CLASSIFICATION IN HIGH DIMENSIONAL FEATURE SPACES

HENDRIK OOSTEWALD VAN DYK

CLASSIFICATION IN HIGH DIMENSIONAL FEATURE SPACES

by

H.O. van Dyk

Student number: 21029288-2007

Dissertation submitted in fulfilment of the requirements for the degree

Master of Engineering

at the

Potchefstroom Campus

of the

North-West University

Supervisor: Professor E. Barnard

May 2009

KLASSIFISERING IN HOË-DIMENSIONELE KENMERK-RUIMTES

deur

H.O. van Dyk

Studente nommer: 21029288-2007

Verhandeling voorgelê vir die graad

Magister in Ingenieurswese

aan die

Potchefstroom kampus

van die

Noordwes-Universiteit

Studieleier: Professor E. Barnard

Mei 2009

SUMMARY

CLASSIFICATION IN HIGH DIMENSIONAL FEATURE SPACES

by

Hendrik Oostewald van Dyk

Supervisor: Professor E. Barnard

School of Electrical, Electronic and Computer Engineering

Masters in Engineering (Computer)

In this dissertation we developed theoretical models to analyse Gaussian and multinomial distributions. The analysis is focused on classification in high dimensional feature spaces and provides a basis for dealing with issues such as data sparsity and feature selection (for Gaussian and multinomial distributions, two frequently used models for high dimensional applications). A Naïve Bayesian philosophy is followed to deal with issues associated with the curse of dimensionality.

The core treatment on Gaussian and multinomial models consists of finding analytical expressions for classification error performances.

Exact analytical expressions were found for calculating error rates of binary class systems with Gaussian features of arbitrary dimensionality and using any type of quadratic decision boundary (except for degenerate paraboloidal boundaries).

Similarly, computationally inexpensive (and approximate) analytical error rate expressions were derived for classifiers with multinomial models.

Additional issues with regards to the curse of dimensionality that are specific to multinomial models (feature sparsity) were dealt with and tested on a text-based language identification problem for all eleven official languages of South Africa.

Keywords: naïve Bayesian, maximum likelihood, curse of dimensionality, Gaussian distribution, multinomial distribution, feature selection, data sparsity, chi-square variates, hyperboloidal decision boundaries.

OPSOMMING

KLASSIFISERING IN HOË-DIMENSIONELE KENMERK-RUIMTES

deur

Hendrik Oostewald van Dyk

Studieleier: Professor E. Barnard

Skool vir Elektriese, Elektroniese en Rekenaaringenieurswese

Magister in Ingenieurswese (Rekenaar)

In hierdie verhandeling ontwikkel ons teoretiese modelle om Gaussise en multinomiale distribusies te bestudeer. Die analise is gefokus op klassifiseerders in hoë dimensionele kenmerk-ruimtes en verteenwoordig 'n grondslag om probleme soos data-skaarsheid en kenmerk-seleksie aan te spreek (waarvan Gaussise en multinomiale distribusies baie populêr is in sekere toepassings). 'n Naïef-Bayes filosofie word gevolg om probleme op te los wat direk verband hou met die vloek van dimensionaliteit.

Die hoofdoel is om analitiese uitdrukkings vir klassifiserings fout-tempos te vind binne die konteks van Gaussise en multinomiale distribusies.

Onder andere het ons presiese uitdrukkings vir fout-tempos gevind wanneer binêre klassifiseerders met Gaussise distribusies gebruik word vir enige dimensie en vir enige kwadratiese beslissings-grens (behalwe vir die gedegenereerde paraboliese beslissings-grense).

Terselfdertyd het ons benaderde uitdrukkings vir die fout-tempos van klassifiseerders met multinomiale modelle gevind.

Tenslotte het ons ekstra teoretiese modelle ontwikkel om kenmerk-skaarsheid probleme op te los vir multinomiale distribusies (een van die probleme wat verband hou met die vloek van dimensionaliteit) en dit toegepas in 'n teksgebaseerde taal-klassifiserings toepassing waar al elf amptelike tale in Suid-Afrika gebruik word.

Kernwoorde: naïef Bayes, maksimum waarskynlikheid, vloek van dimensionaliteit, Gaussise distribusie, multinomiale distribusie, kenmerk-seleksie, data-skaarsheid, Chi-kwadraat veranderlikes, hiperboliese beslissings-grense.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

1.1 CONTEXT

In statistical pattern recognition we are generally concerned with classification problems where some decision boundaries need to be established in a given feature space (the vector space in which we are required to distinguish between patterns) to minimize classification error rates when unseen data (a test set) is provided for prediction.

In general, we distinguish between supervised and unsupervised learning. In supervised learning, we assume that the classification machine is provided with class labels during training, whereas for unsupervised learning no labeling information is provided for the machine to train on. Therefore, unsupervised learning typically requires that a machine should train to see differences in patterns and identify different clusters of vectors in the feature space.

In the context of supervised learning, there are generally two methods of classification, namely density estimation and discriminant analysis. In density estimation we are mainly concerned with estimating a probability density function (pdf) that describes the probability of occurrence of input feature vectors (for a given class) and use it to predict class probabilities (the probability that a given input vector comes from class c_1 for instance). In contrast, discriminant analysis focuses on finding an optimal decision boundary in the feature space to discriminate classes and involves the underlying pdfs indirectly or partially.

Furthermore, we can distinguish between parametric, non-parametric and semi-parametric classifiers. For parametric classifiers a given model (usually a mathematical distribution or a fixed type of decision boundary) is imposed on the the data with a limited number of parameters describing it. For instance, we could assume that the input data from a given class is described by a multivariate Gaussian distribution for which a limited number of parameters needs to be specified; the mean and covariance parameters μ and $\mathbf{\Sigma}$. Another example of a parametric classifier is a linear discriminant classifier where the decision boundary is assumed linear; in this case the parameters of the classifier are simply the components of the finite normal vector describing the linear hyperplane.

Non-parametric classifiers do not have a fixed number of parameters describing the data and grow larger (theoretically, without limit) with an increase in the number of training samples. Examples of purely non-parametric classifiers are methods such as the histogram, k-nearest-neighbor and kernel methods. Finally, the semi-parametric methods also have a growing number of parameters describing the present data set, but their extent is limited in some way and therefore does not grow without limit. Examples include Gaussian mixture models (GMM), where the number of mixture components is limited. Support vector machines (SVM) is another example of a semi-parametric kernel based classifier, where only a limited number of support vectors are used to describe the decision boundary.

It is well known that classifiers in general suffer from the curse of dimensionality. Issues that should be dealt with include compensating for training set sparsity (where the dimensionality of the feature space is very high relative to the number of training samples) and feature selection (where we identify and select a low dimensional subset of features that carry most of the discriminative information). It is easy to see that an increase in dimensionality will impact the reliability of pdf and discriminative boundary estimation due to the data sparsity issue. With an increase of dimensionality, the number of parameters that need to be estimated also increases and with a sparse training set it is very easy to overfit the data; effectively modeling the noise in the sparse data. Regularization (effectively penalizing the machine complexity) is one way of dealing with overfitting problems. Another way (which is the main focus of this dissertation) is to use inherently simple classifiers, such as naive Bayesian (NB) classifiers.

Recent years have seen a resurgence in the use of NB classifiers (Russell and Norvig, 1995; Botha *et al.*, 2006). These classifiers, which assume that all features are statistically independent, are particularly useful in high-dimensional feature spaces (where it is practically infeasible to estimate the correlations between features). Their newfound popularity stems from their use in tasks such as text processing, where such high-dimensional feature spaces arise very naturally. Consider, for example, a task such as text classification, where a natural feature set is the occurrence counts of the distinct words in a document. In this case, the number of dimensions equals the size of the dictionary, which typically contains tens of thousands of entries. Similarly, in text-based language identification (Botha *et al.*, 2006), n-gram frequency counts are frequently used for test vectors (where an n-gram is a sequence of n consecutive letters). High accuracies are achieved by using large values of n, thus creating feature spaces with millions of dimensions.

1.2 PROBLEM STATEMENT

A theoretical analysis of binary (two class) NB classifiers in high dimensional feature spaces and their performance is required in order to gain a better understanding of issues such as error performance and feature selection. Since this problem has not been explored extensively in the past, it is required to set down foundations for simple parametric classifiers with Gaussian and multinomial distributions. The Gaussian distribution is of interest due to its analytical simplicity and importance in statistics (for example multivariate hypothesis testing), while multinomial distributions are widely used in discrete practical applications involving high dimensional feature spaces.

1.3 OVERVIEW OF DISSERTATION

In this dissertation we develop theoretical models to analyse Gaussian and multinomial distributions. It is therefore divided into two parts:

- Classification with correlated Gaussian features (Chapters 3 and 4).
- Classification with high dimensional multinomial features (Chapters 5 and 6).

The analysis is focused on classification in high dimensional feature spaces and provides a basis for dealing with issues such as data sparsity and feature selection (for Gaussian and multinomial distributions, two frequently used models for high dimensional applications). We follow a naive Bayesian philosophy to deal with issues associated with the curse of dimensionality.

We derive analytical expressions for classification error rates in Gaussian and multinomial environments and deal with unique data sparsity issues (in addition to those that can be solved when using simple classifiers, such as NB classifiers) associated with the curse of dimensionality.

The remainder of this dissertation is divided into the following chapters:

- Chapter 2. In this Chapter we discuss relevant research in the literature that is essential to understanding the proposed work in this dissertation.
- Chapter 3. We derive exact analytical solutions for calculating error rates of binary classifiers with arbitrary dimensions given any quadratic decision boundary (except the degenerate paraboloidal boundaries).
- Chapter 4. We compare the analytical error expressions with estimates obtained from Monte-Carlo simulations for binary classifiers with artificially generated Gaussian features. We created both two and twelve dimensional classification problems and tested the error rates for two different decision boundaries: Optimal Bayes boundaries and NB boundaries.
- Chapter 5. We derive approximate analytical expressions for NB classifier error rates for high dimensional multinomial features. We also derive analytical methods for dealing with feature sparsity issues that arise in high dimensional spaces.
- Chapter 6. In this chapter we test the validity and accuracy of error estimates derived for an artificially generated multinomial data set and also investigate the effect that feature selection has on error performance. In addition, we test the theory on feature sparsity compensation on a text-based language identification problem on all eleven official languages of South Africa.
- Chapter 7. We discuss the main results obtained in this dissertation and also propose future research.

CHAPTER TWO

LITERATURE STUDY

2.1 GENERAL BACKGROUND

This section describes the general background required to understand the purpose of the proposed research.

2.1.1 HIGH DIMENSIONAL REGRESSION AND CLASSIFICATION

In applications involving regression or classification with spaces of high dimensionality, one of the most common problems in practice is the so-called curse of dimensionality. However, the main problem involves overfitting and one way to address it is to provide a number of training samples that is exponential in the number of input variables. This problem is easy to understand when we try to divide an n-dimensional feature space into n-dimensional hypercubes with a constant resolution in each dimension (Bishop, 2006). When we do this, the number of hypercubes is exponential in the number of dimensions. It is therefore clear that data sparsity becomes an issue and non-parametric techniques such as histogram methods become practically infeasible (Webb, 2002). In many applications, the dimensionality of the problem is inherently high and it is unrealistic to provide the required number of training samples to compensate for data sparsity.

Some examples where high dimensional feature spaces occur naturally are text-based applications such as topic identification (Rigouste *et al.*, 2005) and language identification (Botha *et al.*, 2006; Hakkinen and Tian, 2001).

2.1.2 REGULARISATION

In order to prevent overfitting, a form of regularisation is required. This means that when we are optimising a classifier, we need to penalise its complexity. Note that there are many ways of defining complexity. In model selection, different measures of complexity are relevant for different models - for example, the number of parameters that need to be estimated is relevant if classifier parameters act independently (Bishop, 2006). Another example would be when one has already selected a parametric

model and defines the complexity of a point estimate on the relevant parameters (such as the parameter vector size).

One way to deal with the problem of regularisation is to follow a Bayesian approach (see, for example Bishop (2006)). This can be illustrated with an example where we try to find the pdf that best describes a set of independent and identically distributed (i.i.d) data points. If we impose a parametric model onto the data (for example, we decide that the data are Gaussian), we have to estimate the parameters of the distribution (for a Gaussian distribution, this will be the mean and covariance matrix). The heart of the Bayesian approach is to assume a prior probability density function for these parameters (which is strictly independent of the data). After choosing such a prior, the posterior distribution of the parameters is inferred using the training data by means of Bayes' theorem. The argument of circumventing regularisation is that if you choose the prior probability correctly (for example, by understanding the context of the problem very well), then it is not necessary to regularise since this information is already incorporated in the prior. For a comprehensive analysis of this example (and many more), refer to Bishop (2006). The Bayesian approach is often criticised for being subjective (as opposed to the frequentist approach), since a prior distribution is strictly chosen before analysing the training data. Therefore, two different researchers can draw two different conclusions with the same set of data. This leads to the concept of using an uninformative prior. Unfortunately a poor selection of priors (including an uninformative prior) will necessarily lead to suboptimal performance. A related reason for criticising the Bayesian approach is that a conjugate prior is often selected purely for the sake of analytical simplicity.

For more than 100 years, there has been a debate between the so-called frequentist and Bayesian supporters. They form different schools in statistical thought and up to recent years the frequentist approach dominated the research community. The Bayesian approach has enjoyed considerable attention in recent years. One reason for this increase might be due to an increase in processing power, since the full Bayesian process requires marginalizing (sum or integrate) over parameters and often requires expensive sampling methods, such as Markov chain Monte Carlo (Bishop, 2006; Webb, 2002).

2.1.3 SUPPORT VECTOR MACHINES

Support vector machines (SVM) are kernel based classifiers, where the kernel function effectively transforms input vector spaces into (possibly) higher dimensional feature spaces (Burges, 1998). Simple radial base kernels, such as Gaussian kernels, transform the input vector space into an infinite dimensional feature space.

In the transformed feature space, the SVM classifier serves as a simple linear classifier that attempts to trade off apparent error rate performance with the size of the classification margin (Webb, 2002; Burges, 1998; Bartlett *et al.*, 2006; Moguerza and Munoz, 2006). This trade-off (margin size vs. apparent error rate) together with the chosen kernel and kernel width serves as a form of regularisation. It can be shown experimentally and intuitively that SVM classifiers perform very well in high dimensional feature spaces without a rigorous mathematical justification (Burges, 1998). One advantage of the SVM is that for a given kernel and error penalty factor (the two regularisation parameters), the optimisation problem is convex and can be optimised without the risk of achieving a suboptimal

local minimum. One disadvantage of the SVM is that it is a machine with a decision boundary and does not provide any posterior probabilities (as opposed to the relevance vector machine) (Bishop, 2006). Another problem is the lack of a tight theoretical error bound on the generalisation performance of SVMs.

To estimate the expected error performance of an SVM, it is required to use methods such as cross validation. These methods require a great deal of training time and are, in general, worth the time for tweaking kernels and error penalty factors. Since 2002, research has been done on optimising SVMs in the prime form (instead of the dual) and when only an approximate solution is required (for example, in cross validation), primal optimisation is superior and allows for fast training (Chapelle, 2007).

In extremely high dimensional problems, cross validation (or even training the classifier once) might not be feasible since SVM training time is done in - at best - O(ND) calculations, where D is the dimensionality of the input vectors and N is the number of training vectors (Burges, 1998). For some specific problems of high dimensionality, there are ways of training SVMs fast and efficiently. A good example of such a specialised case is when the individual input variables are predominantly zero in value (Joachims, 2006). However, many problems remain for which SVM training is simply not feasible for computational reasons.

2.1.4 FEATURE SELECTION

One method of countering the curse of dimensionality (which often leads to overfitting for a given model due to a high number of parameters) is to reduce the dimensionality at a preprocessing stage. In a high dimensional problem there will typically be many input variables that are redundant in the sense that they provide negligible discriminative value. If it is possible to find a way to remove these redundant variables, a classifier can be trained on low dimensional data (which provides parameter estimates with low variances) and the classifier's machine capacity (i.e. the number of parameters) will be reduced considerably (Webb, 2002; Bartlett *et al.*, 2006).

Feature selection requires the introduction of a separability measure that, in some monotonic fashion, describes the discriminative value between different classes when a given subset of input variables are used (Webb, 2002). The ideal separability measure would be the Bayes error rate (or minimum risk) in a classification problem. It is impossible to calculate the absolute minimum error rate unless the underlying distribution of the data is known exactly. In practice, it is only possible to get an estimate or an upper bound on the error performance. One way of estimating the error rate is to train a classifier and use its error rate as an estimate. Unfortunately, this requires training of the classifier every time the feature set changes and it is practically infeasible in high dimensional applications for complex classifiers.

Error analysis is not the only possible separability measure and there are many standard probabilistic dependent measures such as the Bhattacharyya and Chernoff measures (Webb, 2002).

If a proper separability measure is attained, it is still required to find the optimal set of input variables. This can be achieved by searching through all possible input variable sets (with a required dimensionality) and selecting the set that is optimal. A brute force method would be impractical and there are many methods of reducing the number of sets considered. The simplest of these are

the sequential forward and backward selection methods. These two methods do not compensate for correlation between variables and therefore more complex search strategies, such as the Branch and Bound and forward backward methods (Webb, 2002), are required. Another interesting search heuristic, called the multi-start fast forward backward selection (MS-FFWBW) heuristic, is introduced in (Boullé, 2007).

From the discussion above, it is clear that many sets of input variables have to be considered and therefore computationally inexpensive dissimilarity measures (or inexpensive classifiers such as NB classifiers) are required if feature selection is to be done in a realistic time frame.

2.1.5 NAIVE BAYESIAN CLASSIFIERS

The popularity of NB classifiers has increased in recent years (Russell and Norvig, 1995; Van Dyk and Barnard, 2007), because such classifiers are often the only option in high dimensional feature spaces. NB classifiers ignore all correlation between features and are inexpensive to use in high dimensional spaces where it becomes practically infeasible to estimate accurate correlation parameters. An attempt to estimate correlations can often lead to overfitting and decrease the performance (both efficiency and accuracy) of the classifier.

Empirical evidence collected over time suggests that NB classifiers perform well in general - surprisingly so in some cases, considering the fact that dependencies between variables are completely ignored (Hand and Yu, 2001). There is some experimental proof to show that NB classifiers can outperform some more complex classifiers when the dimensionality of the problem increases. One example is an experiment done by Russek *et al.* (1983) on predicting the set of heart disease patients that would die within six months. Studies showed that when only a subset of six variables are used, the predictions of an NB classifier is limited relative to more sophisticated classifiers. On the other hand when a set of 22 variables are considered, the NB classifier performs well. This problem can partly be explained by considering the bias and variance problem on the conditional distributions of the input vector space. Since the number of parameters to estimate is considerably smaller for NB classifiers (than one accounting for all correlation), the variance is low, but the bias will be higher in general (Hand and Yu, 2001). It is perfectly conceivable that in high dimensional spaces the decrease in variance outweighs the problem of bias, where with a more sophisticated classifier the variance would be too high to justify a low bias.

The practical popularity of NB classifiers has not been matched by a similar growth in theoretical understanding. Issues such as feature selection, compensation for training set sparsity and expected learning curves are not well understood for even the simplest cases.

It should also be mentioned that NB classifiers are sometimes confused with linear classifiers. An example of such a misunderstanding is pointed out by Hand and Yu (2001), where NB classifiers are mistaken for having linear decision boundaries (for example, refer to page 106 in Domingos and Pazzani (1997)). This is not true and can be illustrated with a simple example where two Gaussian distributed classes with different diagonal covariance matrices lead to a quadratic decision boundary in general. In fact, linear classification corresponds to equal class covariances, not independent features.

2.2 ESTIMATING ERROR CURVES

We are mainly interested in deriving error bounds or error estimates for binary classification problems with Gaussian and Multinomial features. These estimates will be used to gain valuable insight when applied to NB classification boundaries. In the following two sections we explain the literature relevant to estimating error probabilities in Gaussian and Multinomial environments.

2.2.1 ERROR ESTIMATES FOR BINARY CLASSIFIERS WITH MULTIVARIATE GAUSSIAN FEATURES

There are many sources in the literature that attempt to find measures for calculating error bounds on binary classification problems with multivariate Gaussian distributions. The complexity of such a derivation depends on the form of the decision boundary used, but in general it is not possible to calculate the exact error rate in a closed-form expression. Probably the simplest decision boundaries to work with are linear ones, were the exact error rate for any arbitrary linear bound can be expressed in terms of an error function.

In order to calculate the error performance of a binary NB classifier we turn to basic decision theory were we calculate an NB decision boundary that separates two hyperspace partitions Ω_1 and Ω_2. Whenever an observed feature vector falls within region Ω_1 or Ω_2, we classify the pattern to come from class ω_1 or ω_2 respectively. Therefore we can calculate the classification error rate by computing Eq. 2.1 (Webb, 2002)

$$\epsilon = p(\omega_1) \int_{\Omega_2} p(\mathbf{x}|\omega_1) d\mathbf{x} + p(\omega_2) \int_{\Omega_1} p(\mathbf{x}|\omega_2) d\mathbf{x}, \tag{2.1}$$

where ϵ is the classification error rate, $\mathbf{x}$ is the input vector and $p(\omega_1)$ and $p(\omega_2)$ are the prior probabilities for classes ω_1 and ω_2 respectively. Therefore, the very specific challenge to be addressed, is to calculate the integrals in Eq. 2.1, where $p(\mathbf{x}|\omega_1)$ and $p(\mathbf{x}|\omega_2)$ are correlated Gaussian distributions of arbitrary dimensionality. Since we are working with NB classifiers, the decision boundary will generally be a quadratic surface.

There exist many upper bounds on the Bayes error rate for Gaussian classification problems. Some popular loose bounds that can be calculated efficiently include the Chernoff bound (Chernoff, 1952) and the Bhattacharyya bound (Ito, 1972). Some tighter upper bounds include the equivocation bound (Hellman and Raviv, 1970), Bayesian distance bound (Devijver, 1974), sinusoidal bound (Hashlamoun *et al.*, 1994) and exponential bound (Avi-Itzhak and Diep, 1996). Unfortunately, none of these bounds are useful for the analysis of NB classifiers, since they obtain bounds for the Bayes error rate which do not allow us to investigate the effects of the assumption of no correlation. In order to investigate these effects, we choose to calculate an asymptotically exact error rate. The easiest way to do this, is to do Monte-Carlo simulations where we generate samples from the class distributions and simply count the errors; this is a time-consuming exercise, but does asymptotically converge to the true error rate. Instead, we will derive an exact analytical expression similar to work done by Press (1966) and Ayadi *et al.* (2008).

As proposed in the literature (Press, 1966; Ayadi *et al.*, 2008), we first transform the integrals in Eq. 2.1 into a problem of finding the cumulative distribution (cdf) of a linear combination of non-

central chi-square variates. Therefore we describe the different types of quadratic decision boundaries that can be obtained and describe the approximate solutions obtained in literature.

2.2.1.1 LINEAR COMBINATIONS OF NON-CENTRAL CHI-SQUARE VARIATES

Let us assume that $p(\mathbf{x}|\omega_1)$ and $p(\mathbf{x}|\omega_2)$ are both Gaussian distributions with means μ_1 and μ_2 and covariance matrices Σ_1 and Σ_2 respectively. Therefore

$$p(\mathbf{x}|\omega_i) = \frac{1}{(2\pi)^{D/2}|\mathbf{\Sigma_i}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu_\mathbf{i})^T\mathbf{\Sigma_i^{-1}}(\mathbf{x}-\mu_\mathbf{i})\right), \tag{2.2}$$

where D is the dimensionality of the problem. Unfortunately, the exact values for μ_i and Σ_i are almost never known and need to be estimated, with say $\hat{\mu}_\mathbf{i}$ and $\hat{\mathbf{\Sigma}}_\mathbf{i}$. For NB classifiers, $\hat{\mathbf{\Sigma}}_\mathbf{i}$ is a diagonal matrix. For simplicity we assume that $\hat{\mu}_\mathbf{i} = \mu_\mathbf{i}$ and $\hat{\mathbf{\Sigma}}_\mathbf{i} = \mathbf{\Sigma_i}$ – inaccuracy in estimating the sample means and covariances is best treated as a separate issue.

We can calculate the decision boundary for a binary classification problem. Eq. 2.3 is the simplest way to describe the decision boundary hyperplane in terms of the estimated parameters.

$$p(\omega_1)p(\mathbf{x}|\mu_\mathbf{1}, \mathbf{\Sigma_1}) = p(\omega_2)p(\mathbf{x}|\mu_\mathbf{2}, \mathbf{\Sigma_2}) \tag{2.3}$$

When we take the logarithm on both sides of Eq. 2.3 and use Eq. 2.2, we get the following representation for the decision boundary:

$$\beta_1(\mathbf{x}) = (\mathbf{x}-\mu_\mathbf{1})^T\mathbf{\Sigma_1^{-1}}(\mathbf{x}-\mu_\mathbf{1}) - (\mathbf{x}-\mu_\mathbf{2})^T\mathbf{\Sigma_2^{-1}}(\mathbf{x}-\mu_\mathbf{2}) = t_1, \tag{2.4}$$

where
$t_1 = \log\left(\frac{|\mathbf{\Sigma_2}|}{|\mathbf{\Sigma_1}|}\right) + 2\log\left(\frac{p(\omega_1)}{p(\omega_2)}\right)$.
In the context of Eq. 2.1, it is easy to see that

$$\int_{\Omega_2} p(\mathbf{x}|\omega_1)d\mathbf{x} = p\left(\beta_1(\mathbf{x}) \geq t_1\right), \tag{2.5}$$

where $\mathbf{x} \sim N(\mu_\mathbf{1}, \mathbf{\Sigma_1})$.

To derive a proper error estimate we follow a method similar to that proposed by Ayadi *et al.* (2008) by transforming Eq. 2.4 into a problem of solving for the cumulative distribution of a linear combination of non-central chi-square variates,

$$F(\Phi, \mathbf{m}, t) = p\left(\sum_{i=1}^{D}\phi_i(y_i - m_i)^2 \leq t\right), \tag{2.6}$$

where $\mathbf{y} \sim N(\mathbf{0}, \mathbf{I})$, $F(\Phi, \mathbf{m}, t)$ is a function that we can relate to the error performance, ϕ_i and m_i are variance and bias constants.

Solutions using this transformation are provided by Ayadi *et al.* (2008) for the special case where the optimal Bayesian decision boundaries are used. In this dissertation, we provide a general transformation that applies to all possible quadratic decision boundaries (optimal or not) and apply it to NB decision boundaries.

2.2.1.2 LINEAR DECISION BOUNDARIES

Linear boundaries are special realizations of quadratic boundaries that occur when the covariance matrices of two classes are exactly the same. NB boundaries assume that the covariance matrices are diagonal, but not equal. Therefore NB boundaries will lead to more general quadratic forms. Nonetheless, linear boundaries are often useful in applications involving sparse data sets. The error analysis for classification problems involving Gaussian features and linear boundaries are simple to perform and the integrals in Eq. 2.1 can be solved using only error functions (Ayadi *et al.*, 2008), where

$$Q(z) = \int_z^\infty \frac{1}{\sqrt{2\pi}} e^{-u^2/2} du \tag{2.7}$$

2.2.1.3 ELLIPSOIDAL DECISION BOUNDARIES

Error analysis for ellipsoidal decision boundaries can be transformed into a problem of calculating the cumulative distribution function of a linear sum of non-central chi-square variates discussed earlier where all coefficients are either positive or negative (Ayadi *et al.*, 2008). Therefore the problem can be expressed in a positive definite quadratic form and can be solved efficiently using theorem 1 (Ruben, 1962).

