is used. We generate artificial data sets with correlated and uncorrelated variables and each data set contains three classes. The number of samples per class in each data set ranges from 20 to 100 and we use ten features. We range the SDs of these features from 1 to 25. The class means are chosen from different hypercubes to give well-separated means and all variables are in the range [-1, 1].

The purpose of this experiment is to generate data sets with models that fit the Gaussian classifier and the NB classifier assumptions well. The Gaussian classifier assumes data with Gaussian distributed classes and potentially correlated variables, whereas the NB classifier assumes independent variables of a particular one-dimensional distribution (for simplicity, we have employed Gaussian distributions for those cases as well). We vary the number of samples per class to probe for cases where the model-based assumption is optimally useful.

2.3.2 EXPERIMENT 2

Experiment 2 uses artificial data sets with Gaussian distributed classes and added output noise to illustrate the effect of output noise on the optimal value of k in the kNN classifier. The effect of output noise on the ratio between the error rate of the optimal kNN classifier and the error rate of the 1NN classifier is also illustrated.

We generate two and ten dimensional correlated data sets with noise fractions ranging from 5-25 % ; all the data sets have three classes and the number of samples per class ranges from 20 to 100. The standard deviations of the distributions are varied from 1 to 25 to illustrate the effect of the SD on the optimal k values. The 10-fold cross validation error rates of the optimal kNN classifier and the 1NN classifier are calculated; we compare these error rates for all the data sets.

2.3.3 EXPERIMENT 3

Experiment 3 uses two dimensional Gaussian distributed data with different SDs in the horizontal (x) and vertical (y) directions; these data sets are used to illustrate the effect of the constant distance metric used by the kNN classifier throughout the entire variable space.

Data sets with two and four classes are generated; we vary the number of samples per class from 20 to 100. We compare the 10-fold cross-validation error rates for the model-based and discriminative classifiers to the optimal kNN classifier.

2.3.4 EXPERIMENT 4

When an SVM classifier with Gaussian radial basis function kernel is optimised, two parameters are extremely important: The Gaussian kernel width (g) and the penalty parameter (C). The roles of these two parameters give us a useful hint on the key ingredients to the SVM's typically excellent performance, but also suggest potential weaknesses that will be explored in this experiment. The kernel width (g) is influenced by the scale of the various classes and the complexity measure (C) is influenced by the complexity of the decision boundaries between classes.

When an MLP classifier with one hidden layer (two layers of weights) is optimised, the number of hidden nodes in the hidden layer is similarly important. The number of hidden nodes is an indication of the complexity of the decision boundaries between the various classes.

We will probe these properties of the SVM and MLP classifiers by generating artificial data sets with classes that have varying scale and varying decision boundary complexity. All the data sets in experiment 4 are generated in two and ten dimensions. Each two-dimensional data set is expanded to ten dimensions by adding eight additional variables; this method is used to ensure that the specific properties of the data sets remain the same when the data are projected into higher dimensions. Each additional variable has a zero mean and SD equal to the original pair of variables of the class; these variables don't contribute information that aids in classification - they can thus be regarded as nuisance variables.

We investigate two important data properties: (1) the variation in decision boundary complexity between the various class combinations in a data set and (2) the scale variation of data in a data set, where we will distinguish between inter-class and intra-class scale variations.

2.3.4.1 INTER-CLASS SCALE VARIATION

To probe the effect of scale variation on the performance of classifiers, we start by constructing a data set with four Gaussian distributed classes. To create variation in scale we generate classes 1 and 2 with low SDs and classes 3 and 4 with high SDs. We ensure that the de-

gree of overlap between classes 1 and 2 is approximately the same as the degree of overlap between classes 3 and 4; there is consequently almost no variation in decision boundary complexity between the classes. The parameters used to generate these artificial data sets are summarised in Table 2.1. We will term these data sets artificial set 4.1.

Table 2.1: *Summary of artificial set 4.1 parameters*

Class	**Distribution**	μ**(feature1,feature2)**	**SD**
1	Gaussian	(-50, 0.1)	0.1
2	Gaussian	(-50, -0.1)	0.1
3	Gaussian	(50, 20)	20
4	Gaussian	(50,- 20)	20

To amplify the effect of scale variation and include extreme outliers, we will also generate artificial data with Gaussian and Cauchy distributed classes. We generate two classes with Cauchy class conditional probability density functions and four classes with Gaussian distributed class conditional probability density functions; the variances of the Cauchy distributed classes are extremely large compared to the Gaussian distributed classes. We will term these data sets artificial set 4.2. The parameters used to generate these artificial data sets are summarised in Table 2.2.

Table 2.2: *Summary of artificial set 4.2 parameters*

Class	**Distribution**	μ**(feature1,feature2)**	**SD**
1	Gaussian	(-90, 25)	0.3
2	Gaussian	(-90, 24)	0.3
3	Gaussian	(-90, 23)	0.3
4	Gaussian	(-90, 22)	0.3
5	Cauchy	(100, -100)	10
6	Cauchy	(100, 100)	10

Note that the class overlaps between classes 1-4 are similar and the class overlap between classes 5 and 6 is slightly more.

We generate similar data sets with a slight variation in the mean value of class 4 - we move class 4 closer to class 3. The class overlap between classes 3 and 4 is now bigger than the

class overlaps between classes 1 and 2 and classes 2 and 3. These data sets will allow us to probe the effect of a variation in decision boundary complexity on the MLP and SVM classifiers.

Table 2.3: *Summary of artificial set 4.3 parameters*

Class	Distribution	μ(feature1,feature2)	SD
1	Gaussian	(-90, 25)	0.3
2	Gaussian	(-90, 24)	0.3
3	Gaussian	(-90, 23)	0.3
4	Gaussian	(-90, 22.5)	0.3
5	Cauchy	(100, -100)	10
6	Cauchy	(100, 100)	10

Note that the class overlaps between classes 1 and 2 and classes 2 and 3 are similar while the overlap between classes 3 and 4 is bigger; we thus also include a slight variation in decision boundary complexity.

2.3.4.2 DECISION BOUNDARY COMPLEXITY

We generate an artificial data set to probe the effect of varying decision boundary complexity on classification performance. The data set consists of six Gaussian distributed classes; these six classes all have different means but the same SD. This allows us to focus on the variation in decision boundary complexity. We will term these data sets artificial set 4.4. The parameters used to generate these artificial data sets are summarised in Table 2.4.

Table 2.4: *Summary of artificial set 4.4 parameters*

Class	Distribution	μ(feature1,feature2)	SD
1	Gaussian	(-1, 25)	0.3
2	Gaussian	(-1, 24)	0.3
3	Gaussian	(-1, 23)	0.3
4	Gaussian	(-1, 22.5)	0.3
5	Gaussian	(0, 24.5)	0.3
6	Gaussian	(1, 22.5)	0.3

Note that the SDs are the same for all classes; there is thus no variation in scale. The

class overlap between the various classes varies, which in turn causes a variation in decision boundary complexity between the various class combinations.

2.3.4.3 INTRA-CLASS SCALE VARIATION

We generate similar data sets to artificial sets 4.1 and 4.3 with slight variations: we will assign only two class labels to these data sets. This will allow us to investigate the effect of intra-class scale variation on the classification performance of the MLP and SVM classifiers. The parameters used to generate these artificial data sets are summarised in Tables 2.5 and 2.6; we will term these data sets artificial sets 4.5 and 4.6.

Table 2.5: *Summary of artificial set 4.5 parameters*

Class	**Distribution**	μ**(feature1,feature2)**	**SD**
1	Gaussian	(-50, 0.1)	0.1
2	Gaussian	(-50, -0.1)	0.1
1	Gaussian	(50, 20)	20
2	Gaussian	(50,- 20)	20

Table 2.6: *Summary of artificial set 4.6 parameters*

Class	**Distribution**	μ**(feature1,feature2)**	**SD**
1	Gaussian	(-90, 25)	0.3
2	Gaussian	(-90, 24)	0.3
1	Gaussian	(-90, 23)	0.3
2	Gaussian	(-90,- 22.5)	0.3
1	Cauchy	(100, -100)	10
2	Cauchy	(100, 100)	10

2.3.4.4 VARIATION IN DECISION BOUNDARY COMPLEXITY AND SCALE

To probe the simultaneous effect of inter-class scale variation and variation in decision boundary complexity on the classification performances of the SVM and MLP classifiers, we generate artificial data sets with five Gaussian distributed classes. We will term these data sets artificial set 4.7. The parameters used to generate these artificial data sets are summarised in Table 2.7.

Table 2.7: *Summary of artificial set 4.7 parameters.*

Class	**Distribution**	μ**(feature1,feature2)**	**SD**
1	Gaussian	(-50, 12)	5
2	Gaussian	(-50, -12)	5
3	Gaussian	(-50, -0.2)	0.1
4	Gaussian	(50, 0)	0.1
5	Gaussian	(50, 0.2)	0.1

It is important to note that the class overlap between classes 1 and 2 is much smaller than the class overlaps between classes 3-5. These data sets simulate both variation in scale and variation in decision boundary complexity simultaneously.

We will introduce a slight variation to these data sets by labelling the data with only two classes; this will allow us to probe the simultaneous effect of intra-class scale variation and variation in decision boundary complexity. We will term these data sets artificial set 4.8. The parameters used to generate these artificial data sets are summarised in Table 2.8.

Table 2.8: *Summary of artificial set 4.8 parameters*

Class	**Distribution**	μ**(feature1,feature2)**	**SD**
1	Gaussian	(-50, 12)	5
2	Gaussian	(-50, -12)	5
1	Gaussian	(-50, -0.2)	0.1
2	Gaussian	(50, 0)	0.1
1	Gaussian	(50, 0.2)	0.1

2.3.5 EXPERIMENT 5

In this experiment we investigate the relationship between data set complexity and classification performance. We simulate data set complexity by sampling data from a Gaussian mixture model distribution; the complexity of the data set is determined by the number of mixtures per class and by the SDs of the mixtures. If there is significant overlap between the mixtures in a class the mixtures will fuse together to create a larger mixture. From the perspective of decision-boundary complexity, the effective number of mixtures per class is

consequently less than the actual number of mixtures per class.

Data sets with two and ten features are generated; the two-dimensional data have SDs in the range [0.1, 1]. The ten-dimensional data have SDs in the range [1, 10]. The SDs of the ten-dimensional data are a factor ten bigger than those used for the two-dimensional data; this ensures that the degree of overlap in two dimensions and in ten dimensions is similar.

Data sets with 10 and 50 groups per class are generated; all data sets contain three classes and ten samples per group. The mean values of the groups are chosen randomly and all features are in the range [-1, 1].

2.4 RESULTS

The results of the experiments in Section 2.3 are summarised in this section. Throughout our discussion, high-dimensional data are defined as data with a small number of samples in each class per dimension.

2.4.1 EXPERIMENT 1

The classification results of experiment 1 are given in Figures 2.1 and 2.2; we abbreviate samples per class as spc. Figures 2.1 and 2.2 show that the Gaussian classifier achieves the lowest error rate over all the correlated data sets in this experiment. This result is not surprising in itself since the data are in fact normally distributed; the interesting results are contained in: (1) the extent to which the discriminative classifiers underperform the appropriate model-based classifier, and (2) the dependence of this underperformance on factors such as data overlap and the size of the training set.

We see that all the discriminative classifiers perform considerably worse than the model-based classifier in the ten-dimensional space. It is thus not safe to assume that discriminative classifiers will perform comparably to a Gaussian classifier on correlated high-dimensional data with a small number of samples per class.

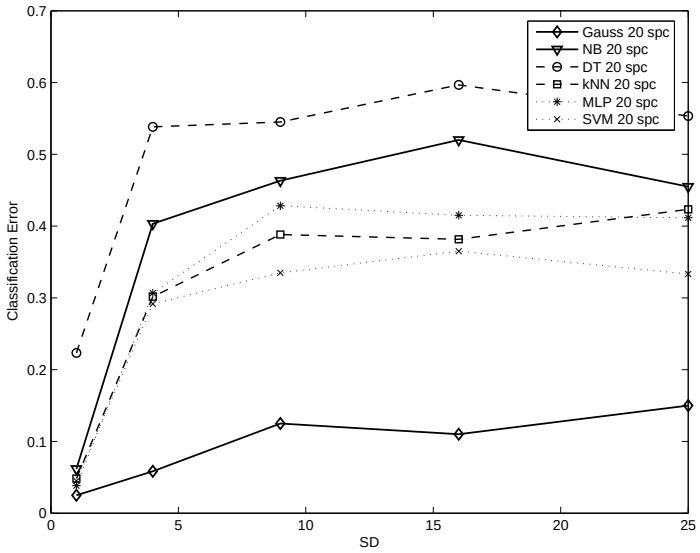

Figure 2.1: *Classification results of correlated ten-dimensional data (20 samples per class)*

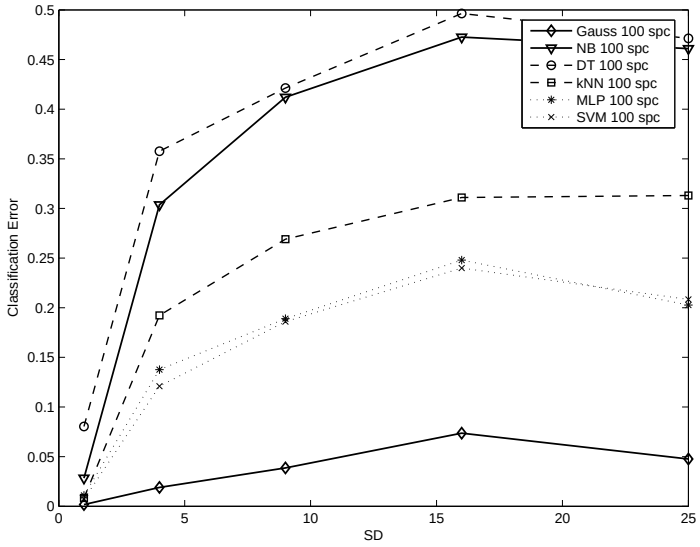

Figure 2.2: *Classification results of correlated ten-dimensional data (100 samples per class)*

Figures 2.3 and 2.4 show that the NB classifier has the lowest error rate over all the uncorrelated data sets used in this experiment, whereas all the other classifiers had substantial error rates for at least some experimental conditions. It is thus not safe to assume that discriminative classifiers will perform comparable to a NB classifier on uncorrelated high-dimensional data with a small number of samples per class.

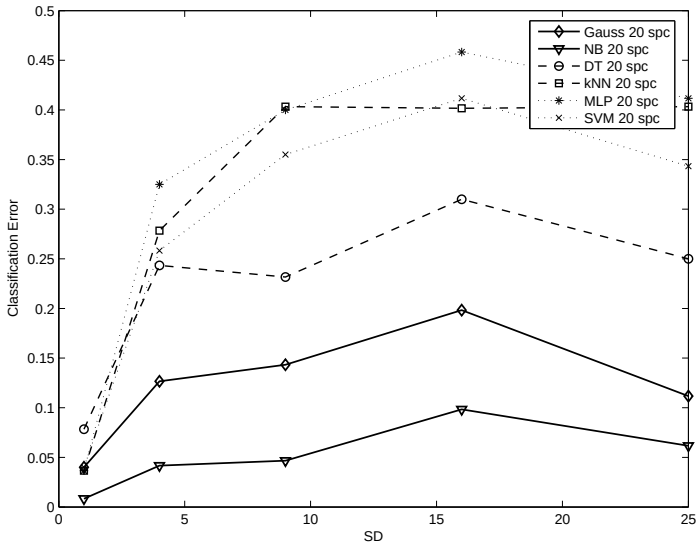

Figure 2.3: *Classification results of uncorrelated 10 dimensional data (20 samples per class)*

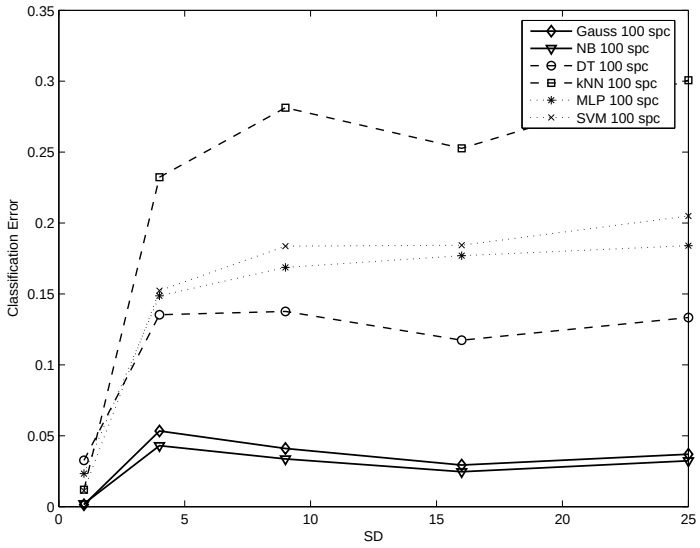

Figure 2.4: *Classification results of uncorrelated 10 dimensional data (100 samples per class)*

2.4.2 EXPERIMENT 2

The results of experiment 2 are given in Figures 2.5-2.8. Figure 2.5 shows that the optimal value of k for the kNN classifier increases monotonically as the (output) noise in the data increases, whereas the optimal k value seems to decrease (though not as predictably) when the SD increases. At first glance these results seem contradictory, since the SD can also be viewed as a form of noise – specifically, input noise. However, these results are actually consistent, and provide an important hint on the choice of k: whereas increasing SD does

create increasing overlap of the different classes, samples that overlap tend to lie at the edges of these distributions. Output noise, on the other hand, permeates the entire feature space – hence, a larger k value is required to properly smooth over these samples as the noise percentage increases.

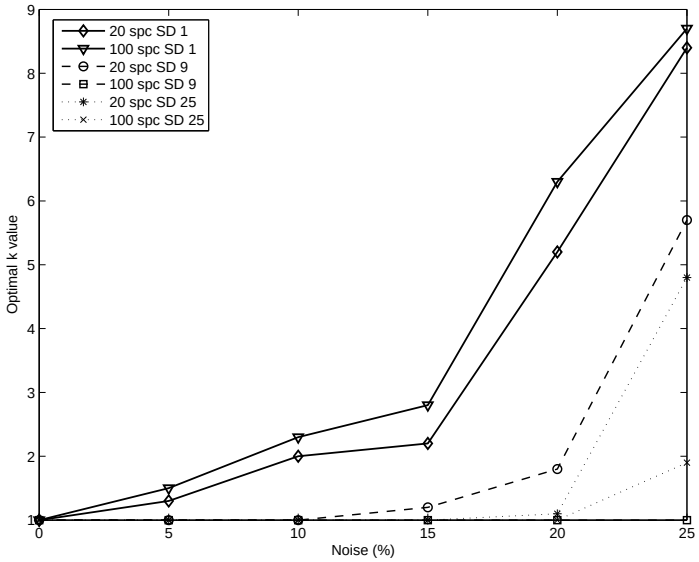

Figure 2.5: *kNN classification results of correlated noisy two-dimensional data*

Figure 2.6 shows that in a high-dimensional feature space with high SD, the optimal k values are close to 1. This might be because, for large k, the contributing samples may be so far away from the sample as to be meaningless.