Theorem 1. (All ϕ_i positive) For $\mathbf{y} \sim N(\mathbf{0}, \mathbf{I})$ and $F(\Phi, \mathbf{m}, t)$ as defined in Eq. 2.6, we have

$$F(\Phi, \mathbf{m}, t) = \sum_{i=0}^{\infty} \alpha_i F_{D+2i}(t/p), \text{ if } \phi_i > 0 \ \ \forall i \in \{1, ..., D\},$$

where $F_n(x)$ is defined to be the cdf of a central chi-square distribution with n degrees of freedom, p is any constant satisfying

$$0 < p \leq \phi_i \quad \forall i \in \{1, ..., D\},$$

and α_i can be calculated with the recurrence relations

$$\begin{aligned}
\alpha_0 &= \exp\left(-\frac{1}{2}\sum_{j=1}^{D} m_j^2\right) \sqrt{\prod_{j=1}^{D} p/\phi_j} \\
\alpha_i &= \frac{1}{2i}\sum_{j=0}^{i-1} \alpha_j g_{i-j} \\
g_r &= \sum_{i=1}^{D} (1 - p/\phi_i)^r + rp \sum_{i=1}^{D} \frac{m_i^2}{\phi_i} (1 - p/\phi_i)^{r-1}
\end{aligned}$$

Also, the α coefficients above will always converge and

$$\sum_{i=0}^{\infty} \alpha_i = 1$$

Finally, a bound can be placed on the error from summing only k terms as follows:

$$\begin{aligned} 0 &\leq F(\Phi, \mathbf{m}, t) - \sum_{i=0}^{k-1} \alpha_i F_{D+2i}(t/p) \\ &\leq (1 - \sum_{i=1}^{k-1} \alpha_i) F_{D+2k}(t/p) \end{aligned}$$

Proof. Refer to Ruben (1962) for a proof.
For optimal convergence in the above series we select $p = \inf\{\phi_1, ..., \phi_D\}$, the largest possible value for p.

A useful recurrence relation for calculating $F_n(x)$ is as follows:

$$\begin{aligned} F_1(x) &= \text{erf}(\sqrt{x/2}) \\ F_2(x) &= 1 - \exp(-x/2) \\ F_{n+2}(x) &= F_n(x) - \frac{(x/2)^{n/2} e^{-x/2}}{\Gamma(n/2+1)} \end{aligned} \tag{2.8}$$

A similar derivation can be performed when all the ϕ_i values are negative (see section 3.2.2)

2.2.1.4 HYPERBOLOIDAL DECISION BOUNDARIES

Hyperboloidal decision boundaries occur most frequently in high dimensional spaces and error analysis can be transformed into a problem of calculating the cdf of a linear sum of non-central chi-square variates where some coefficients are positive and others negative (Ayadi *et al.*, 2008). Although much research has been done on solving the definite quadratic form (as for the elliptic boundary discussed above), finding an exact analytical expression for this indefinite quadratic form has been unsuccessful (see Press (1966); Ayadi *et al.* (2008); Shah (1963); Raphaeli (1996)). The existing solutions all lead to estimates, bounds or unwieldy solutions (and unusable for NB error analysis).

The basic method used to solve the indefinite form is to group all the positive and negative ϕ_i terms together (see Eq. 2.6) and calculate the cdf as follows:

$$\begin{aligned} F(\Phi, \mathbf{m}, t) &= p\left(\sum_{i=1}^{d_1} \phi'_i (y_i - m'_i)^2 - \sum_{j=1}^{d_2} \phi^*_j (y_{d_1+j} - m^*_j)^2 \leq t\right), \\ \phi'_i, \phi^*_j > 0 &\qquad \forall i \in \{1, ..., d_1\}, \forall j \in \{1, ..., d_2\}, \end{aligned}$$

where $d_1 + d_2 = D$ and $\Phi = (\phi'_1, \phi'_2, ..., -\phi^*_1, -\phi^*_2, ...)$.

This cumulative probability can be expressed as follows (Ayadi *et al.*, 2008):

$$F(\Phi, \mathbf{m}, t) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \alpha'_i \alpha^*_j \int_0^{\infty} \int_0^{(t+u)/p_1} \frac{1}{p_2} f_{d_1+2i}(v) f_{d_2+2j}(u/p_2) dv du, \tag{2.9}$$

where $f_n(x)$ is defined to be the pdf of a central chi-square variate with n degrees of freedom. We cal-

culate the α'_i and α^*_j coefficients by applying theorem 1 to $F(\Phi', \mathbf{m}', t)$ and $F(\Phi^*, \mathbf{m}^*, t)$ respectively (p_1 and p_2 are also calculated from theorem 1).

The literature provides no analytical solution to the double integral in Eq. 2.9 and approximations can be found in Ayadi *et al.* (2008) and Press (1966). Numerical integration is also infeasible due to the improper nature of the integral. Therefore, one of the main technical challenges in this dissertation is to provide an exact analytical solution to the above double integral.

2.2.1.5 CYLINDRICAL DECISION BOUNDARIES

For cylindrical decision boundaries, some of the ϕ_i terms in Eq. 2.6 are zero. It is easy to see that these terms can be removed from the equation entirely and one therefore reduces the dimensionality of the problem, without incurring any problems.

2.2.1.6 PARABOLOIDAL DECISION BOUNDARIES

Error analysis with paraboloidal decision boundaries can be transformed into the following quadratic form:

$$F(\Phi, \mathbf{m}, t) = p\left(\sum_{i\in I}\phi_i(y_i - m_i)^2 + \sum_{j\in J}\phi_j y_j \leq t\right), \quad I \cap J = \emptyset, \tag{2.10}$$

where $I \subset \{1, 2, ..., D\}$ and $J \subset \{1, 2, ..., D\}$. Therefore, paraboloidal decision boundaries are a degenerate case where some of the y_i terms have only a linear term and not a quadratic term. This problem can be seen as a limiting case of either the ellipsoidal or hyperboloidal case by simply adding a small $\delta\phi_i$ value to the linear terms (Ayadi *et al.*, 2008). Unfortunately, an exact solution to this problem does not yet exist.

2.2.2 ERROR ESTIMATES FOR BINARY CLASSIFIERS WITH MULTINOMIAL FEATURES

Very little research has been done on accurate error estimates for NB classifiers with multinomial features, even though there are many high dimensional practical applications where such an analysis would be useful (especially in text-processing) (Botha *et al.*, 2006; Hakkinen and Tian, 2001; Rigouste *et al.*, 2005). One of the main objectives of this dissertation is to derive such an error estimate that can for example be used in feature selection.

CHAPTER THREE

NAIVE BAYESIAN CLASSIFIERS WITH CORRELATED GAUSSIAN FEATURES: A THEORETICAL APPROACH

The main contribution of this chapter is that we are able to derive exact analytic expressions for the NB error rates for correlated Gaussian features (of arbitrary dimensionality) with quadratic decision boundaries in general, whereas previous authors were able to do so only in terms of computationally expensive series expansions (Shah, 1963) or imprecise approximations (Ayadi *et al.*, 2008).

The rest of this chapter is organized as follows. In Section 3.1, we derive the equations needed to transform the classification problem into one represented as a linear combination of chi-square variates. In Section 3.2, we discuss all possible quadratic decision boundaries obtained in the context of the work done in Section 3.1 and we show the exact solution to the cdf for most of these boundaries. Finally, in Section 3.3 we provide a proof for one of the theorems presented in Section 3.2.

None of the theory developed in Sections 3.1 and 3.2 is limited to NB classifiers and applies to quadratic discriminant analysis (QDA) in general. To be concrete, Sections 3.1 and 3.2 focus on methods for calculating $\int_{\Omega_2} p(\mathbf{x}|\omega_1)d\mathbf{x}$ in Eq. 2.1. It is easy to calculate $\int_{\Omega_1} p(\mathbf{x}|\omega_2)d\mathbf{x}$ by simply reversing the roles of ω_1 and ω_2.

3.1 LINEAR COMBINATIONS OF NON-CENTRAL CHI-SQUARE VARIATES

Let us assume that $p(\mathbf{x}|\omega_1)$ and $p(\mathbf{x}|\omega_2)$ are both Gaussian distributions with means μ_1 and μ_2 and covariance matrices Σ_1 and Σ_2 respectively (see Eq. 2.2).

Unfortunately, the exact values for μ_i and Σ_i are almost never known and need to be estimated, with say $\hat{\mu}_\mathbf{i}$ and $\hat{\mathbf{\Sigma}}_\mathbf{i}$. For NB classifiers, $\hat{\mathbf{\Sigma}}_\mathbf{i}$ is a diagonal matrix. For simplicity we assume that $\hat{\mu}_\mathbf{1} = \mu_\mathbf{1}$ and $\hat{\mu}_\mathbf{2} = \mu_\mathbf{2}$ – inaccuracy in estimating the sample means is best treated as a separate issue.

To revisit Chapter 2, we can use the parameter estimates to calculate the decision boundary for a binary classification problem. As already discussed, Eq. 2.3 is the simplest way to describe the decision boundary hyperplane in terms of the estimated parameters and is repeated here for convenience.

$$p(\omega_1)p(\mathbf{x}|\hat{\mu}_\mathbf{1}, \hat{\mathbf{\Sigma}}_\mathbf{1}) = p(\omega_2)p(\mathbf{x}|\hat{\mu}_\mathbf{2}, \hat{\mathbf{\Sigma}}_\mathbf{2}) \tag{3.1}$$

When we take the logarithm on both sides of Eq. 3.1 and use Eq. 2.2, we get the following representation for the decision boundary:

$$\beta_1(\mathbf{x}) = (\mathbf{x} - \hat{\mu}_\mathbf{1})^T \hat{\mathbf{\Sigma}}_\mathbf{1}^{-1}(\mathbf{x} - \hat{\mu}_\mathbf{1}) - (\mathbf{x} - \hat{\mu}_\mathbf{2})^T \hat{\mathbf{\Sigma}}_\mathbf{2}^{-1}(\mathbf{x} - \hat{\mu}_\mathbf{2}) = t_1, \tag{3.2}$$

where
$t_1 = \log\left(\frac{|\hat{\mathbf{\Sigma}}_\mathbf{2}|}{|\hat{\mathbf{\Sigma}}_\mathbf{1}|}\right) + 2\log\left(\frac{p(\omega_1)}{p(\omega_2)}\right)$.
In the context of Eq. 2.1, it is easy to see that

$$\int_{\Omega_2} p(\mathbf{x}|\omega_1)d\mathbf{x} = p\left(\beta_1(\mathbf{x}) \geq t_1\right), \tag{3.3}$$

where $\mathbf{x} \sim N(\mu_\mathbf{1}, \mathbf{\Sigma_1})$.

In the rest of this section, we focus our efforts on transforming Eq. 3.2 into a much more usable form,

$$F(\Phi, \mathbf{m}, t) = p\left(\sum_{i=1}^{D} \phi_i(y_i - m_i)^2 \leq t\right), \tag{3.4}$$

where $\mathbf{y} \sim N(\mathbf{0}, \mathbf{I})$, $F(\Phi, \mathbf{m}, t)$ is a function that we can relate to the error (see Section 3.2), ϕ_i and m_i are variance and bias constants. We do the transformation in four steps as follows.

3.1.1 SHIFT MEANS BY $\mu_\mathbf{1}$

We define $\mathbf{z} = \mathbf{x} - \mu_\mathbf{1}$ and with a little manipulation (and assuming $\hat{\mu}_\mathbf{1} = \mu_\mathbf{1}$ and $\hat{\mu}_\mathbf{2} = \mu_\mathbf{2}$) we can rewrite Eq. 3.2 as follows:

$$\begin{aligned}
\beta_2(\mathbf{z}) &= \mathbf{z}^T\mathbf{B_1}\mathbf{z} - 2\mathbf{b_1^T}\mathbf{z} = t_2 \\
\mathbf{B_1} &= \hat{\mathbf{\Sigma}}_\mathbf{1}^{-1} - \hat{\mathbf{\Sigma}}_\mathbf{2}^{-1} \\
\mathbf{b_1^T} &= (\mu_\mathbf{1} - \mu_\mathbf{2})^T\hat{\mathbf{\Sigma}}_\mathbf{2}^{-1} \\
t_2 &= t_1 + (\mu_\mathbf{1} - \mu_\mathbf{2})^T\hat{\mathbf{\Sigma}}_\mathbf{2}^{-1}(\mu_\mathbf{1} - \mu_\mathbf{2}) \\
\mathbf{z} &\sim N(\mathbf{0}, \mathbf{\Sigma_1})
\end{aligned} \tag{3.5}$$

Note that $\mathbf{B_1}$ is in general not a positive-definite matrix, but is symmetric and can be diagonalised.

3.1.2 ROTATE MATRICES TO DIAGONALISE $\mathbf{\Sigma_1}$

Since $\mathbf{z}$ is centered at the origin, we can rotate $\mathbf{\Sigma_1}$ to be diagonal, as long as we rotate the decision boundary too. We define $\mathbf{v} = \mathbf{U}_{\omega_\mathbf{1}}^\mathbf{T}\mathbf{z}$, where $\mathbf{U}_{\omega_\mathbf{1}}$ is the eigenvector matrix of $\mathbf{\Sigma_1}$ satisfying

$\mathbf{U}_{\omega_1}^{\mathbf{T}} \mathbf{\Sigma_1} \mathbf{U}_{\omega_1} = \mathbf{\Lambda}_{\omega_1}$,
$\mathbf{\Lambda}_{\omega_1} = \text{diag}(\lambda_{\omega_1,1}, ...\lambda_{\omega_1,D})$,
where $\lambda_{\omega_1,1}, ..., \lambda_{\omega_1,D}$ are the eigenvalues of $\mathbf{\Sigma_1}$. From this we can derive Eq. 3.6.

$$\begin{aligned} \beta_3(\mathbf{v}) &= \mathbf{v}^T \mathbf{B_2} \mathbf{v} - 2\mathbf{b_2^T} \mathbf{v} = t_2 \\ \mathbf{B_2} &= \mathbf{U}_{\omega_1}^{\mathbf{T}} (\hat{\mathbf{\Sigma}}_{\mathbf{1}}^{-\mathbf{1}} - \hat{\mathbf{\Sigma}}_{\mathbf{2}}^{-\mathbf{1}}) \mathbf{U}_{\omega_1} \\ \mathbf{b_2^T} &= (\mu_1 - \mu_2)^T \hat{\mathbf{\Sigma}}_{\mathbf{2}}^{-\mathbf{1}} \mathbf{U}_{\omega_1} \\ \mathbf{v} &\sim N(\mathbf{0}, \mathbf{\Lambda}_{\omega_1}) \end{aligned} \tag{3.6}$$

3.1.3 SCALE DIMENSIONS TO NORMALIZE ALL VARIANCES IN $\mathbf{\Sigma_1}$

We assume that $\mathbf{\Lambda}_{\omega_1}$ is positive definite and therefore none of the eigenvalues are zero. If some of the eigenvalues are zero, the dimensionality of the problem can either be reduced or the classification problem is trivial (if ω_2 has a variance in this dimension or a different mean). (Of course, an NB classifier may not be responsive to this state of affairs, and therefore perform sub-optimally. However, we do not consider this degenerate special case below.)

We define $\mathbf{u} = \mathbf{\Lambda}_{\omega_1}^{-\mathbf{1/2}} \mathbf{v}$ and derive Eq. 3.7.

$$\begin{aligned} \beta_4(\mathbf{u}) &= \mathbf{u}^T \mathbf{B} \mathbf{u} - 2\mathbf{b_3^T} \mathbf{u} = t_2 \\ \mathbf{B} &= \mathbf{\Lambda}_{\omega_1}^{\mathbf{1/2}} \mathbf{U}_{\omega_1}^{\mathbf{T}} (\hat{\mathbf{\Sigma}}_{\mathbf{1}}^{-\mathbf{1}} - \hat{\mathbf{\Sigma}}_{\mathbf{2}}^{-\mathbf{1}}) \mathbf{U}_{\omega_1} \mathbf{\Lambda}_{\omega_1}^{\mathbf{1/2}} \\ \mathbf{b_3^T} &= (\mu_1 - \mu_2)^T \hat{\mathbf{\Sigma}}_{\mathbf{2}}^{-\mathbf{1}} \mathbf{U}_{\omega_1} \mathbf{\Lambda}_{\omega_1}^{\mathbf{1/2}} \\ \mathbf{u} &\sim N(\mathbf{0}, \mathbf{I}) \end{aligned} \tag{3.7}$$

3.1.4 ROTATE MATRICES TO DIAGONALIZE THE QUADRATIC BOUNDARY

Now that $\mathbf{u}$ is normally distributed with mean $\mathbf{0}$ and covariance $\mathbf{I}$, it is possible to rotate $\mathbf{B}$ until it is diagonal without inducing any correlation between random variates. Therefore, we define $\mathbf{U_B}$ and $\mathbf{\Lambda_B}$ to be the eigenvector matrix and diagonal eigenvalue matrix of $\mathbf{B}$ respectively.

We finally define $\mathbf{y} = \mathbf{U_B^T} \mathbf{u}$ and derive Eq. 3.8.

$$\begin{aligned} \beta(\mathbf{y}) &= \mathbf{y}^T \mathbf{\Lambda_B} \mathbf{y} - 2\mathbf{b^T} \mathbf{y} = t_2 \\ \mathbf{b^T} &= (\mu_1 - \mu_2)^T \hat{\mathbf{\Sigma}}_{\mathbf{2}}^{-\mathbf{1}} \mathbf{U}_{\omega_1} \mathbf{\Lambda}_{\omega_1}^{\mathbf{1/2}} \mathbf{U_B} \\ \mathbf{y} &\sim N(\mathbf{0}, \mathbf{I}) \end{aligned} \tag{3.8}$$

It is easy to derive the values for $\mathbf{\Phi}$, $\mathbf{m}$ and t in Eq. 3.4 using Eq. 3.8. These values are given in Eq. 3.9.

$$\begin{aligned} \phi_i &= \lambda_{B,i} \quad \forall i \in \{1, ..., D\} \\ m_i &= \frac{b_i}{\lambda_{B,i}} \quad \forall i \in \{1, ..., D\} \\ t &= t_2 + \sum_{i=1}^{D} \frac{b_i^2}{\lambda_{B,i}}. \end{aligned} \tag{3.9}$$

It is possible for some of the $\lambda_{B,i}$ values to be zero in which case some of the m_i coefficients become infinite or undefined (this is also the case for t). This happens when some of the random variates only have a linear component in Eq. 3.8 or if the variates make no discriminative difference (in which case b_i is also zero). These cases are discussed in the next section.

3.2 DECISION BOUNDARIES AND THEIR SOLUTIONS

In this section we discuss all classes of quadratic boundaries derivable from the theory developed in Section 3.1. We also give analytical solutions to the error rate expressions associated with each decision boundary (except for paraboloidal decision boundaries discussed later).

3.2.1 LINEAR DECISION BOUNDARIES

Linear decision boundaries are the simplest case to solve and occur when $\mathbf{\Lambda_B} = \mathbf{B} = \mathbf{0}$. From Eq. 3.7 it is easy to see that $\hat{\mathbf{\Sigma}}_\mathbf{1} = \hat{\mathbf{\Sigma}}_\mathbf{2}$ for this to be true and it follows that

$$\begin{aligned} \int_{\Omega_2} p(\mathbf{x}|\omega_1) d\mathbf{x} &= p(-2\mathbf{b^T y} > t_2) \\ -2\mathbf{b^T y} &\sim N(0, 4\mathbf{b^T b}) \end{aligned} \tag{3.10}$$

From Eq. 3.10 it is easy to prove that

$$\int_{\Omega_2} p(\mathbf{x}|\omega_1) d\mathbf{x} = \frac{1}{2} \operatorname{erfc}\left(\frac{t_2}{\sqrt{8\mathbf{b^T b}}}\right) \tag{3.11}$$

3.2.2 ELLIPSOIDAL DECISION BOUNDARIES

Ellipsoidal decision boundaries occur when either $\mathbf{B}$ or $-\mathbf{B}$ is positive definite. In other words the eigenvalues $\lambda_{B,1}, ..., \lambda_{B,D}$ are either all negative or all positive. This is a special case that occurs in NB classifiers when one class consistently has a larger variance than the other class for all dimensions. Since $\mathbf{m}$ (see Eq. 3.9) is defined (none of the eigenvalues are zero), we can attempt to solve Eq. 3.4. Many solutions have been proposed for this problem (see, for example Shah (1963)), but the one that we find most efficient is proposed by Ayadi *et al.* (2008); Ruben (1962) and is restated in Section 2.2.1.3 (see theorem 1).

We discussed analytical solutions for the case where all ϕ_i's are greater than zero. A similar

statement can be made for all ϕ_i's less than zero, yielding:

$$\begin{aligned} &\int_{\Omega_2} p(\mathbf{x}|\omega_1)d\mathbf{x} \\ &= \begin{cases} F(-\Phi, \mathbf{m}, -t) & \sup\{\phi_1, ..., \phi_D\} < 0 \\ 1 - F(\Phi, \mathbf{m}, t) & \inf\{\phi_1, ..., \phi_D\} > 0 \end{cases} \end{aligned} \tag{3.12}$$

3.2.3 HYPERBOLOIDAL DECISION BOUNDARIES

Hyperboloidal decision boundaries occur when $\mathbf{B}$ is indefinite and invertible. Therefore, some of the eigenvalues of $\mathbf{B}$ will be positive and others negative, but none of them zero. This is the most frequently occurring case and also the most difficult to solve. Although much research has been done on solving the definite quadratic form (as for the elliptic boundary discussed above), finding an exact analytical expression for the indefinite quadratic form has been unsuccessful (see Press (1966); Ayadi *et al.* (2008); Shah (1963); Raphaeli (1996)). The existing solutions all lead to estimates, bounds or unwieldy solutions (and unusable for NB error analysis). In contrast, we propose a solution that is exact and efficient.

Theorem 2. For $\mathbf{y} \sim N(\mathbf{0}, \mathbf{I})$ and $F(\Phi, \mathbf{m}, t)$ as defined in Eq. 3.4, we can rewrite $F(\Phi, \mathbf{m}, t)$ as follows:

$$\begin{aligned} F(\Phi, \mathbf{m}, t) &= p\left(\sum_{i=1}^{d_1} \phi_i'(y_i - m_i')^2 - \sum_{j=1}^{d_2} \phi_j^*(y_{d_1+j} - m_j^*)^2 \leq t\right), \\ \phi_i', \phi_j^* > 0 &\qquad \forall i \in \{1, ..., d_1\}, \forall j \in \{1, ..., d_2\}, \end{aligned}$$

where $d_1 + d_2 = D$. From this, we can show that

$$F(\Phi, \mathbf{m}, t) = 1 - \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} \alpha_i'\alpha_j^* \Upsilon_{d_1+2i, d_2+2j}(t/p), \quad t \geq 0$$

where we calculate the α_i' and α_j^* coefficients by applying theorem 1 (with common value p) to $F(\Phi', \mathbf{m}', t)$ and $F(\Phi^*, \mathbf{m}^*, t)$ respectively. Note that the α_i' and α_j^* coefficients are independent of t. p can be any arbitrary constant satisfying

$$0 < p \leq \phi_i', \phi_j^* \;\; \forall i \in \{1, ..., d_1\}, \forall j \in \{1, ..., d_2\}$$

$\Upsilon_{k_1,k_2}(z)$ can be calculated using the following recurrence relations.

$$\begin{aligned} \Upsilon_{1,0}(z) &= \frac{1}{\sqrt{\pi}}\Gamma(1/2, z/2) \\ \Upsilon_{1,1}(z) &= \frac{1}{2}\left[1 - \frac{z}{2}\left(K_0(\frac{z}{2})\mathbf{L}_{-1}(\frac{z}{2}) + \mathbf{L}_0(\frac{z}{2})K_{-1}(\frac{z}{2})\right)\right] \\ \Upsilon_{2,k_2}(z) &= 2^{-k_2/2}e^{-z/2} \\ \Upsilon_{k_1,k_2}(z) &= \Upsilon_{k_1-2,k_2}(z) + D_{k_1,k_2}(z) \\ \Upsilon_{k_1,k_2}(z) &= \Upsilon_{k_1,k_2-2}(z) - D_{k_1,k_2}(z), \end{aligned}$$

where

$$D_{k_1,k_2}(z) = \frac{e^{-z/2}}{2^{(k_1+k_2)/2-1}\Gamma(k_1/2)}\psi(1-\frac{k_1}{2}, 2-\frac{k_1+k_2}{2}; z)$$

$\Gamma(a)$ is the gamma function and $\Gamma(a,x)$ is the upper incomplete gamma function. $K_n(x)$ is the modified Bessel function of the second kind and $\mathbf{L}_n(x)$ is the modified Struve function. $\psi(a,b;z)$ is the Tricomi confluent hypergeometric function (also known as the $U(a,b;z)$ function discussed by Slatêr (1960)).

Finally, a bound can be placed on the error from summing only K and L terms in the positive and negative domains, respectively:

$$\begin{aligned} 0 &\leq 1 - \sum_{i=0}^{K}\sum_{j=0}^{L} \alpha_i'\alpha_j^* \Upsilon_{d_1+2i,d_2+2j}(t/p) - F(\Phi, \mathbf{m}, t) \\ &\leq \left(1 - \sum_{i=0}^{K-1} \alpha_i'\right)\left(\sum_{j=0}^{L-1} \alpha_j^*\right) \Upsilon_{d_1+2K,d_2+2L}(t/p) \\ &\quad +1 - \sum_{j=0}^{L-1} \alpha_j^* \end{aligned}$$

Proof. Refer to Section 3.3

It becomes impractical to calculate $D_{k_1,k_2}(z)$ for large values of k_1 and k_2 and therefore the following recurrence relations become useful

$$\begin{aligned} D_{k_1,k_2}(z) &= \frac{1}{4-2k_1}\left[(4-k_1-k_2-2z)D_{k_1-2,k_2}(z) + zD_{k_1-4,k_2}(z)\right] \\ D_{k_1,k_2}(z) &= \frac{1}{4-2k_2}[(4-k_1-k_2+2z)D_{k_1,k_2-2}(z) - zD_{k_1,k_2-4}(z)] \\ D_{k_1,k_2}(z) &= \frac{1}{2}(D_{k_1-2,k_2}(z) + D_{k_1,k_2-2}(z)) \end{aligned} \tag{3.13}$$

Although it is theoretically possible to use only the first two recurrence relations in Eq. 3.13, numerical experiments show that when combined, quantization noise will increase rapidly with each iteration. Therefore we use the first two recurrence relations independently and fill all the remaining gaps with the third recurrence relation in Eq. 3.13. Notice that theorem 2 only applies for cases where $t \geq 0$. A symmetric argument can be expressed for cases where $t < 0$. Finally, we conclude that

$$\begin{aligned} &\textstyle\int_{\Omega_2} p(\mathbf{x}|\omega_1)d\mathbf{x} \\ &= \begin{cases} F(-\Phi, \mathbf{m}, -t) & t < 0 \\ 1 - F(\Phi, \mathbf{m}, t) & t \geq 0 \end{cases} \end{aligned} \tag{3.14}$$

3.2.4 CYLINDRICAL DECISION BOUNDARIES

Cylindrical decision boundaries occur when some of the eigenvalues $\lambda_{B,i}$ and their corresponding linear parts b_i are zero. It is fairly easy to see from Eq. 3.8 that these features can simply be dropped

Table 3.1: All possible quadratic decision boundaries

Boundary type	$\mathbf{\Lambda_B}$	$\mathbf{b}$	Φ	$\mathbf{m}$
Linear	$\mathbf{0}$	$\mathbf{b} \in \Re^D$	$\mathbf{0}$	m_i undef $\forall i \in \{1, ..., D\}$
Ellipsoidal	pos./neg. def.	$\mathbf{b} \in \Re^D$	$\phi_i > 0 \; \forall i \in \{1, ..., D\}$	$\mathbf{m} \in \Re^D$
Hyperboloidal	indef., $\lambda_{ii} \neq 0$	$\mathbf{b} \in \Re^D$	$\Phi \in \Re^D, \phi_i \neq 0 \; \forall i \in \{1, ..., D\}$	$\mathbf{m} \in \Re^D$
Cylindrical	$\lambda_{ii} = 0$	$b_i = 0$	$\Phi \in \Re^D, \phi_i = 0$	m_i undef
Paraboloidal	$\lambda_{ii} = 0$	$b_i \neq 0$	$\Phi \in \Re^D, \phi_i = 0$	m_i undef

and the dimensionality decreased.

3.2.5 PARABOLOIDAL DECISION BOUNDARIES

Paraboloidal decision boundaries occur when some of the eigenvalues $\lambda_{B,i}$ are zero, but their corresponding linear parts b_i are non-zero. In the context of NB classifiers, this only happens when some of the estimated variances (in a given dimension) are identical for ω_1 and ω_2, but their means differ. Unfortunately, an exact solution for this problem does not yet exist. Therefore, as a temporary solution, we simply add a small disturbance $\delta\lambda_i$ to Eq. 3.8 to get an approximate hyperboloidal or ellipsoidal decision boundary. This is a degenerate case, and we discuss its practical relevance below.

In Table 3.1, we summarise all the different decision boundaries that can be obtained and show their meaning in terms of Eq. 3.8 and 3.9.

3.3 PROOF OF THEOREM 2

We can manipulate Eq. 2.9 as follows:

$$\begin{aligned}
F(\Phi, \mathbf{m}, t) &= \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} \alpha_i' \alpha_j^* \int_0^{\infty} \int_0^{(t+u)/p_1} \frac{1}{p_2} f_{d_1+2i}(v) f_{d_2+2j}(\frac{u}{p_2}) dv du \\
&= \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} \alpha_i' \alpha_j^* \int_{-\infty}^{\infty} \int_{-\infty}^{(t+u)/p_1} \frac{1}{p_2} f_{d_1+2i}(v) f_{d_2+2j}(\frac{u}{p_2}) dv du \\
&= \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} \alpha_i' \alpha_j^* \int_{-\infty}^{\infty} \int_{-\infty}^{t} \frac{f_{d_1+2i}(\frac{u+q}{p_1}) f_{d_2+2j}(\frac{u}{p_2})}{p_1 p_2} dq du \\
&= \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} \alpha_i' \alpha_j^* \int_{-\infty}^{t} \int_{-\infty}^{\infty} \frac{f_{d_1+2i}(\frac{u+q}{p_1}) f_{d_2+2j}(\frac{u}{p_2})}{p_1 p_2} du dq \qquad (3.15)
\end{aligned}$$

In step 2, the integral boundaries are shifted from 0 to $-\infty$, since $f_i(x) = 0$ for $x \leq 0$. In step 3, we substituted $q = p_1 v - u$. Since the integration boundaries are independent of q and u, we change the order in which we integrate in step 4.

Let us define $g_{i,p}(x) = \frac{1}{p} f_i(\frac{x}{p})$. It is clear that $\int_{-\infty}^{\infty} g_{i,p}(x) dx = 1$ (and can be interpreted as a pdf), since $f_i(x)$ is the chi-square pdf. Now we can rewrite Eq. 3.15 as follows:

$$F(\Phi, \mathbf{m}, t) = \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} \alpha_i' \alpha_j^* \int_{-\infty}^{t} g_{d_1+2i,p_1}(-u) * g_{d_2+2j,p_2}(u) dq, \quad (3.16)$$

where $*$ is the convolution operation on the two pdf's and represents a new pdf in terms of the q random variate. Therefore $\int_{-\infty}^{\infty} g_{d_1+2i,p_1}(-u) * g_{d_2+2j,p_2}(u) dq = 1$ and we can rewrite Eq. 3.16 as follows:

$$\begin{aligned} F(\Phi, \mathbf{m}, t) &= \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} \alpha_i' \alpha_j^* \left[1 - \int_{t}^{\infty} g_{d_1+2i,p_1}(-u) * g_{d_2+2j,p_2}(u) dq\right] \\ &= 1 - \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} \alpha_i' \alpha_j^* \int_{t}^{\infty} g_{d_1+2i,p_1}(-u) * g_{d_2+2j,p_2}(u) dq, \quad (3.17) \end{aligned}$$

where we made use of the fact that $\sum_{i=0}^{\infty}\sum_{j=0}^{\infty} \alpha_i' \alpha_j^* = 1$, since $\sum_{i=0}^{\infty} \alpha_i' = 1$ and $\sum_{j=0}^{\infty} \alpha_j^* = 1$ (see theorem 1).

From now on, we assume $t \geq 0$ and solve for $\int_{t}^{\infty} g_{d_1+2i,p_1}(-u) * g_{d_2+2j,p_2}(u) dq$.

3.3.1 COMPUTING THE CONVOLUTION

With simple manipulation and substituting $f_k(x) = \frac{x^{k/2-1} e^{-x/2}}{2^{k/2}\Gamma(k/2)} U(x)$, where $U(x)$ is the unit step function, we obtain the following equation.

$$\begin{aligned} g_{k_1,p_1}(-u) * g_{k_2,p_2}(u) &= \frac{e^{-q/2p_1}}{2^{(k_1+k_2)/2} p_1^{k_1/2} p_2^{k_2/2} \Gamma(\frac{k_1}{2})\Gamma(\frac{k_2}{2})} \times \\ &\int_{-\infty}^{\infty} (u+q)^{k_1/2-1} u^{k_2/2-1} e^{\frac{-(p_1+p_2)}{2p_1p_2}u} U(u+q)U(u) du, \quad (3.18) \end{aligned}$$

where $k_1 = d_1 + 2i$ and $k_2 = d_2 + 2j$.

Since we assume $t \geq 0$, we know that $q \geq 0$ and therefore we can ignore the unit step functions and start integrating from 0 as follows:

$$\begin{aligned} g_{k_1,p_1}(-u) * g_{k_2,p_2}(u) &= \frac{e^{-q/2p_1}}{2^{(k_1+k_2)/2} p_1^{k_1/2} p_2^{k_2/2} \Gamma(\frac{k_1}{2})\Gamma(\frac{k_2}{2})} \times \\ &\int_{0}^{\infty} (u+q)^{k_1/2-1} u^{k_2/2-1} e^{\frac{-(p_1+p_2)}{2p_1p_2}u} du \quad (3.19) \end{aligned}$$

Next, we substitute $r = u/q$ and simplify to get the following equation.

$$\begin{aligned} g_{k_1,p_1}(-u) * g_{k_2,p_2}(u) &= \frac{e^{-q/2p_1} p_1^{k_2/2-1} p_2^{k_1/2-1}}{2\Gamma(\frac{k_1}{2})\Gamma(\frac{k_2}{2})(p_1+p_2)^{(k_1+k_2)/2-1}} \times \\ &\quad (\frac{p_1+p_2}{2p_1p_2}q)^{b-1} \int_0^\infty (r+1)^{b-a-1} r^{a-1} e^{\frac{-(p_1+p_2)}{2p_1p_2}qr} dr \\ a &= k_2/2 \\ b &= (k_1+k_2)/2 \end{aligned} \tag{3.20}$$

Note that the integral in Eq. 3.20 can be interpreted as the integral definition of the Tricomi confluent hypergeometric function $\psi(a, b; z)$ (see Gradshteyn and Ryzhik (1980)).

$$\psi(a,b;z) = \frac{1}{\Gamma(a)} \int_0^\infty (t+1)^{b-a-1} t^{a-1} e^{-zt} dt \quad \mathrm{Re}(b) > \mathrm{Re}(a) > 0 \tag{3.21}$$

It is clear that $a = k_2/2 = d_2/2 + j > 0$, $b - a = (k_1+k_2)/2 - k_2/2 = k_1/2 = d_1/2 + i > 0$ and therefore the definition is valid. Also note that $\psi(a, b; z)$ is defined for all $a, b, z \in R$ even though the integral definition is only valid for $\mathrm{Re}(b) > \mathrm{Re}(a) > 0$.

Now we can replace the integral in Eq. 3.20 with the confluent hypergeometric function.

$$\begin{aligned} g_{k_1,p_1}(-u) * g_{k_2,p_2}(u) &= \frac{e^{-q/2p_1} p_1^{k_2/2-1} p_2^{k_1/2-1}}{2\Gamma(\frac{k_1}{2})(p_1+p_2)^{(k_1+k_2)/2-1}} \times \\ &\quad (\frac{p_1+p_2}{2p_1p_2}q)^{b-1} \psi(a,b;\frac{p_1+p_2}{2p_1p_2}q) \end{aligned} \tag{3.22}$$

Finally, we make use of the following Kummer transformation (see Abramowitz and Stegun (1972)).

$$\psi(1+a-b, 2-b; z) = z^{b-1}\psi(a,b;z) \tag{3.23}$$

Therefore, our final expression for the convolution is given in Eq. 3.24.

$$\begin{aligned} g_{k_1,p_1}(-u) * g_{k_2,p_2}(u) &= \frac{p_1^{k_2/2-1} p_2^{k_1/2-1}}{2\Gamma(\frac{k_1}{2})(p_1+p_2)^{(k_1+k_2)/2-1}} e^{-q/2p_1} \times \\ &\quad \psi(1-\frac{k_1}{2}, 2-\frac{k_1+k_2}{2}; \frac{p_1+p_2}{2p_1p_2}q) \end{aligned} \tag{3.24}$$

3.3.2 COMPUTING THE CUMULATIVE DISTRIBUTION

Let us define $\Upsilon_{k_1,k_2}^{p_1,p_2}(z)$ as follows:

$$\Upsilon_{k_1,k_2}^{p_1,p_2}(z) = \frac{p_1^{k_2/2} p_2^{k_1/2-1}}{2\Gamma(\frac{k_1}{2})(p_1+p_2)^{(k_1+k_2)/2-1}} \int_z^\infty e^{-\tau/2} \psi(1-\frac{k_1}{2}, 2-\frac{k_1+k_2}{2}; \frac{p_1+p_2}{2p_2}\tau) d\tau \tag{3.25}$$

From this definition we are trying to compute $\Upsilon_{k_1,k_2}^{p_1,p_2}(\frac{t}{p_1})$.

$$\Upsilon_{k_1,k_2}^{p_1,p_2}(\frac{t}{p_1}) = \int_t^\infty g_{d_1+2i,p_1}(-u) * g_{d_2+2j,p_2}(u)dq \tag{3.26}$$

It is also easy to see that when $p_1 = p_2 = p$, $\Upsilon_{k_1,k_2}^{p,p}(z)$ is independent of p and we use the simplified notation $\Upsilon_{k_1,k_2}(z)$.

$$\begin{aligned}\Upsilon_{k_1,k_2}(z) &= \frac{p^{k_2/2}p^{k_1/2-1}}{2\Gamma(\frac{k_1}{2})(p+p)^{(k_1+k_2)/2-1}}\int_z^\infty e^{-\tau/2}\psi(1-\frac{k_1}{2},2-\frac{k_1+k_2}{2};\frac{p+p}{2p}\tau)d\tau \\ &= \frac{1}{2^{(k_1+k_2)/2}\Gamma(\frac{k_1}{2})}\int_z^\infty e^{-\tau/2}\psi(1-\frac{k_1}{2},2-\frac{k_1+k_2}{2};\tau)d\tau \end{aligned}\tag{3.27}$$

3.3.2.1 COMPUTING $\Upsilon_{1,0}^{p_1,p_2}(z)$

We calculate $\Upsilon_{1,0}^{p_1,p_2}(z)$.

$$\begin{aligned}\Upsilon_{1,0}^{p_1,p_2}(z) &= \frac{p_2^{-1/2}}{2\Gamma(\frac{1}{2})(p_1+p_2)^{-1/2}}\int_z^\infty e^{-\tau/2}\psi(\frac{1}{2},\frac{3}{2};\frac{p_1+p_2}{2p_2}\tau)d\tau \\ &= \frac{p_2^{-1/2}}{2\Gamma(\frac{1}{2})(p_1+p_2)^{-1/2}}\int_z^\infty e^{-\tau/2}(\frac{p_1+p_2}{2p_2}\tau)^{-1/2}d\tau \\ &= \frac{1}{\sqrt{2}\Gamma(\frac{1}{2})}\int_z^\infty e^{-\tau/2}(\tau)^{-1/2}d\tau \\ &= \frac{1}{\Gamma(\frac{1}{2})}\int_{z/2}^\infty e^{-t}t^{-1/2}dt \\ &= \frac{1}{\sqrt{\pi}}\Gamma(\frac{1}{2},\frac{z}{2}) \end{aligned}\tag{3.28}$$

In step two, we made use of the identity $\psi(\frac{1}{2},\frac{3}{2},x) = x^{-1/2}$, which follows from the definition of ψ. In step 4 we substituted $t = \frac{\tau}{2}$. In step 5 we notice that the integral represents the standard definition of the upper incomplete gamma function $\Gamma(s,x) = \int_x^\infty t^{s-1}e^{-t}dt$ and we also use $\Gamma(1/2) = \sqrt{\pi}$.

Note that $\Upsilon_{1,0}^{p_1,p_2}(z)$ is independent of p_1 and p_2 and therefore we can use the simplified notation $\Upsilon_{1,0}(z)$

$$\Upsilon_{1,0}(z) = \frac{1}{\sqrt{\pi}}\Gamma(\frac{1}{2},\frac{z}{2}) \tag{3.29}$$

3.3.2.2 COMPUTING $\Upsilon_{2,k_2}^{p_1,p_2}(z)$

We calculate $\Upsilon_{2,k_2}^{p_1,p_2}(z)$.

$$\begin{aligned}\Upsilon_{2,k_2}^{p_1,p_2}(z) &= \frac{p_1^{k_2/2}}{2(p_1+p_2)^{k_2/2}}\int_z^\infty e^{-\tau/2}\psi(0,1-\frac{k_2}{2};\frac{p_1+p_2}{2p_2}\tau)d\tau \\ &= \frac{p_1^{k_2/2}}{2(p_1+p_2)^{k_2/2}}\int_z^\infty e^{-\tau/2}d\tau \\ &= (\frac{p_1}{p_1+p_2})^{k_2/2}e^{-z/2} \end{aligned} \tag{3.30}$$

In step two, we made use of the identity $\psi(0,b,x)=1$.

For the special case where $p_1=p_2$,

$$\Upsilon_{2,k_2}(z) = 2^{-k_2/2}e^{-z/2} \tag{3.31}$$

3.3.2.3 COMPUTING $\Upsilon_{1,1}^{p_1,p_2}(z)$

We calculate $\Upsilon_{1,1}^{p_1,p_2}(z)$.

$$\begin{aligned}\Upsilon_{1,1}^{p_1,p_2}(z) &= \frac{p_1^{1/2}p_2^{-1/2}}{2\Gamma(\frac{1}{2})}\int_z^\infty e^{-\tau/2}\psi(\frac{1}{2},1;\frac{p_1+p_2}{2p_2}\tau)d\tau \\ &= \frac{(p_1p_2)^{1/2}}{\Gamma(\frac{1}{2})(p_1+p_2)}\int_{\frac{p_1+p_2}{2p_2}z}^\infty e^{-\frac{p_2}{p_1+p_2}t}\psi(\frac{1}{2},1;t)dt \\ &= \frac{(p_1p_2)^{1/2}}{\pi(p_1+p_2)}\int_{\frac{p_1+p_2}{2p_2}z}^\infty e^{(\frac{1}{2}-\frac{p_2}{p_1+p_2})t}K_0(t/2)dt \end{aligned} \tag{3.32}$$

In step 2 we substituted $t=\frac{p_1+p_2}{2p_2}\tau$ and in step 3 we made use of the identity $\psi(\frac{1}{2},1,x)=\pi^{-1/2}e^{x/2}K_0(\frac{x}{2})$, where K_0 is the modified Bessel function of the second kind (see Abramowitz and Stegun (1972)).

Next, we make use of the following identity (Gradshteyn and Ryzhik, 1980).

$$\int_0^\infty e^{-\alpha x}K_0(\beta x)dx = \frac{\arccos(\alpha/\beta)}{\sqrt{\beta^2-\alpha^2}} \quad 0<\beta,\alpha<\beta,\mathrm{Re}(\alpha+\beta)>0 \tag{3.33}$$

From Eq. 3.32, $\alpha=\frac{p_2}{p_1+p_2}-\frac{1}{2}$ and $\beta=\frac{1}{2}$. Since both p_1 and p_2 are positive constants (see theorem 1), we know that $-\frac{1}{2}<\alpha<\frac{1}{2}$ and therefore Eq. 3.33 applies for all positive p_1 and p_2 values.

When we combine Eq. 3.32 and 3.33, we get the following expression.

$$\Upsilon_{1,1}^{p_1,p_2}(z) = \frac{1}{\pi}\arccos(\frac{2p_2}{p_1+p_2}-1) - \frac{(p_1p_2)^{1/2}}{\pi(p_1+p_2)}\int_0^{\frac{p_1+p_2}{2p_2}z}e^{(\frac{1}{2}-\frac{p_2}{p_1+p_2})t}K_0(t/2)dt \tag{3.34}$$

We are only able to get a closed-form expression (in terms of special functions) for the integral in Eq. 3.34 when $p_1=p_2=p$, in which case we can simplify as follows:

$$\Upsilon_{1,1}(z) = \frac{1}{2} - \frac{1}{2\pi}\int_0^z K_0(t/2)dt \tag{3.35}$$

Next, we use the following identity (Gradshteyn and Ryzhik, 1980).

$$\begin{aligned}
\int_0^1 x^v K_v(ax)dx &= 2^{v-1}a^{-v}\pi^{1/2}\Gamma(v+\frac{1}{2})[K_v(a)\mathbf{L}_{v-1}(a) + \mathbf{L}_v(a)K_{v-1}(a)] \quad \mathrm{Re}(v) > -\frac{1}{2} \\
\therefore \int_0^1 K_0(ax)dx &= \frac{\pi}{2}[K_0(a)\mathbf{L}_{-1}(a) + \mathbf{L}_0(a)K_{-1}(a)] \\
\therefore \frac{1}{z}\int_0^z K_0(\frac{a}{z}t)dt &= \frac{\pi}{2}[K_0(a)\mathbf{L}_{-1}(a) + \mathbf{L}_0(a)K_{-1}(a)] \\
\therefore \int_0^z K_0(\alpha t)dt &= \frac{\pi z}{2}[K_0(\alpha z)\mathbf{L}_{-1}(\alpha z) + \mathbf{L}_0(\alpha z)K_{-1}(\alpha z)] \\
\therefore \int_0^z K_0(\frac{t}{2})dt &= \frac{\pi z}{2}[K_0(\frac{z}{2})\mathbf{L}_{-1}(\frac{z}{2}) + \mathbf{L}_0(\frac{z}{2})K_{-1}(\frac{z}{2})],
\end{aligned} \tag{3.36}$$

where $L_n(x)$ is the modified Struve function.

We conclude that

$$\Upsilon_{1,1}(z) = \frac{1}{2}\left[1 - \frac{z}{2}\left(K_0(\frac{z}{2})\mathbf{L}_{-1}(\frac{z}{2}) + \mathbf{L}_0(\frac{z}{2})K_{-1}(\frac{z}{2})\right)\right] \tag{3.37}$$

3.3.2.4 RECURSIVE SOLUTION FOR $\Upsilon^{p_1,p_2}_{k_1-2,k_2}(z)$

Let us rewrite $\Upsilon^{p_1,p_2}_{k_1,k_2}(z)$ as follows:

$$\begin{aligned}
\Upsilon^{p_1,p_2}_{k_1,k_2}(z) &= A^{p_1,p_2}_{k_1,k_2}\int_z^\infty e^{-\tau/2}\psi(a,b;\alpha\tau)d\tau \\
&= \frac{A^{p_1,p_2}_{k_1,k_2}}{\alpha}\int_{\alpha z}^\infty e^{-\frac{u}{2\alpha}}\psi(a,b;u)du \\
A^{p_1,p_2}_{k_1,k_2} &= \frac{p_1^{k_2/2}p_2^{k_1/2-1}}{2\Gamma(\frac{k_1}{2})(p_1+p_2)^{\frac{k_1+k_2}{2}-1}} \\
a &= 1 - \frac{k_1}{2} \\
b &= 2 - \frac{k_1+k_2}{2} \\
\alpha &= \frac{p_1+p_2}{2p_2}
\end{aligned} \tag{3.38}$$

We can also get an expression for $\Upsilon^{p_1,p_2}_{k_1-2,k_2}(z)$.

$$\begin{aligned}
\Upsilon^{p_1,p_2}_{k_1-2,k_2}(z) &= \frac{A^{p_1,p_2}_{k_1-2,k_2}}{\alpha}\int_{\alpha z}^{\infty} e^{-\frac{u}{2\alpha}}\psi(a+1,b+1;u)du \\
A^{p_1,p_2}_{k_1-2,k_2} &= \frac{p_1^{k_2/2}p_2^{k_1/2-2}}{2\Gamma(\frac{k_1}{2}-1)(p_1+p_2)^{\frac{k_1+k_2}{2}-2}} \\
&= (\frac{p_1+p_2}{p_2})(\frac{k_1}{2}-1)\frac{p_1^{k_2/2}p_2^{k_1/2-1}}{2\Gamma(\frac{k_1}{2})(p_1+p_2)^{\frac{k_1+k_2}{2}-1}} \\
&= -2\alpha a A^{p_1,p_2}_{k_1,k_2} \qquad (3.39)
\end{aligned}$$

We now compute the integral using integration by parts and using $\frac{d}{dx}\psi(a,b;x) = -a\psi(a+1,b+1;x)$ (Abramowitz and Stegun, 1972).

$$\begin{aligned}
\int_{\alpha z}^{\infty} e^{-\frac{u}{2\alpha}}\psi(a,b;u)du &= \left[\psi(a,b;u)\int e^{-\frac{u}{2\alpha}}du\right]_{\alpha z}^{\infty} - \int_{\alpha z}^{\infty}\left[\frac{d}{du}\psi(a,b;u)\int e^{-\frac{u}{2\alpha}}du\right]du \\
&= \left[-2\alpha e^{-\frac{u}{2\alpha}}\psi(a,b;u)\right]_{\alpha z}^{\infty} - 2\alpha a\int_{\alpha z}^{\infty} e^{-\frac{u}{2\alpha}}\psi(a+1,b+1;u)du \qquad (3.40)
\end{aligned}$$

Before we can go further, we need to find $\lim_{x\to\infty} e^{-\beta x}\psi(a,b,x) \quad \beta > 0$. First we look at the asymptotic properties of $\psi(a,b,x)$ (Abramowitz and Stegun, 1972).

$$\psi(a,b,x) = x^{-a}\left[\sum_{n=0}^{R-1}\frac{(a)_n(1+a-b)_n}{n!}(-x)^{-n} + O(|x|^{-R})\right] \qquad x\to\infty, \qquad (3.41)$$

where $(a)_n$ represents the falling factorial $(a)_n = a(a-1)...(a-n+1)$.

Therefore

$$\begin{aligned}
\lim_{x\to\infty}[e^{-\beta x}\psi(a,b,x)] &= \lim_{x\to\infty}\left[e^{-\beta x}x^{-a}\sum_{n=0}^{R-1}\frac{(a)_n(1+a-b)_n}{n!}(-x)^{-n} + O(|x|^{-R})\right] \\
&= 0 \qquad \beta > 0 \qquad (3.42)
\end{aligned}$$

Now we can simplify Eq. 3.40.

$$\int_{\alpha z}^{\infty} e^{-\frac{u}{2\alpha}}\psi(a,b;u)du = 2\alpha e^{-\frac{z}{2}}\psi(a,b;\alpha z) - 2\alpha a\int_{\alpha z}^{\infty} e^{-\frac{u}{2\alpha}}\psi(a+1,b+1;u)du \qquad (3.43)$$

If we combine Eq. 3.38 and 3.43 we get the following expression.

$$\Upsilon^{p_1,p_2}_{k_1,k_2}(z) = 2A^{p_1,p_2}_{k_1,k_2}e^{-\frac{z}{2}}\psi(a,b,\alpha z) - 2aA^{p_1,p_2}_{k_1,k_2}\int_{\alpha z}^{\infty} e^{-\frac{u}{2\alpha}}\psi(a+1,b+1;u)du \qquad (3.44)$$

If we combine this result with Eq. 3.39, we get the final recurrence relation.

$$\begin{aligned}\Upsilon^{p_1,p_2}_{k_1,k_2}(z) &= 2A^{p_1,p_2}_{k_1,k_2}e^{-\frac{z}{2}}\psi(a,b,\alpha z) + \frac{A^{p_1,p_2}_{k_1-2,k_2}}{\alpha}\int_{\alpha z}^{\infty} e^{-\frac{u}{2\alpha}}\psi(a+1,b+1;u)du \\ &= 2A^{p_1,p_2}_{k_1,k_2}e^{-\frac{z}{2}}\psi(a,b,\alpha z) + \Upsilon^{p_1,p_2}_{k_1-2,k_2}(z) \\ &= \Upsilon^{p_1,p_2}_{k_1-2,k_2}(z) + D^{p_1,p_2}_{k_1,k_2}(z), \end{aligned} \tag{3.45}$$

where

$$\begin{aligned} D^{p_1,p_2}_{k_1,k_2}(z) &= 2A^{p_1,p_2}_{k_1,k_2}e^{-\frac{z}{2}}\psi(a,b,\alpha z) \\ &= \frac{p_1^{k_2/2}p_2^{k_1/2-1}}{\Gamma(\frac{k_1}{2})(p_1+p_2)^{\frac{k_1+k_2}{2}-1}}e^{-\frac{z}{2}}\psi(1-\frac{k_1}{2},2-\frac{k_1+k_2}{2};\frac{p_1+p_2}{2p_2}z) \end{aligned} \tag{3.46}$$

For the special case were $p_1 = p_2 = p$, we get the following simplified recurrence relation.

$$\begin{aligned}\Upsilon_{k_1,k_2}(z) &= \Upsilon_{k_1-2,k_2}(z) + D_{k_1,k_2}(z) \\ D_{k_1,k_2}(z) &= \frac{e^{-\frac{z}{2}}}{2^{\frac{k_1+k_2}{2}-1}\Gamma(\frac{k_1}{2})}\psi(1-\frac{k_1}{2},2-\frac{k_1+k_2}{2};z) \end{aligned} \tag{3.47}$$

3.3.2.5 RECURSIVE SOLUTION FOR $\Upsilon^{p_1,p_2}_{k_1,k_2-2}(z)$

First we make use of the following recurrence relation (Abramowitz and Stegun, 1972).

$$\begin{aligned}\psi(a,b;x) &= \psi(a,b+1;x) + \frac{d}{dx}\psi(a,b,z) \\ &= \psi(a,b+1;x) - a\psi(a+1,b+1,z), \end{aligned} \tag{3.48}$$

using the identity $\frac{d}{dx}\psi(a,b,x) = -a\psi(a+1,b+1,x)$.