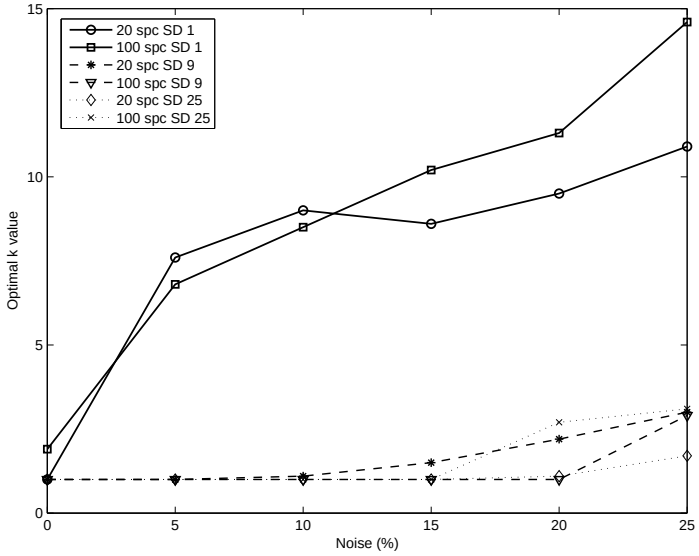

Figure 2.6: *kNN classification results of correlated noisy ten-dimensional data*

Figure 2.6 also shows that the optimal k value continues to increase reasonably monotonically with the noise percentage in 10 dimensions both for high and low overlap.

How significant are the differences between the accuracies obtained with the various values for k? Figures 2.7 and 2.8 show the error rates of the 1NN classifier divided by those of the optimal kNN classifier for each of the cases corresponding to Figures 2.5 and 2.6.

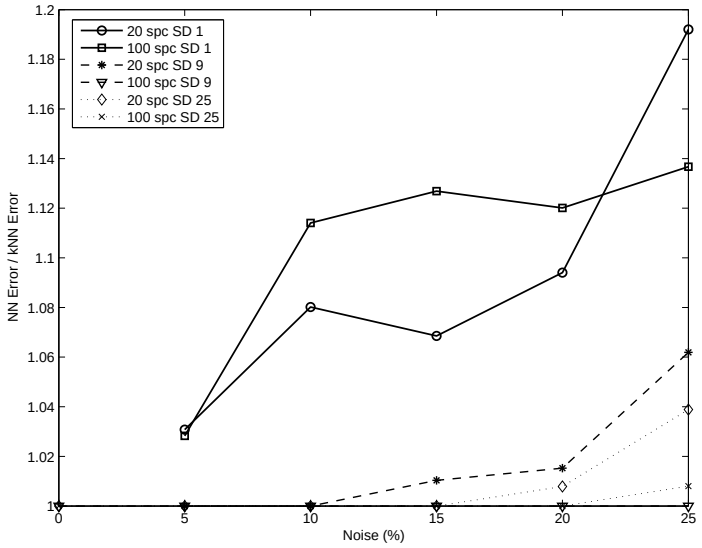

Figure 2.7: *Error rate ratios of correlated noisy two-dimensional data*

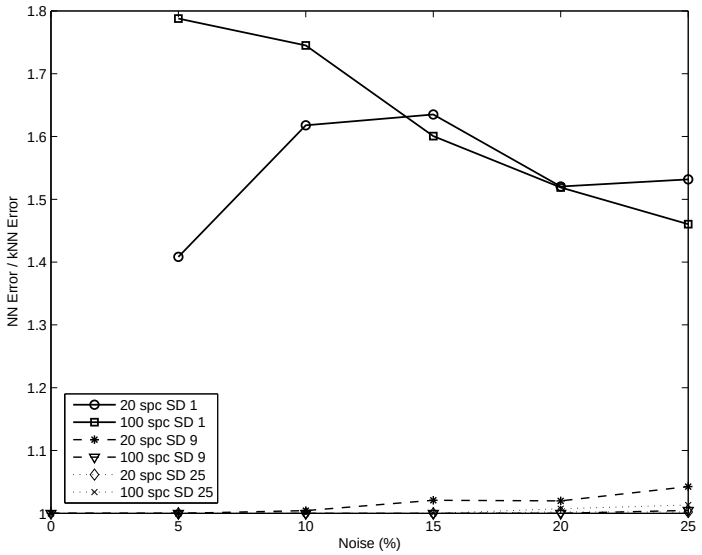

Figure 2.8: *Error rate ratios of correlated noisy ten-dimensional data*

In the vast majority of cases, for a SD of 1 and dimensionality of 10, the 1NN classifier has

more than 1.5 times the error rate of the optimal classifier, suggesting that these differences are indeed significant.

2.4.3 EXPERIMENT 3

Two-dimensional data sets with uncorrelated class-conditional densities were used in experiment 3. The SDs of the classes in the data sets were different in the horizontal (x) and vertical (y) directions. Figure 2.9 is a scatter plot of a four-class data set (with 100 samples per class) that was used in one of the experimental runs.

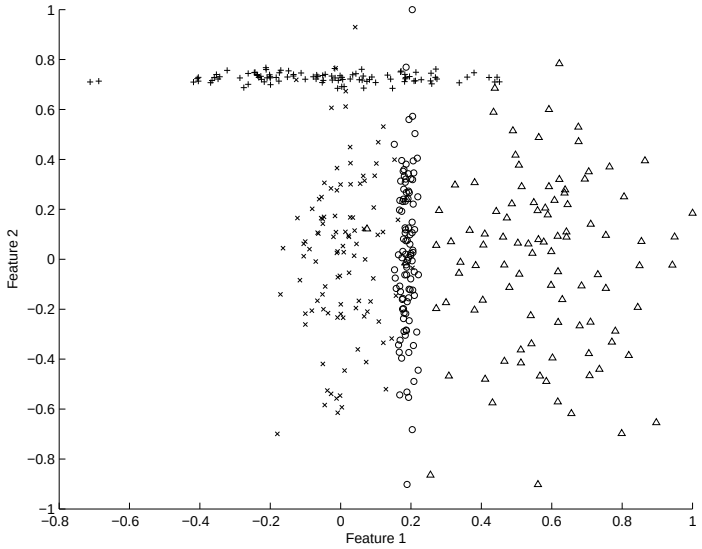

Figure 2.9: *Scatter plot of four-class 100 samples/class two-dimensional data*

Figure 2.9 shows that the classes marked with 'X', 'O' and 'Δ' all have the same SD in the y direction (Feature 2) but have different SDs in the x direction (Feature 1). The class marked by '+' has a very large SD in the x direction and a very small SD in the y direction.

The kNN classification results of uncorrelated two-dimensional artificial data sets similar to the data set illustrated in Figure 2.9 are summarised in Figures 2.10 and 2.11. Figures 2.10 and 2.11 show that the kNN classifier has high error rates for these artificial data sets and that the classification results become worse, compared to the other classifiers, when the number of classes increase. These results show that the kNN classifier is best employed in cases where the "natural" metric is fairly constant throughout the feature space.

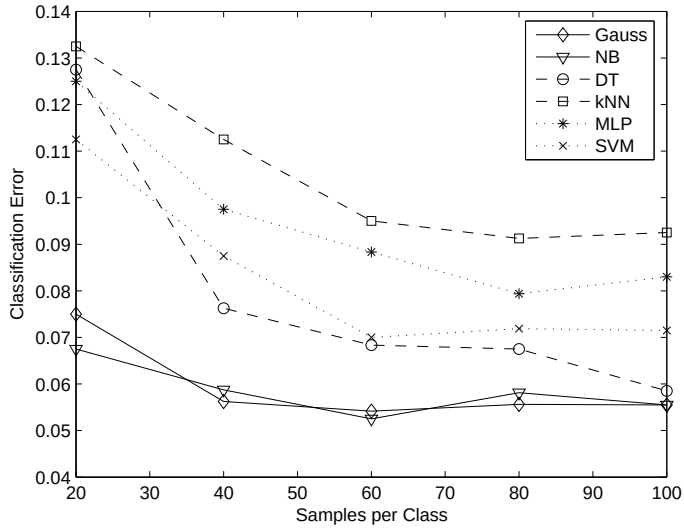

Figure 2.10: *kNN classification results of uncorrelated two-class data*

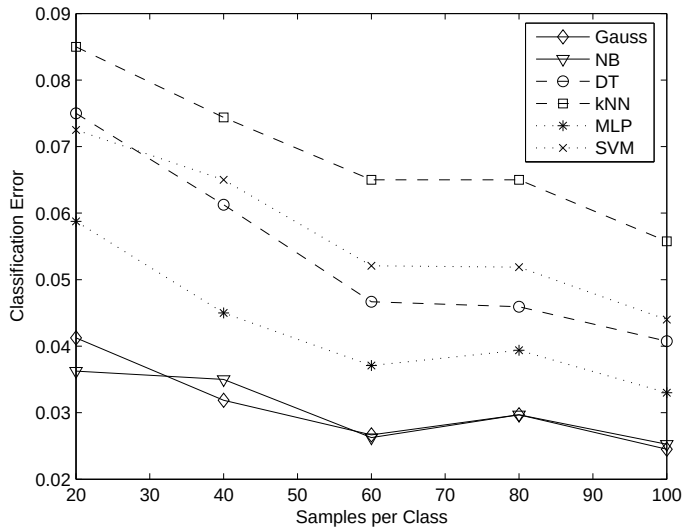

Figure 2.11: *kNN classification results of uncorrelated four-class data*

2.4.4 EXPERIMENT 4

The classification results for experiment 4 are summarised in Figures 2.12 - 2.23. (We denote dimensionality as D.)

2.4.4.1 INTER-CLASS SCALE VARIATION

Figures 2.12 - 2.15 show that the MLP classifier has the highest classification error rate of all the classifiers for artificial sets 4.1 and 4.3. Artificial sets 4.1 and 4.3 were both generated

to simulate large variations in scale between various classes. These results suggest that the classification error rate of a MLP increases as the variation in scale between classes increases.

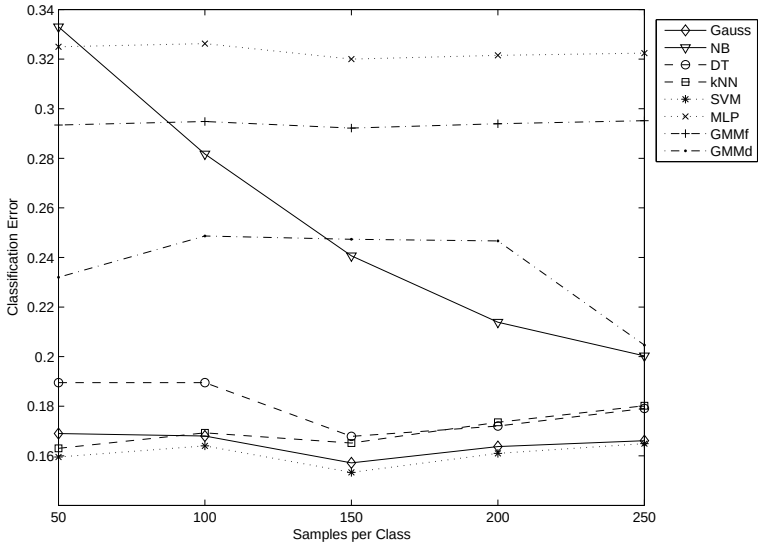

Figure 2.12: *Classification results of artificial set 4.1 (2 D)*

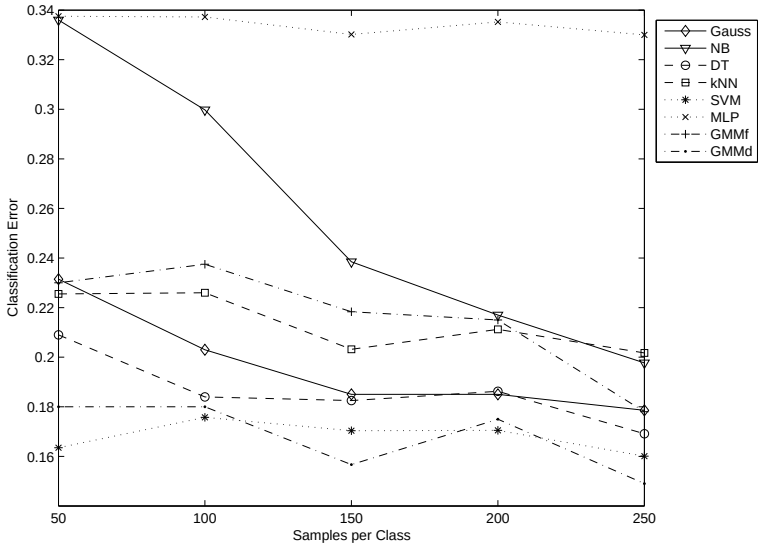

Figure 2.13: *Classification results of artificial set 4.1 (10 D)*

Figures 2.12 and 2.13 show that the NB classifier error rate decreases significantly as the number of samples per class increases in artificial set 1. Figures 2.14 and 2.15, on the other hand, show that the classification error of the NB classifier increases slightly as the number of samples per class increases. Both sets of data have large inter-class scale variation; the distributions of the classes within these sets of data do, however, differ significantly. Artificial set 4.1 contains only Gaussian distributed classes while artificial set 4.3 contains classes

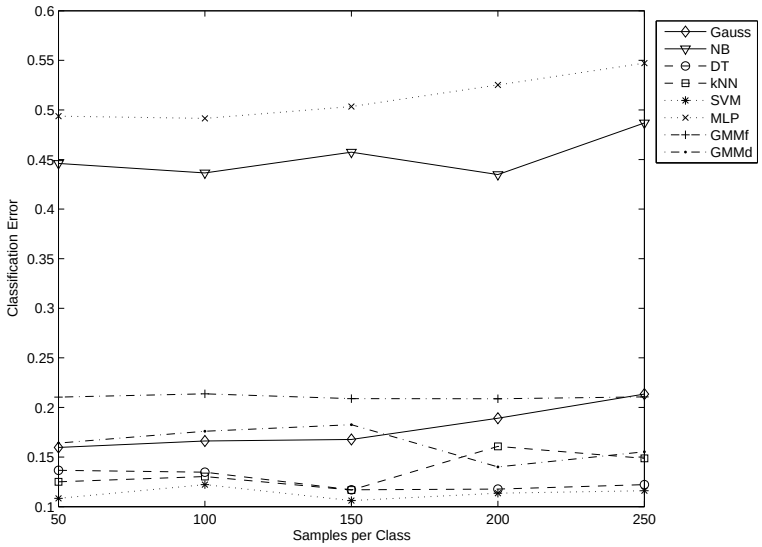

Figure 2.14: *Classification results of artificial set 4.3 (2 D)*

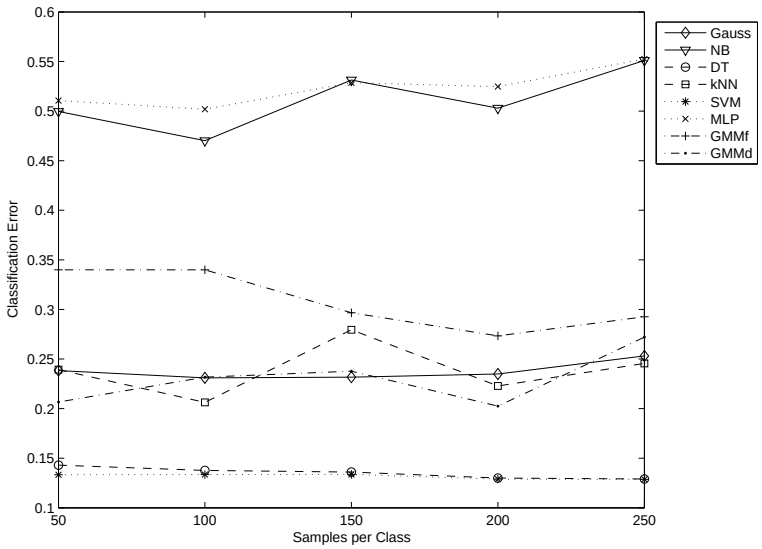

Figure 2.15: *Classification results of artificial set 4.3 (10 D)*

with Gaussian and Cauchy distributions. The NB classification performance of artificial set 4.1 improves since the data sets contain only Gaussian distributed classes - the assumptions made by the NB classifier thus fit the distributions of the classes and the data are modelled more accurately with an increase in training data. The Cauchy distributed classes in artificial set 4.3, on the other hand, do not fit the assumption of normality made by the NB classifier; this explains why the NB classification performance does not improve with an increase in training data.

2.4.4.2 DECISION BOUNDARY COMPLEXITY

Figures 2.16 and 2.17 show that the MLP has relatively good classification performance on artificial set 4.4.

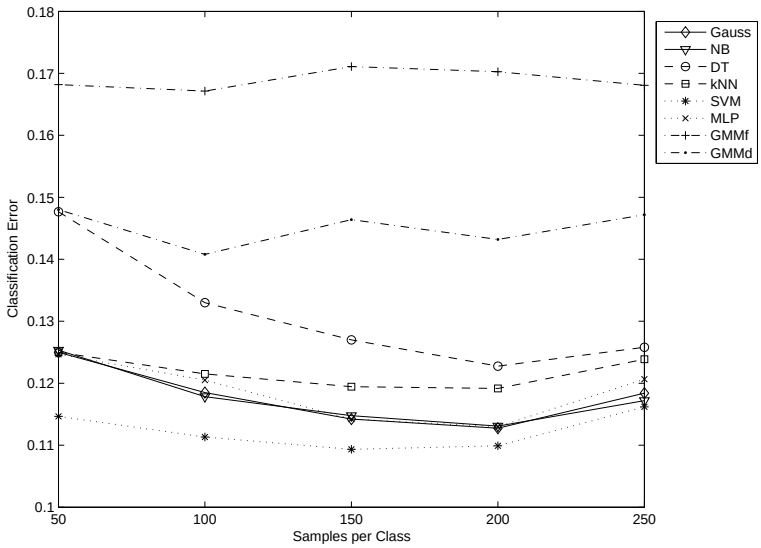

Figure 2.16: *Classification results of artificial set 4.4 (2 D)*

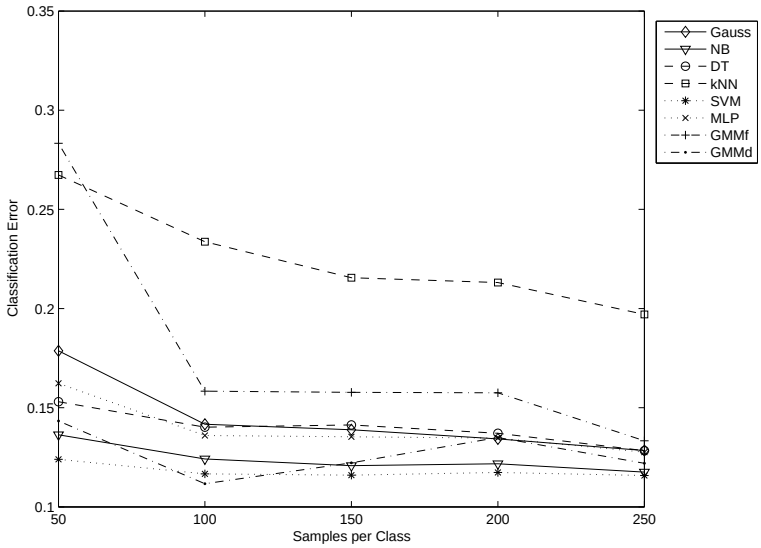

Figure 2.17: *Classification results of artificial set 4.4 (10 D)*

The results suggest that the classification performance of the MLP is not compromised when the variation in decision boundary complexity increases if there is no significant scale variation. We also see that the classification performance of the SVM classifier is not influenced by a variation in decision boundary complexity; we note, however, that the kNN

seems to be sensitive to a variation in decision boundary complexity in the ten-dimensional case.