When we substitute Eq. 3.48 into Eq. 3.38 we get the following.

$$\begin{aligned}\Upsilon^{p_1,p_2}_{k_1,k_2}(z) &= \frac{A^{p_1,p_2}_{k_1,k_2}}{\alpha}\int_{\alpha z}^{\infty} e^{-\frac{u}{2\alpha}}[\psi(a,b+1;u) - a\psi(a+1,b+1;u)]du \\ &= \frac{A^{p_1,p_2}_{k_1,k_2}}{\alpha}\int_{\alpha z}^{\infty} e^{-\frac{u}{2\alpha}}\psi(a,b+1;u)du - \frac{aA^{p_1,p_2}_{k_1,k_2}}{\alpha}\int_{\alpha z}^{\infty} e^{-\frac{u}{2\alpha}}\psi(a+1,b+1;u)]du \end{aligned} \tag{3.49}$$

When we combine Eq. 3.39, 3.45 and 3.49, we get the following expressions.

$$\begin{aligned}\Upsilon^{p_1,p_2}_{k_1,k_2}(z) &= \frac{A^{p_1,p_2}_{k_1,k_2}}{\alpha}\int_{\alpha z}^{\infty} e^{-\frac{u}{2\alpha}}\psi(a,b+1;u)du + \frac{1}{2\alpha}\Upsilon^{p_1,p_2}_{k_1-2,k_2}(z) \\ \therefore (2\alpha)\Upsilon^{p_1,p_2}_{k_1,k_2}(z) &= 2A^{p_1,p_2}_{k_1,k_2}\int_{\alpha z}^{\infty} e^{-\frac{u}{2\alpha}}\psi(a,b+1;u)du + [\Upsilon^{p_1,p_2}_{k_1,k_2}(z) - D^{p_1,p_2}_{k_1,k_2}(z)] \\ \therefore (2\alpha-1)\Upsilon^{p_1,p_2}_{k_1,k_2}(z) &= 2A^{p_1,p_2}_{k_1,k_2}\int_{\alpha z}^{\infty} e^{-\frac{u}{2\alpha}}\psi(a,b+1;u)du - D^{p_1,p_2}_{k_1,k_2}(z) \end{aligned} \tag{3.50}$$

Next, we note that

$$\begin{aligned}
\Upsilon^{p_1,p_2}_{k_1,k_2-2}(z) &= \frac{A^{p_1,p_2}_{k_1,k_2-2}}{\alpha}\int_{\alpha z}^{\infty} e^{-\frac{u}{2\alpha}}\psi(a,b+1;u)du \\
A^{p_1,p_2}_{k_1,k_2-2} &= \frac{p_1^{k_2/2-1}p_2^{k_1/2-1}}{2\Gamma(\frac{k_1}{2})(p_1+p_2)^{\frac{k_1+k_2}{2}-2}} \\
&= (\frac{p_1+p_2}{p_1})\frac{p_1^{k_2/2}p_2^{k_1/2-1}}{2\Gamma(\frac{k_1}{2})(p_1+p_2)^{\frac{k_1+k_2}{2}-1}} \\
&= \frac{2\alpha}{2\alpha-1}A^{p_1,p_2}_{k_1,k_2} \\
\therefore \Upsilon^{p_1,p_2}_{k_1,k_2-2}(z) &= \frac{2A^{p_1,p_2}_{k_1,k_2}}{2\alpha-1}\int_{\alpha z}^{\infty} e^{-\frac{u}{2\alpha}}\psi(a,b+1;u)du
\end{aligned} \tag{3.51}$$

Finally, if we substitute Eq. 3.51 into Eq. 3.50, we get the final recurrence relation.

$$\begin{aligned}
(2\alpha-1)\Upsilon^{p_1,p_2}_{k_1,k_2}(z) &= (2\alpha-1)\Upsilon^{p_1,p_2}_{k_1,k_2-2}(z) - D^{p_1,p_2}_{k_1,k_2}(z) \\
\Upsilon^{p_1,p_2}_{k_1,k_2}(z) &= \Upsilon^{p_1,p_2}_{k_1,k_2-2}(z) - \frac{1}{2\alpha-1}D^{p_1,p_2}_{k_1,k_2}(z) \\
&= \Upsilon^{p_1,p_2}_{k_1,k_2-2}(z) - \frac{p_2}{p_1}D^{p_1,p_2}_{k_1,k_2}(z)
\end{aligned} \tag{3.52}$$

For the special case were $p_1 = p_2 = p$, we get the following simplified recurrence relation.

$$\Upsilon_{k_1,k_2}(z) = \Upsilon_{k_1,k_2-2}(z) - D_{k_1,k_2}(z) \tag{3.53}$$

3.3.2.6 ERROR BOUND

The error bound in Ayadi *et al.* (2008) that can be obtained by summing only K and L terms respectively, applies in our case as well, since we have found closed-form expressions for the same quantities.

3.4 CONCLUSION

In this chapter, we derived analytical solutions for calculating error probabilities in correlated Gaussian feature spaces for arbitrary quadratic decision boundaries.

Unfortunately, we still do not have a proper solution for the paraboloidal decision boundaries and we suggested a method for approximating such boundaries with hyperboloidal or ellipsoidal boundaries; this method has also been proposed by Ayadi *et al.* (2008). It should be noted that this method is not without problems, since the α_i terms in theorem 1 take longer to converge when an exceptionally small ϕ_i value or large m_i value is present. From Eq. 3.9 it is clear that a small value for $\lambda_{B,i}$ will produce a small value for ϕ_i and a large value for m_i.

For future work, we propose to find an exact analytical solution for the error rates obtained when paraboloidal decision boundaries occur. Although these boundaries are themselves degenerate (requiring exactly equal class covariances), the same computational issues arise when the hyperboloidal boundaries are almost paraboloidal (i.e. when the relevant class covariances are close).

CHAPTER FOUR

EXPERIMENTS AND RESULTS FOR NAIVE BAYESIAN CLASSIFICATION WITH GAUSSIAN FEATURES

In this chapter, we test the validity of the error estimates derived in Chapter 3 and experiment with their implications. In Section 4.1 we investigate two artificial case studies (a two and twelve dimensional problem) where we compare NB classifier performance with optimal Bayes performance and also maximum likelihood (ML) estimates. This is followed by a discussion of the results in Section 4.2.

4.1 EXPERIMENTS ON THEORETICAL ERROR ESTIMATES

In this section we compare the error performance of simple binary classifiers on problems with different dimensionalities to both the Bayes error rate and that obtained using NB classifiers. These error rates will be obtained using two methods: Monte-Carlo simulations and the analytical methods proposed in Chapter 3.

4.1.1 EXAMPLE 1: A TWO DIMENSIONAL CLASSIFICATION PROBLEM

For this example we will explore the error rates of a two dimensional Gaussian binary classification problem with parameters

$$\mu_{\mathbf{1}} = \begin{bmatrix} 1 \\ \\ 1 \end{bmatrix} \quad \mathbf{\Sigma_1} = \alpha \begin{bmatrix} 3 & -1 \\ \\ -1 & 3 \end{bmatrix},$$

$$\mu_{\mathbf{2}} = \begin{bmatrix} -1 \\ \\ -1 \end{bmatrix} \quad \mathbf{\Sigma_2} = \alpha \begin{bmatrix} 5 & -2 \\ \\ -2 & 1 \end{bmatrix},$$

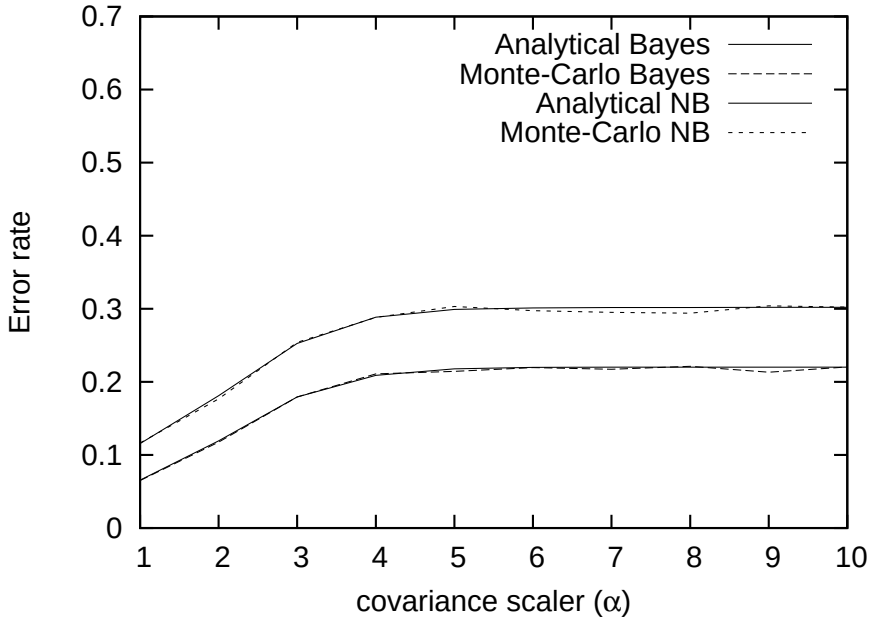

Figure 4.1: *NB and Bayes error rates for two dimensional problem in Example 1 with increasing class covariances.*

where α is a covariance scale factor. Figure 4.1 shows the Bayes and NB error rates obtained with the analytical model developed and Monte-Carlo simulations. The NB classifier's parameters were calculated using diagonal covariance matrices and setting all diagonal components equal to those of the true covariance matrices. It is possible to select better NB parameter estimates, but it is beyond the scope of this experiment.

For this experiment $p(\omega_1) = p(\omega_2) = 0.5$ and 10000 samples in total were generated for the simulations.

Figure 4.2 shows the analytical results obtained for $\alpha = 1$ where we estimate both the full covariance and diagonal covariance (NB) parameters using a varying number of training samples (with one standard deviation error bars).

It is clear from this experiment that the low dimensional classifier with ML full covariance parameters provides superior performance to the classifier with diagonal covariance (NB) parameters for all sample sizes, and that our analytic estimates agree with those obtained by Monte-Carlo simulation.

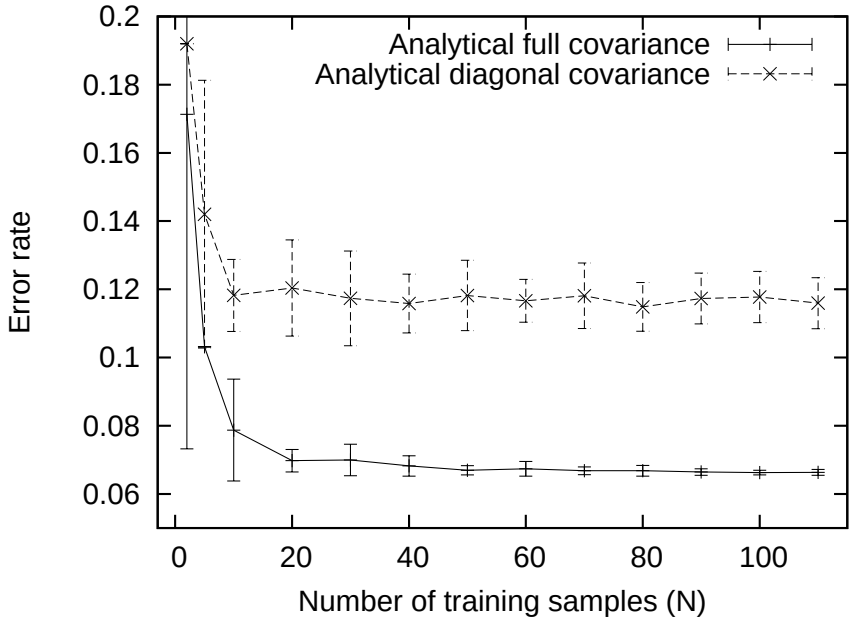

Figure 4.2: *Error rates using diagonal and full covariance ML estimates for the two dimensional problem in Example 1 while increasing the number of training samples.*

4.1.2 EXAMPLE 2: A TWELVE DIMENSIONAL CLASSIFICATION PROBLEM

Now we explore a higher dimensional problem (twelve dimensional) to illustrate the power of NB classifiers. For this example we define

$$\mu_{\mathbf{1}} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix} \quad \mathbf{\Sigma_1} = \alpha \begin{bmatrix} 5 & -1 & 0 & \dots & 0 \\ -1 & 5 & -1 & & 0 \\ \vdots & \ddots & \ddots & \ddots & \\ 0 & & -1 & 5 & -1 \\ 0 & \dots & 0 & -1 & 5 \end{bmatrix},$$

$$\mu_{\mathbf{2}} = \begin{bmatrix} -1 \\ -1 \\ \vdots \\ -1 \\ -1 \end{bmatrix} \quad \mathbf{\Sigma_2} = \alpha \begin{bmatrix} 6 & -2 & 0 & \dots & 0 \\ -2 & 4 & -2 & & 0 \\ \vdots & \ddots & \ddots & \ddots & \\ 0 & & -2 & 6 & -2 \\ 0 & \dots & 0 & -2 & 4 \end{bmatrix},$$

where α is a covariance scale factor. Figure 4.3 shows the Bayes and NB error rates obtained with the analytical model developed and Monte-Carlo simulations. For this experiment $p(\omega_1) = p(\omega_2) = 0.5$ and 10000 samples in total were generated for the simulations.

Figure 4.4 shows the analytical results obtained for $\alpha = 1$ where we estimate both the full covariance and diagonal covariance parameters using a varying number of training samples (with one standard deviation error bars).

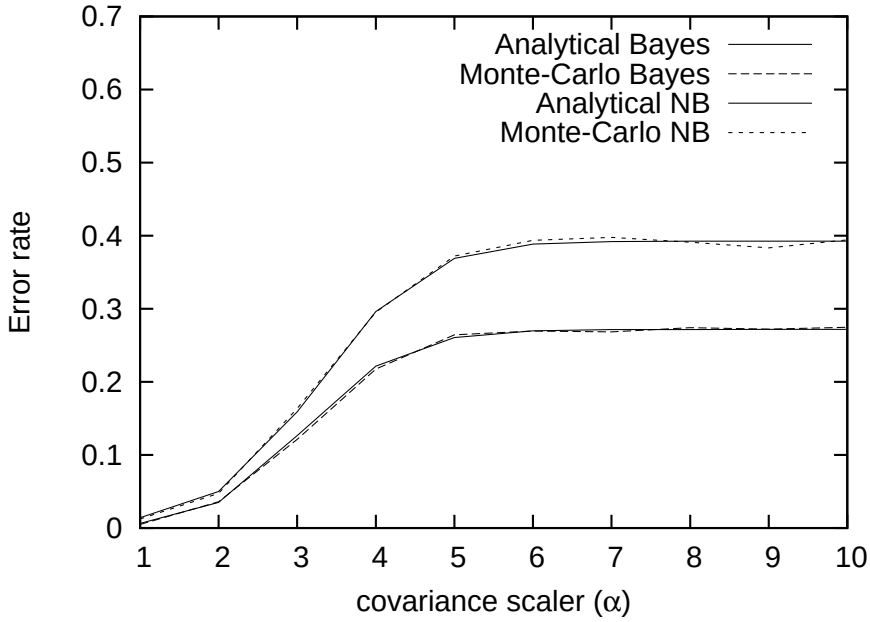

Figure 4.3: *NB and Bayes error rates for twelve dimensional problem in Example 2 with increasing class covariances.*

It is clear from Figure 4.4 that for high dimensional problems, data sparsity becomes an issue and NB classifiers perform better. This is due to the high variance in the full covariance estimate due to too many parameters. NB classifiers are robust for sparse problems and for this specific problem, NB classifiers perform relatively well even when more than a hundred training samples are provided.

4.2 CONCLUSION

In this chapter we tested the validity of the analytical methods developed in Chapter 3 for calculating exact error rates when hyperboloidal decision boundaries are used with correlated Gaussian distributed data sets. It is evident from these results that the analytical errors are exact since the error estimates derived from Monte-Carlo simulations approach these curves exactly (compare these results, for instance, with those obtained by Ayadi *et al.* (2008)).

Note that in these experiments we only tested hyperboloidal decision boundaries. Error expressions for linear and ellipsoidal decision boundaries are considered "easy" and have been verified successfully in the literature. Also, we did not consider paraboloidal or cylindrical decision boundaries since they are degenerate forms that almost never occur in practice.

Finally, we investigated the effects of assuming uncorrelated features on classification performance in a low and high dimensional (two and twelve dimensional) Gaussian problem. From these results it is clear that diagonal covariance parameter estimates suffer less from the curse of dimensionality than those of full covariance estimates (due to the high number of parameters required) and will show superior performance for sparse data sets.

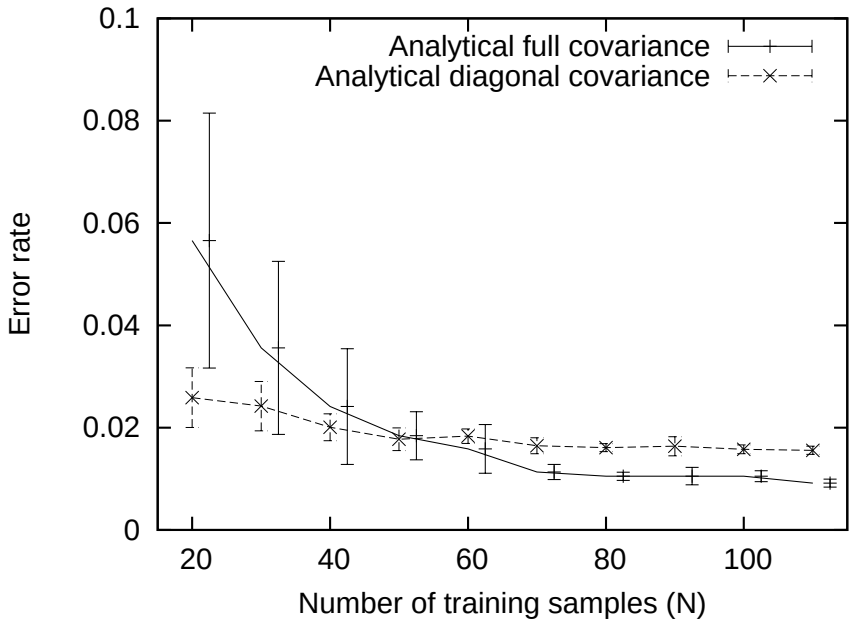

Figure 4.4: *Error rates using diagonal and full covariance ML estimates for the twelve dimensional problem in Example 2 while increasing the number of training samples.*

CHAPTER FIVE

NAIVE BAYESIAN CLASSIFIERS WITH CORRELATED MULTINOMIAL FEATURES: A THEORETICAL APPROACH

One of the main contributions in this chapter includes the derivation of accurate estimates of NB error rates for correlated multinomial features. Recent years have seen a resurgence in the use of NB classifiers with multinomial features (Russell and Norvig, 1995; Botha *et al.*, 2006). Their newfound popularity stems from their use in tasks such as text processing, where such high-dimensional feature spaces arise very naturally.

Consider, for example, a task such as text classification, where a natural feature set is the counts of the distinct words in a document. In this case, the number of dimensions equals the size of the dictionary, which typically contains tens of thousands of entries. Similarly, in text-based language identification (Botha *et al.*, 2006), n-gram frequency counts are often used for test vectors (where an n-gram is a sequence of n consecutive letters). High accuracies are achieved by using large values of n, thus creating feature spaces with millions of dimensions.

The practical popularity of NB classifiers has not been matched by a similar growth in theoretical understanding. Issues such as feature selection, compensation for training-set sparsity and expected learning curves are not well understood for even the simplest cases. In this chapter we therefore develop a theoretical model for calculating error estimates and compensating for data sparsity for the specific case of frequency counts.

To be precise, we assume that there are D distinct entities (such as different n-grams) with fixed class-conditional probabilities of occurring, but only one entity can occur at a time. The process of observing only one entity from the class-conditional probability mass function is called a multinomial trial (or a Bernoulli trial for the special case where $D = 2$); these multinomial trials are repeated m times to create a D dimensional feature vector where each feature is simply the frequency (or total number of occurrences) of the respective entity. This is known as a multinomial process (or a

Bernoulli process for $D = 2$)(Rice, 1988). If all the samples are drawn independently (for example, the probability of observing a given entity is unrelated to any previous observations), the D dimensional feature vectors will take on a multinomial distribution (Rice, 1988). It is important to realize that the classifiers that are used in this chapter and Chapter 6 are naive (see Eq. 5.1), even though there are (in reality) correlation between different features. The most obvious source of correlation (which is independent of the data set) is due to the fact that all the features (frequency of each entity) must add to m (there are only $D - 1$ degrees of freedom). Another source of correlation (dependent on the data set) is due to the fact that the multinomial trials are dependent on each other. In fact, even though we use NB classifiers, it remains important to deal with these correlations to estimate classification error rates (see Sections 5.1.2 and 5.1.3).

We are mainly interested in developing a class separability measure that can be used for feature selection. This separability measure will be in the form of an error estimate for the appropriate classifier. After developing this separability measure, a search algorithm can be used for feature selection. Some examples of search algorithms are the *Branch and Bound* procedure (if we assume that features cannot make negative contributions to error – a possibly inaccurate assumption), *Sequential forward selection* and *Plus l take away r selection* (Webb, 2002). Other search methods such as the *Best individual N* method may not be useful since, as we already mentioned, the different features are correlated. These algorithms are tools for searching through $\binom{D}{d}$ possible feature combinations (a very large number in high dimensional spaces), where d is the number of features selected. Therefore it is crucial to find a computationally efficient separability measure (or error estimate). In Section 5.1.3 we introduce a method to reduce the number of calculations from $O(D^2)$ to $O(D)$ for the proposed error estimate.

The class separability measure that we will use assumes that features are generated from a multinomial distribution with known entity probabilities. It then estimates a distribution for the likelihood function of each class. By observing the overlap of these probability functions, we can calculate an estimate on the error rate of the given classifier.

5.1 MULTINOMIAL LIKELIHOOD DISTRIBUTION ESTIMATION

Let us assume that we have a collection of D entities and that a test sample consists of m independent multinomial trials, where each entity has a probability of p_{dc_r} and $\sum_{d=1}^{D} p_{dc_r} = 1$ for a class c_r. Let the frequencies of each entity (over m trials) represent a D-dimensional feature space. Assuming that all test samples and all multinomial trials are independent, we can calculate the likelihood of any given test vector, given class c_r:

$$p(\bar{x}|c_r) = \frac{m!}{x_1!x_2!...x_D!} p_{1c_r}^{x_1} p_{2c_r}^{x_2} ... p_{Dc_r}^{x_D}, \tag{5.1}$$

where $\bar{x}$ is the input vector, x_d is the frequency count for entity d, m is the number of multinomial trials done and p_{dc_r} is the probability of entity d occurring in a multinomial trial for class c_r.

Notice that the factors $m!$ and $x_d!$ are common to all class likelihood functions for any input vector. Therefore we can ignore these factors and define the modified log likelihood as

$$L(\bar{x}|c_r) = \sum_{d=1}^{D} \alpha_{dr} x_d, \tag{5.2}$$

where $\alpha_{dr} = log(p_{dc_r})$.

Next, we wish to calculate the distribution of this likelihood function given that $\bar{x}$ is sampled from class c_t. Therefore, the probability density function that we wish to estimate is $p(L(\bar{x}|c_r)|c_t)$.

It is clear that $L(\bar{x}|c_r)$ represents a linear combination of multinomial variables that are correlated. It can be shown theoretically (refer to Morris (1975) for a treatment on the central limit theorem for linear combinations of correlated multinomial variables) and through experiments (see Section 6.1.1) that $L(\bar{x}|c_r)$ is approximately Gaussian in the limit where D becomes infinity. Exactly how high D should be before the likelihood becomes practically Gaussian depends on how strongly correlated the variables are and the linear coefficients α_{dr}.

If we assume a Gaussian distribution and estimate the mean and variance of $p(L(\bar{x}|c_r)|c_t)$, we have an estimate for the overall distribution. Therefore we can use the overlap between different likelihood distributions as a separability measure. By taking the expectation of this expression, we find that the mean and variance of the modified class log likelihood functions are

$$\mu = E[L(\bar{x}|c_r)] \tag{5.3}$$

$$\sigma^2 = E[L^2(\bar{x}|c_r)] - E^2[L(\bar{x}|c_r)] \tag{5.4}$$

where all expected values are calculated from the multinomial distribution of class c_t.

The mean of $L(\bar{x}|c_r)$ is given by

$$\mu = E\left(\sum_{d=1}^{D} \alpha_{dr} x_d\right) = m \sum_{d=1}^{D} \alpha_{dr} p_{dc_t}. \tag{5.5}$$

It is therefore straightforward to calculate the mean of all likelihood functions in $O(D)$ computations.

Unfortunately, it is not straightforward to calculate the variance of the likelihood function, since the variables over all dimensions are correlated. Below, we will show an easy way to compensate for the correlation; however, we first derive the estimate that arises when feature correlations are neglected (therefore we assume that samples are generated independently with D degrees of freedom), and then we find the estimate when all correlations are considered.

5.1.1 VARIANCE WITHOUT CORRELATION

If we assume that all variables x_d are uncorrelated, we can calculate the variance of $L(\bar{x}|c_r)$ as

$$\begin{aligned}\sigma_u^2 &= \text{Var}\left(\sum_{d=1}^{D} \alpha_{dr} x_d\right) \\ &= \sum_{d=1}^{D} \alpha_{dr}^2 \,\text{Var}(x_d) \\ &= m \sum_{d=1}^{D} p_{dc_t} q_{dc_t} \alpha_{dr}^2, \end{aligned} \tag{5.6}$$

where σ_u^2 represents the uncorrelated variance (variance of the likelihood when independent binomial samples are tested). From Eq. 5.6 it is clear that σ_u^2 can be calculated in $O(D)$ computations.