We can investigate the simultaneous effect of inter-class scale variation and variation in decision boundary complexity by comparing Figures 2.15 and 2.18, since artificial set 4.3 has the same scale variation as artificial set 4.2 but a higher variation in decision boundary complexity.

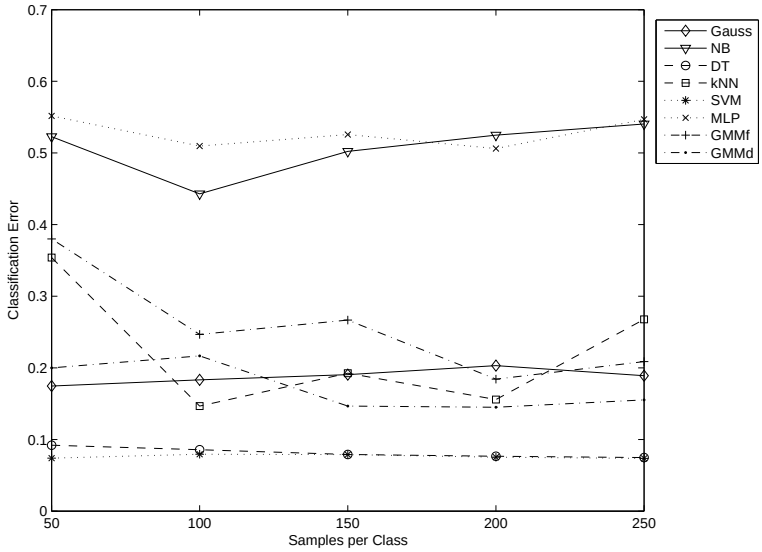

Figure 2.18: *Classification results of artificial set 4.2 (10 D)*

We see that the classification performance of the MLP is approximately the same for both sets of data; these results suggest that the variation in decision boundary complexity doesn't influence the classification performance of a MLP classifier, even when the variation in scale is extremely high.

2.4.4.3 INTRA-CLASS SCALE VARIATION

Figures 2.19 and 2.20 show that the MLP has relatively poor classification performance on artificial set 4.5; this suggests that the MLP is also sensitive to intra-class scale variation, since the data in artificial set 4.5 consist of two classes with two groups per class.

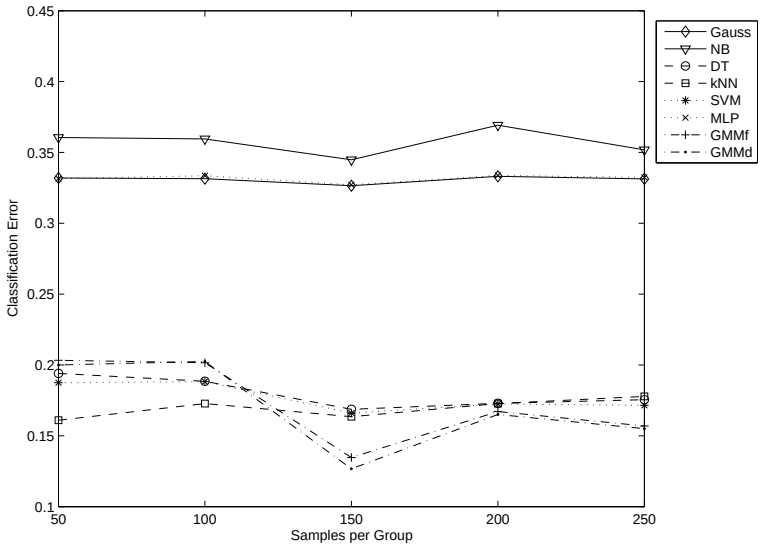

Figure 2.19: *Classification results of artificial set 4.5 (2 D)*

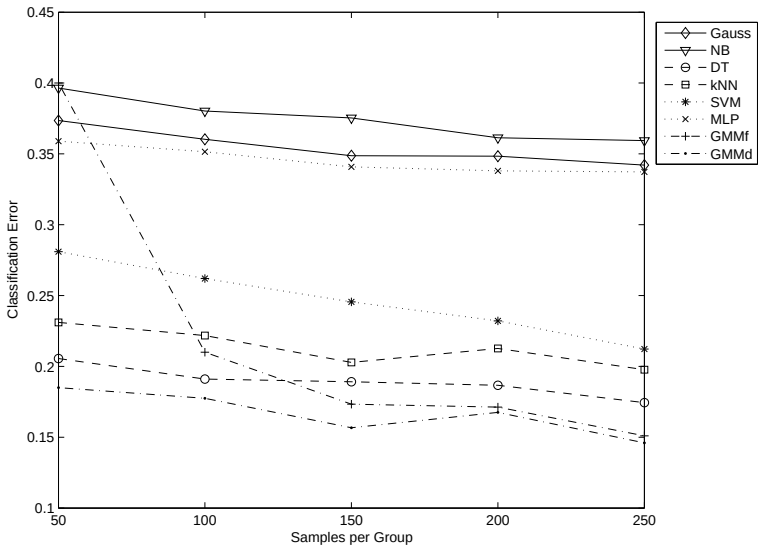

Figure 2.20: *Classification results of artificial set 4.5 (10 D)*

If we compare the classification results of Figure 2.15 and Figure 2.21 we see that the classification error rates of the SVM have increased significantly in Figure 2.21; the relative SVM classification error rates for artificial set 4.6 are much higher than for artificial set 4.3. Note that inter-class scale variation was simulated in artificial set 4.3 and intra-class scale variation was simulated in artificial set 4.6; these results thus imply that SVM classification error rate increases as intra-class scale variation increases.

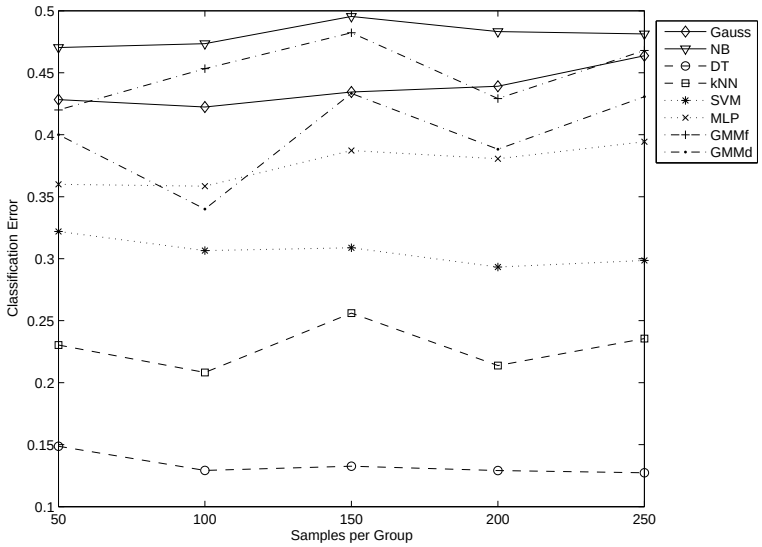

Figure 2.21: *Classification results of artificial set 4.6 (10 D)*

Figures 2.22 and 2.23 show that the SVM classification error rates for artificial set 4.8 are significantly higher than for artificial set 4.7. Artificial set 4.7 has high inter-class scale variation and high variation in decision boundary complexity while artificial set 4.8 has high intra-class scale variation and high variation in decision boundary complexity. These results verify that SVM classification error rate is increased by an increase in intra-class scale variation.

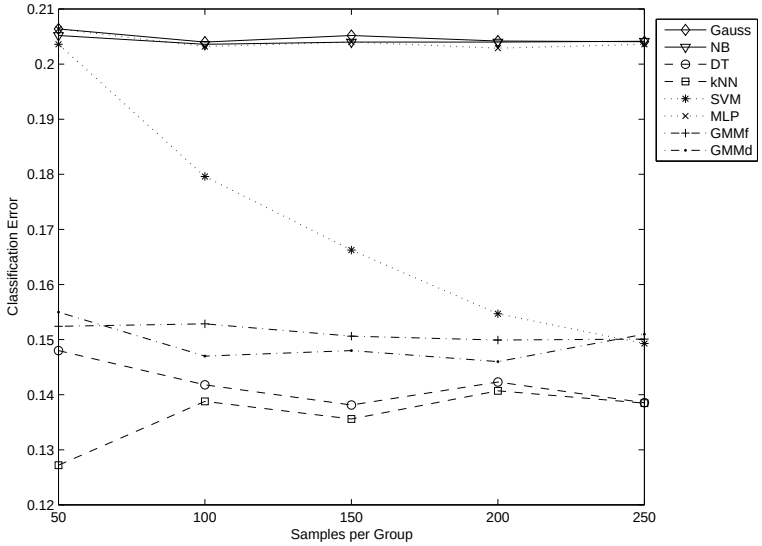

Figure 2.22: *Classification results of artificial set 4.8 (2 D)*

If we compare the relative SVM classification performance of artificial set 4.8 (Figure 2.22) to that of artificial set 4.5 (Figure 2.19) we see that the relative SVM performance

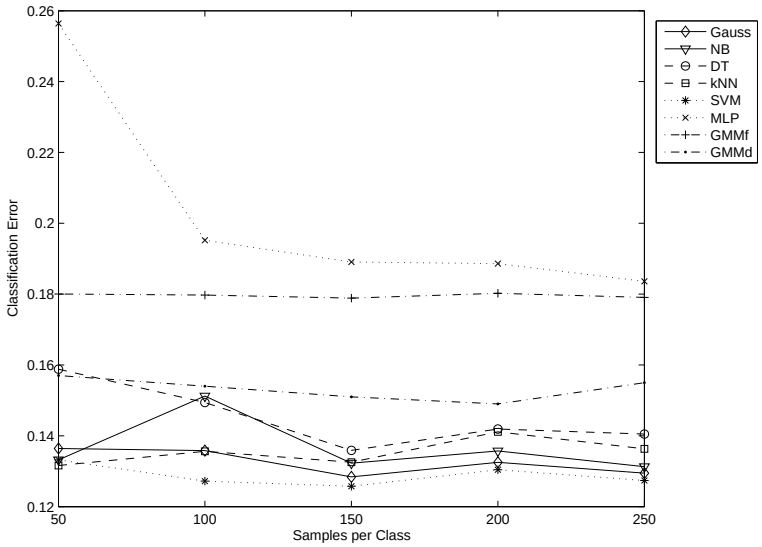

Figure 2.23: *Classification results of artificial set 4.7 (2 D)*

is considerably worse for artificial set 4.8. Artificial set 4.8 simulated both intra-class scale variation and variation in decision boundary complexity, while artificial set 4.5 simulated only intra-class scale variation. These results suggest that the negative effect of the intra-class scale variation on the SVM classification performance is amplified by the additional variation in decision boundary complexity.

We finally note that the DT classifier had extremely good classification performance in general on all the data sets employed in this experiment, especially on the data sets containing Cauchy distributed data, see e.g. Figures 2.15 and 2.18. The DT classifier had the best relative classification performance on artificial set 4.6 (see Figure 2.21); the data sets in artificial set 4.6 contained a combination of Gaussian and Cauchy distributed classes.

The DT classifier has two desirable properties for the data used in this experiment: (1) the DT distinguishes between the importance of features (in terms of classification) and defines relationships between variables; these properties allow the DT classifier to do well on data sets that have uncorrelated variables and a high number of noisy features; (2) the DT classifier is discriminative (it doesn't make any assumptions regarding the distribution of the data)

The first property explains why the DT classifier performed very well (compared to the other classifiers) on the ten-dimensional data - the data sets used in this experiment had only two features that contributed to classification, the ten-dimensional data thus had eight noisy features. The second property also explains the good overall classification performance of the DT classifier, since Cauchy distributed classes were used in some of the data sets in this experiment and no assumptions were made regarding these class-conditional probability density functions by the DT classifier.

The excellent classification performance of the DT classifier on artificial set 4.6 (Figure 2.21) might be attributed to three data properties: (1) the data sets contain high intra-class scale variation; (2) the data sets contain Cauchy distributed data; (3) the data sets contain large proportions of noisy features.

We have shown that the classification performances of the MLP and SVM classifiers are greatly decreased by the intra-class scale variation, whereas the localized nature of DT discriminants allows it to handle scale variations successfully. The classification performances of the model-based classifiers (Gauss and NB) are greatly decreased by the Cauchy distributed data while the DT does not make any assumptions regarding the distribution of the data. The DT classifier determines the importance of features when performing classification; this property of the DT classifier makes it very robust against noisy features, and gives it an advantage over the other classifiers on artificial set 4.6. The combination of these three data properties might explain the excellent classification performance of the DT classifier on artificial set 4.6.

2.4.5 EXPERIMENT 5

The results of experiment 5 are given in Figures 2.24-2.27. Figure 2.24 shows that the GMMd, GMMf and kNN classifiers achieve the lowest classification error rates for two-dimensional data with ten groups per class. This result is not surprising in itself since the data sets were generated from Gaussian mixtures. The interesting result is the good classification performance of the kNN classifier. The sparseness of the data (each group contains only ten samples) might explain why the kNN has better overall classification performance than the

SVM and MLP classifiers since the SVM and MLP classifiers require more data points to obtain accurate decision boundaries.

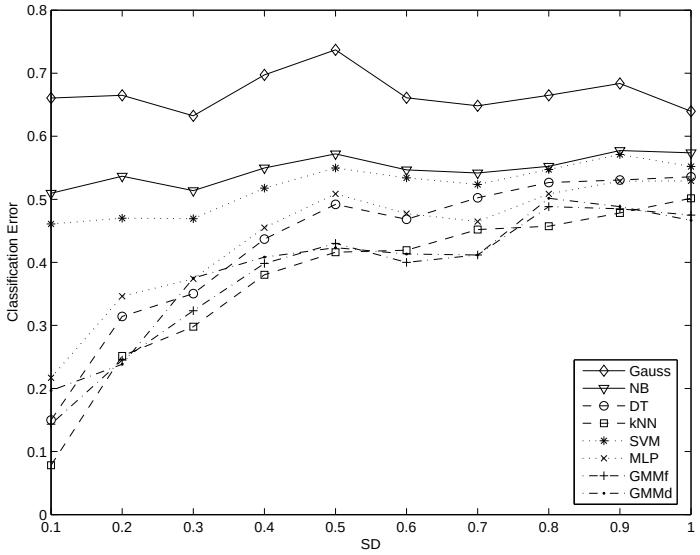

Figure 2.24: *Classification results of GMM data (2 D, ten groups per class)*

Figure 2.25 shows that the GMMd classifier achieves the lowest error rate for ten-dimensional data with ten groups per class.

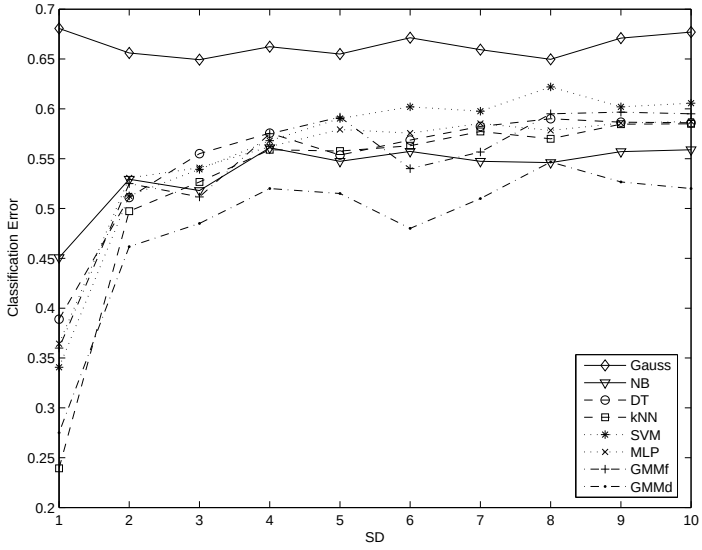

Figure 2.25: *Classification results of GMM data (10 D, 10 groups per class)*

We see that the error rates of the kNN and GMMf classifiers relative to the GMMd have increased with dimensionality. The sparsity of data in the feature space increases as the dimensionality of the data increases; this sparseness of the data, combined with the effect

of so many mixtures, causes the discriminative classifiers to underperform the GMMd classifier by a considerable margin. The GMMd classifier performs considerably better than the GMMf classifier, as would be expected given that the samples within each mixture component are uncorrelated. The GMMf requires more samples to obtain accurate estimates of the mixture parameters since the GMMf has more parameters than the GMMd to estimate; it cannot obtain accurate model parameters since the data is so sparse in ten dimensions.

Figures 2.26 and 2.27 show that the results obtained for ten groups per class also hold for 50 groups per class.

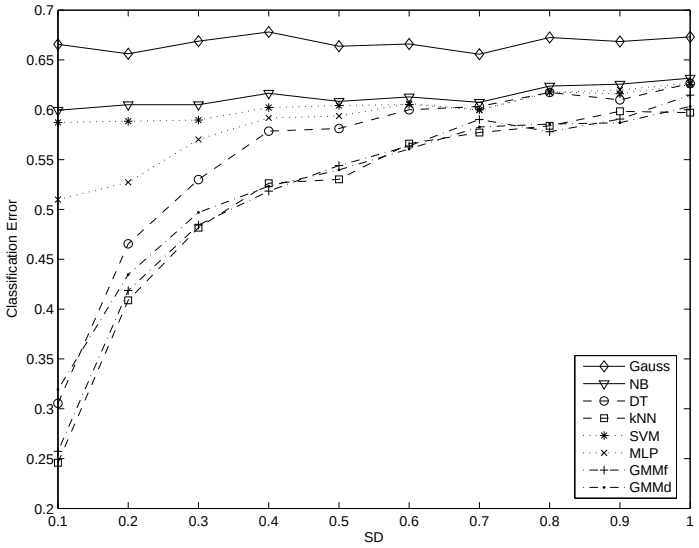

Figure 2.26: *Classification results of GMM data (2 D, 50 groups per class)*

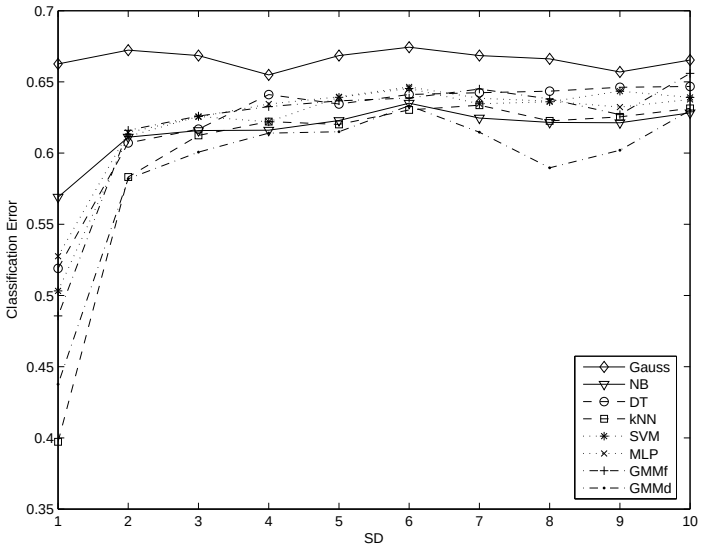

Figure 2.27: *Classification results of GMM data (10 D, 50 groups per class)*

2.5 CONCLUSION

We have studied several examples where data sets do not behave according to "conventional wisdom". We have shown classification problems where model-based classifiers outperform several discriminative classifiers by a wide margin, and where kNN classifiers, even with optimised k, perform poorly in comparison with the other classifiers studied.

At least some of these observations can be understood by reference to the detailed properties of the particular classifiers employed. For example, we have seen that kNN classifiers are best employed in cases where the "natural" metric is fairly constant throughout feature space, and that the optimal value for k depends on the effective output noise, rather than the input noise (which produces a different form of class overlap). We have also seen that model-based classifiers are a viable alternative to discriminative classifiers when the amount of training data is severely limited (relative to the dimensionality of the feature space), and the parametric form of the assumed model is a sufficiently good fit for the actual data distribution.

Similarly, we have shown that the classification performance of the MLP is influenced by the inter-class and intra-class scale variation in a data set. We have also shown that the SVM

classification performance is degraded by an increase in intra-class scale variation and that the negative effect of the intra-class scale variation is amplified by an additional variation in decision boundary complexity. Finally, we have shown that the DT classifier performs very well (compared to the other classifiers) on data that contain noisy features which are uncorrelated with more informative features. We have also seen that the DT classifier performance is not influenced by any scale variations or distributions of data.

All these specific results show that data properties influence even non-parametric classifiers in much the same way that the parametric fit can influence the performance of parametric classifiers. However, in order to use such insights for practical purposes, we need measures that allow us to predict how well a classifier will perform on an arbitrary classification task. That task is taken up in the next chapter.

CHAPTER THREE

DATA MEASURES

3.1 INTRODUCTION

In Chapter 2 we identified several important data properties that influence classification performance; in this chapter we will develop data measures that are specifically tailored to measure such data properties. Previous empirical studies have shown that data measures can be employed to give valuable insight into data set properties [11, 29]; these studies have, however, failed to explain how these properties influence classification performance. A true understanding of this relationship is required to develop a successful meta-classification system.

We will use the data properties that we identified in Chapter 2 to guide us in developing data measures; these measures will allow us to define the relationship between data characteristics and classifier performance. We group these measures into the following categories: standard measures, data sparseness measures, statistical measures, information theoretic measures, decision boundary measures, topology measures and noise measures.

Each section in this chapter will discuss a group of measures in detail. We will conclude this chapter with a summary of all these measures and their relationships to the data properties that influence classification performance.

3.2 STANDARD MEASURES

We list three generic measures that serve to normalise many of our other measures in Table 3.1.

Table 3.1: *Standard measures*

Measure	Data property
d	Number of features
C	Number of classes
N	Number of samples

We have illustrated in Section 2.4.1 that the number of samples per dimension can have a great influence on selecting model-based classifiers over discriminative classifiers, and in general the number of dimensions plays a critical role in the selection of classifiers. The number of classes in a data set influences the sparsity of data in a class; any classifier requires samples from each class, either to model the class-conditional probability density function (model-based) or to determine the decision boundaries between classes (discriminative). Similarly, the number of samples per class influences classification performance to a great extent, since it determines the amount of information available for the purposes of training. We will discuss these three standard features in more detail in the next section.

3.3 DATA SPARSENESS

In this section we will investigate the relationship between the dimensionality of data and the number of samples required to model the data accurately; this relationship is not trivial and we will define measures that capture some of the relevant factors.

In Section 3.3.3 we will develop a single measure to quantify whether the number of samples in a data set is sufficient to model the data accurately; this measure will measure how sparse

data is by taking the dimensionality, number of classes and number of samples in a data set into account.

3.3.1 RELATIONSHIP BETWEEN DIMENSIONALITY, DATA SET SIZE AND NUMBER OF CLASSES

The amount of data required to obtain a given level of classification accuracy is typically a monotonic function of the number of dimensions in the feature space, for a given data family. In typical cases, this relationship between dimensionality (d) and the number of samples (N) can be linear, quadratic or exponential, as we show below. Specific data properties can be used to decide which one of these relationships is more accurate. The way in which the number of classes (C) is factored into this relationship depends on the type of relationship.

We will use theoretical properties of classifiers to describe each of the three types of relationship. We will use Gaussian data distributions to illustrate how the first two cases arise, but it should be clear that these relationships hold for much wider families of distributions.

3.3.1.1 LINEAR RELATIONSHIP

If the variables in a data set are uncorrelated, the NB classifier will be a suitable candidate for classification. The NB assumes that the features are independent and, for data with a Gaussian distribution, requires a SD for each of these features; the NB also requires mean values of the features and prior probabilities for each class. The number of parameters that must thus be estimated is therefore $2dC + C$. The relationship between d and the number of samples required to model the data accurately for a NB classifier is thus a linear function of the dimensionality.

To test if a linear relationship holds between d and N we will employ the same tests that are required to test the important data properties of a NB classifier. These properties are the normality of the data and the correlation between features; they are sufficient, but not necessary tests for a linear relationship. We will discuss how to measure these two properties in Section 3.4.

3.3.1.2 QUADRATIC RELATIONSHIP

If correlations between variables exist and the covariances of all the classes are close to the pooled covariance matrix then the normal-based linear classifier will be a suitable candidate for classification.

The normal-based linear classifier assumes that all the class covariance matrices are similar, d^2 variables must thus be determined; C class means and C class priors are also required. The total number of parameters that must thus be estimated is $d^2 + dC + C$ - thus, a quadratic function of d.

To test if this quadratic relationship between d and N holds we will measure the homogeneity of class covariance matrices as well as the normality of the class data. We will discuss these measures in Section 3.4. The requirements are again sufficient, but not necessary.

If the class covariance matrices are not (approximately) equal, and the data are normally distributed, the standard Gaussian classifier will be a suitable candidate for classification. A Gaussian classifier must determine the full covariance matrix of each class, C class means and C class priors; the total number of parameters that must be estimated is thus $Cd^2 + dC + C$, which is once again a quadratic function of d (there is also a linear relationship between the required number of samples and C).

To test if this type of quadratic relationship between N and d holds and if a linear relationship between N and C holds, we will measure the normality of the class-conditional probability density functions.

3.3.1.3 EXPONENTIAL RELATIONSHIP

If no assumptions concerning the distributions of the class-conditional probability density functions can be made, it will not be safe to assume a linear or quadratic relationship between N and d since an exponential relationship between N and d can arise [30]. It is useful to think of a histogram approach to understand this relationship. If we want to construct a histogram from data with at least one sample in each bin and with D_{steps} discrete

steps per feature, we will require at least D_{steps}^{d} samples. The number of samples required to model the data accurately is in this case an exponential function of d.

In practice, pattern-recognition problems tend to have sufficient structure in the relationships between features to ensure that exponential growth does not occur. However, the existence of this upper bound must be kept in mind when analyzing data properties.

How do we decide which of the three relationships between N and d is most appropriate?

- A linear relationship can be tested by employing tests for multivariate normality and correlation.
- Quadratic relationships can be tested by testing for multivariate normality and the homogeneity of class covariance matrices.
- If the linear and quadratic relationships don't hold, an exponential relationship between N and d is possible.

3.3.2 MINIMUM NUMBER OF SAMPLES

After we have determined the relationship between d and N we need to quantify whether there are enough samples in the training set to model the structure of the data accurately. For each of the four relationships mentioned above, we define a measure (N_{min}), which sets the scale for the minimum number of samples that is required to model the data accurately.

If the data are normally distributed and uncorrelated, a linear relationship between d and N will exist and the minimum number of samples that are required will be in the order of

$$N_{l(\min)} = 2dC + C. \tag{3.1}$$

If the data are normally distributed, correlated and the classes have homogeneous covariance matrices, then a quadratic relationship will exist between d and N and the minimum number of samples that are required will be proportional to

$$N_{q1(\min)} = d^2 + dC + C. \tag{3.2}$$

If the data are normally distributed, correlated and the classes have non-homogeneous covariance matrices, then a quadratic relationship will exist between d and N and the minimum number of samples that are required will be on the order of

$$N_{q2(\min)} = Cd^2 + \ dC + C. \tag{3.3}$$

If the data are not normally distributed, an exponential relationship between d and N will be assumed and the number of samples that are required may be as plentiful as

$$N_{e(\min)} = D_{steps}^{d}, \tag{3.4}$$

where D_{steps} is the discrete number of steps per feature.

3.3.3 DATA SPARSENESS MEASURE

We will now quantify if the number of samples are sufficient to model the data accurately by defining a ratio between the actual number of samples and the minimum number of samples that are required. We define a measure of data sparsity as follows:

$$DSR = \frac{N}{N_{\min}}, \tag{3.5}$$

where N_{min} is the appropriate minimum number of samples measure and N the actual number of samples in the data set.

We also define a measure to indicate if the number of samples are sufficient by inverting equation (3.4) as follows:

$$DS = \sqrt[d]{N}, \tag{3.6}$$

where N is the number of samples in the data set and d the dimensionality of the data set.

3.4 STATISTICAL MEASURES

In this section we will propose statistical measures to measure the correlation between features, the multivariate normality of class-conditional probability density functions and the homogeneity of class covariance matrices.

3.4.1 CORRELATION

Correlation is a very important property in classifiers such as the NB and DT classifiers. The NB classifier assumes that all the variables in a data set are uncorrelated while the DT classifier only allows correlation between certain variables and assumes that other variables are uncorrelated. We will use the following measure (proposed by [2]) to quantify the correlation between features in a data set:

$$p = \frac{1}{T} \sum_{i=1}^{C} \sum_{j=1}^{d-1} \sum_{k=j+1}^{d} |p_{jk}|, \tag{3.7}$$

where $|p_{jk}|$ is the absolute value of the Pearson correlation coefficient between features j and k, T is the total number of correlation coefficients added together, C is the number of classes and d is the number of features.

The measure p is the average absolute correlation coefficient value between all variable pairs for all classes. This measure gives us an indication of the interdependence between all features and is strictly zero if all the features are uncorrelated and equal to unity if all the features are identical. Values of p close to unity indicate that features are highly correlated and suggest that there is redundant information since the correlated variables share similar information.

3.4.2 NORMALITY

Measures such as skewness and kurtosis are not robust in the sense that distributions exist that are incorrectly identified as normal distributions. These measures also do not provide any information on the rejection of a hypothesis of normality.

The BHEP test for multivariate normality is a robust test and has the following desirable properties [31, 32]:

- Affine invariant
- Consistent against non-normal distributions
- Can be applied to data sets of any size and dimensionality.

The BHEP test for multivariate normality (MVN) calculates a weighted L^2-distance between the true characteristic function of a normal distribution and the empirical characteristic function obtained from the data. The calculation of this measure is rather involved; we refer the reader to [31] for a full discussion of this test. We will use this weighted distance measure as a measure of normality and indicate it as *MVN*.

3.4.3 HOMOGENEITY OF COVARIANCE MATRICES

The geometric mean ratio between the pooled covariance matrix and the individual class covariance matrices can be used to evaluate the homogeneity of class covariance matrices. The individual class matrices can be tested for homogeneity by making use of Box's M test statistic [2]. The M test statistic is defined as:

$$M = \gamma \sum_{i=1}^{C} (n_i - 1) \log |S_i^{-1} S|, \tag{3.8}$$

where

$$\gamma = 1 - \frac{2d^2 + 3d - 1}{6\,(d+1)\,(C-1)} \left[\sum_i \frac{1}{n_i - 1} - \frac{1}{n - C} \right], \tag{3.9}$$

and n_i is the number of samples in class i, S is the pooled covariance matrix and S_i^{-1} is the inverse of the class covariance matrix of class i, d and C are the same as defined in Section 3.2.

The M statistic can be used in the following expression (proposed by [2]) to give a measure of homogeneity of the class covariance matrices:

$$SDR = \exp\left(\frac{M}{d \sum_{i=1}^{C} (n_i - 1)} \right). \tag{3.10}$$

The value of SDR is strictly equal to unity if all the class covariances are equal to the pooled covariance matrix, the value of SDR increases as the class covariances become more non-homogeneous.

3.5 INFORMATION THEORETIC MEASURES

It is important to note that the statistical measures that were discussed in the previous section are all based on the assumption that the features are continuous. Information measures are, however, suited for continuous and categorical variables [2].

The mutual information between classes and features, $M(C, X)$, can be used to determine the intrinsic dimensionality of a data set. We will measure how many features are not contributing significantly to classification by measuring the importance of features with their values of $M(C, X)$.

We calculate a cumulative distribution function of the mutual information between class and features (ordered from most to least significant) to determine how many features are required to represent 90% of the total mutual information between class and features. We define the intrinsic dimensionality as the number of features required to represent 90% of the mutual information between class and features. We denote this measure as ID and the ratio between ID and the true dimensionality as IDR.

If IDR is low (close to $1/d$) there are numerous redundant features, which may be caused by highly correlated features; this suggests that an eigenvalue transformation should be considered. If IDR is close to unity most features contain a significant amount of classification information and the classification problem is described well by the features.

3.6 DECISION BOUNDARY MEASURES

In this section we propose data measures that characterise the decision boundaries of classification problems.

3.6.1 LINEAR SEPARABILITY

We measure the linear separability of classification problems by employing a linear-discriminative classifier described in [20].

The linear discrimination function is a linear combination of the variables in a sample. An optimal hyperplane is selected to discriminate between data of different classes in a d-dimensional feature space. The linear discriminant rule maximises the distance between classes in a least-square sense by optimising the weight and bias terms with a sum-of-squares error function.

We use the 10-fold cross-validation error rate of this linear classifier as a measure of linear separability; we denote this error rate as $L1$.

3.6.2 VARIATION IN DECISION BOUNDARY COMPLEXITY

We have shown in Chapter 2 that a variation in decision boundary complexity between classes can influence classification performance; we will use the linear classifier error rates between the different class combinations of a data set to define a measure of variation in decision boundary complexity. We calculate the SD of the linear classifier error rates of all the class combinations with the maximum-likelihood estimates given by:

$$\hat{\mu}_e = \frac{1}{n}\sum_{i=1}^{n} e_i, \tag{3.11}$$

$$\sigma_e = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(e_i - \hat{\mu})^2}, \tag{3.12}$$

where e_i is the linear classification error rate of the i^{th} class combination, $\hat{\mu}_e$is the mean error rate of the n class combinations and σ_e is the SD of the error rates between all the class combinations.

We define σ_e as a measure of variation in decision boundary complexity and denote this measure as $L2$.

3.6.3 COMPLEXITY OF DECISION BOUNDARIES

We use an ϵ-neighbourhood pretopology approach proposed by [11, 33], to grow successive adherence subsets from points in each class. Each adherence subset is grown to the highest

order such that it includes only points of the same class.

A sample in each class is randomly selected and an Euclidean distance measure is used to compute the nearest neighbours of these points. If the nearest neighbour of a selected sample is of the same class it is included in the adherence subset; the next nearest neighbour of this centre is then calculated again. The adherence subset grows by repeating this process until a sample from a different class is encountered. The final result is that all samples are grouped into hyper-spheres that contain samples of the same class. A good topological description of a data set is given by the sizes and centres of all the retained adherence subsets.

The interleaving of retained adherence subsets of different classes gives us a good indication of the decision boundary complexity; we will make us of a minimum-spanning tree (MST) proposed by [11] to quantify the degree of interleaving between retained subsets. The MST connects all the samples in a data set to their nearest neighbour regardless of class. The connections can thus either be between samples of the same class or samples of different classes.

We can obtain a measure of interleaving between the retained subsets by employing the MST on the centres of these subsets and then counting the number of connections between centres of the same class and centres of different classes. The complexity of the decision boundaries can be measured by the amount of inter-class centre connections; we consequently define the following measure of decision boundary complexity:

$$DBC = \frac{N_{inter}}{N_{retained}} \quad , \tag{3.13}$$

where N_{inter} is the number of inter-class connections made by the MST on the retained subset centres and $N_{retained}$ is the total number of subsets retained by the ϵ-neighbourhoods algorithm.

3.7 TOPOLOGY MEASURES

In this section we will focus on measures that attempt to explain the topology of a data set; we will regard the retained subsets of the ϵ-neighbourhoods approach as hyper-spheres.

3.7.1 NUMBER OF GROUPS

The number of retained adherence subsets and the size of these subsets give us a good indication of whether data are clustered together in feature space or distributed in other more obscure structures. If data are clustered together, fewer subsets will be retained and subsets will have higher orders. We define the following measure to give us an indication of how much data are clustered together:

$$T1 = \frac{N_{retained}}{N}, \tag{3.14}$$

where $N_{retained}$ is the number of retained adherence subsets and N is the number of samples in the data set.

Measure $T1$ gives us an indication of how many groups per class occur in the data, since groups of data will be clustered together and each group will belong to a different hypersphere. We have shown in Section 2.4.5 that this is an important measure for the GMM classifier. This measure can also give us an indication of central tendency in data since the size of adherence subsets are larger for data with central tendency.

3.7.2 NUMBER OF SAMPLES PER GROUP

The number of samples in the retained adherence subsets gives us an indication of what the sizes of groups in the data are. The average size of these subsets can be seen as a measure of the average number of samples per group; we propose the following measure to give us an indication of the number of samples per group in a data set:

$$T_2 = \frac{1}{N_{retained}} \sum_{i=1}^{N_{retained}} S_i, \tag{3.15}$$

where $N_{retained}$ is the number of retained adherence subsets and S_i is the number of samples in adherence subset i.

3.7.3 VARIATION IN FEATURE SD

We have shown in Chapter 2 that the variation in SD of features in a class can give model-based classifiers an advantage over discriminative classifiers if the distribution of the data fits the model-based classifier assumptions. We will calculate the variation of feature SDs in a class by calculating the SD of the feature SDs for each class; we use the equations (3.11) and (3.12) to calculate these SDs. We denote this SD of feature SDs as measure $T3$.

3.7.4 SCALE VARIATION

We have shown in Chapter 2 that scale variation influences the classification performances of the MLP and SVM classifiers significantly. The scale of data in various parts of the feature space of a data set can be measured by the density of the retained hyper-spheres obtained by the pretopology ϵ-neighbourhoods approach. We define the density of a retained subset as follows:

$$\rho = \frac{N_{sphere}}{V_{sphere}}, \tag{3.16}$$

where N_{sphere} is the number of samples in a retained hyper-sphere and V_{sphere} is the volume of the retained hyper-sphere. The radius of the hyper-sphere is the Euclidean distance from the sphere centre to the furthest sample in the sphere.

If samples are far apart in feature space the hyper-spheres that encapsulate these samples will have a large volume and the densities of these spheres will be low. If samples are close together in feature-space the hyper-spheres containing these samples will have smaller volumes and the density of these spheres will be higher. The variation of the densities of these hyper-spheres can thus give us an indication of the scale differences in feature space.