5.1.2 VARIANCE WITH CORRELATION

Let us now calculate the complete equation for the variance of $L(\bar{x}|c_r)$ that takes correlation into consideration. From the definition of variance, we can write

$$\sigma^2 = E\left[\left(\sum_{d=1}^{D} \alpha_{dr} x_d\right)^2\right] - E^2\left[\sum_{d=1}^{D} \alpha_{dr} x_d\right] \tag{5.7}$$

From Eq. 5.7 we can rewrite the variance in terms of the multinomial covariance matrix $\mathbf{\Sigma}$:

$$\sigma^2 = \bar{\alpha}_{\mathbf{r}}^{\mathbf{T}} \mathbf{\Sigma} \bar{\alpha}_{\mathbf{r}} \tag{5.8}$$

where $\bar{\alpha}_{\mathbf{r}}^{\mathbf{T}} = [\alpha_{1r}, \alpha_{2r}, ..., \alpha_{Dr}]$. The diagonal elements of $\mathbf{\Sigma}$ are $\sigma_{dd} = m p_{dc_t} q_{dc_t}$ and the off-diagonal elements represent the covariance terms $\sigma_{de} = -m p_{dc_t} p_{ec_t}$.

We can also rewrite Eq. 5.8 in terms of the uncorrelated variance as follows:

$$\sigma^2 = \sigma_u^2 - m \sum_{d} \sum_{e;e \neq d} \alpha_{dr} \alpha_{er} p_{dc_t} p_{ec_t} \tag{5.9}$$

From Eq. 5.9 it is clear that σ^2 can be calculated in $O(D^2)$ computations, which is computationally expensive when D is large (which is often the case), especially for feature selection applications. In the next section we will show a different way to estimate the correlated variance accurately in $O(D)$ computations, which is useful for very large values of D.

5.1.3 COMPENSATING FOR CORRELATION

It is not surprising (see Section 6.1.2.2 in Chapter 6) that the uncorrelated assumption gives inaccurate results, since the features are constrained to sum to a constant value. We now propose a method to compensate for the constraint violation that arises from assuming uncorrelated variables. When assuming that all variables are uncorrelated, we treat each variable as an independent binomial variable. In the multinomial case, we know that $\sum x_d = m$. Therefore, the technique of compensation will

calculate $\sum x_d = m + \Delta m$ for the uncorrelated assumption (where the variables are not constrained to sum to m) and add or subtract the necessary Δm to compensate.

We can express Eq. 5.2 as

$$L(\bar{x}|c_r) = \sum_{d=1}^{D} L_d(x_d|c_r) \tag{5.10}$$

where $L_d(x_d|c_r) = \alpha_{dr} x_d$. Now, if we add any compensation value Δm_d to x_d, we get

$$L_d(x_d + \Delta m_d|c_r) = L_d(x_d|c_r) + \Delta m_d \alpha_{dr} \tag{5.11}$$

The true variance (with all correlations considered) can also be expressed as

$$\sigma^2 = E\left[\left(\sum_{d=1}^{D} L_d(x_d|c_r) - \mu\right)^2\right] \tag{5.12}$$

From Eq. 5.12 we can also write an approximate expression for the true variance $\sigma^2_{\Delta m}$ when $\sum x_d = m + \Delta m$:

$$\sigma^2_{\Delta m} = E\left[\left(\sum_{d=1}^{D} L_d(x_d|c_r) + \Delta m \sum_{d=1}^{D} \alpha_{dr} p_{dc_t} - \mu\right)^2\right], \tag{5.13}$$

where we used Eq. 5.11 and made the approximation $\Delta m_d = \Delta m p_{dc_t}$. By expanding the square in Eq. 5.13 we obtain

$$\sigma^2_{\Delta m} = \sigma^2 + \Delta^2 m \left(\sum_{d=1}^{D} \alpha_{dr} p_{dc_t}\right)^2 \tag{5.14}$$

Next, independent binomial features (from which σ^2_u is calculated) will not necessarily sum to m, instead $\sum x_d = m + \Delta m$ with a distribution mean and variance for $\Delta m = \sum x_d - m$ given by

$$\begin{aligned} \text{Mean}(\Delta m) &= m \sum_{d=1}^{D} p_{dc_t} - m = 0 \\ \text{Var}(\Delta m) &= \sum_{d=1}^{D} \text{Var}(x_d) = m \sum_{d=1}^{D} p_{dc_t} q_{dc_t} \end{aligned} \tag{5.15}$$

The uncorrelated variance σ^2_u can be expressed in terms of $\sigma^2_{\Delta m}$ by summing over the probability mass function of Δm:

$$\sigma^2_u = \sum_{\Delta m} \sigma^2_{\Delta m} p(\Delta m) \tag{5.16}$$

Combining equation 5.14 and 5.16 we get

$$\sigma^2_u = \sigma^2 + \left(\sum_{d=1}^{D} \alpha_{dr} p_{dc_t}\right)^2 \sum_{\Delta m} (\Delta m)^2 p(\Delta m) \tag{5.17}$$

and we notice that:

$$\sum_{\Delta m} (\Delta m)^2 p(\Delta m) = \text{Var}(\Delta m) = m \sum_{d=1}^{D} p_{dc_t} q_{dc_t} \tag{5.18}$$

Therefore, the true variance is expressed in terms of the uncorrelated variance as

$$\sigma^2 = \sigma_u^2 - m\left(\sum_{d=1}^{D}\alpha_{dr}p_{dc_t}\right)^2\left(\sum_{d=1}^{D}p_{dc_t}q_{dc_t}\right) \tag{5.19}$$

Notice the similarities between Eqs. 5.9 and 5.19 and also that Eq. 5.19 can be calculated in $O(D)$ computations.

As we will see below, experimental evidence shows that Eq. 5.19 is accurate for low values of mp_{dc_t}. This is a reasonable condition for high dimensional applications such as text-based language identification.

5.1.4 ADDING AND REMOVING FEATURES

Since we are interested in feature selection, we need a mechanism to add and remove features from the analysis. All the derivations thus far (Eqs. 5.1 to 5.19) assume that all features are accounted for.

The solution to the problem is simple: we simply define a frequency feature x_R that represents the sum of all frequency counts that are removed from the analysis. We therefore define the following parameters:

$$\begin{aligned} p_{Rc_r} &= 1 - \sum_{d\subset C} p_{dc_r} \\ p_{Rc_t} &= 1 - \sum_{d\subset C} p_{dc_t} \\ x_R &= m - \sum_{d\subset C} x_d \\ \alpha_{Rr} &= log(p_{Rc_r}) \end{aligned} \tag{5.20}$$

where C is the subset of all features that are included in the analysis.

The analysis is practically the same as above, except for the fact that p_{Rc_t} can grow large, depending on how many features are used, and Eq. 5.19 might be inaccurate. Therefore we need to take correlation into consideration between features $d \subset C$ and R.

Eqs. 5.21, 5.22 and 5.23 are the new formulas that are equivalent to Eqs. 5.2, 5.5 and 5.6 respectively:

$$L(\bar{x}|c_r) = \sum_{d\subset C}(\alpha_{dr}x_d) + \alpha_{Rr}x_R \tag{5.21}$$

$$\mu = m\sum_{d\subset C} p_{dc_t}\alpha_{dr} + mp_{Rc_t}\alpha_{Rr} \tag{5.22}$$

$$\begin{aligned} \sigma_u^2 = m\sum_{d\subset C}[p_{dc_t}q_{dc_t}\alpha_{dr}^2 - 2p_{dc_t}p_{Rc_t}\alpha_{dr}\alpha_{Rr}] \\ +mp_{rc_t}q_{rc_t}\alpha_{Rr}^2 \end{aligned} \tag{5.23}$$

Notice that Eqs. 5.21 to 5.23 can all be calculated in $O(D_C)$ computations, where D_C is the number of features considered (length of subset C). It is also important to notice that σ_u^2 ignores all correlation

between features in subset C, but takes all correlation into consideration with feature R. We can therefore use a modified version of Eq. 5.19 to include all correlation. The new version of Eq. 5.19 can be expressed as

$$\sigma^2 = \sigma_u^2 - m\left(\sum_{d\subset C}\alpha_{dr}\frac{p_{dc_t}}{q_{Rc_t}}\right)^2\left(\sum_{d\subset C}p_{dc_t}q_{dc_t} - mp_{Rc_t}q_{Rc_t}\right) \tag{5.24}$$

Eq. (5.24) can also be calculated in $O(D_C)$ computations.

5.2 ERROR ESTIMATION FROM LIKELIHOOD DISTRIBUTIONS

Now that we have a Gaussian model with mean and variance estimates for all the likelihood functions, we are in a position to calculate an error rate estimate for all the different classes. If we use the likelihood classifier for discrimination, the optimal class choice (for minimum error) is given by Webb (2002):

$$c = \max_{i=1,\ldots,C} L_i + log(p_i), \tag{5.25}$$

where $L_i = L(\bar{x}|c_i)$ and $p_i = p(c_i)$ is the prior probability of class c_i.

The probability of detecting class i, when the true class is j, is:

$$p_{i|j} = p(c = i|c_j) \tag{5.26}$$

We can combine Eqs. 5.25 and 5.26 to get:

$$\begin{aligned} p_{i|j} = \quad & p[(L_i + log(p_i) \geq L_1 + log(p_1)) \cap \\ & (L_i + log(p_i) \geq L_2 + log(p_2)) \cap \\ & \vdots \\ & (L_i + log(p_i) \geq L_C + log(p_C))|c_j] \end{aligned} \tag{5.27}$$

It is not easy to calculate Eq. 5.27 for $C > 2$ and from now on we consider only binary classifiers ($C = 2$):

$$p_{i|j} = \prod_{k=1}^{2} p_{ik|j}, \tag{5.28}$$

where

$$p_{ik|j} = p[(L_{ik} \geq T_{ik})|c_j], \tag{5.29}$$

$L_{ik} = L_i - L_k$ and $T_{ik} = log(p_k) - log(p_i)$. Notice that $p_{ik|j} = 1$ for $i = k$ and that the expression in Eq. 5.28 is exact (for $C = 2$).

In the previous section we estimated the probability density function of L_i for $i = 1, 2, ..., C$, which we can use to find an estimate for $p_{ik|j}$. In order to do this we need a distribution estimate for

$$L_{ik} = L_i - L_k = \sum_{d=1}^{D}(\alpha_{di} - \alpha_{dk})x_d \tag{5.30}$$

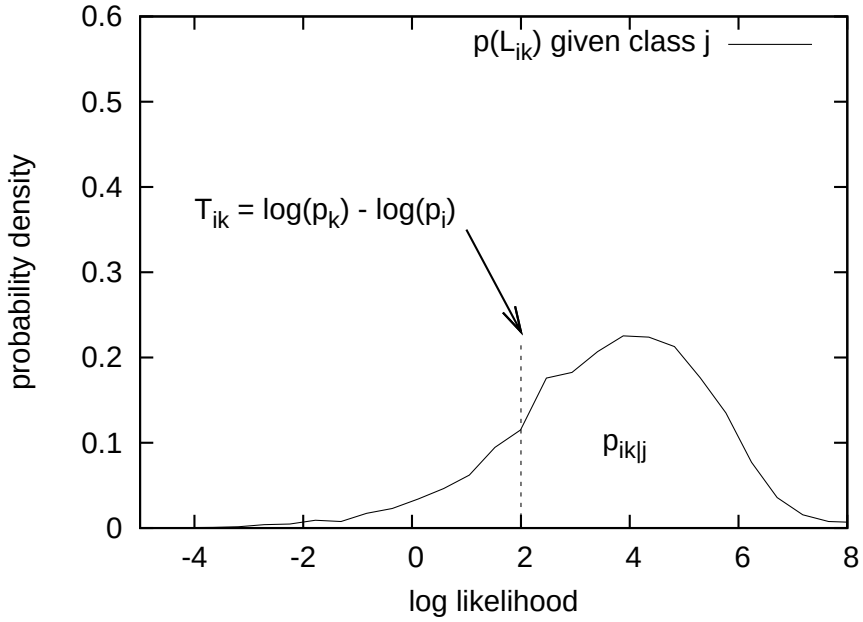

Figure 5.1: *Estimating $p_{ik|j}$ from the probability density function of L_{ik}.*

Notice that Eq. 5.30 has the same functional form as Eq. 5.2. Therefore we can use Eq. 5.5, 5.19 and 5.24 to estimate the distribution of L_{ik} by simply substituting α_{dr} with $(\alpha_{di} - \alpha_{dk})$.

Figure 5.1 shows how we can calculate $p_{ik|j}$ from the distribution of L_{ik}:

$$p_{ik|j} = \frac{1}{2} - \frac{1}{2}\,\mathrm{erf}[\frac{T_{ik} - \mu_{ik|j}}{\sigma_{ik|j}\sqrt{2}}] \tag{5.31}$$

where $\mu_{ik|j}$ and $\sigma_{ik|j}$ are the mean and variance estimates for L_{ik} given class j.

Finally, we can calculate the overall error estimate of the classifier and use it as a dissimilarity measure for feature selection:

$$\epsilon = 1 - \sum_{i=1}^{C} p_{i|i} p_i \tag{5.32}$$

5.3 COMPENSATING FOR UNSEEN ENTITIES

A common problem in high dimensional spaces with multinomial features is that all entity probabilities can not be estimated directly from the data, since not all entities are observed in the training set. We estimate probabilities by counting entity frequencies, yet some entities will never occur in a training set. In this section we therefore derive a method for calculating expected probabilities for entities never observed in the training set for a given class. First, let us derive entity probability estimates for seen entities.

5.3.1 FEATURE PROBABILITY ESTIMATES FOR SEEN ENTITIES

We estimate the entity probabilities in each class by counting the number of entity occurrences in the class training set and dividing it by the total number of entities observed in the training set. This is the same method used by Botha and Barnard (2008) (see also Jelinek and Mercer (1980)) for the specific application of text-based language identification.

$$p_{dc_i} = \frac{f_{dc_i}}{|\mathbf{f_{c_i}}|}, \tag{5.33}$$

where p_{dc_i} is the estimated probability for entity d occurring in class c_i (in one multinomial trial), f_{dc_i} is the total number of times that entity d occurred in the training set of class c_i and $|\mathbf{f_{c_i}}|$ represents the sum of all the entity counts and is equal to the number of multinomial trials run in the training set for class c_i.

Since we have a training set of limited size, we know that there is a given probability of observing an entity in a test set that has never been observed before. We call this probability $P(\overline{o_{c_i}}|s_{c_i})$ for reasons explained in Section 5.3.2. Therefore

$$\begin{aligned} \sum_{d \in D} p_{dc_i} &= \sum_{d \in D_o} p_{dc_i} + P(\overline{o_{c_i}}|s_{c_i}) = 1 \\ \therefore \sum_{d \in D_o} p_{dc_i} &= 1 - P(\overline{o_{c_i}}|s_{c_i}) \end{aligned} \tag{5.34}$$

where D_o is the set of all observed entities and D is the set of all possible entities that can exist in c_i.

From Eq. 5.33 it can be seen that if we have never observed entity d before, then its probability estimate is 0 and the log likelihood for class c_i will be $-\infty$ from Eq. 5.2. This is clearly not optimal and we need to find a better estimate for these unseen entities. Botha and Barnard (2008) used a method where penalty factors were added to the log likelihood for unseen entities and empirical experiments (using cross-validation methods) showed that there are optimal choices for these penalties that drastically improves classification performance. These penalty factors depend on factors such as the window size used (the number of multinomial trials used for classifying) and also the dimensionality of the problem. We therefore replace the estimate in Eq. 5.33 with a modified estimator to allow for the possibility of unseen entities:

$$p_{dc_i} = [1 - P(\overline{o_{c_i}}|s_{c_i})]\frac{f_{dc_i}}{|\mathbf{f_{c_i}}|} \tag{5.35}$$

5.3.2 FEATURE PROBABILITY ESTIMATES FOR UNSEEN ENTITIES

Let us define a four dimensional joint probability mass function that represents the probability of each entity in class c_i for one multinomial trial:

$$P_{c_i}(f, s_{c_j}, o_{c_i}, o_{\overline{c_i}}), \tag{5.36}$$

where f represents the entity seen, s_{c_j} is a discrete variate with C possible values (C being the number of classes) representing the class from which f is observed, o_{c_i} is a Boolean variable indicating whether f has been observed before in the training set for class c_i and $o_{\overline{c_i}}$ is a Boolean variable indicating whether f has been observed before in the training set outside of class c_i. Therefore, f and s_{c_j} provides information regarding the current occurrence of f (in the test set), whereas o_{c_i} and $o_{\overline{c_i}}$

conveys past information (obtained from the train set) with regards to f.

From now on we abuse the notation by using the symbols o_{c_i} and $\overline{o_{c_i}}$ when we refer to $o_{c_i} = \text{true}$ and $o_{c_i} = \text{false}$ respectively. We refer to $o_{\overline{c_i}}$ and $\overline{o_{\overline{c_i}}}$ in a similar fashion.

We have already shown a method for calculating $P_{c_i}(f|o_{c_i})$ using Eq. 5.33. Note that in this special case where we have observed the entity before, the probability is estimated using Eq. 5.35 and is considered independent of whether we are currently observing it from c_i or if it has been observed outside of c_i before. Also note that for this entity $P_{c_i}(f, \overline{o_{c_i}}) = 0$, since we already know that the entity has been observed in c_i.

For unobserved entities, we are interested in calculating $P_{c_i}(f|\overline{o_{c_i}}, o_{\overline{c_i}})$ and $P_{c_i}(f|\overline{o_{c_i}}, \overline{o_{\overline{c_i}}})$.

5.3.2.1 CALCULATING $P_{c_i}(f)$ IF ONLY OBSERVED OUTSIDE OF c_i BEFORE

We now focus on calculating the probability of a specific entity f occurring in class c_i, when all we know is that we have never before seen it in the training set for c_i, yet we have seen it in some of the other classes.

Since we are observing this entity from an unknown class, we have to marginalise over s_{c_j}.

$$\begin{aligned} P_{c_i}(f|\overline{o_{c_i}}, o_{\overline{c_i}}) &= \sum_{j=1}^{C} P_{c_i}(f, s_{c_j}|\overline{o_{c_i}}, o_{\overline{c_i}}) \\ &= \sum_{j=1}^{C} P_{c_i}(f|s_{c_j}, \overline{o_{c_i}}, o_{\overline{c_i}}) P(s_{c_j}|\overline{o_{c_i}}, o_{\overline{c_i}}) \end{aligned} \tag{5.37}$$

Next, we expand $P(s_{c_j}|\overline{o_{c_i}}, o_{\overline{c_i}})$ by applying Bayes' theorem.

$$\begin{aligned} P(s_{c_j}|\overline{o_{c_i}}, o_{\overline{c_i}}) &= \frac{P(\overline{o_{c_i}}|s_{c_j}, o_{\overline{c_i}}) P(s_{c_j}|o_{\overline{c_i}})}{\sum_{k=1}^{C} P(\overline{o_{c_i}}|s_{c_k}, o_{\overline{c_i}}) P(s_{c_k}|o_{\overline{c_i}})} \\ &= \frac{P(\overline{o_{c_i}}|s_{c_j}, o_{\overline{c_i}}) P(o_{\overline{c_i}}|s_{c_j}) P(s_{c_j})}{\sum_{k=1}^{C} P(\overline{o_{c_i}}|s_{c_k}, o_{\overline{c_i}}) P(o_{\overline{c_i}}|s_{c_k}) P(s_{c_k})} \end{aligned} \tag{5.38}$$

We note that $P(s_{c_j})$ represents the probability of entity f being observed from class c_j and is therefore simply the prior probability $p(c_j)$.

Finally, we combine Eqs. 5.37 and 5.38.

$$\begin{aligned} P_{c_i}(f|\overline{o_{c_i}}, o_{\overline{c_i}}) &= \frac{1}{N_{c_i}} \sum_{j=1}^{C} P_{c_i}(f|s_{c_j}, \overline{o_{c_i}}, o_{\overline{c_i}}) P(\overline{o_{c_i}}|s_{c_j}, o_{\overline{c_i}}) P(o_{\overline{c_i}}|s_{c_j}) p(c_j) \\ N_{c_i} &= \sum_{k=1}^{C} P(\overline{o_{c_i}}|s_{c_k}, o_{\overline{c_i}}) P(o_{\overline{c_i}}|s_{c_k}) p(c_k) \end{aligned} \tag{5.39}$$

By making the following assumptions we can simplify Eq. 5.39:

- f is conditionally independent of $o_{\overline{c_i}}$ with respect to s_{c_j} and $\overline{o_{c_i}}$. If we know that we are observing f from class c_j, then we assume that its probability is weakly correlated with its probability of occurrence in other classes.
- $\overline{o_{c_i}}$ is conditionally independent of $o_{\overline{c_i}}$ with respect to s_{c_j}. If we know that we are observing f

from class c_j, then we do not expect the probability that f has never been seen in class c_i to be dependent on its probability to have occurred outside of class c_i.

- $\sum_{j=1}^{C} P_{c_i}(f, s_{c_j}|\overline{o_{c_i}}, o_{\overline{c_i}}) \approx P_{c_i}(f, s_{c_i}|\overline{o_{c_i}}, o_{\overline{c_i}})$. This is a strong assumption, since its validity depends on the application and how disjoint different class entities are. We are therefore assuming that given we have never seen f in c_i before (although we have seen it in other classes) and we know that we are currently observing it from c_j, where $i \neq j$, then the probability that this entity occurs in c_i is negligible.

With these three assumptions, we can simplify Eq. 5.39 as follows:

$$\begin{aligned} P_{c_i}(f|\overline{o_{c_i}}, o_{\overline{c_i}}) &= \frac{1}{N_{c_i}} P_{c_i}(f|s_{c_i}, \overline{o_{c_i}}) P(\overline{o_{c_i}}|s_{c_i}) P(o_{\overline{c_i}}|s_{c_i}) p(c_i) \\ N_{c_i} &= \sum_{k=1}^{C} P(\overline{o_{c_i}}|s_{c_k}) P(o_{\overline{c_i}}|s_{c_k}) p(c_k) \end{aligned} \tag{5.40}$$

From Eq. 5.40, we need to learn the following parameters from the training set.

- $P_{c_i}(f|s_{c_i}, \overline{o_{c_i}})$. We need to calculate the expected probability of the next entity f that occurs for the first time in c_i.

- $P(\overline{o_{c_i}}|s_{c_k})$. We need to calculate the probability that the next entity has never been seen in class c_i if we are observing it in class c_k. If $i = k$, we expect this probability to be smaller than for the case where $i \neq k$.

- $P(o_{\overline{c_i}}|s_{c_k})$. We need to calculate the probability that the next entity has been observed outside of class c_i if we are observing it from class c_k. Clearly this probability will frequently be lower for $i = k$ in comparison to the case where $i \neq k$.

In Chapter 6 we will show methods for calculating these parameters in a text-based language identification case study.

5.3.2.2 HANDLING ENTITIES THAT HAVE NEVER BEEN OBSERVED BEFORE

We can follow the same analysis as in Section 5.3.2.1 for calculating $P_{c_i}(f|\overline{o_{c_i}}, \overline{o_{\overline{c_i}}})$. Unfortunately this will not give us any useful results, due to the symmetry in the expected entity probability for each individual class, since this entity did not occur anywhere in the training set.

By the same logic, the number of completely unobserved entities that will occur in a test vector (m multinomial trials) will be the same for all classes. Therefore, the best approach is to ignore these entities completely, since their contributions for all classes are the same.

5.4 CONCLUSION

In this chapter we derived analytical approximations for solving error probabilities with NB multinomial distributed data sets for high dimensional feature spaces. We also derived compensation techniques for correlated features due to the fact that the total number of features that occur in a test vector must sum to m, where m is the total number of multinomial trials in such a test vector.

We also showed how to adapt these equations in order to calculate error estimates while changing the number of features considered. These equations can, for instance, be used as a dissimilarity measure in feature selection applications.

Finally we derived analytical methods that can be used to compensate for feature sparsity issues that arise in high dimensional spaces. We effectively derived an equation that can be used to calculate entity probability estimates for unobserved entities.

In Chapter 6 we will test the theory proposed in this chapter.

CHAPTER SIX

EXPERIMENTS AND APPLICATION SPECIFIC THEORY FOR MULTINOMIAL FEATURES

In this chapter, we test the theory developed in Chapter 5. It is divided into two independent case studies.

- In Section 6.1 we test the validity of the error analysis done in Section 5.2 for an artificially generated multinomial classification problem.
- In Section 6.2, we test the theory in Section 5.3 on dealing with data sparsity for high dimensional spaces on a text-based language identification problem for all eleven official languages of South Africa. Although most of the theory on dealing with data sparsity is covered in Chapter 5, extra theory will be included that is specific to text-based language identification.

6.1 EXPERIMENTS ON SYNTHETIC MULTINOMIAL FEATURE SETS

In this section we investigate the validity of the various approaches to calculating error probabilities (derived in Chapter 5) with a simulated problem consisting of two multinomial classes with entity probabilities, as shown in Figure 6.1.

These two classes are generated over a feature space of 500 dimensions and all tests are done both empirically and with the theoretical model provided in Chapter 5. Empirical distributions are calculated by generating 10000 multinomial samples of each class and drawing a histogram for all the likelihood functions. Error analysis is done by estimating the error rate of the 10000 samples of each class empirically. All tests are done with $m = 10$ multinomial trial repetitions.

6.1.1 PROBABILITY DISTRIBUTIONS OF LIKELIHOOD FUNCTIONS

The theoretical distributions of two likelihood functions are compared to the empirical histograms. The main reason for doing so is to illustrate the fact that the likelihood functions are indeed approximately Gaussian.

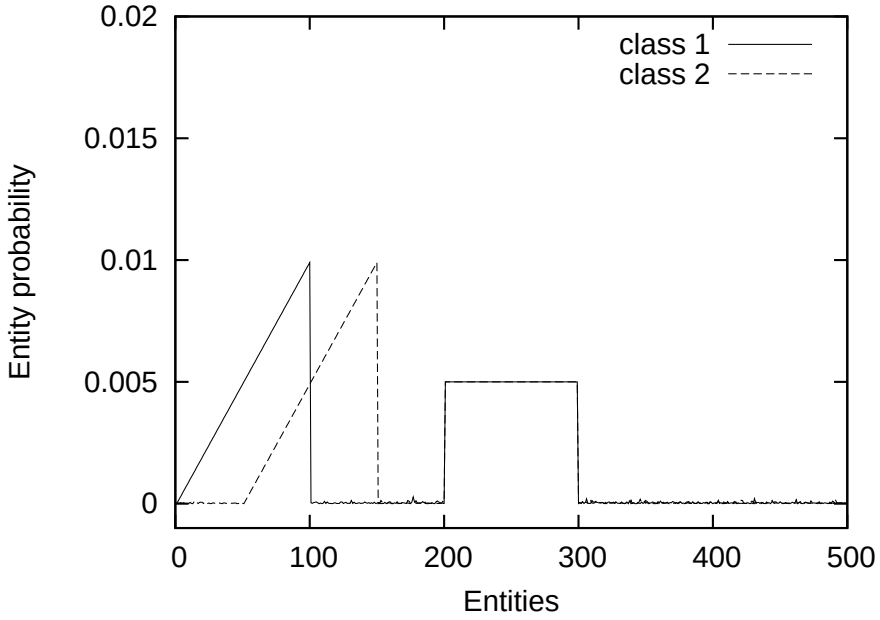

Figure 6.1: *Two classes generated with different entity probabilities.*

Figure 6.2 shows the empirical and theoretical distributions given by $p(L(\bar{x}|c_1)|c_1)$ and $p(L(\bar{x}|c_2)|c_1)$, where c_1 and c_2 are classes one and two presented in Figure 6.1. Also, for this test, only the first 200 features are used and $m = 10$. Therefore, these are the distributions of the likelihood functions for c_1 and c_2, while the true class generator of the input vector is c_1. It is expected that $p(L(\bar{x}|c_1)|c_1)$ tends to higher likelihood values than $p(L(\bar{x}|c_2)|c_1)$ since the true vector is from class c_1.