We calculate the SD of the sphere densities of a data set to give us an indication of the variation in sphere density in a data set and consequently a measure of variation in scale through the feature space. The SD of sphere densities will give us a measure of both intra-class and inter-class scale variation. We will denote the SD of sphere densities as measure $T4$.

3.8 NOISE MEASURES

In Chapter 2 we identified and investigated three types of noise. We defined the overlap of class samples as a form of input noise, we defined incorrectly labelled samples as a form of output noise and we mentioned that the ten-dimensional data sets in experiment 4 contained a large proportion of features that didn't contribute to classification; we will call this phenomenon feature noise. In this section we will propose measures to measure each of these types of noise.

3.8.1 INPUT NOISE

To determine input noise we will determine the amount of overlap between features of different classes; we will follow an approach suggested by [11] with two slight variations - we will rotate the feature axes with an eigenvalue transformation and also consider the number of dimensions in which overlap occurs. The reason for the eigenvalue transformation is to decorrelate the data as much as possible since correlation can create the false impression that overlap between features exists (if only one feature is considered at a time).

The maximum and minimum values of a feature in each class are used to define boundaries for a feature; if the feature value of a sample lies in the boundaries of another class's feature values, then we will assume that this sample contributes to overlap in this specific feature. We will count for each sample in how many dimensions it overlaps and then normalise the total overlap with *Nd*. We will denote this measure of input noise as measure $N1$. The value of $N1$ will be unity if all the samples overlap in all dimensions with samples of different classes, conversely the value of $N1$ will be zero if none of the features of any sample overlaps with feature values of samples from different classes.

3.8.2 OUTPUT NOISE

To determine output noise we will use the nearest neighbour classification error rate. Incorrectly labelled samples will typically lie closer to samples from different groups than samples of their own group, this proximity of points in different classes will thus influence the error rate of the nearest neighbour classifier. We will use 10-fold cross-validation to find

an approximation to the true error rate of the nearest-neighbour classifier.

The nearest neighbour error rate will also give us an indication of the amount of input noise present in a data set, since samples in regions that are highly overlapped will also be misclassified more often. The nearest neighbour error rate can thus be seen as a measure of the sum of input and output noise present in the data. We will denote the nearest neighbour error rate as $N2$.

3.8.3 FEATURE NOISE

The intrinsic dimensionality measure that we proposed in Section 3.5 can be used to measure the proportion of features that don't contribute to classification. We propose the following measure as a measure of feature noise:

$$ID2 = \frac{d - ID}{d} \quad , \tag{3.17}$$

where d is the dimensionality of the data and ID is the intrinsic dimensionality measure.

3.9 SUMMARY OF MEASURES

Table 3.2 summarises the relationship between data measures proposed in this chapter and the properties of data they measure.

Table 3.2: *Summary of proposed data measures and corresponding data properties*

Data properties	**Measures**
Standard measures	
Dimensionality	d
Number of samples	N
Number of classes	C
Data sparseness measures	
Data sparseness ratio	DSR
Data sparseness	DS
Statistical measures	
Correlation of features	p
Multivariate normality	MVN
Homogeneity of class covariances	SDR
Information theoretic measures	
Intrinsic dimensionality	ID
Intrinsic dimensionality ratio	IDR
Decision boundary measures	
Linear separability	L1
Variation in decision boundary complexity	L2
Decision boundary complexity	DBC
Topology measures	
Normalised measure of groups per class	T1
Number of samples per group	T2
Variation in feature SD	T3
Variation in scale	T4
Noise measures	
Input noise	N1
Output noise	N2
Feature noise	ID2

Table 3.3 summarises the relationship between data measures proposed in this chapter and data properties investigated in Chapter 2. The data properties and measures are grouped according to the experiments in Sections 2.3.1 - 2.3.5.

Table 3.3: *Data properties and measures applicable to classification experiments*

Data properties	**Measures**
Experiment1	
Multivariate normality	MVN
Correlation of features	p
Variation in feature SD	T3
Data sparseness ratio	DS, DSR
Experiment 2	
Input noise	N1
Output noise	N2
Experiment 3	
Variation in scale	T4
Experiment 4	
Variation in scale	T4
Variation in decision boundary complexity	L2
Correlation of features	p
Intrinsic dimensionality	ID, IDR, ID2
Experiment 5	
Groups per class	T1
Samples per group	T2
Interleaving of groups of different classes	DBC
Variation in feature SD	T3

3.10 CONCLUSION

We identified properties of data that influence classifier performance in Chapter 2 and in this chapter we developed measures to measure each of these properties from data. It is clear that measuring and interpreting these properties is not trivial and in the next chapter we will evaluate the contribution of each of the measures proposed in this chapter to understanding classifier performance.

CHAPTER FOUR

ANALYSIS OF DATA MEASURES

4.1 INTRODUCTION

In Chapter 3 we have developed several measures to measure data properties that influence classifier performance. In this chapter we will use artificial data from the classification experiments in Chapter 2 and additional artificial data to evaluate the efficacy of the data measures proposed in the previous chapter.

4.2 EXPERIMENTAL DESIGN

In this section we will discuss the design of experiments that will be performed to evaluate the efficacy of the data measures. We will refer to the experiments in Chapter 2 as classification experiments and to the experiments in this chapter as measures experiments.

We will use the methods discussed in Section 2.2.1 to generate additional artificial data sets; the important properties of these artificial data sets are: (1) the distribution of the data, (2) the correlation between variables, (3) the relative sizes of the SD for different variables, (4) the degree of SD for each variable (input noise) and (5) the degree of output noise.

We will distinguish between two types of feature SDs: constant feature SD will refer to the case where the SDs of features in a class are similar, and varying SD will refer to the case where the SDs of features in a class are all significantly different. We will generate data sets with Gaussian, uniform and Gaussian mixture distributions.

4.2.1 MEASURES EXPERIMENT 1

We will evaluate the efficacy of the correlation (p), multivariate normality (MVN) and variation in feature SD ($T3$) measures in this experiment.

4.2.1.1 CORRELATION AND NORMALITY

We generate additional artificial data sets with uniform, Gaussian and Gaussian mixture model class-conditional probability density functions to investigate the correlation and multivariate normality measures. The attributes of these artificial sets of data are summarised in Table 4.1.

Table 4.1: *Attributes of artificial data sets 1-4*

Artificial set	Distribution	Correlation	SD type	SDs
1	Uniform	Uncorrelated	C	1-25
2	Uniform	Correlated	C	1-25
3	GMM	Uncorrelated	C	0.1-0.5
4	GMM	Correlated	C	0.1-0.5

We will compare the measurement values of p and *MVN* for these artificial data sets and the data sets used in classification experiment 1.

4.2.1.2 VARIATION IN FEATURE SD

We investigate the variation in feature SD measure by comparing the values of measurement $T3$ for: (1) the data used in classification experiment 1 (with varying feature SD), (2) uniformly distributed data with constant feature SD values and (3) Gaussian data with constant feature SD values. These artificial sets are summarised in Table 4.2.

Table 4.2: *Attributes of artificial data sets 5-6*

Artificial set	Distribution	Correlation	SD type	SDs
5	Gaussian	Uncorrelated	C	1-25
6	Gaussian	Correlated	C	1-25

4.2.2 MEASURES EXPERIMENT 2

We will investigate the input noise ($N2$), variation in feature SD ($T3$) and output noise ($N1$) data measures in this experiment.

4.2.2.1 INPUT NOISE

We investigate the relationship between the measure of input noise and feature SD by using artificial data sets used in classification experiment 2. We vary the feature SD values from 1-25.

4.2.2.2 OUTPUT NOISE

We illustrate the effect of output noise on the measure of output noise by using the data used in classification experiment 2 with varying degrees of output noise. We vary the percentage of output noise from 5-25%.

4.2.3 MEASURES EXPERIMENT 3

We will investigate the measures of linear separability ($L1$) and inter-class scale variation ($T4$) in this experiment.

4.2.3.1 LINEAR SEPARABILITY

We investigate the effectiveness of the linear separability measure by using the data sets in artificial set 5; we vary the degree of SD to probe the effect of overlap on the linear separability measure.

4.2.3.2 INTER-CLASS SCALE VARIATION

We analyse the measure of inter-class scale variation by making use of artificial set 5 and the artificial data used in classification experiment 3.

4.2.4 MEASURES EXPERIMENT 4

We will investigate the measures of variation in decision boundary complexity ($L2$), inter-class scale variation ($T4$), intra-class scale variation ($T4$) and feature noise ($ID2$) in this experiment.

4.2.4.1 VARIATION IN DECISION BOUNDARY COMPLEXITY AND INTER-CLASS SCALE VARIATION

To analyse the efficacy of the variation in decision boundary complexity measure and inter-class scale variation measure we will use artificial sets of data that were used in classification experiment 4. We will compare these measures for data with large-scale variations (artificial set 4.1) and large variation in decision boundary complexity (artificial set 4.4).

4.2.4.2 INTRA-CLASS SCALE VARIATION

To evaluate the efficacy of the variation in scale measure on data with intra-class scale variation we compare artificial set 4.4 (with only two classes) to artificial set 4.5. Artificial set 4.4 has no variation in scale and high variation in decision boundary complexity, while artificial set 4.5 has high variation in scale and almost no variation in decision boundary complexity.

4.2.4.3 FEATURE NOISE

We evaluate the efficacy of the measure of feature noise by using artificial sets 4.1, 4.4 and 4.5. The dimensionalities of these artificial sets will be increased from two to ten dimensions by adding additional features that don't contribute to classification performance. We compute the values of measure $ID2$ for dimensionalities from two to ten.

4.2.5 MEASURES EXPERIMENT 5

We will investigate the groups per class measure ($T1$) in this experiment. We will also show how this measure can be used to measure the interleaving of groups in a class.

4.2.5.1 GROUPS PER CLASS

We investigate the groups per class measure by making use of the GMM artificial data sets that were used in classification experiment 5; these data sets have 10, 50 and 100 groups per class.

4.2.5.2 INTERLEAVING OF GROUPS

We evaluate the efficacy of the measure ($T1$) to measure how much the groups of data (in this case Gaussian mixtures) are interleaved in a data set. We make use of the data used in classification experiment 5 as well as two additional artificial sets of data. These additional artificial sets are summarised in Table 4.3.

Table 4.3: *Attributes of artificial data sets 7-8*

Artificial set	Distribution	Correlation	SD type	SDs
7	GMM	Uncorrelated	C	1-10
8	GMM	Correlated	C	1-10

The GMM data sets that were used in classification experiment 5 are identical to these artificial sets of data – except for the selection of the group means. The group means of the data sets in classification experiment 5 were chosen randomly, whereas the groups in a class are selected close to the class mean for artificial sets 7 and 8 – the groups of data in these data sets are thus not as dispersed as the data sets that were used in classification experiment 5.

4.3 RESULTS

The results of the experiments that were designed in Section 4.2 are summarised in this section.

4.3.1 MEASURES EXPERIMENT 1

4.3.1.1 CORRELATION AND NORMALITY

The data measures obtained from the experiments in Section 4.2.1.1 are given in Figures 4.1 and 4.2.

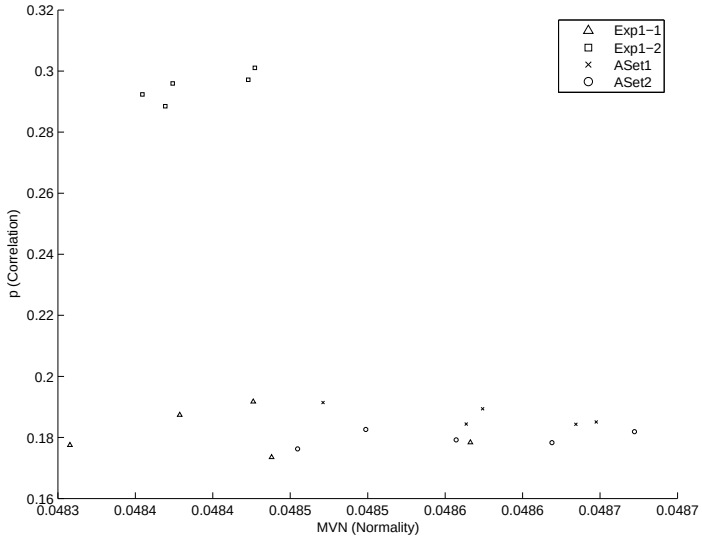

Figure 4.1: *Data measures of uniform and Gaussian data*

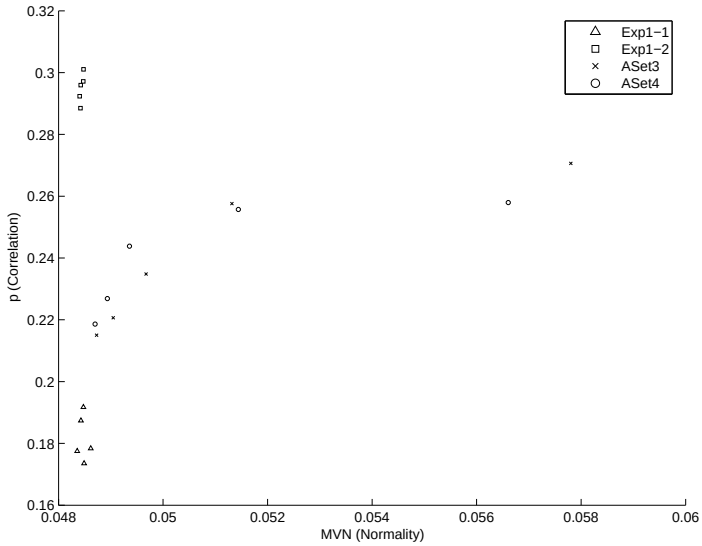

Figure 4.2: *Data measures of Gaussian and GMM data*

Figure 4.1 shows that the uncorrelated Gaussian distributed data used in classification experiment 1 (Exp1-1) and the similar correlated data (Exp1-2) are separated by the correlation measure p. It is interesting to observe that the p measure of the uncorre-

lated uniform data (ASet1) and correlated uniform data (ASet2) are close together – the correlated data sets have slightly higher p values. This might suggest that correlated and uncorrelated uniform data have very similar properties, when assessed with this measure.

We observe that the Gaussian distributed data sets (Exp1-1 and Exp1-2) are separated from the uniformly distributed data sets (ASet1 and ASet2) by the MVN measure; it is, however, interesting to note that the MVN measures of the uniform and Gaussian data are very close. This is reasonable given that Gaussian and uniformly distributed data have very similar properties such as short tails and no extreme outliers; this might be an important property for model-based classifiers that assume Gaussian distributed data.

Figure 4.2 shows that the correlations of the GMM distributed data sets (ASet3 and ASet4) are influenced by the SDs of the features in each group (the markers to the right have lower SDs than the markers to the left); this suggests that the correlation measure p is not invariant to SD for GMM distributed data. We also see that the p measures of the correlated and uncorrelated data GMM data sets are not clearly separated; this suggests that the correlation measure p is not invariant against distribution type - as we have seen in Figure 4.1.

We see that the MVN measures of the GMM data are in some instances very far from normality and in other instances very close to normality; mixture data with large standard deviations can be close to the normally distributed data by these measures. This phenomenon is encountered when the standard deviations of mixtures are large enough so that they overlap significantly; the shape of the distributions converges to a single mixture - the distribution is then similar to a Gaussian distribution. If the SDs of the mixtures are small, the mixture structure of the data is more prominent; this explains why the MVN measure moves away from normality as the SDs of the mixtures decrease.

4.3.1.2 VARIATION IN FEATURE SD

An important property of the artificial data used in classification experiment 1 is the variation in feature SDs of the Gaussian distributed classes. The SD values of the features were randomly chosen between zero and the specified SD value, consequently the data sets

have a large variation in feature SD.

This property is important since it gives the Gaussian and NB classifiers an advantage over other classifiers. Classification becomes more difficult for other classifiers (the decision boundaries become more complex and the data are sparse) while the classification difficulty remains the same for the Gaussian and NB classifiers since they can estimate the exact class conditional probability density functions (the properties of the data match their assumptions) with a small amount of data.

Figure 4.3 shows the results of the variation in feature SD measure ($T3$) for data sets with varying degrees of feature SD variation.

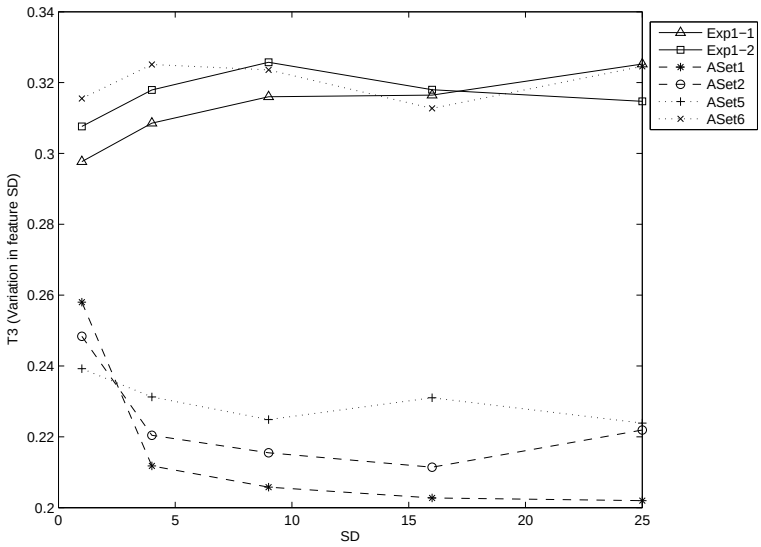

Figure 4.3: *Data measures of variation in feature SD*

We see that the values for $T3$ are small for correlated and uncorrelated uniformly distributed data (ASet1 and ASet2); this implies small variation in feature SDs. The artificial data sets ASet1 and ASet2 were generated with constant feature SD values and we see that the values of measure $T3$ are small for these data sets. This implies that measure $T3$ is able to measures the variation in feature SD accurately.

The interesting result is that of the correlated GMM data (ASet6), since a stretch matrix with equal values on the diagonal components was used to introduce SD into the features;

we thus expect no variation in feature SD. We did however multiply the stretched data with a rotation matrix (generated by the Gram Schmidt procedure) to introduce correlation. If we calculate the eigenvalues of the **A** matrix after rotation we will find that the eigenvalues of the resulting matrix generally differ from one another.

It seems as if the eigenvalues of the correlated Gaussian data covariance matrix are still very similar (ASet2) after rotation but the addition of mixtures causes the eigenvalues to vary more; the result is a variation in feature SD, which results in the high $T3$ measurement values for the data in ASet6. We see that the values of $T3$ are small for the uncorrelated GMM data (MData8), which is as expected, since all the feature SD values are equal.

4.3.2 MEASURES EXPERIMENT 2

4.3.2.1 INPUT NOISE

The results of the experiments explained in Section 4.2.2.1 are given in Figure 4.4. Figure 4.4 illustrates the relationship between the input noise measure ($N2$) and the SD values of features in a class.

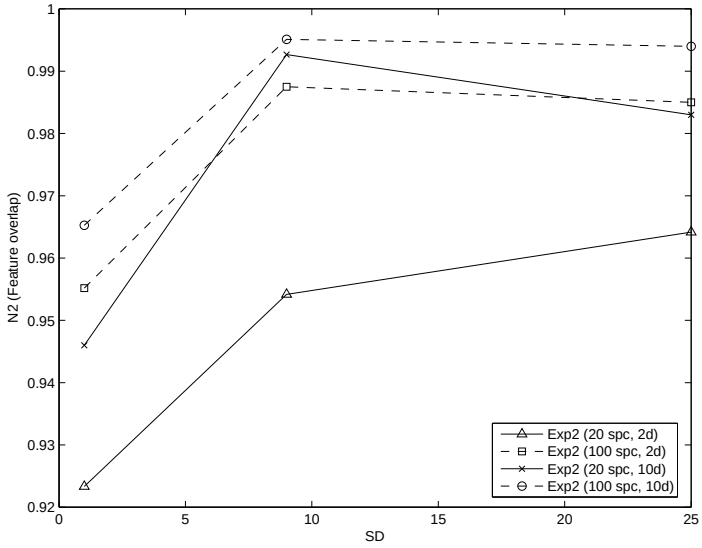

Figure 4.4: *Input noise data measures*

We see that there is a monotonic relationship between the SD of features and the measure $N2$ for SDs 1-9; as the SD becomes larger than 9 almost all of the samples are marked as overlapped for the ten-dimensional data (thus a $N2$ value close to unity). We can safely say

that the value of $N2$ increases monotonically with SD until it approaches the limit where all of the samples overlap in all feature dimensions.

We also note that the data sets with more samples per class have higher values for $N2$; this may be due to the fixed size of the feature space. The feature spaces for the 20 samples per class and 100 samples per class data are exactly the same size (all features are in the region [-1, 1]); if the number of samples increases and the volume of the feature space remains constant, then more overlap of samples will occur.

We finally observe that the $N2$ values for the ten-dimensional data are influenced less by the number of samples per class than the two-dimensional data; we might attribute this to the fact that the volume of a ten-dimensional feature space is much larger than the area of a two-dimensional feature space. The number of samples per class thus has a much bigger influence on the overlap of the two-dimensional data, since the area of the feature space is significantly smaller.

4.3.2.2 OUTPUT NOISE

Figure 4.5 illustrates the relationship between the measure of output noise ($N1$) and the percentage of incorrectly labelled samples in a data set (output noise).

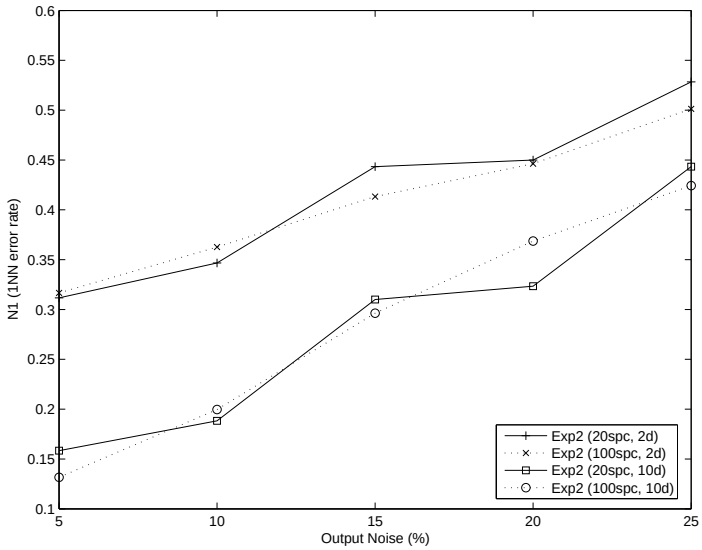

Figure 4.5: *Output noise data measures*

We see that measure $N1$ increases monotonically as output noise increases; the error rate of the nearest neighbour classifier thus increases in proportion to the number of switched class labels.

An interesting observation is that the error rates of the two-dimensional data sets are significantly more than the ten-dimensional data. Figure 4.5 shows that the overlap between features is significantly more for lower dimensional data; this increased overlap in lower dimensions (due to a smaller feature space) causes the error rates of the two-dimensional data to be higher than the error rates of the ten-dimensional data. The measure $N1$ is thus a measure of the total amount of noise, since it includes both the input noise and the output noise of a data set.

4.3.3 MEASURES EXPERIMENT 3

4.3.3.1 LINEAR SEPARABILITY

Figure 4.6 illustrates the relationship between the linear separability measure ($L1$) and the SD of features in a data set.

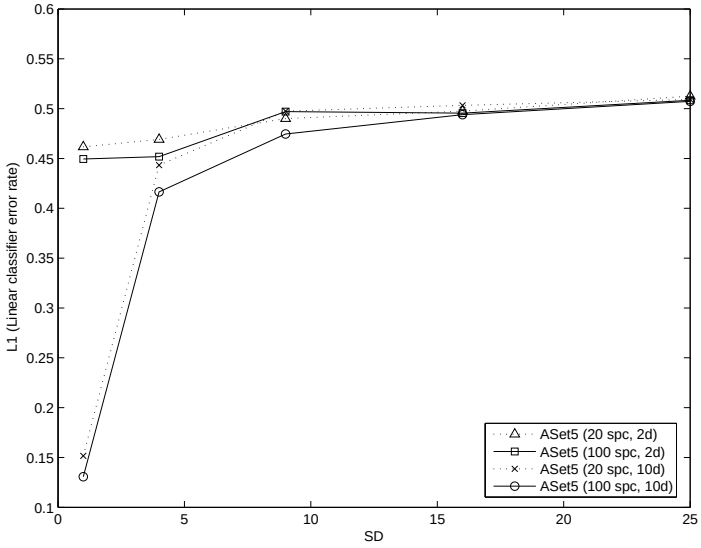

Figure 4.6: *Linear classifier error rate*

We see that the linear classifier error rate ($L1$) increases as the SDs of the data sets increase due to an increase in class overlap. We also note that the linear classifier error rates are not influenced significantly by the number of samples per class; the dimensionality, however,

influences the linear classification error rate to a great extent for SDs between 1 and 4. The linear classification error rate converges for all samples per class and all dimensionalities as the SD increases to 25. This convergence is caused by the class overlap that starts to permeate through the entire feature space as the SD increases; the noise caused by this overlap then tends to be more like a form of output noise. The linear classifier error rate successfully measures an increase in class overlap for SDs 1-4 - which is the region where class overlap is still at the decision boundaries. These results show that this measure can effectively measure input noise.

4.3.3.2 INTER-CLASS SCALE VARIATION

Figure 4.7 illustrates the values of the linear separability measure ($L1$) and the inter-class scale variation measure ($T4$) on the artificial data sets used in classification experiment 3 and ASet5.

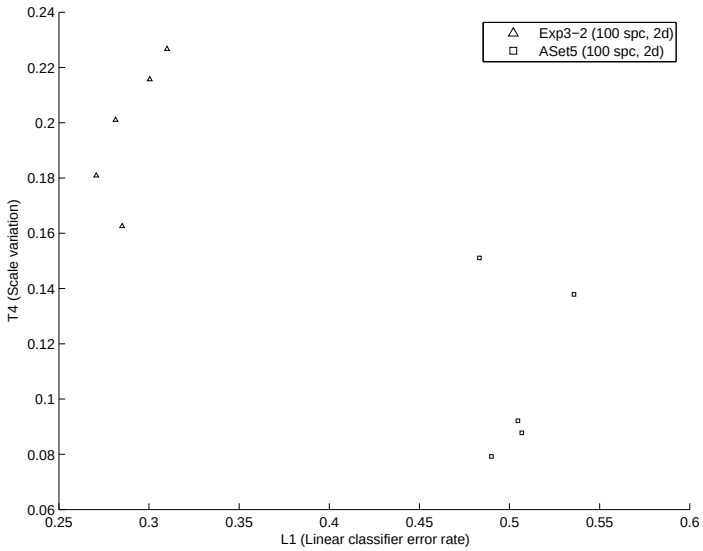

Figure 4.7: *Scale variation and linear separability data measures*

Measure $T4$ indicates that the data sets that were used in classification experiment 3 have high variations in scale; this measure is correct since the data sets were designed with classes that have large variations in feature SDs. Measure $T4$ has low values for the data sets used in ASet5, which is appropriate since these data sets were generated with equal SDs for all features.

We observe that measure $L1$ has low values for the data sets that were used in classification experiment 3; measure $L1$ thus suggests that the data sets are quite separable linearly – this is as expected since the classes were generated with a small degree of overlap (see Figure 2.9). Measure $L1$ has low values for the data sets in artificial set 5; this is appropriate since the data are two-dimensional, which leads to a high degree of overlap between the various classes in this case.

4.3.4 MEASURES EXPERIMENT 4

4.3.4.1 VARIATION IN DECISION BOUNDARY COMPLEXITY AND INTER-CLASS SCALE VARIATION

Figure 4.8 shows the measures of scale variation ($T4$) and variation in decision boundary complexity ($L2$) for the data sets that were employed in classification experiment 4.

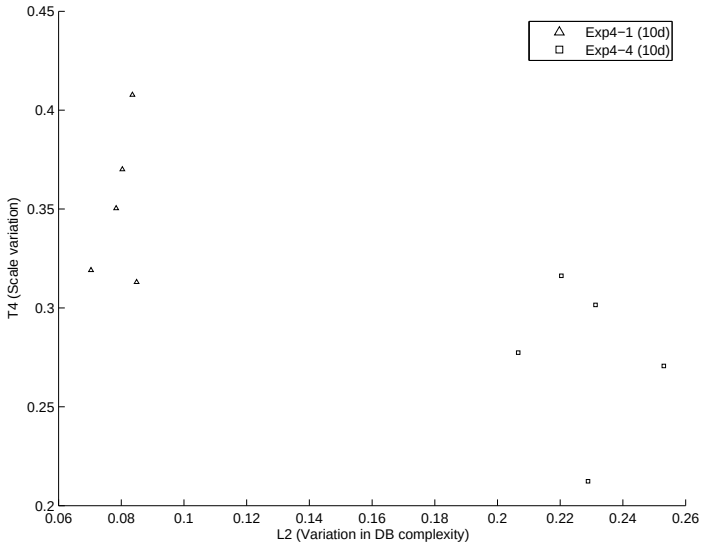

Figure 4.8: *Scale variation and variation in decision boundary complexity data measures*

Measure $T4$ indicates that all the data sets in artificial set 4.1 have relatively high variations in scale while all the data sets in artificial set 4.4 have relatively low variations in scale; this is as expected, since artificial set 4.1 has classes with varying degrees of SDs while artificial set 4.4 has classes with equal feature SDs.

Measure $L2$ indicates that the data sets in artificial set 4.1 all have relatively low variations in decision boundary complexities while the data sets in artificial set 4.4 all have relatively high

variations in decision boundary complexity. The data sets in artificial set 4.1 were generated in such a way that all the classes had the same degree of overlap; this is reflected in measure $L2$. The data sets in artificial set 4.4 were all generated with varying degrees of overlap between the various classes; measure $L2$ measures this relationship appropriately.

4.3.4.2 INTRA-CLASS SCALE VARIATION

Figure 4.9 shows the measures of scale variation ($T4$) and variation in decision boundary complexity ($L2$) for artificial sets 4.4 and 4.5. Note that a slight modification has been made to artificial set 4.4 – the classes were relabelled so that the data set contains only two classes.

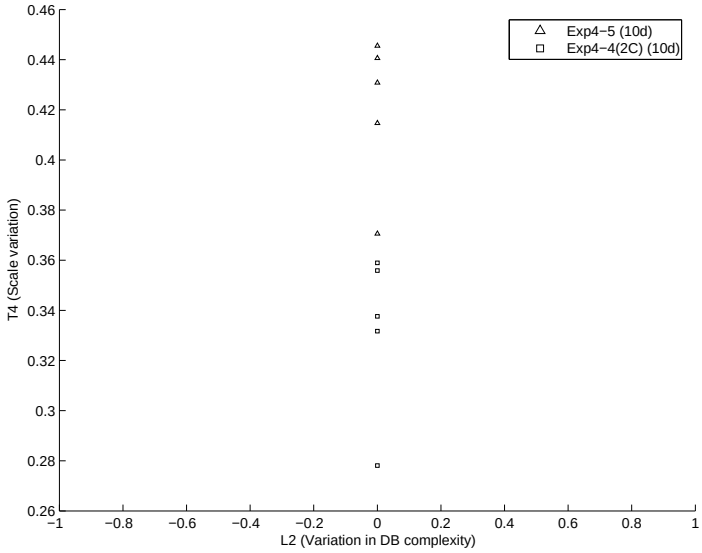

Figure 4.9: *Intra-class scale variation measures*

Measure $L2$ has values of zero for all the data sets since all the data sets used contain only two classes; there is thus only one decision boundary. We see that measure $T4$ predicts higher variations in scale for all the data sets in artificial set 4.5; this is as expected, since the data sets in artificial set 4.4 have Gaussian distributed classes with equal covariance matrices while the data in artificial set 4.5 have Gaussian distributed classes with varying feature SDs.

4.3.4.3 INTRINSIC DIMENSIONALITY

Figure 4.10 shows the measurement values of feature noise $ID2$ for varying dimensionalities.

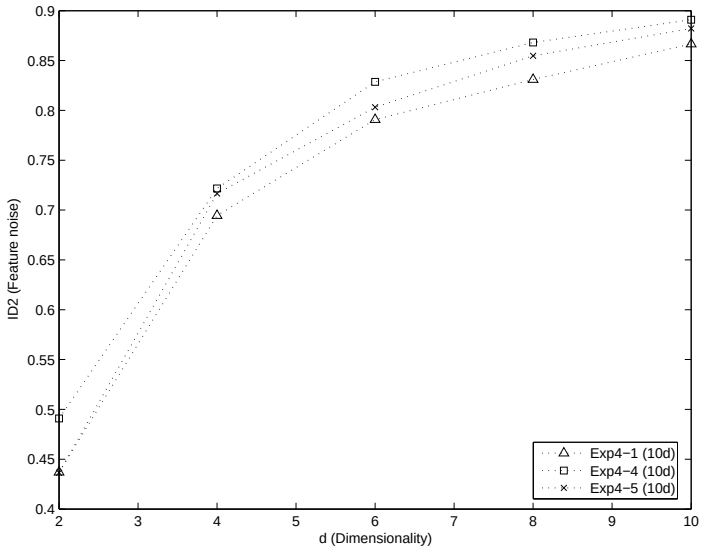

Figure 4.10: *Feature noise data measures*

We see that the predicted feature noise ($ID2$) increases monotonically as the dimensionality of the data increases. All the data sets were generated with two features that contribute to classification; the additional features were all noisy ones. Figure 4.10 thus shows that the feature noise is correctly measured by measure $ID2$.

4.3.5 MEASURES EXPERIMENT 5

4.3.5.1 GROUPS PER CLASS

Figure 4.11 shows the measurement values of the number of groups per class measure ($T1$) for varying feature SDs. We see that the data sets with the highest $T1$ values are the artificial data sets with 100 groups per class; these data sets were used in classification experiment 5. We see that the data sets with the lowest $T1$ values are the artificial sets with 10 groups per class. Figure 4.11 thus shows that the measure $T1$ is an effective measure of the number of groups per class. We also note that the measure $T1$ decreases as the SD increases; this can be attributed to an increase in overlap between groups - this overlap causes the groups of data to fuse together and form fewer but larger groups of data.

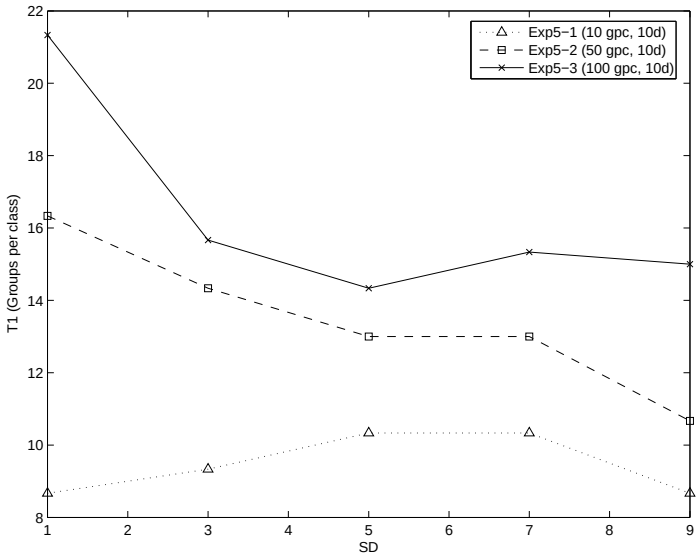

Figure 4.11: *Groups per class data measures*

4.3.5.2 INTERLEAVING OF GROUPS

Figure 4.12 shows the measurement values of the measurement $T1$ on two sets of data that were used in classification experiment 5 and two additional artificial sets that are summarised in Section 4.2.5.2.

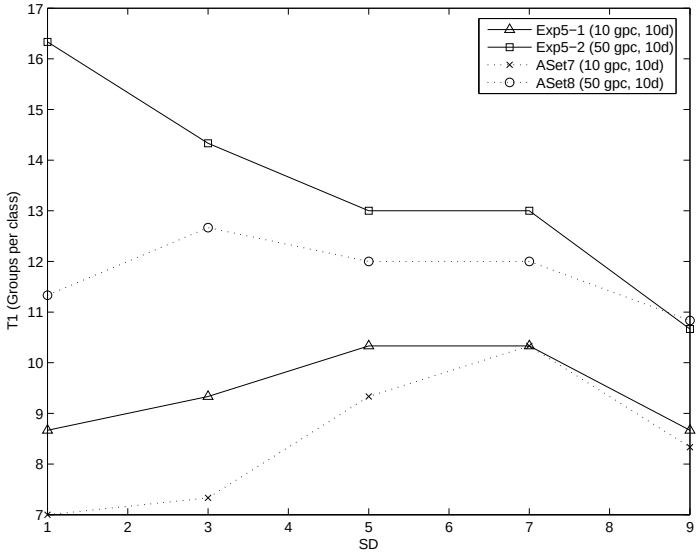

Figure 4.12: *Relationship between group interleaving and groups per class measure*

We see that the data sets that were employed in classification experiment 5 have higher values for $T1$; these data sets have random group means while the data sets from ASet7 and ASet8

have group means near the class means. The groups in the data sets from experiment 5 are thus more dispersed, which causes the mixture structure of the data to be more prominent. This shows that measure $T1$ also takes the degree to which groups in a class are interleaved into account; this is an important characteristic for the GMM classifier.

4.4 CONCLUSION

We have used artificial data from previous experiments (with very specific properties) as well as additional artificial data to probe the effectiveness of the data measures that were proposed in Chapter 3. We have shown that these data measures can successfully measure important data properties that were identified in Chapter 2. However, we have also seen that some of these measures are sensitive to factors that are incidental to their main focus – for example, the effect of dimensionality on feature overlap ($N1$).

We will use these measures to construct a meta-classification system in the next chapter, to see whether a combination of features can be constructed that predicts classifier performance in the face of such variability.

CHAPTER FIVE

META-CLASSIFICATION

5.1 INTRODUCTION

In this chapter we will construct a meta-classifier to predict the classification performances of ten real-world data sets. We will construct a meta-training data set by utilising artificial data sets with various data properties; we will then employ a nearest-neighbour classifier to find the most similar artificial data set to the real-world data sets. The reason for this methodology is twofold: (1) to predict the classification performances of real-world data sets by using the error rates of artificial data sets with similar data properties, and (2) to deduce information regarding the structures and properties of real-world data sets.