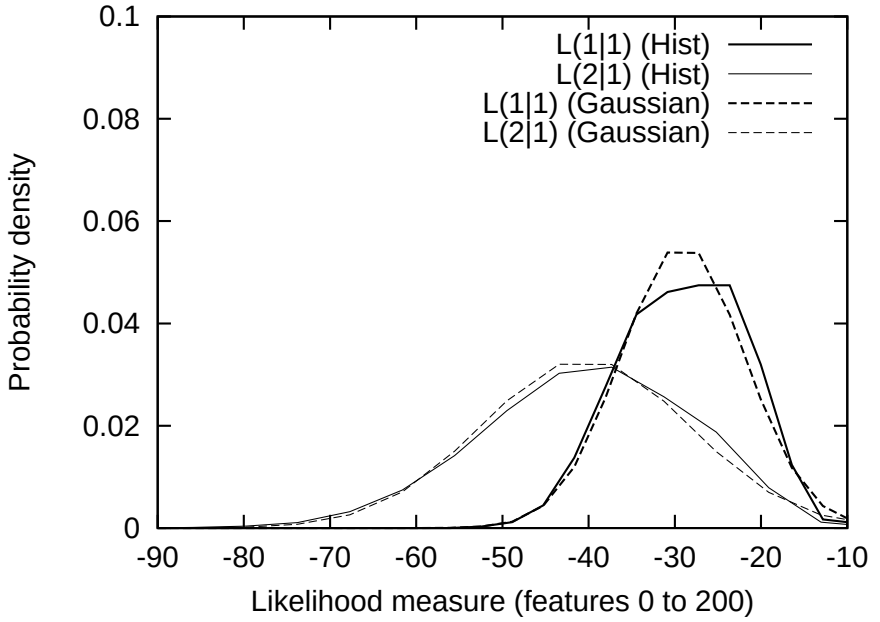

Figure 6.2: *Likelihood distributions of classes 1 and 2 for features 0 to 200*

It should be noted that only the first 150 features are useful for classification, since classes one and two are identically distributed from features 150 and onwards.

Notice that the overlap between the likelihood functions (in Figure 6.2) of classes one and two

suggests that the error rate could be high. However, this is not necessarily the case, since the two likelihood functions are correlated. In fact, the correct likelihood functions to use for error analysis are given by Eq. 5.30.

6.1.2 ERROR ANALYSIS

The empirical error rate (10000 test samples per class are used) of the NB classifier is compared with the theoretical error rate predicted in Chapter 5. For all the tests, we use the two classes described in Figure 6.1 with $m = 10$. We assume equal priors for both classes.

6.1.2.1 EFFECTS OF FEATURE ADDITION ON LIKELIHOOD MEANS

Figure 6.3 shows the predicted likelihood mean values of L_{12} (described in Section 5.1) for input vectors from classes c_1 and c_2 when features are incrementally added into the analysis (see Eqs. 5.30 and 5.31). Notice that L_{ik} is symmetrical to L_{ki} in Eq. 5.30 and therefore we only consider L_{12} and not L_{21}. Also notice that the mean value of L_{12} does not change after feature 150 since the two classes are identical afterward, even though classes one and two have a dense probability space between features 200 and 300 (see Figure 6.1).

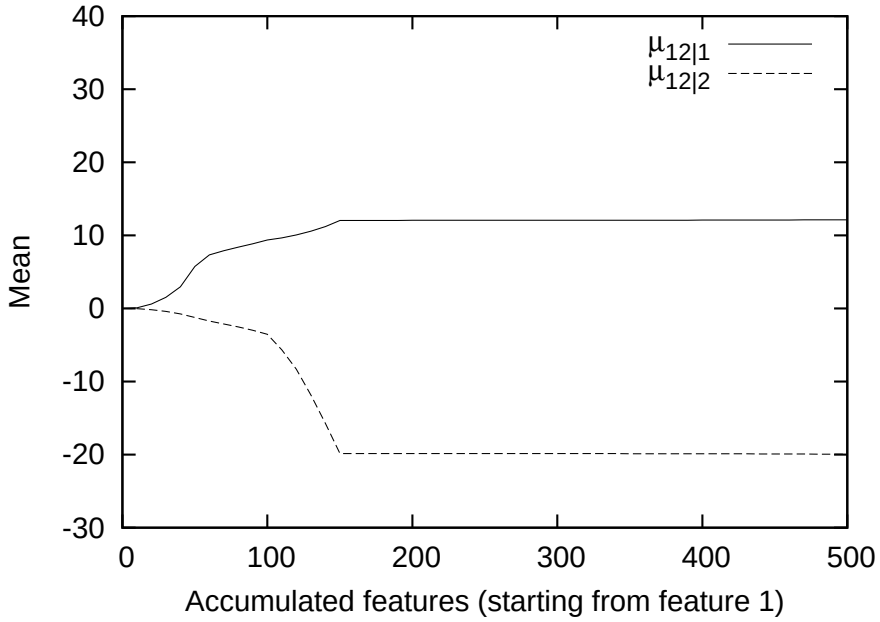

Figure 6.3: *Mean curves for the modified difference likelihood function L_{12} for input vectors from classes c_1 and c_2 while incrementally adding features*

6.1.2.2 EFFECTS OF FEATURE ADDITION ON LIKELIHOOD VARIANCE

Figures 6.4 and 6.5 show the predicted likelihood variances (described in Section 5.1.3) for L_{12} given input vectors from classes c_1 and c_2 respectively. Again, the features 150 to 500 have very little influence on the variances since the two classes are distributed identically on these features.

One deduction that might be surprising from Figure 6.4 is that the variance starts decreasing when adding features beyond about 100. This is understandable, since the cross-correlations between different features are negative (refer to Eq. 5.8).

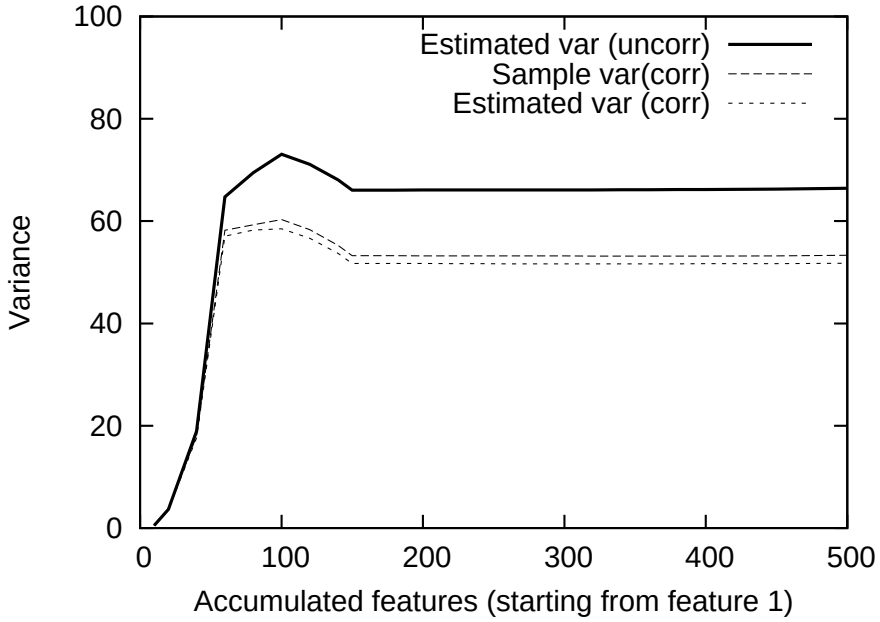

Figure 6.4: *Variance curves for L_{12}, given c_1, when incrementally adding features. Sampled values are compared to those computed from two different approximations.*

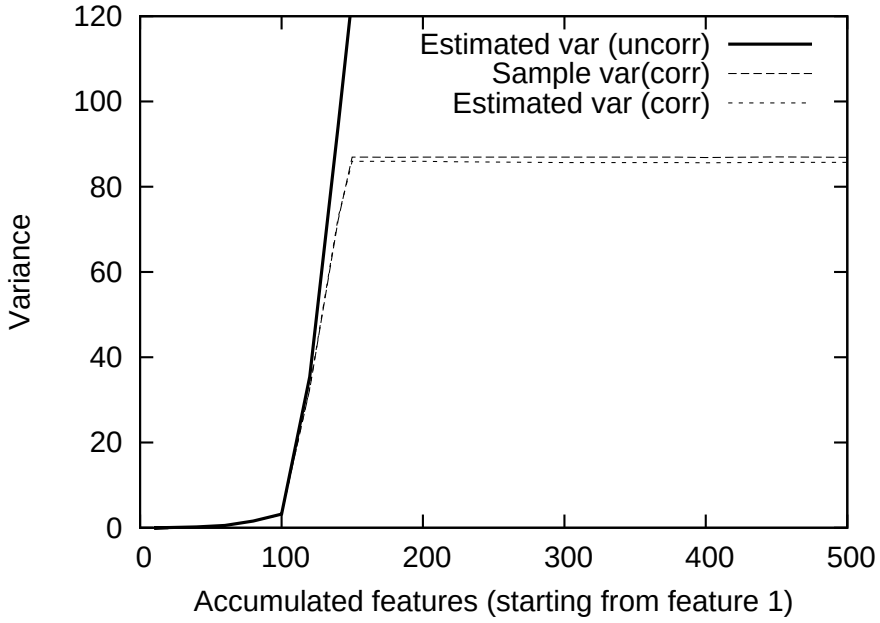

Figure 6.5: *Variance curves for L_{12}, given c_2, when incrementally adding features*

6.1.2.3 EFFECTS OF FEATURE ADDITION ON CLASSIFICATION ERROR RATE

Now we investigate the effect that feature selection has on the dissimilarity measure and error estimate ϵ (see Eq. 5.32). Figure 6.6 shows the empirical error rate (on a test set of 10000 samples per class) and the estimated error rate, while incrementally adding features.

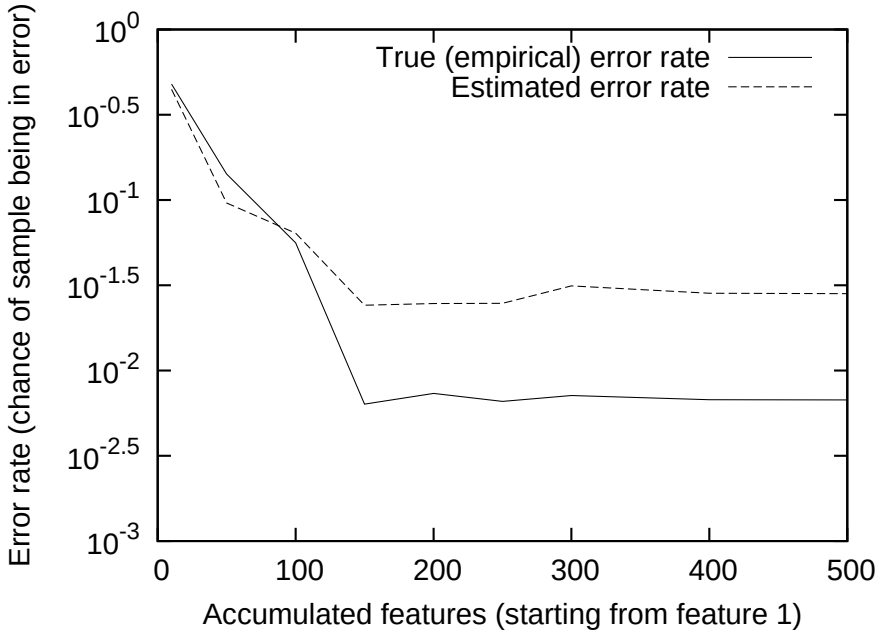

Figure 6.6: *Classification error rate ϵ of Bayesian classifier while incrementally adding features*

In Figure 6.6, the estimated error rate has the correct overall shape, but is proportionally less accurate when a large number of features are employed. This is a consequence of our assumption that all distributions are Gaussian. In fact, the distributions are somewhat skewed, especially for small values of m. The normal assumption is consequently less accurate for small error rates. Even though ϵ is a rather inaccurate estimate for the true error rate, it can still be used as a good dissimilarity measure for feature selection.

6.2 EXPERIMENTS AND THEORY ON TEXT-BASED LANGUAGE IDENTIFICATION FOR ALL ELEVEN OFFICIAL LANGUAGES OF SOUTH AFRICA

We apply the theory developed in Section 5.3 to compensate for data sparsity for text-based language identification for all eleven official languages of South Africa. This is a typical application where multinomial features are present. This section is divided into four subsections.

- Section 6.2.1 is concerned with the experimental setup for the text-based language identification application.
- Section 6.2.2 explains the issues that need to be resolved regarding the curse of dimensionality.
- Section 6.2.3 is a theoretical section that builds on Section 5.3 for calculating probability estimates of unseen features. The theory developed in this section is separate from Section 5.3

since it is specific to text-based language identification for the eleven official languages in South Africa.

- Section 6.2.4 focuses on the classification performance of the text-based language identification system and the improvements gained from finding good probability estimates of unseen features.

6.2.1 EXPERIMENTAL SETUP

6.2.1.1 TEXT CORPUS

The text corpus we used is identical to the one described by Botha and Barnard (2008). Text was used from various domains for all the official languages of South Africa using data obtained from Professor. D.J. Prinsloo of the University of Pretoria and also using a web crawler (Botha and Barnard, 2005). The data were pre-processed in UTF-8 format to handle different special characters that are unique for different languages (Botha and Barnard, 2008). We also removed all punctuation characters and made no distinction between lower and uppercase letters.

Nine of the official languages are from the Bantu family and two from the Germanic family (Afrikaans and English). The Bantu family can be subdivided into the Nguni subfamily, Sotho subfamily and two languages that are rather distant from these subfamilies, namely tshiVenda and Xitsonga. The Nguni subfamily consists of four languages that are very closely related (and poor classification performance is expected between these languages) namely isiNdebele, isiXhosa, isiZulu and siSwati. Finally, there are three languages from the Sotho subfamily, namely Sepedi, Sesotho and Setswana.

6.2.1.2 FEATURES USED

For this experiment, we assume a multinomial feature set where each dimension is described by counting 6-gram letter occurrences. Therefore, each entity is described by a unique six letter combination. We choose 6-grams because of its superior performance in text-based language identification as proposed in the literature (Botha and Barnard, 2008).

In total there are 42 unique letters in all the South African languages (Botha and Barnard, 2008), which implies a maximum dimensionality of $42^6 \approx 5.5 \times 10^9$.

6.2.1.3 DETAILS ON TRAINING AND TESTING

For all experiments we used ten fold cross-validation to test classification performance and we tried to ensure that each subset had data from a unique domain. This ensures that any test subset consists of data that is from a different domain than the training set, as far as possible.

We perform tests on 200K, 400K, 800K, 1.6M and 2.0M letters per language. More specifically, if we use the 200K data set, we divide it into ten 20K segments (for each language) and perform cross validation.

Tests are also performed on varying window sizes of 15, 100 and 300 characters. The window size is simply the number of characters that we regard as one test sample. For example if we refer to a

window size of 15, we classify the language based on 15 characters only. Also, since we use 6-grams, we can obtain $15 - 6 + 1 = 10$ entities from 15 characters.

Finally, we assume equal prior probabilities for all languages.

6.2.2 CHALLENGES REGARDING THE CURSE OF DIMENSIONALITY

It is clear that the dimensionality of the problem (5.5×10^9) is extremely high in comparison to the amount of training data (200K, 400K, etc). Therefore we expect the curse of dimensionality to play a role in two ways.

1. Parameter variability. Due to the data sparsity, we expect high variability in parameter estimates. We partially counter this problem by assuming independence between features and therefore using a NB classifier with multinomial features.

2. Feature sparsity. It is clear that in our training set, we do not observe all the entities (6-grams) that can occur in any given language. Therefore we use the techniques described in Section 5.3 to counter this problem. Note that this problem can be regarded as the limiting case of parameter variability where frequency counts would predict 0 probability of occurrence for unseen entities, whereas the true probability is larger, albeit small.

6.2.3 APPLICATION SPECIFIC THEORY FOR ESTIMATING UNSEEN ENTITY PROBABILITIES

It is clearly important to obtain non-zero entity probability estimates for unseen entities as described in Section 5.3. To do this we measure the probabilities described at the end of Section 5.3.2.1, namely $P_{c_i}(f|s_{c_i}, \overline{o_{c_i}})$, $P(\overline{o_{c_i}}|s_{c_k})$ and $P(o_{\overline{c_i}}|s_{c_k})$. We can generally divide these parameters into two classes:

- Intra-class parameters. These probabilities are independent of classes outside of c_i. We can measure $P_{c_i}(f|s_{c_i}, \overline{o_{c_i}})$ and $P(\overline{o_{c_i}}|s_{c_i})$ without taking c_k into consideration for $k \neq i$.

- Inter-class parameters. For these measurements, we need to consider how languages affect parameter estimates of other languages. These parameters include $P(\overline{o_{c_i}}|s_{c_k})$, $P(o_{\overline{c_i}}|s_{c_k})$ and $P(o_{\overline{c_i}}|s_{c_i})$, where $i \neq k$. Technically $P(o_{\overline{c_i}}|s_{c_i})$ is dependent on the other classes, since we are measuring probabilities of a entity occurring outside of c_i.

6.2.3.1 MEASURING $P_{c_i}(f|s_{c_i}, \overline{o_{c_i}})$ AND $P(\overline{o_{c_i}}|s_{c_i})$

$P_{c_i}(f|s_{c_i}, \overline{o_{c_i}})$ and $P(\overline{o_{c_i}}|s_{c_i})$ can be measured by observing the cumulative curve, which we call $F_{c_i}(x)$, representing the number of entities that are present in the training set while adding new multinomial trials. Figure 6.7 shows these cumulative curves for some of the South African languages on the 200K training set. Note that the number of multinomial trials in this figure is only 180K, since we reserve 20K for testing. It should also be mentioned that we randomly add new multinomial trials from the training set to average over all the domains present.

Much can be learned from the cumulative curve $F_{c_i}(x)$ if we approximate it with a continuous best fit. For instance, the slope of the curve represents the frequency with which new entities are

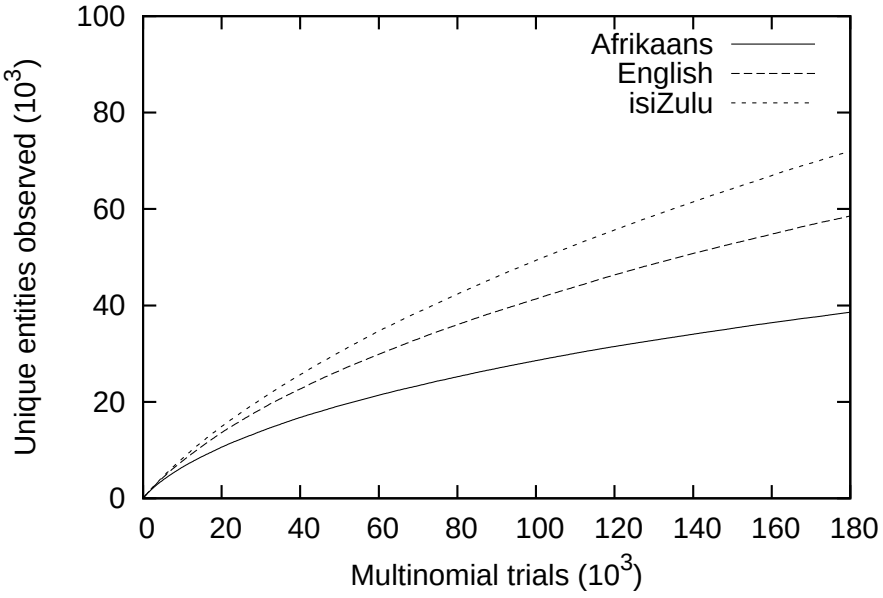

Figure 6.7: *The cumulative count of new entities observed in languages, while increasing the number of training samples*

observed in class c_i. Therefore the expected probability of observing a new entity in class c_i after observing x multinomial trials (from the train set) is given by the following equation.

$$P(\overline{o_{c_i}}|s_{c_i}, x) = f_{c_i}(x) = \frac{d}{dx}F_{c_i}(x) \tag{6.1}$$

We can also get a functional expression for the derivative in terms of the number of entities already seen.

$$P(\overline{o_{c_i}}|s_{c_i}, y) = g_{c_i}(y) = f_{c_i}(F_{c_i}^{-1}(y)), \tag{6.2}$$

where $F_{c_i}^{-1}$ is the inverse function of F_{c_i}.

To clarify the difference between $P(\overline{o_{c_i}}|s_{c_i}, x)$ and $P(\overline{o_{c_i}}|s_{c_i}, y)$, the first probability is in terms of the total number of multinomial trials observed in the training set (represented by x), and the second probability is in terms of the total number of unique entities observed (represented by y), where $y = F_{c_i}(x)$.

The rate (with regards to y) at which the expected probability of observing a new entity changes gives the expected probability of the next entity.

$$P_{c_i}(f|s_{c_i}, \overline{o_{c_i}}, y) = h_{c_i}(y) = -\frac{d}{dy}g_{c_i}(y) \tag{6.3}$$

We can calculate $P_{c_i}(f|s_{c_i}, \overline{o_{c_i}})$ and $P(\overline{o_{c_i}}|s_{c_i})$ by finding an analytical expression for $F_{c_i}(x)$ and observing its derivatives at the end of the curve, since we are interested in entity probabilities for a test vector after all the training data has been observed. Therefore, we get the following two expressions.

$$\begin{aligned} P_{c_i}(f|s_{c_i}, \overline{o_{c_i}}) &= h_{c_i}(y_{\max}) \\ P(\overline{o_{c_i}}|s_{c_i}) &= g_{c_i}(y_{\max}), \end{aligned} \tag{6.4}$$

where $y_{\max}$ is the total number of unique entities observed in the training set.

6.2.3.2 FINDING AN ANALYTICAL EXPRESSION FOR $F_{c_i}(x)$

In order to find the best functional form for $F_{c_i}(x)$ we plot $g_{c_i}(y)$. This can be done by obtaining an estimate for $f_{c_i}(x) = \frac{d}{dx}F_{c_i}(x)$ (we simply calculate average slopes every 100 samples) and plotting it against y (the number of unique entities observed). Figure 6.8 shows this plot for Afrikaans when using a 200K letter training set. Note that we plotted the log of $g_{c_i}(y)$ against y and not x. It appears that this figure becomes linear as y becomes large. In fact, this is true for all eleven languages (see appendix A.1 for these plots in all languages).

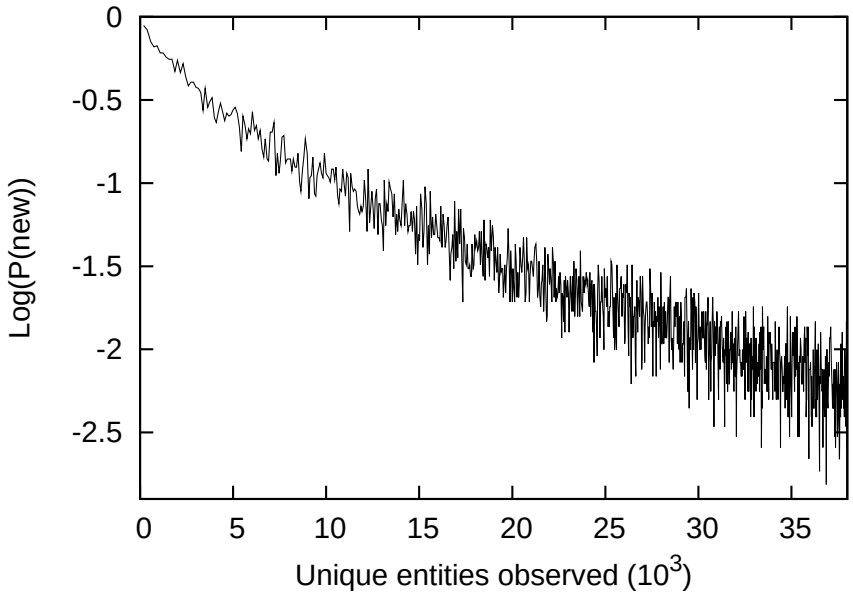

Figure 6.8: *The probability of observing a new unique entity in Afrikaans, while increasing the training set size*

We therefore fit an exponential curve to approximate $g_{c_i}(y)$.

$$\begin{aligned} \log(g_{c_i}(y)) &= \alpha y + log(A) \\ \therefore g_{c_i}(y) &= Ae^{\alpha y} \end{aligned} \tag{6.5}$$

Now we can finally get an expression for $y = F_{c_i}(x)$ by solving the following non-linear differ-

ential equation.

$$\begin{aligned} \frac{dy}{dx} &= Ae^{\alpha y} \\ \therefore log(\frac{dy}{dx}) &= log(A) + \alpha y \\ \therefore \frac{d^2y/dx^2}{dy/dx} &= \alpha \frac{dy}{dx} \\ \therefore \frac{d^2y}{dx^2} &= \alpha(\frac{dy}{dx})^2 \end{aligned} \tag{6.6}$$

In step two we took the logarithm on both sides of the equation and in step three we differentiated on both sides. Next, we solve for $u = \frac{dy}{dx}$.

$$\begin{aligned} \frac{du}{dx} &= \alpha u^2 \\ \therefore u &= -\frac{a}{\alpha}(ax+b)^{-1} \end{aligned} \tag{6.7}$$

Therefore we get the following expression for y.

$$y = \int u dx = -\frac{a}{\alpha}\int (ax+b)^{-1} dx = -\frac{1}{\alpha}\log(ax+b) + c \tag{6.8}$$

We can also solve for a as follows:

$$\begin{aligned} u &= Ae^{\alpha y} \\ \therefore -\frac{a}{\alpha}(ax+b)^{-1} &= Ae^{\alpha(-\frac{1}{\alpha}\log(ax+b)+c)} \\ &= Ae^{\alpha c}(ax+b)^{-1} \\ \therefore a &= -\alpha A e^{\alpha c} \end{aligned} \tag{6.9}$$

Substituting for a into $y = F_{c_i}(x)$ reveals:

$$\begin{aligned} F_{c_i}(x) &= -\frac{1}{\alpha}\log(-\alpha A e^{\alpha c}x + b) + c \\ &= -\frac{1}{\alpha}\log(e^{\alpha c}[-\alpha A x + be^{-\alpha c}]) + c \\ &= -\frac{1}{\alpha}\log(-\alpha A x + be^{-\alpha c}) \end{aligned} \tag{6.10}$$

If we substitute for $B = be^{-\alpha c}$ we get the final analytical expression for $y = F_{c_i}(x)$.

$$F_{c_i}(x) = -\frac{1}{\alpha}\log(-\alpha A x + B) \tag{6.11}$$

6.2.3.3 OPTIMISING FOR α, A AND B

We optimise α, A and B to fit $F_{c_i}(x)$ using a least square error approach. First we get good initial values for α and A by fitting the linear curve in Figure 6.8 and setting $B = 1$. These initial conditions are good enough to ensure that the closest local minimum of the squared error is also the

global minimum. After this, we optimise parameters α, A and B using a numerical steepest gradient descent approach. This is a first-order optimisation algorithm and only the error function's first order derivatives with respect to α, A and B need to be calculated. These derivatives are as follows:

$$\begin{aligned} E &= \frac{1}{2}\sum_{n=1}^{N}[f_{c_i}(x_n) - y_n]^2 \\ \therefore \frac{\partial E}{\partial \alpha} &= \frac{1}{\alpha}\sum_{n=1}^{N}[f_{c_i}(x_n) - y_n][f_{c_i}(x_n) + \frac{Ax_n}{B - \alpha A x_n}] \\ \frac{\partial E}{\partial A} &= \sum_{n=1}^{N} x_n \frac{f_{c_i}(x_n) - y_n}{B - \alpha A x_n} \\ \frac{\partial E}{\partial B} &= \frac{1}{\alpha}\sum_{n=1}^{N}\frac{f_{c_i}(x_n) - y_n}{\alpha A x_n - B}, \end{aligned} \tag{6.12}$$

where E is the square error, N is the total number of samples considered and (x_n, y_n) are the empirical samples.