We will explain the construction of the meta-classification system in Section 5.2, and we will evaluate the efficacy of the meta-classifier in Section 5.3. We will explain the predictions of the meta-classifier in Section 5.4 by making use of data measures and we will conclude on our findings in Section 5.5.

5.2 CONSTRUCTION OF META-CLASSIFIER

The flow diagram in Figure 1 illustrates the process used to predict and evaluate the classification performances of real-world data sets.

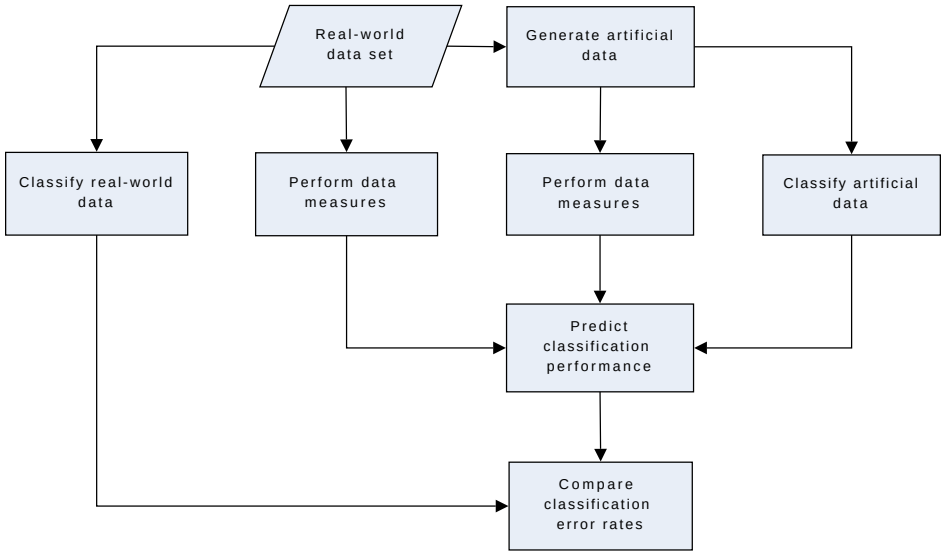

Figure 5.1: *Flow diagram of meta-classification system*

The data measures discussed in Chapter 3 are employed on a real-world data set and artificial data sets are generated with exactly the same dimensionality, number of samples and number of classes; these artificial data sets contain various data properties that were identified in Chapter 2. Data measures are employed on these artificial data sets and the 10-fold cross validation classification error rates of the artificial data sets are determined.

A weighted Euclidean distance is used to compare the data measures of the real-world data set to the data measures of the artificial data sets. The artificial data set closest to the real-world data set (in terms of Euclidean distance) is considered as the data set with the most similar data properties; the classification error rates of this artificial data set are used as the predicted error rates of the real-world data set.

The classification error rates of the real-world data set are estimated by performing 10-fold cross-validation; these error rates are used to evaluate the efficacy of the meta-classifier by comparing them to the predicted classification error rates.

We give a detailed discussion of the components in the meta-classifier in Sections 5.2.1 - 5.2.5.

5.2.1 DATA MEASURES

We will employ all of the data measures proposed in Chapter 3 (see Table 3.2) on ten real-world data sets and on artificial data sets that will be discussed next. These measures will be used to make predictions regarding the classification performance of the real-world data sets.

5.2.2 META-TRAINING DATA

We will construct a meta-training set by employing the above-mentioned data measures on the artificial sets described in sections 2.3 and 4.2. Versions of these artificial sets are generated for each real-world data set with exactly the same dimensionality (d), number of classes (C) and number of samples (N); by doing this we effectively ensure that all the artificial data measures are normalised in terms of these three parameters, which were seen to be potentially problematic in Chapter 4.

We create a meta-training set by using the measurement values of each artificial data set as input and the 10-fold cross-validation classification results as output. We will use this meta-training set to predict the classification error rates of the real-world data sets.

5.2.3 META-TESTING DATA

The meta classifier will be tested by employing the data measures on the ten real-world data sets. Each of the real-world data sets will produce an observation that can be classified by the meta-classifier. The 10-fold cross-validation error rates of the classifiers will be the desired outputs of these observations, which will be compared to the classification error rates predicted by the meta-classifier; these predictions will be discussed in more detail in section 5.3.

5.2.4 PREDICTIONS

Error rates will be predicted for each real-world data set by using a nearest-neighbour approach. The Euclidean distances between the real-world data set measures and each of the artificial data set measures are calculated in order to determine which artificial data set is

most similar (in terms of these measured data properties) to the real-world data set. The classification error rates of the most similar artificial data set will be used as the predicted classification error rates of the real-world data set.

5.2.5 META-CLASSIFIER PERFORMANCE MEASURE

We will evaluate the performance of the meta-classifier by calculating the Pearson correlation coefficient between the true error rates and predicted error rates of each real-world data set; these results will be discussed in the next section.

5.3 EVALUATION OF META-CLASSIFIER PERFORMANCE

The real-world data sets that will be used to evaluate the efficacy of the meta-classifier are summarised in Table 5.1. (We denote the number of numerical features as $d(Num)$ and the number of categorical features as $d(Cat)$.)

Table 5.1: *Summary of real-world data sets*

Data set	d(Num)	d(Cat)	d	N	C
Iris	4	-	4	150	4
Balance-scale	4	-	4	625	3
Diabetes	4	4	8	768	2
Tic-tac-toe	-	9	9	958	2
Heart	7	6	13	270	2
Australian	6	9	15	690	2
Vehicle	18	-	18	846	4
German	7	13	20	1000	2
Ionosphere	34	-	34	351	2
Sonar	60	-	60	208	2

These data sets were obtained from the UCI Machine Learning repository [17] - the Diabetes, Heart, Australian, Vehicle and German data sets were studied in the Statlog project [2].

5.3.1 REAL-WORLD CLASSIFICATION RESULTS

The classification results of the ten real-world data sets, when classified with several of the classifiers described previously, are given in Table 5.2.

Table 5.2: *Classification error rates of real-world data sets*

Data set	NB	Gauss	GMMd	GMMf	kNN	DT	SVM	MLP
Iris	0.0467	0.0200	0.0400	0.0333	0.0333	0.0600	0.0267	0.0400
Balance-s.	0.0960	0.0983	0.2720	0.0832	0.0976	0.2176	0.0000	0.0512
Diabetes	0.2422	0.2579	0.2566	0.2695	0.2500	0.2630	0.2305	0.2227
Tic-tac-toe	0.2265	0.3011	25.00	0.2140	0.0313	0.0438	0.0939	0.0167
Heart	0.1667	0.1704	0.1519	0.1814	0.1926	0.2037	0.1519	0.1667
Australian	0.2290	0.2103	0.1942	0.2029	0.1478	0.1507	0.1464	0.1217
Vehicle	0.5627	0.1451	0.5638	0.1525	0.2943	0.2731	0.1478	0.1690
German	0.2510	0.2890	0.3200	0.3220	0.2690	0.2600	0.2120	0.2490
Ionosphere	0.1738	0.0765	0.3589	0.3049	0.1311	0.1168	0.0884	0.0855
Sonar	0.3173	0.3500	0.1680	0.3269	0.1490	0.2933	0.2260	0.1490

We will compare these error rates to the predicted error rates to evaluate the performance of the meta-classifier.

5.3.2 WEIGHTED DATA MEASURES

We found that certain data measures are more informative when characterising data sets; these measures should thus be weighted more heavily when calculating the Euclidean distances between measurement observations.

We obtained measurement weights empirically by employing a hill climbing procedure with search directions parallel to the coordinate axes. All weights were initialised with values of one and the optimal weights were determined by iterating the weight of each measure from 1-10 while keeping the other measurement weights constant. The optimal weight of a measure was determined by evaluating the average correlation coefficient between the true and predicted classification error rates of the ten real-world data sets when using the specific weight value in the Euclidean distance measure. After an optimal weight was found for a measure, the optimal weight was fixed and used to determine the optimal weights of the remaining measures.

The most informative measures and their corresponding weights that were obtained by using this procedure are given in Table 5.3.

Table 5.3: *Most informative data measures*

Measure	Weight
p	10
$N1$	4
$T3$	4
MVN	4
$L1$	2
$T4$	2
$T2$	2
$ID2$	2

These measurements are not necessarily more efficient than the other proposed measures. They are more informative since the data properties that they measure influence classifier behaviour more than the other measured data properties; these weights thus give us insight into which data properties are more important in terms of classification.

5.3.3 NORMALISATION OF DATA MEASURES

We generate artificial data sets for each real-world data set with exactly the same dimensionality, number of classes and number of samples to construct a meta-training set for each real-world data set; this procedure ensures that all the data measures are normalised in terms of dimensionality, number of classes and data set size. All of the data measures are also designed to give values that don't exceed unity to a great extent. These normalisation procedures enable us to compare data measures of data sets with an Euclidean distance measure without bias.

5.3.4 META-CLASSIFIER PREDICTIONS

The artificial data sets with the most similar measurement values (in terms of weighted Euclidean distance) to each real-world data set are given in Table 5.4; these artificial sets were used to make classification predictions for the real-world data sets. We use the same names for the artificial sets in section 4.2 and we denote the artificial sets in Section 2.3 with the sub-experiment they belong to; we also denote groups per class with gpc.

An interesting observation is that seven of the ten data sets are judged to be closest to GMM distributed data and the other three data sets are closest to Gaussian distributions. This suggests that real-world data may have a tendency to be distributed in mixtures of Gaussians or in Gaussian distributed classes when processed with realistic feature-extraction algorithms. None of the nearest data sets contained uniform or Cauchy distributed data.

Table 5.4: *Nearest data sets*

Data set	**Artificial Set**
Iris	6
Balance-scale	Exp5 (100 gpc)
Diabetes	Exp5 (100 gpc)
Tic-Tac-Toe	Exp4-4 (2C)
Heart	Exp5 (100 gpc)
Australian	Exp5 (50 gpc)
Vehicle	6
German	Exp5 (10 gpc)
Ionosphere	Exp4-4 (6C)
Sonar	Exp5 (10 gpc)

The data properties of each of these artificial sets are summarised in Table 5.5. We denote variation in feature SD with V and no variation in feature SD with C.

Table 5.5: *Data properties of nearest artificial data sets*

Artificial set	**Distribution**	**Correlation**	**SD type**	**SD**
6	Gaussian	Correlated	C	1
Exp5 (100 gpc)	GMM	Uncorrelated	C	1
Exp5 (100 gpc)	GMM	Uncorrelated	V	5
Exp4-4 (2C)	GMM	Uncorrelated	V	0.3
Exp5 (100 gpc)	GMM	Uncorrelated	V	5
Exp5 (50 gpc)	GMM	Uncorrelated	V	0.8
6	Gaussian	Correlated	V	1
Exp5 (10 gpc)	GMM	Uncorrelated	C	2
Exp4-4 (6C)	Gaussian	Uncorrelated	V	0.3
Exp5 (10 gpc)	GMM	Uncorrelated	C	0.25

We see that nine of the ten nearest data sets in this study have SDs equal to or smaller than unity. Real-world data may thus tend to have standard deviations typically smaller or equal to unity, which implies that the conditioning of the effective covariance matrices is fairly

regular.

The predicted error rates that were obtained by using the error rates of these nearest artificial data sets are given in Table 5.6.

Table 5.6: *Predicted error rates of real-world data sets*

Data set	NB	Gauss	GMMd	GMMf	kNN	DT	SVM	MLP
Iris	0.1867	0.0400	0.1533	0.0400	0.1133	0.1667	0.0667	0.0667
Balance-s.	0.6333	0.6400	0.6083	0.6250	0.6117	0.6700	0.6250	0.6117
Diabetes	0.4838	0.5275	0.4925	0.5175	0.4975	0.5050	0.4788	0.4888
Tic-tac-toe	0.3052	0.3156	0.1885	0.1865	0.2063	0.1323	0.2531	0.1479
Heart	0.4500	0.4150	0.4100	0.3850	0.4550	0.5100	0.4350	0.4150
Australian	0.3786	0.4757	0.3271	0.2443	0.1414	0.3757	0.2671	0.3086
Vehicle	0.0248	0.0000	0.0248	0.0000	0.0142	0.1168	0.02005	0.0224
German	0.3600	0.4330	0.3560	0.3740	0.3690	0.4110	0.3390	0.3420
Ionosphere	0.1893	0.4567	0.1864	0.4522	0.4096	0.1808	0.1525	0.2175
Sonar	0.0150	0.3550	0.0000	0.0050	0.0000	0.1350	0.0000	0.0050

We need to compare these predicted error rates to the real-world data set classification error rates in order to evaluate the accuracy of these predictions; we will perform this evaluation in the next section.

5.3.5 EVALUATION OF PERFORMANCE

The Pearson correlation coefficients between the cross-validation classification error rates (Table 5.2) and the predicted error rates (Table 5.6) are calculated for each data set; the correlation coefficients give us an indication of how accurately the measurements can explain the behaviour of all the classifiers. We summarise these correlation coefficients in Figure 5.2.

We see in Figure 5.2 that the Tic-tac-toe data set is the only one with a negative correlation coefficient; this is to be expected, since this data set is the only one that contains only categorical features. Closer evaluation of the classification error rates in Table 5.2 reveal that the NB, Gaussian, GMMd and GMMf classifiers have very poor classification performance for this data set; this is due to the fact that these classifiers are not suited for categorical data.

All the data sets in the meta-training set contain continuous variables, which explains why the predictions of these error rates are not accurate.

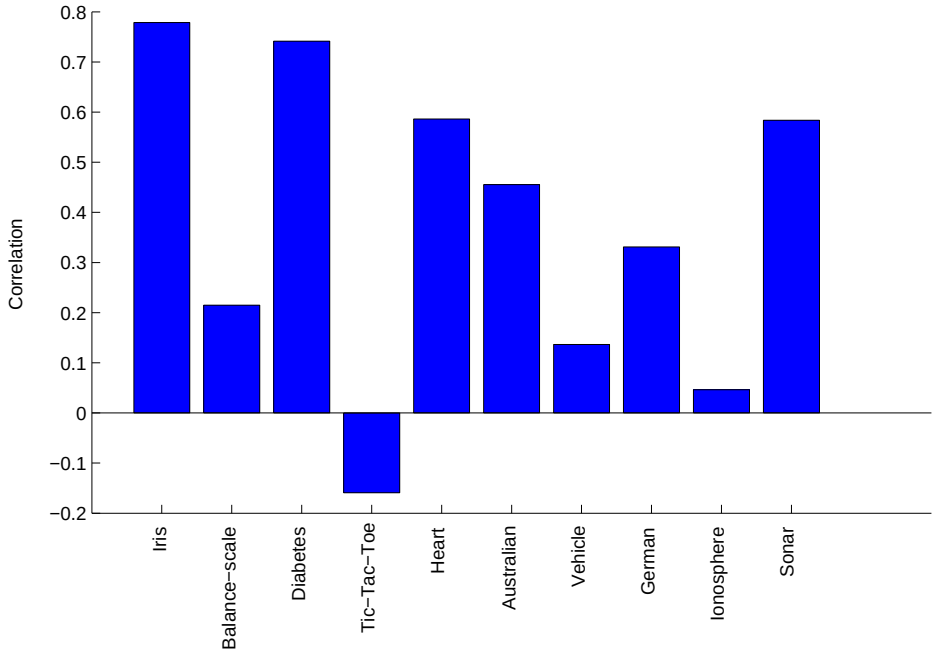

Figure 5.2: *Correlation coefficients of real-world data sets*

The Ionosphere data set has the lowest correlation coefficient of the non-categorical data sets. If we investigate the classification results and the predicted classification error rates more closely we find that the predicted error rate of the Gaussian classifier differs significantly from the 10-fold cross-validation error rate. If we calculate the correlation coefficient excluding the Gaussian classifier we obtain a correlation coefficient of 0.2861.

The two data sets with the highest correlation coefficients are the Iris and Diabetes data sets. The artificial data set nearest to the Iris data set has Gaussian distributed classes with feature SDs close to unity; the nearest data set to the Diabetes has GMM distributed classes with 100 groups per class with feature SDs between 0 and 5. What is interesting is that the Diabetes data set contains four numerical and four categorical features. If we evaluate the classification error rates we observe that these four categorical features do not influence the model-based classifiers too negatively compared to the discriminative classifiers; this explains why the correlation coefficient is still very good even though the data set contains categorical attributes.

The remaining data sets have correlation coefficients between 0.1364 and 0.5862; we will investigate some of these data sets further in the next section.

We calculate the correlation coefficients for each classifier across the ten real-world data sets to give us an indication of how well the data measures describe the properties of each classifier. The Pearson correlation coefficients between the predicted and 10-fold cross-validation classification error rates of each classifier are given in Figure 5.3.

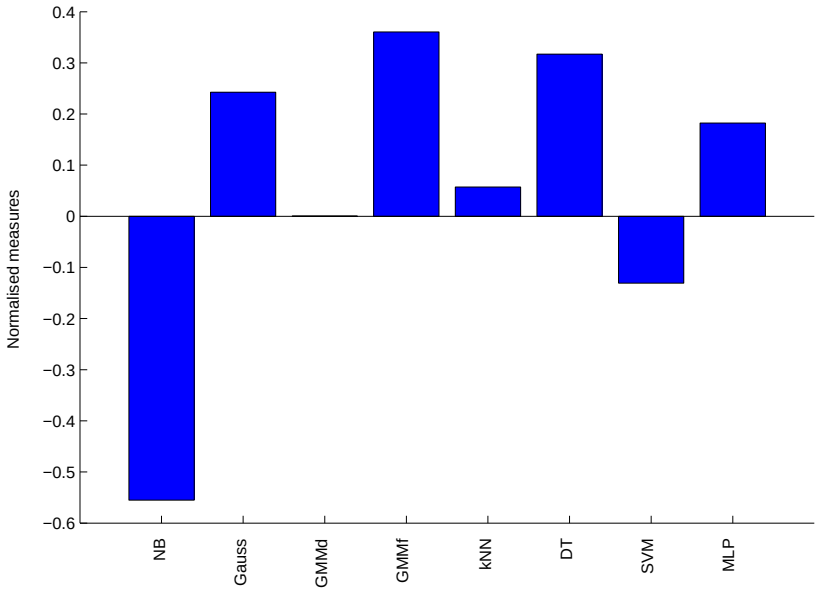

Figure 5.3: *Correlation coefficients of classifiers*

We see that the NB and SVM classifiers have negative correlation coefficients across all the real-world data sets, and that the observed correlation values are generally lower than the values across the different classifiers for a fixed data set. This suggests that our features are more successful in predicting the relative performance of different classifiers across the same data set than error rates across data sets. These results are not surprising in the light of the tremendous variability of data sets. Fortunately, the prediction of relative classifier performance is the more interesting task from a practical perspective.

We will evaluate the actual measurement values for several of the real-world data sets in the next section to gain insight into the predictions of the meta-classifier.

5.4 DISCUSSION OF PREDICTIONS

In this section we will explain the predictions of the meta-classifier by evaluating the normalised values of the most informative data measures. We will use the Iris, Diabetes, Heart, Tic-tac-toe and Ionosphere data sets to illustrate the relationships between these measures and classification performance.

5.4.1 NORMALISATION OF MEASURES

We need to normalise the obtained measurement values in order to compare these measures across real-world data sets with different sizes, dimensionalities and classes. We normalise each measure by dividing its values with the maximum value obtained from the meta-training set. Each measure is thus normalised relative to N, d and C and scaled in to the range [0, 1], where 0 will be the lowest value in the meta-training set for a measure and 1 will be the highest value for a measure in the meta-training set.

The only exception to this normalisation procedure is the MVN measure. We found that the normality of MVN measure is linearly correlated to the number of samples per class; as the number of samples increases (for any type of data distribution), the normality of the data increases (according to MVN) and the value of MVN consequently decreases (since smaller MVN values signify data closer to normality) - this is clearly not correct if the data are not normally distributed.

We normalise the MVN measures by multiplying the measurement values with the N/C ratio; after we normalise with this ratio the normality measure only increases with an increase in N if the data distribution is normal. Since the MVN measurement is invariant to dimensionality, we can compare these normalised MVN values across all the real-world data sets.

5.4.2 MEASUREMENT RESULTS

The most informative data measures (summarised in Table 5.4) that were obtained for each real-world data set are given in Figures 5.4 and 5.5. These measures are normalised as

discussed in the previous section.

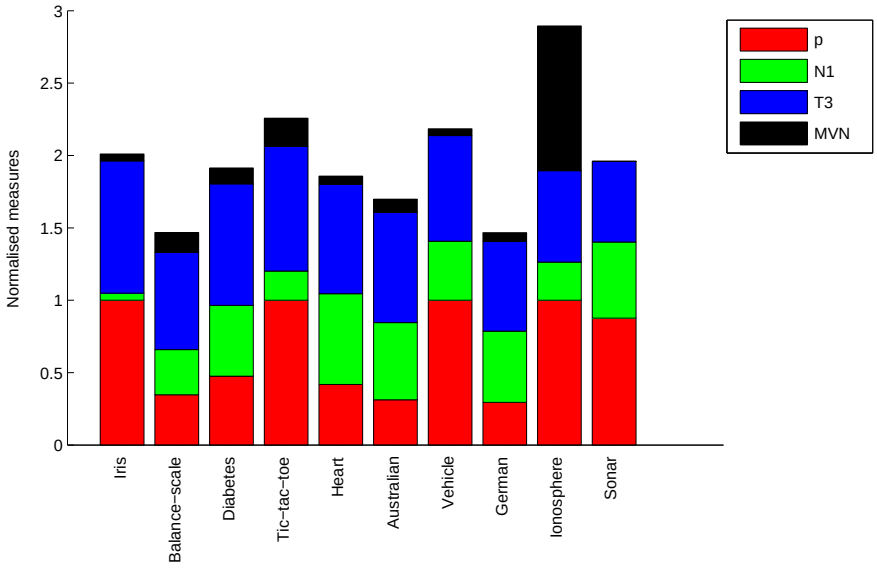

Figure 5.4: *Informative data measures of real-world data sets*

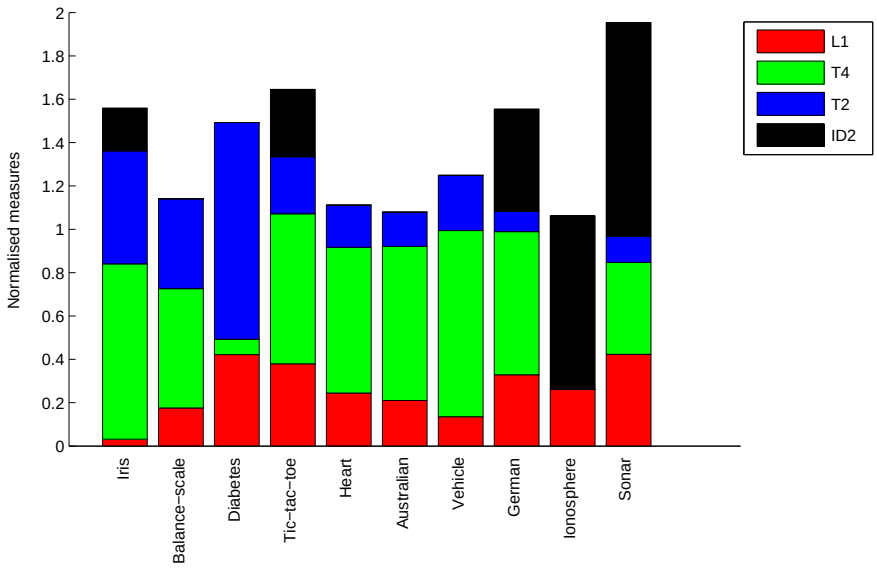

Figure 5.5: *Informative data measures of real-world data sets (continued)*

5.4.2.1 IRIS

We see in Figure 5.4 that the Iris data set has a high correlation measure value (p) and a low multivariate normality value (MVN), which indicates that the class-conditional probability density functions are close to normality. The nearest-neighbour error rate ($N1$) is very low,

which suggests that the data set contains very little input and output noise. This low measure of noise explains the low linear classification error rate ($L1$) that is shown in Figure 5.5. We also note that the variation in feature SD ($T3$) is relatively high for the Iris data set.

We see in Figure 5.5 that the Iris data set contains a relatively high proportion of features that don't contribute to classification ($ID2$) and the average sphere size ($T2$) is relatively high, which implies that the data has a high degree of central tendency, since larger adherence subsets are formed when data are clustered together.

The nearest data set to the Iris data set has correlated Gaussian class conditional probability density functions with similar feature SDs. We have explained in Section 4.3.1.2 that the eigenvalues of the correlated artificial data may, however, still differ significantly.

All of the measured data properties of the Iris data set are suited for any one of the classifiers in this study; this is verified by the good classification performance of all the classifiers on the Iris data set in Table 5.2.

5.4.2.2 DIABETES

Figure 5.4 shows that the Diabetes data set has relatively low correlation between features and the class-conditional probability density functions are close to multivariate normality (low (MVN) value). We see in Figure 5.5 that the average sphere size measure is very high; this implies a measure of central tendency in the data.

Table 5.2 shows that the classification error rate of the NB classifier is lower than all the classifiers except for the MLP and SVM classifiers; this shows that the assumptions of normality and uncorrelated features are valid ones for the Diabetes data set as indicated by the data measures. We see in Figure 5.5 that the variation in scale is very low; this explains the excellent classification performances of the MLP and SVM classifiers.

5.4.2.3 HEART

Figure 5.4 shows that the Heart data set has a relatively low measure of feature correlation and the small (MVN) measure indicates that the class-conditional probability density functions are close to normality. We also observe that the average sphere size measure is small, which implies that there are more than one retained hyper-sphere per class and thus more than one group per class in this data set.

We see in Table 5.2 that the NB and GMMd classifiers perform better than the Gaussian and GMMf classifiers - this suggests that the features are highly uncorrelated as measured. The classification performance of the GMMd classifier is better than the performance of the NB classifier; this verifies that there are more than one group per class in the data. The NB and the Gaussian classifiers perform better than the kNN and DT classifiers, which shows that the class-conditional probability density functions are close to normality even though there are several groups per class.

5.4.2.4 TIC-TAC-TOE

Figure 5.4 shows that the correlation between the variables in the Tic-tac-toe data set are very high; we also see that the feature noise is relatively high. This high feature noise may be a consequence of the high correlation between features.

Table 5.2 shows that the classification performance of the DT classifier is better than all the other classifiers except for the MLP and kNN classifiers; this good classification performance of the DT classifier may be attributed to the relatively high feature noise, since this feature noise does not affect its performance as much as the other classifiers.

5.4.2.5 IONOSPHERE

Figure 5.4 shows that the MVN measure of the Ionosphere data set is very high; this implies that the class-conditional probability density functions are not close to normality. We also see that the correlations between features are extremely high.

Figure 5.5 shows that the Ionosphere data set has virtually no scale variation and a very small average sphere size (this is because no adherence subsets were retained by the ϵ-neighbourhood pretopology algorithm). For an adherence subset to be retained, the subset must have at least more than $(N/C)/d$ samples. We can thus deduce that the Ionosphere data set consists of data that have almost no central tendency; the data points are thus scattered through feature space. We see that the measure of feature noise is very high, which is not surprising in view of the scattered nature of the data.

The low classification error rate of the Gaussian classifier given in Table 5.2 is extremely surprising, since the MVN and $T2$ measures imply that the data are not close to normality and that the data are scattered through the feature space. These results shows that there might be more complex relationships in the data that must still be investigated. We see, however, that the classification error rate of the NB classifier is significantly higher than that of the Gaussian classifier - this may be attributed to the high feature correlation as suggested by the correlation measure. We finally observe that the MLP and SVM classifiers perform very well, which might be attributed to the low variation in scale through the feature space.

5.5 CONCLUSION

We have illustrated how the data measures proposed in Chapter 2 can be employed to characterise a data set and how these data measures can be used to predict the classification performance of real-world data. Further, we have illustrated that these measures can give us valuable insights into the properties and structures of real-world data; these insights are extremely valuable in the case of high dimensional data.

Positive correlation coefficients were obtained between the true and predicted classification error rates of all the non-categorical real-world data sets. These results show that the meta-classifier captured important characteristics of the relationship between data and classifier performance. The performance of the meta-classifier across all real-world data sets for each classifier, however, suggests that further insight into the properties of data is required to fully describe the relationship between data characteristics and classifier performance.

CHAPTER SIX

CONCLUSION

6.1 INTRODUCTION

It is clear from this research that none of the current classifiers is optimal under all circumstances. Understanding the relationship between data characteristics and the performance of classifiers is therefore crucial to the selection of the optimal classifier for a classification task. We have investigated this relationship in our research and we summarise our findings in the next section.

6.2 SUMMARY OF WORK

We have shown in that "conventional wisdoms" regarding classification selection are not applicable to all types of data. We illustrated scenarios where model-based classifiers outperform discriminative classifiers significantly; similarly we have identified data properties that cause highly-rated discriminative classifiers to perform poorly in comparison with the other classifiers studied.

We used theoretical properties of classifiers to guide us in the development of data measures that describe the relationship between data characteristics and classifier performance, and we

constructed a meta-classification system using these measures to predict the classification performance of real-world classification tasks without training classifiers. The performance of this system shows that some important characteristics of the relationship between data and classifiers are successfully captured by these data measures - only one of the ten real-world data sets used in this study had a negative correlation between the true and predicted classification error rates.

We illustrated how the meta-classifier can be used to explain classification predictions of real-world data sets by evaluating the data measures. Evaluation of these measures gave us valuable insights into the properties and structures of the real-world data sets that were studied. We were able to quantify properties such as data sparseness, correlation between features, multivariate normality of class-conditional probability density functions, homogeneity of class covariance matrices, intrinsic dimensionality, variation in feature SD through feature space, input and output noise, linear separability, variation in scale through feature space, decision boundary complexity, variation in decision boundary complexity, significant number of groups per class and the proportion of noisy features.

We used artificial data sets to perform experiments under controlled circumstances. Control over the properties of data was extremely important, since the structure and properties of real-world data sets are very complex and we do not fully understand the relationship between data properties and how they influence classification performance. The artificial data sets assisted us in (1) probing the data properties that influence classifier performance, (2) developing and verifying data measures and (3) constructing meta-classification data sets.

6.3 FURTHER APPLICATION AND FUTURE WORK

We would like to address the following issues in our future endeavours:

- Creation of a theoretical framework that models the relationship between data characteristics and classifier performance. This framework will allow us to compare classifiers theoretically and explain classifier behaviour from a more fundamental perspective.

- Construction of hybrid classifiers using this theoretical framework. These hybrid classifiers will allow us to gain increased performance over existing classifiers.

- Generation of artificial data sets in a more systematic fashion to gain a more comprehensive view of the space of possible classification problems, and to assist us in improving the performance of the meta-classification system.

6.4 CONTRIBUTIONS AND SHORTCOMINGS

The contributions that were made by this research are summarised as follows:

- We have shown scenarios for which conventional wisdoms regarding the relative performance of various classification systems are inappropriate.

- We have identified new data characteristics that influence classifier performance. These include properties such as the spatial variability of the mean intra-class distance, which are seen to be quite important in practice and had not been described previously.

- We have developed novel data measures to measure these data characteristics and have investigated their performance as indicators of these data properties. Some of these measures function well in isolation, whereas others need to be normalised by other variables (such as the dimensionality of the feature space).

- We have developed a meta-classification system that describes important aspects of the relationships between data characteristics and classifier performance. This system allows us to gain insight into the properties and structures of real-world data and allows us to predict classification performance without training classifiers on the real-world data.

We have identified the following shortcomings in our research:

- The correlation values between the predicted and cross-validation classification error rates of some real-world data sets were very low. This implies that not all of the data characteristics that influence classification performance were captured by the employed data measures.

- The correlation values for fixed classifiers across all real-world data sets were generally lower than the observed correlation values across different classifiers for fixed data sets. This implies that the employed data measures did not capture all of the necessary data properties to characterise the behaviour of the classifiers that were studied.

These shortcomings are not surprising in the light of the tremendous variability of the real-world data sets that were used. Further insights into the relationship between data characteristics and the performance of classifiers are required to address these shortcomings; we will search for such insights in our future endeavours.

6.5 CONCLUSION

This research has shown that the optimal choice of classifier depends on the data set employed and that a true understanding of data characteristics and their influence on classification performance is required to select the optimal classifier for a classification task. We have developed data measures that characterise real-world classification problems and we have shown that these measures can successfully be employed to predict classification performance.

Classification has great theoretical interest and practical importance. This work has given new perspectives on classification, and we hope that this will lead to further progress in this field.

REFERENCES

[1] A.K. Jain, R.P.W. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4–37, 2000.

[2] D. Michie, D.J. Spiegelhalter, and C.C. Taylor, *Machine learning, neural and statistical classification*, Ellis Horwood Limited, Hemel Hempstead, 1994.

[3] R.P.W. Duin, "A note on comparing classifiers," *Pattern Recognition Letters*, vol. 17, no. 5, pp. 529–536, 1996.

[4] P.B. Brazdil, J. Gama, and B. Henery, "Characterizing the applicability of classification algorithms using meta-level learning," *in Proceedings of the European Conference on Machine Learning*, vol. 784, pp. 83–102, 1994.

[5] D.M.J. Tax and R.P.W. Duin, "Characterizing one-class datasets," *in Proceedings of the Sixteenth Annual Symposium of the Pattern Recognition Association of South Africa*, pp. 21–26, 2005.

[6] D.H. Wolpert, "The lack of a priori distinctions between learning algorithms," *Neural Computation*, vol. 8, no. 7, pp. 1341–1390, 1996.

[7] D.H. Wolpert, "The bayesian and computational learning theories," *NASA Ames Research Center, CA, MS 269-1*, Oct. 2000.

[8] C.J.C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[9] B. Pfahringer, H. Bensusan, and C. Giraud-Carrier, "Meta-learning by landmarking various learning algorithms," *in Proceedings of the Seventeenth International Conference on Machine Learning*, vol. 951, no. 2000, pp. 743–750, 2000.

[10] A. van den Bosch, "Wrapped progressive sampling search for optimizing learning algorithm parameters," *in Proceedings of the Sixteenth Belgian-Dutch Conference on Artificial Intelligence*, pp. 219–226, 2004. [Online]. Available: http://ilk.uvt.nl/ antalb/paramsearch. [Accessed: August 24, 2007].

[11] T.K. Ho and M. Basu, "Complexity measures of supervised classification problems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 289–300, 2002.

[12] J.R. Quinlan, *C4. 5: programs for machine learning*, Morgan Kaufmann Publishers, San Francisco, CA, 1993.

[13] P.B. Brazdil and C. Soares, "Zoomed ranking: Selection of classification algorithms based on relevant performance information," *in Proceedings of the Fourth European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 126–135, 2000.

[14] P.B. Brazdil, C. Soares, and J.P. da Costa, "Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results," *Machine Learning*, vol. 50, no. 3, pp. 251–277, 2003.

[15] I.H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann, San Francisco, 2nd ed., 1999.

[16] G. Lindner and R. Studer, "AST: Support for algorithm selection with a cbr approach," *in Proceedings of the Third European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 418–423, 1999.

[17] C.L. Blake and C.J. Merz, "UCI repository of machine learning databases," 1998. [Online]. Available: http://mlearn.ics.uci.edu/MLRepository. [Accessed: August 24, 2007].

[18] C. Giraud-Carrier and F. Provost, "Toward a justification of meta-learning: Is the no free lunch theorem a show-stopper?," *in Proceedings of the ICML-2005 Workshop on Meta-learning*, pp. 12–19, 2005.

[19] M. Nakazawa, T. Kohnosu, T. Matsushima, and S. Hirasawa, "On the complexity of hypothesis space and the sample complexity for machine learning," *in Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, vol. 1, pp. 132–137, 1994.

[20] A.R. Webb, *Statistical Pattern Recognition*, John Wiley, NJ, 2nd edition, 2000.

[21] C.M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, NY, 1995.

[22] J.G Proakis and M. Salehi, *Communication systems engineering*, Prentice-Hall Inc, NJ, 2nd edition, 2002.

[23] G.H. John and P. Langley, "Estimating continuous distributions in bayesian classifiers," *in Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pp. 338–345, 1995.

[24] D.W. Aha, D. Kibler, and M.K. Albert, "Instance-based learning algorithms," *Machine Learning*, vol. 6, no. 1, pp. 37–66, 1991.

[25] C. Chang and C. Lin, "LIBSVM: a library for support vector machines," 2001. [Online]. Available: http://www.csie.ntu.edu.tw/cjlin/libsvm. [Accessed: August 8, 2007].

[26] C.M. van der Walt, "Maximum likelihood gaussian classifier," 2007. [Online]. Available: http://www.patternrecognition.co.za. [Accessed: August 8, 2007].

[27] I. Nabney, *NETLAB: Algorithms for Pattern Recognitions*, Springer, 2002. [Online]. Available: http://www.ncrg.aston.ac.uk/netlab. [Accessed: August 8, 2007].

[28] J.H. Mathews and K.D. Fink, *Numerical Methods Using MATLAB*, Pretince-Hall, New Jersey, 3rd edition, chapter 8, pp. 401-405, 1999.

[29] R. Engels and C. Theusinger, "Using a data metric for preprocessing advice for data mining applications," *in Proceedings of the European Conference on Artificial Intelligence*, pp. 430–434, 1998.

[30] B.W. Silverman, *Density Estimation for Statistics and Data Analysis*, Chapman Hall, 1986.

[31] N. Henze and T. Wagner, "A new approach to the bhep tests for multivariate normality," *Journal of Multivariate Analysis*, vol. 62, no. 1, pp. 1–23, 1997.

[32] G.J. Szkely and M.L. Rizzo, "A new test for multivariate normality," *Journal of Multivariate Analysis*, vol. 93, no. 1, pp. 58–80, 2005.

[33] F. Lebourgeois and H. Emptoz, "Pretopological approach for supervised learning," *Proceedings of the Thirteenth International Conference on Pattern Recognition*, pp. 256–260, 1996.