Figure 6.9 shows the plot for the empirical $F_{c_i}(x)$ and the analytical best fit for a 200K character Afrikaans training set. Refer to Section A.2 for these figures in all eleven official languages of South Africa.

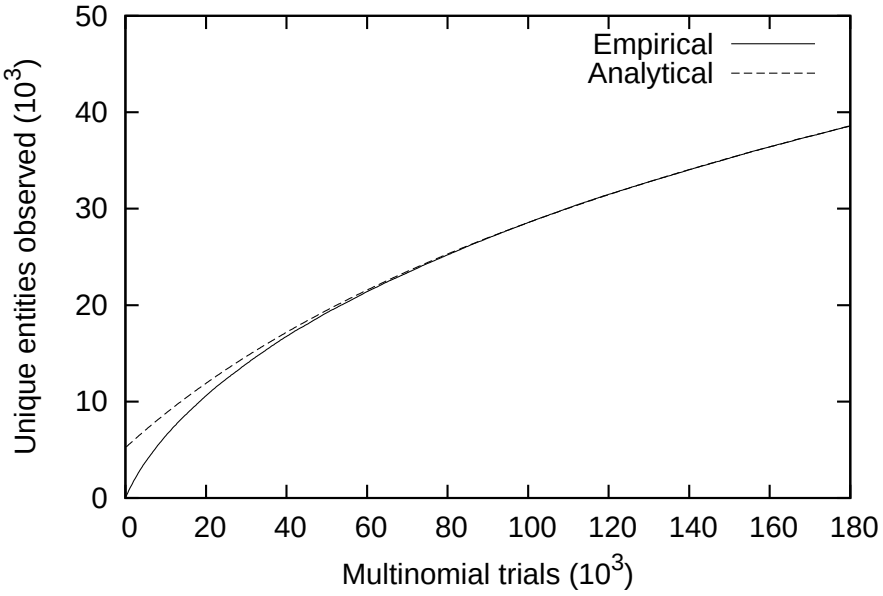

Figure 6.9: *The cumulative number of unique entities in Afrikaans, while increasing the training set size*

We can express $P_{c_i}(f|s_{c_i}, \overline{o_{c_i}})$ and $P(\overline{o_{c_i}}|s_{c_i})$ in terms of α and A as follows (see Eq. 6.4).

$$\begin{aligned} P_{c_i}(f|s_{c_i}, \overline{o_{c_i}}) &= h_{c_i}(y_{\max}) = -\frac{d}{dy}g_{c_i}(y_{\max}) = -\alpha A e^{\alpha y_{\max}} \\ P(\overline{o_{c_i}}|s_{c_i}) &= g_{c_i}(y_{\max}) = A e^{\alpha y_{\max}}, \end{aligned} \tag{6.13}$$

6.2.3.4 MEASURING $P(\overline{o_{c_i}}|s_{c_k})$

We are interested in finding the probability that a entity has never been observed inside of class c_i if it is currently observed from class c_k, where $i \neq k$. We can measure this quantity by accumulating the entities that are observed from the training set of class c_k that have never been observed inside of class c_i. For example in the figures below, we used a 200K character training set for each language. The optimal method for plotting these curves would be to first observe all the entities that occur in all languages except c_k and then incrementally add entities to c_k and see if they do not occur in class c_i. For convenience of implementation we differentially increase the number of entities for all classes together, though. $P(\overline{o_{c_i}}|s_{c_k})$ is simply the derivative of this curve at $x = x_{\max}$, where x_{max} is the last sample in the training set.

Figures 6.10 and 6.11 show these cumulative curves for $c_i = \text{Afrikaans}$, $c_k = \text{English}$ and $c_i = \text{isiNdebele}$, $c_k = \text{isiZulu}$ respectively.

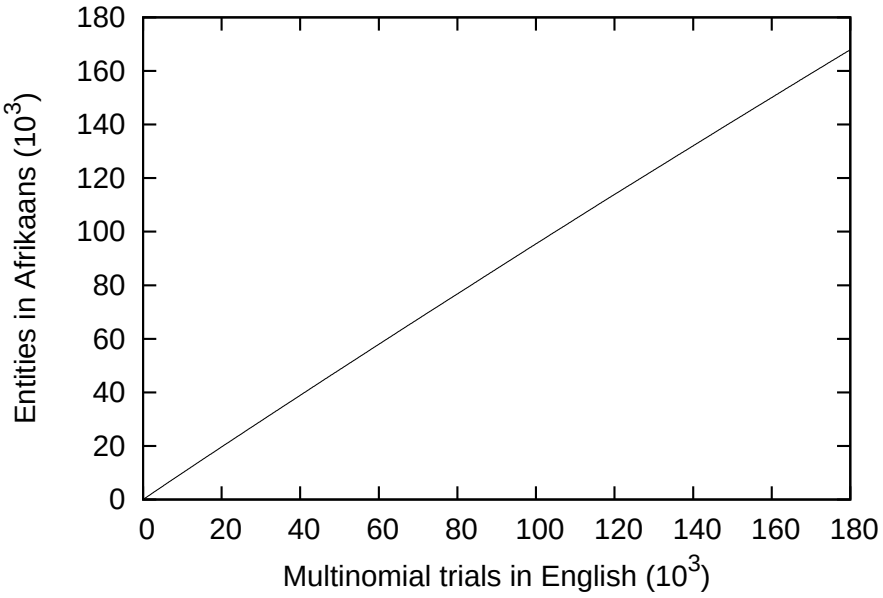

Figure 6.10: *The cumulative number of entities seen in English that have never been observed in Afrikaans, while increasing the training set size*

If c_i and c_k are completely disjoint we expect the cumulative curve to be linear with unit slope; Figure 6.10 is close to being a linear curve with unit slope and therefore it is clear that Afrikaans is practically disjoint from English. In contrast, isiNdebele is very closely related to isiZulu as suggested in Figure 6.11.

6.2.3.5 MEASURING $P(o_{\overline{c_i}}|s_{c_k})$

We are interested in finding the probability that an entity has been observed outside of class c_i if it is currently observed from class c_k. We can measure this quantity by accumulating the entities that are observed from the training set of class c_k that have also been observed outside of class c_i. For example in all the figures shown below, we used a 200K character training set for each language. The optimal method for plotting these curves would be to first observe all the entities that occur in all languages except c_k and then incrementally add entities to c_k and see if they also occur outside of

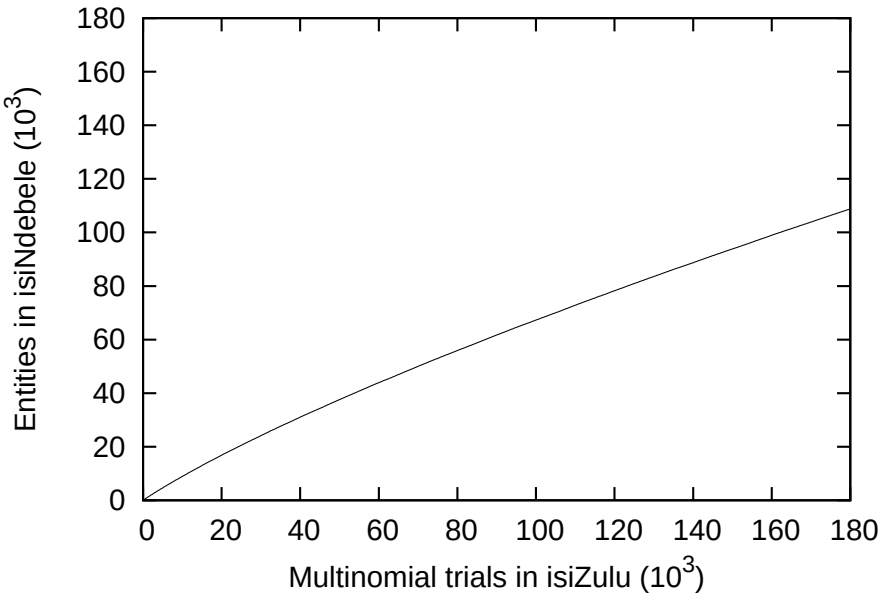

Figure 6.11: *The cumulative number of entities seen in isiZulu that have never been observed in isiNdebele, while increasing the training set size*

class c_i. For convenience of implementation we differentially increase the number of entities for all classes together, though. $P(o_{\overline{c_i}}|s_{c_k})$ is simply the derivative of this curve at $x = x_{\max}$, where x_{max} is the last sample in the training set.

Figures 6.12 and 6.13 show these cumulative curves for Afrikaans and isiZulu respectively, where we assume $i = k$.

It is clear from Figures 6.12 and 6.13 that $P(o_{\overline{c_i}}|s_{c_i})$ is much larger for isiZulu than Afrikaans, since Afrikaans is distant from all other South African languages, whereas isiZulu is closely related to other Nguni languages (isiNdebele, isiXhosa and siSwati).

We can get similar cumulative plots to derive $P(o_{\overline{c_i}}|s_{c_k})$, where $i \neq k$. Figures 6.14 and 6.15 show these plots for $c_i = \text{Afrikaans}$, $c_k = \text{English}$ and $c_i = \text{isiNdebele}$, $c_k = \text{isiZulu}$ respectively.

6.2.3.6 REGULARISATION OF UNSEEN ENTITY PARAMETERS AND EXPERIMENTS

In the previous sections we showed methods for calculating $P_{c_i}(f|\overline{o_{c_i}}, o_{\overline{c_i}})$ (see Eq. 5.40). These are the entity probabilities that we assign to each entity that has never been observed before in class c_i, but are present in other classes (which we will refer to as the penalty factors from now on). Remember that we ignore all entities that have never been seen at all. Also note that these probabilities are independent of the window size (number of text characters used per test vector). Table 6.1 shows these values for all languages with different training set sizes.

Note that in this table, we averaged over all language penalty factors to get one penalty factor for all languages. This provides a form of regularisation that is required due to high variability of these penalty factors. Table 6.2 shows how the regularised penalty factor consistently outperforms those that are unique for each individual language. Error rates are calculated using 10 fold cross-validation and we use different character window sizes. Further results on the error performance and penalty factors are shown later.

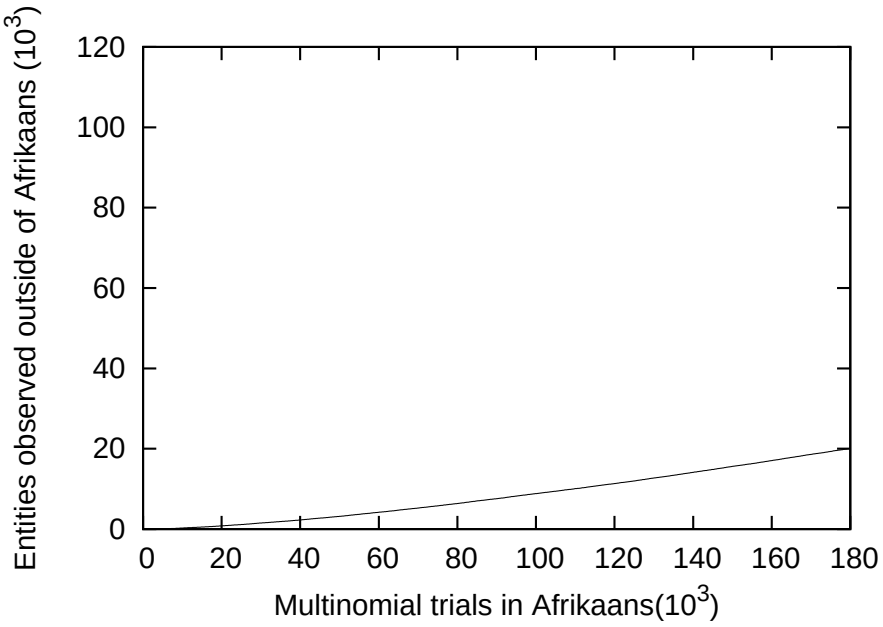

Figure 6.12: *The cumulative number of entities seen in Afrikaans that have also been observed outside of Afrikaans, while increasing the training set size*

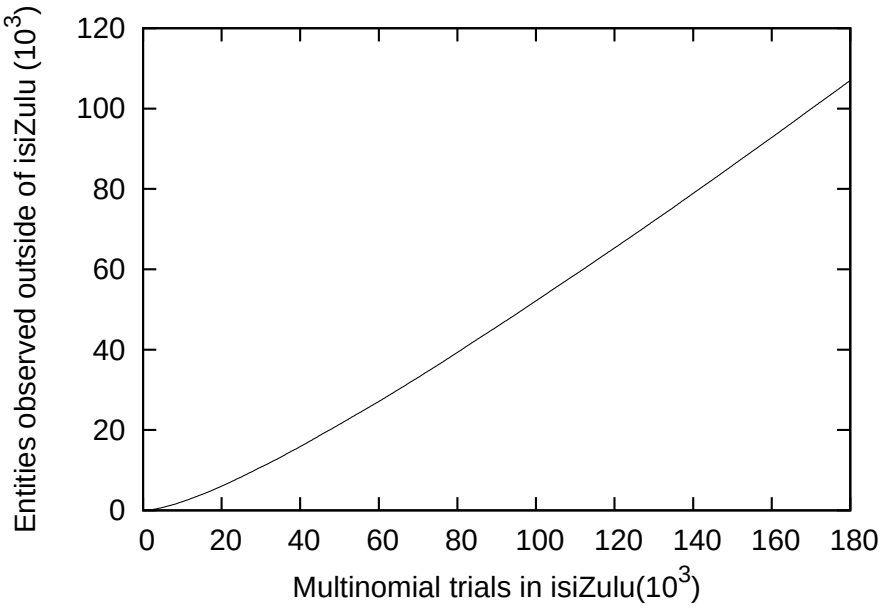

Figure 6.13: *The cumulative number of entities seen in isiZulu that have also been observed outside of isiZulu, while increasing the training set size*

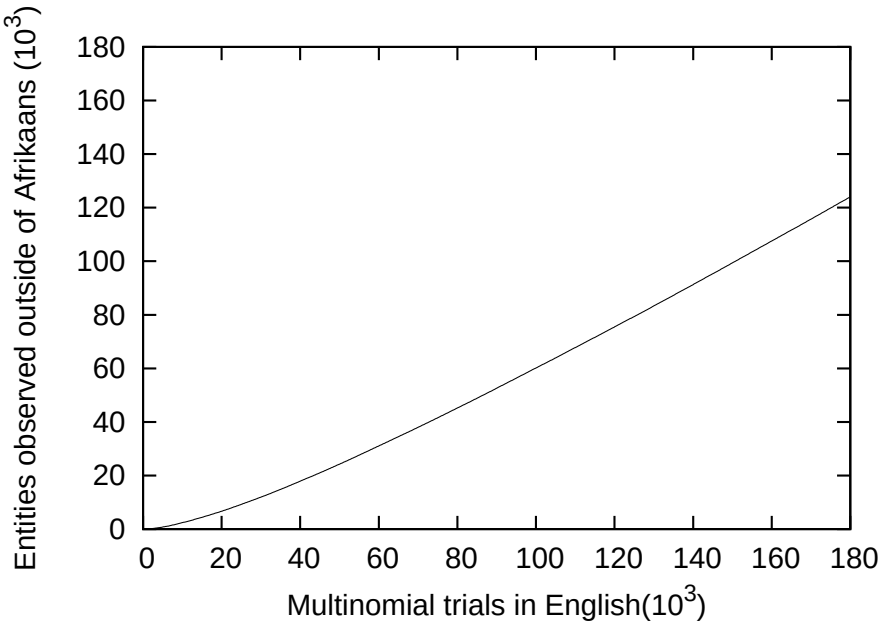

Figure 6.14: *The cumulative number of entities seen in English that have also been observed outside of Afrikaans, while increasing the training set size*

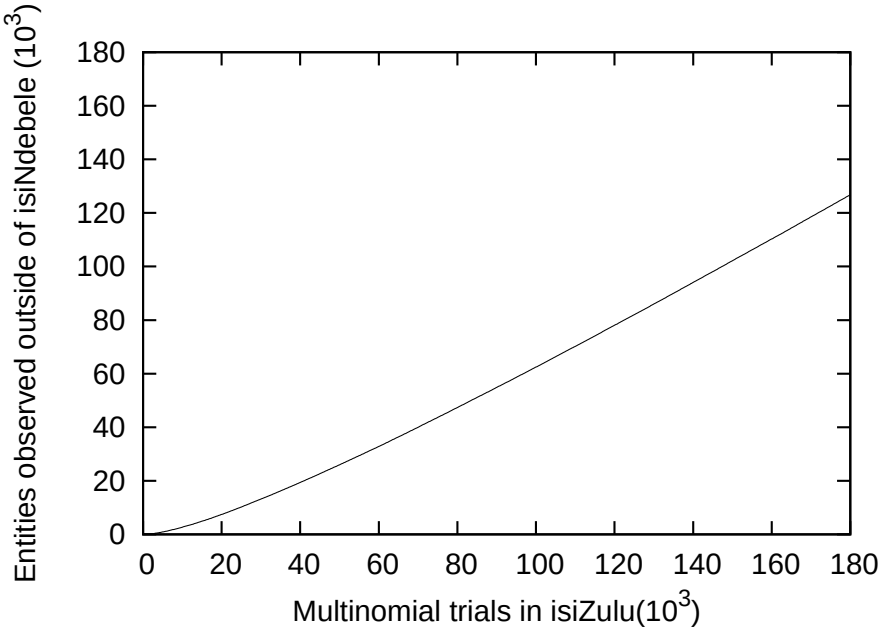

Figure 6.15: *The cumulative number of entities seen in isiZulu that have also been observed outside of isiNdebele, while increasing the training set size*

Table 6.1: Penalty factors for unseen entities in all classes while varying the training set size (independent of window size)

Language	**Penalty factors**				
Training Set size (per language)	**200K**	**400K**	**800K**	**1.6M**	**2.0M**
Afrikaans	7.41e-9	4.14e-9	2.27e-9	1.43e-9	1.08e-9
English	2.08e-8	9.62e-9	4.46e-9	2.02e-9	1.48e-9
isiNdebele	6.31e-8	2.79e-8	1.10e-8	4.04e-9	3.21e-9
isiXhosa	6.78e-8	3.19e-8	1.39e-8	5.58e-9	4.19e-9
isiZulu	7.83e-8	3.38e-8	1.35e-8	4.96e-9	3.54e-9
siSwati	2.23e-8	9.72e-9	4.04e-9	2.43e-9	1.94e-9
Sepedi	4.34e-8	1.61e-8	5.86e-9	2.06e-9	1.45e-9
Sesotho	4.79e-8	1.91e-8	6.76e-9	2.30e-9	1.66e-9
Setswana	4.79e-8	1.87e-8	6.52e-9	1.98e-9	1.42e-9
tshiVenda	2.24e-8	8.84e-9	4.00e-9	1.92e-9	1.46e-9
Xitsonga	2.94e-8	1.15e-8	5.21e-9	2.10e-9	1.52e-9
Average	4.10e-8	1.74e-8	7.04e-9	2.80e-9	2.09e-9

Table 6.2: Effect of regularised common penalty factor on classification performance

	Classification error rate (%)					
	200K		**400K**		**800K**	
Window size (characters)	direct	regularised	direct	regularised	direct	regularised
15	20.23	20.20	18.35	18.31	16.35	16.33
100	3.29	3.16	1.97	1.91	1.46	1.40
300	1.13	1.01	0.64	0.60	0.59	0.54
	1.6M		**2.0M**			
Window size (characters)	direct	regularised	direct	regularised		
15	16.36	16.35	15.65	15.64		
100	1.56	1.53	1.49	1.46		
300	0.60	0.59	0.67	0.67		

6.2.4 CLASSIFICATION PERFORMANCE OF 6-GRAM NAIVE BAYESIAN CLASSIFIER

If we combine all the theory above for calculating feature parameters, we can calculate error curves for the 6-gram NB classifier on all eleven South African languages.

6.2.4.1 ERROR CURVES FOR VARIOUS TRAINING SET SIZES

Figures 6.16, 6.17 and 6.18 show the error performance for different window sizes and different training set sizes. These figures also show the expected error rates when we do not compensate for feature sparsity ($P_{c_i}(f|\overline{o_{c_i}}, o_{\overline{c_i}}) =$ 1e-99).

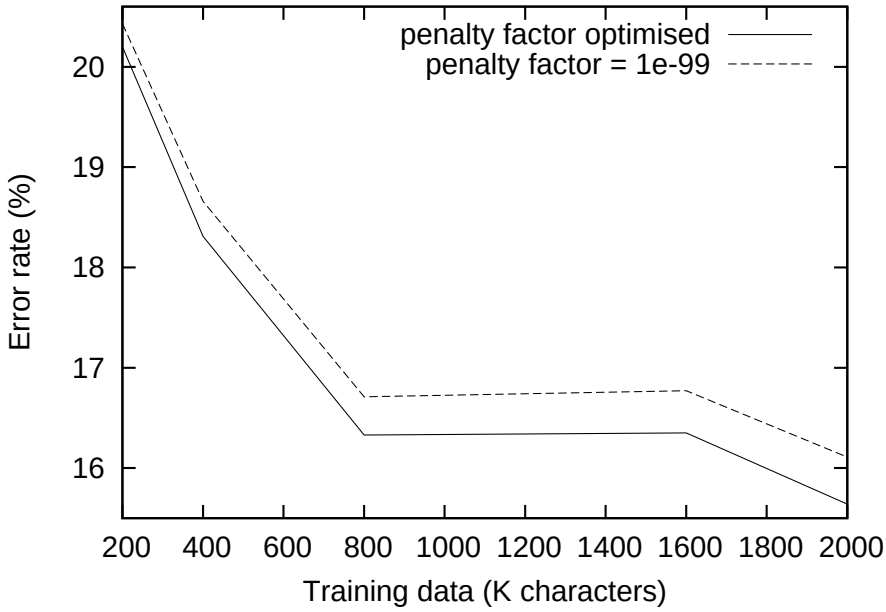

Figure 6.16: *6-gram NB classifier error performance while varying the training set size with a window size of 15 characters*

Note that all the error rates in Figures 6.16 to 6.18 start converging after 800K training samples. This is most likely due to the fact that the parameter variability remains constant for a given training set size and is independent of the number of multinomial trials (the window size).

Also note that the 300 character window size shows an increase in error rates for large training sets. This is most likely due to impurities in the test data (wrongly labeled languages) that is not equally distributed over all the test set sizes (the 2000K test set contains data from domains in the Bantu families, with many English words, that is not present in the 1600K or lower test sets). Otherwise we would expect the 15 and 100 character window sizes to show proportional error increases. The error increase due to impurities is expected to be constant for different window sizes and is therefore not noticeable for the 15 and 100 character window sizes (since an error increase of say 0.1% would hardly be noticeable for the 15 character window size where the error is in the range of 17%).

Finally, the error performances show greater sensitivity to larger window sizes since one would expect a larger number of unseen entities to occur in these windows.

Table 6.3 summarises the error rates obtained when compensating for feature sparsity as shown in Figures 6.16, 6.17 and 6.18.

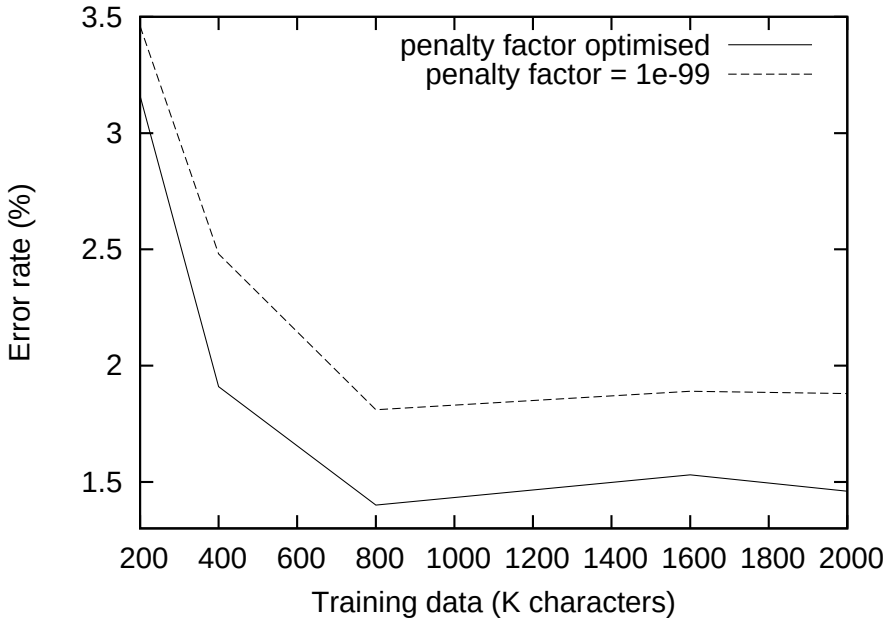

Figure 6.17: *6-gram NB classifier error performance while varying the training set size with a window size of 100 characters*

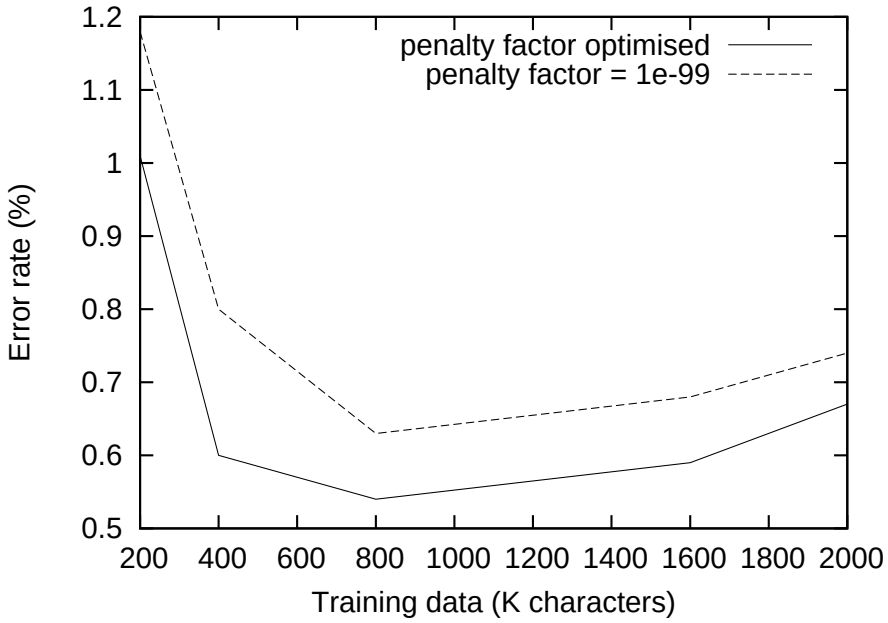

Figure 6.18: *6-gram NB classifier error performance while varying the training set size with a window size of 300 characters*

Table 6.3: Classifier performance for varying training set sizes and window sizes

error rate (%)	**200K**	**400K**	**800K**	**1.6M**	**2.0M**
15 Chars	20.20	18.31	16.33	16.35	15.64
100 Chars	3.16	1.91	1.40	1.53	1.46
300 Chars	1.01	0.60	0.54	0.59	0.67

6.2.4.2 CONFUSION MATRICES AND INTERPRETATION

It is clear from section 6.2.4.1 that the language classifier performs significantly worse for a 15 character window size and the error rate decreases as we increase the character window size.

Table 6.4 shows the confusion matrix for classifying on 15 characters with a training set size of 200K characters per language, where the rows represent the true classes and the columns the predicted classes. Table 6.5 shows the error probabilities calculated from the confusion matrix for each language.

Table 6.4: Confusion matrix for language identification trained on 200K characters, for a window size of 15 characters

	Afrikaans	**English**	**isiNdebele**	**isiXhosa**	**isiZulu**	**siSwati**	**Sepedi**	**Sesotho**	**Setswana**	**tshiVenda**	**Xitsonga**
Afrikaans	**98.11**	1.28	0.06	0.05	0.08	0.04	0.06	0.08	0.08	0.08	0.09
English	1.25	**97.65**	0.06	0.07	0.16	0.07	0.18	0.14	0.13	0.11	0.20
isiNdebele	0.36	0.52	**63.84**	12.39	16.72	3.31	0.39	0.50	0.41	0.63	0.94
isiXhosa	0.60	0.40	16.29	**58.52**	19.01	2.18	0.32	0.52	0.38	0.58	1.22
isiZulu	0.44	0.99	14.27	15.09	**64.09**	3.18	0.33	0.40	0.20	0.31	0.71
siSwati	0.02	0.25	2.22	0.84	1.72	**94.26**	0.13	0.13	0.09	0.15	0.20
Sepedi	0.12	0.47	0.30	0.23	0.47	0.14	**73.26**	11.03	13.31	0.25	0.43
Sesotho	0.32	0.42	0.20	0.24	0.32	0.06	10.31	**74.22**	13.11	0.38	0.43
Setswana	0.49	0.27	0.17	0.13	0.30	0.14	15.45	16.19	**65.87**	0.42	0.58
tshiVenda	0.34	0.71	0.40	0.37	0.41	0.26	0.38	0.56	0.56	**94.72**	1.31
Xitsonga	0.44	0.80	0.61	0.53	0.84	0.44	0.52	0.66	0.56	1.41	**93.20**

Table 6.5: Error rates for languages calculated from the confusion matrix in Table 6.4

Afrikaans	**English**	**isiNdebele**	**isiXhosa**	**isiZulu**	**siSwati**	**Sepedi**	**Sesotho**	**Setswana**	**tshiVenda**	**Xitsonga**	**Overall**
1.89	2.35	36.16	41.48	35.91	5.71	26.74	25.78	34.13	5.28	6.80	20.20

Table 6.6: Error rates for subsets, using the 200K training set on a 15 character window.

Subset	1	2	3	4	5	6	7	8	9	10
Average error	22.19	19.83	19.93	18.67	18.47	20.48	23.23	19.67	18.00	21.56

The confusion matrix and error rates in Tables 6.4 and 6.5 are the average performance rates over all ten test subsets that we use for cross-validation. The predicted error rate for each subset is shown in Table 6.6 and the standard deviation across the subsets is 1.60%.

It is clear that most of the errors that occur in Table 6.4 are due to confusion between languages in the same subfamilies. The confusion matrix for classification between the different subfamilies is

Table 6.7: Confusion matrix for language families trained on 200K characters, for a window size of 15 characters

	Afrikaans	**English**	**Nguni family**	**Sotho family**	**tshiVenda**	**Xitsonga**
Afrikaans	**98.11**	1.28	0.23	0.22	0.08	0.09
English	1.25	**97.65**	0.36	0.45	0.11	0.20
Nguni family	0.36	0.54	**96.98**	0.95	0.42	0.77
Sotho family	0.31	0.39	0.9	**97.58**	0.35	0.48
tshiVenda	0.34	0.71	1.44	1.5	**94.72**	1.31
Xitsonga	0.44	0.80	2.42	1.74	1.41	**93.20**

shown in Table 6.7. The error rate between language families is 3.24% (equal prior probability for each language), and is much lower that the error rate between languages as expected.

6.2.4.3 COMPARING ERROR PERFORMANCE FOR DIFFERENT PENALTY FACTORS

We test the NB classifier's performance for different penalty factors and compare it to the performance obtained when using the estimated penalty factors.

Figures 6.19 to 6.21 shows the error performances obtained (with 15, 100 and 300 character window sizes) for the text-based language identification system, while varying the penalty factors from 1e-15 to 1e-6 for the 200K training set. Similar figures are shown for the 400K, 800K, 1.6M and 2.0M training set sizes in Section A.3

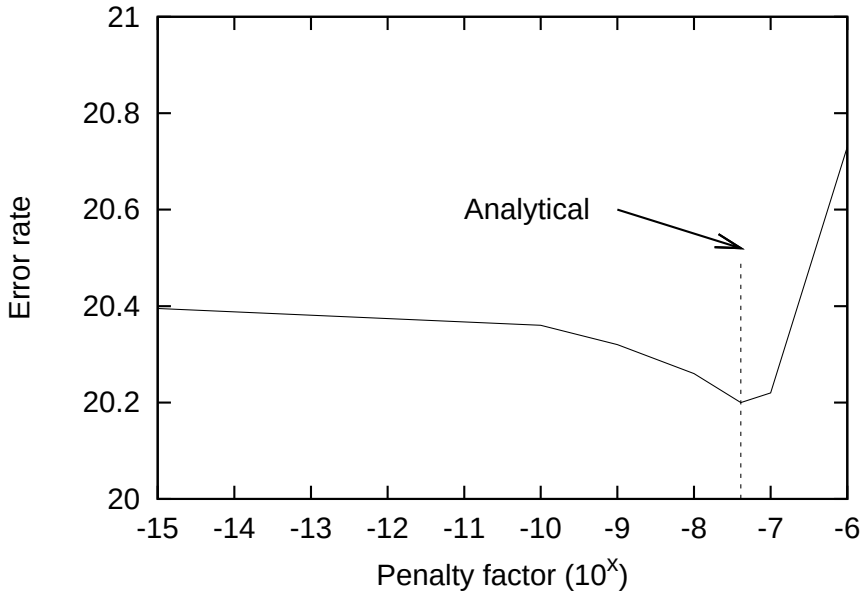

Figure 6.19: *6-gram NB classifier error performance while varying the unseen entity penalty factor for a 15 character window size, 200K characters training set*

From these figures it is clear that the optimal penalty factors do not depend strongly on the window size used for classification. The classifier performance is suboptimal for very small penalty factors (say 1e-99), since any unseen entity would dominate the log likelihood probabilities and the classi-

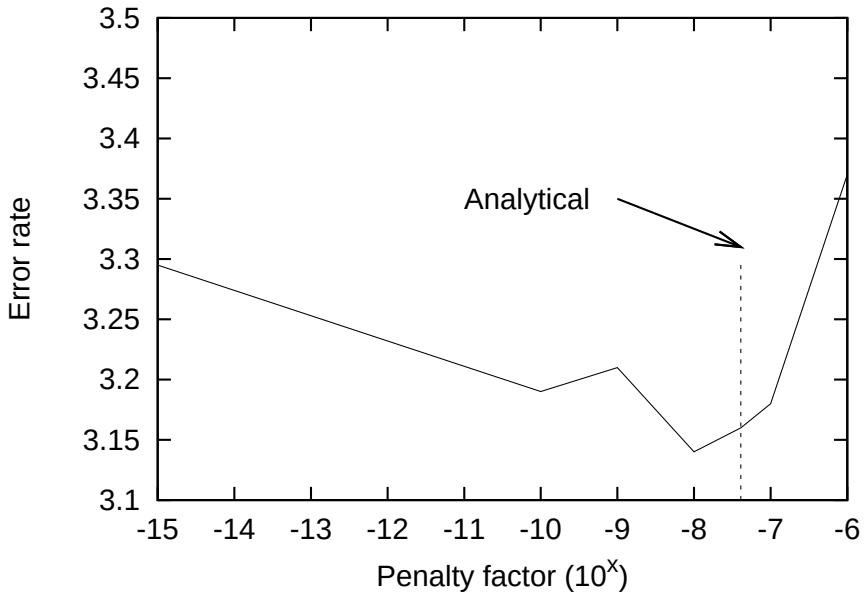

Figure 6.20: *6-gram NB classifier error performance while varying the unseen entity penalty factor for a 100 character window size, 200K characters training set*

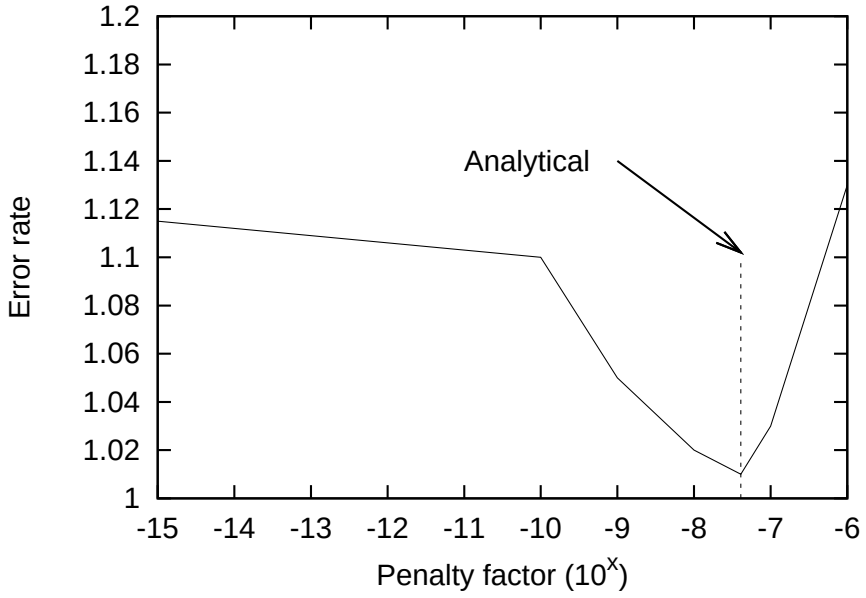

Figure 6.21: *6-gram NB classifier error performance while varying the unseen entity penalty factor for a 300 character window size, 200K characters training set*

fier would effectively choose the class with the least number of unknown entities. In contrast, the classifier's performance would deteriorate significantly when we use large penalty factors (say 1e-6 and higher) since we are effectively favoring unseen entities. In the limiting case where we assigning a penalty factor of 1.0 or ignore the unseen entity terms from Eq. 5.2 ($log(1.0) = 0$) the classifier would choose the class with the highest number of unseen entities; clearly nonsensical.

6.3 CONCLUSION

We have found that we can derive a useful error estimate for feature selection in NB classifiers with multinomial features by approximating the likelihood functions with Gaussian probability distributions. Even though the error estimate is somewhat inaccurate for low error rates, it still serves as a good dissimilarity measure between the different class distributions. The strong point of the analysis above is that the means and variances of all the likelihood distributions are calculated accurately, independently of the final distribution. The only assumption made on the distributions is that they are composed of a sum of multinomial variables.

Usually, the likelihood distributions become Gaussian (and therefore provide accurate error analysis) when many multinomial trials are taken (high value for m), the dimensionality is high (many features included in the analysis) and the total frequency count on the used feature space is not too low.

Unfortunately, for applications such as text-based language identification, we cannot assume perfect Gaussian distributions. For example, we would like to use small values for m. In Section 6.1 we used $m = 10$ and the effect of this can be seen by observing skewed probability curves that would typically result in poor error estimates for low error rates (since the tail of the distribution is somewhat inaccurate). In future research we will investigate whether more accurate estimates can be derived by correcting for this deviation from normality.

In the analysis for finding error curves, all the entity probabilities were assumed to be fixed. In real life applications, these probabilities have to be estimated from training data and are therefore random variables themselves. When the problem expands into extremely high dimensional spaces the entity probability estimates become inaccurate. To model this fact, one could follow a Bayesian approach and take expectations over the distributions of the entity probabilities (using, for example, Dirichlet conjugate priors). We prefer not to take this course for reasons described in Section 2.1.2. Instead, we deal with the data sparsity problem by assuming independence between features and are able to find parameter estimates for entities that have never been seen in the training set by following a frequentist approach. Even though the entity probability estimates become inaccurate (with an increase in dimensionality), the distributions of different classes might become more mutually exclusive and therefore the NB classifier that is trained on possibly inaccurate probabilities might still perform very well (and often much better than a lower dimensional classifier). For example, in text-based language identification, we can increase the performance by increasing the dimensionality of the problem (increasing n for the n-grams), even though the entity probability estimates become inaccurate (Botha and Barnard, 2008).

CHAPTER SEVEN

CONCLUSION

7.1 DISCUSSION

In this dissertation we developed theoretical models to analyse Gaussian and multinomial distributions. The analysis is focused on classification in high dimensional feature spaces and provides a basis for dealing with issues such as data sparsity and feature selection (for Gaussian and multinomial distributions, two frequently used models for high dimensional applications). We followed a Naive Bayesian philosophy to deal with issues associated with the curse of dimensionality.

The core focus of this dissertation was on calculating analytical error rates for high dimensional feature spaces with the following posterior distributions:

- Correlated Gaussian distributions. We derived exact analytical solutions for calculating error rates of binary classifiers with arbitrary dimensions given any quadratic decision boundary (except for the degenerate paraboloidal boundaries).

- Multinomial distributions. These feature spaces arise when we count frequencies of feature occurrences in a number of multinomial trials. We derived approximate analytical expressions for the error rates in high dimensional feature spaces.

We compared the analytical error expressions with estimates obtained from Monte-Carlo simulations for the following two case studies:

- Binary classifiers with artificially generated Gaussian features. We created both two and twelve dimensional classification problems and tested the error rates for two different decision boundaries: Optimal Bayes boundaries and NB boundaries. These tests showed that the analytical expressions are exact and also illustrated that NB classifiers are robust in sparse data environments.

- Binary classifiers with artificially generated multinomial data sets. We created a 500 dimensional problem, where a given entity probability was assigned to each of the 500 features for

each class, and showed that the analytical error estimates are comparable to those obtained from empirical simulations.

It is interesting to compare the behaviour of classifiers with Gaussian and multinomial features in high dimensional spaces. Both present problems regarding data sparsity and we decrease parameter variability (at the cost of high biases) by assuming that all features are independent. In fact, we often do not have a choice but to assume independence in very sparse problems since the distributions describing correlations would present high variances. With this in mind, we should realise that multinomial features are still uniquely different from Gaussian features: With Gaussian features we always observe some measurement for all the features and with a sufficient number of training samples, we can get accurate NB Gaussian parameter estimates in each dimension, regardless of the dimensionality. On the other hand, in the multinomial case we use feature counts over m multinomial trials (for one vector) and most of these features do not occur in such a vector (feature counts are predominantly zero and some of the features have never even been observed in the training set).

Therefore multinomial features present an extra data sparsity issue which we call feature sparsity. This type of sparsity arises since certain features have never been observed in the training set before. The conventional maximum likelihood estimates for the entity probabilities for such features would result in 0 probability. It turns out that the classification error rates of NB classifiers with multinomial features are somewhat sensitive to the entity probability estimates that we use for unseen features. Therefore, the final set of theory deals with feature sparsity by providing proper non-zero entity probability estimates for unseen features.

Finally, we applied the theory on dealing with feature sparsity in a text-based language identification application for all eleven official languages of South Africa and showed that the entity probability estimates for unseen features performed close to optimally (if we assume that all unseen features have the same entity probability).

7.2 FUTURE WORK

The theories developed in this dissertation can be regarded as fundamental tools for investigating classifier behaviours in high dimensional feature spaces. Further research includes:

- A detailed comparison between NB and optimal Bayes classifiers for high dimensional Gaussian data sets.
- Feature selection for multinomial and Gaussian distributed data.
- A comparison between the frequentist theory developed for dealing with feature sparsity and a Bayesian approach (for multinomial distributions).
- We can compare error performances, for high dimensional multinomial distributions, between NB classifiers and SVM classifiers. Although SVM classifiers might be impractical (due to slow training time) for extremely high dimensional feature spaces, there are techniques for decreasing SVM training times in spaces where input variables are predominantly zero in value (Joachims, 2006).

APPENDIX A

TEXT-BASED LANGUAGE IDENTIFICATION FIGURES

A.1 PROBABILITY OF OBSERVING NEW 6-GRAMS IN LANGUAGES WITH A 200K CHARACTER TRAINING SET

In this section we plot the approximate probabilities for observing new 6-gram entities in all the 11 official languages of South Africa, while incrementally increasing the training set sizes up to 200K characters per language. Refer to Section 6.2.3.2 for more detail on the significance of these plots.

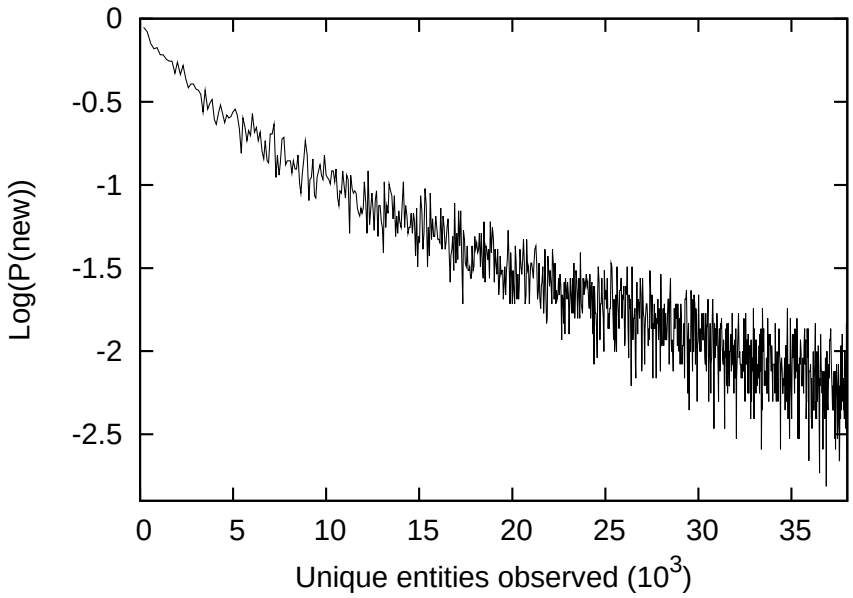

Figure A.1: *The probability of observing a new unique entity in Afrikaans, while increasing the training set size*

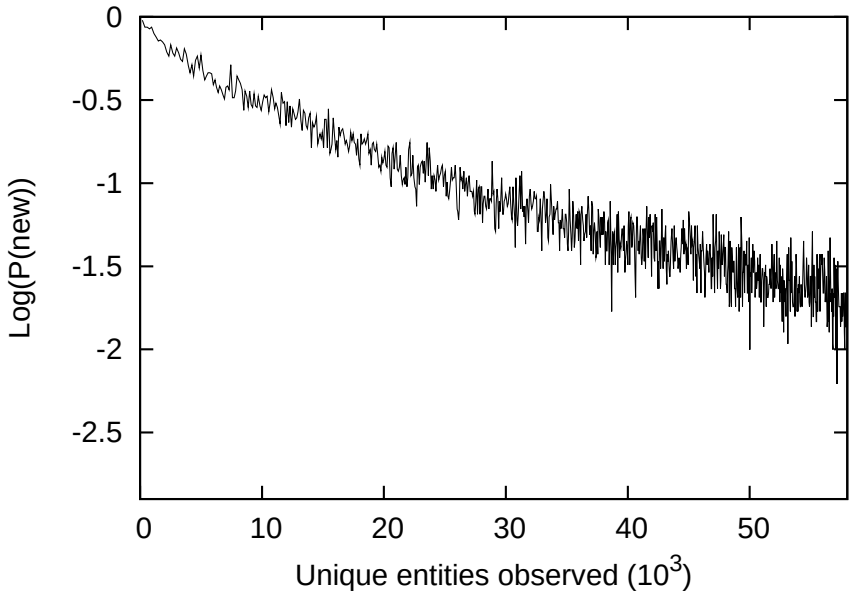

Figure A.2: *The probability of observing a new unique entity in English, while increasing the training set size*

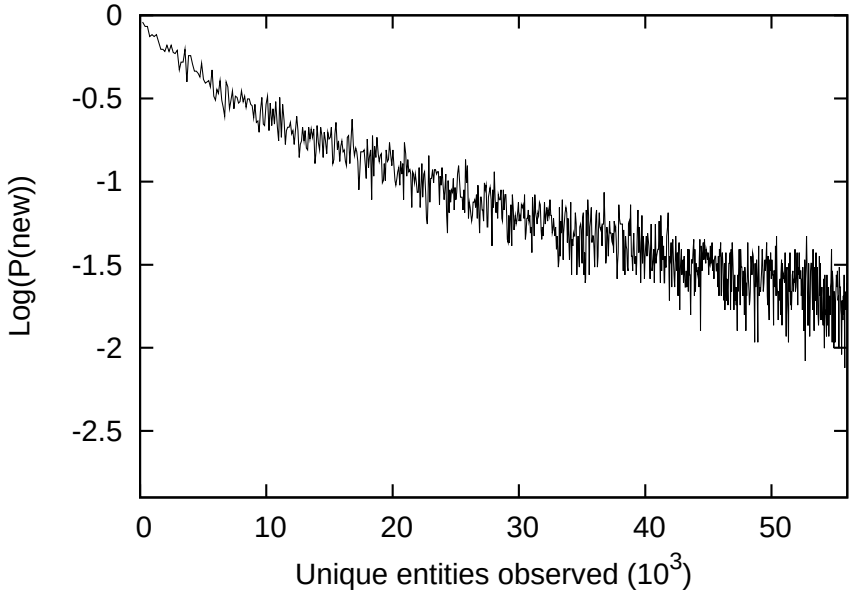

Figure A.3: *The probability of observing a new unique entity in isiNdebele, while increasing the training set size*

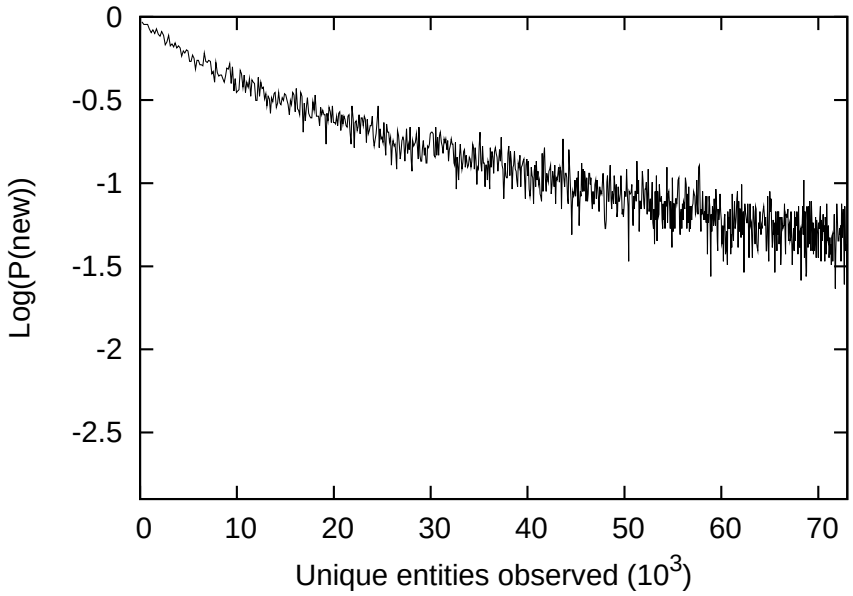

Figure A.4: *The probability of observing a new unique entity in isiXhosa, while increasing the training set size*

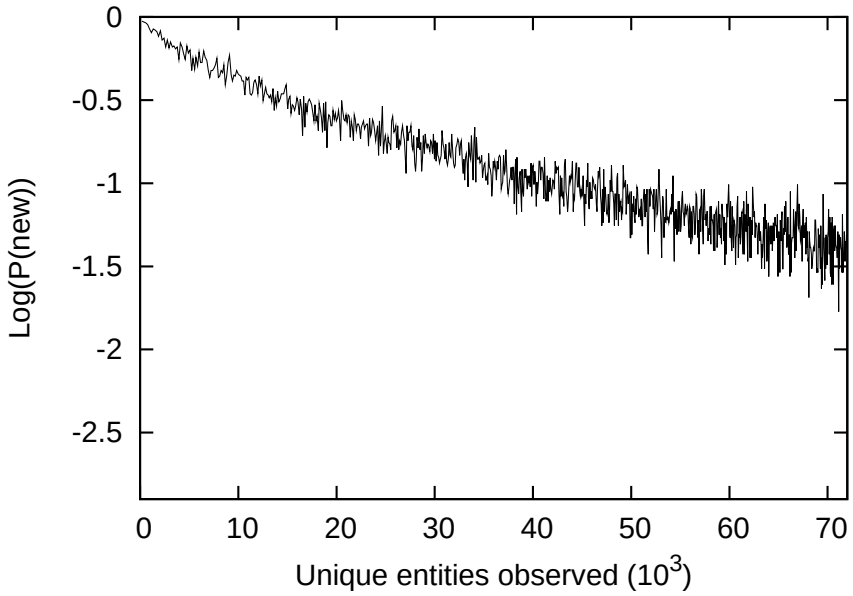

Figure A.5: *The probability of observing a new unique entity in isiZulu, while increasing the training set size*

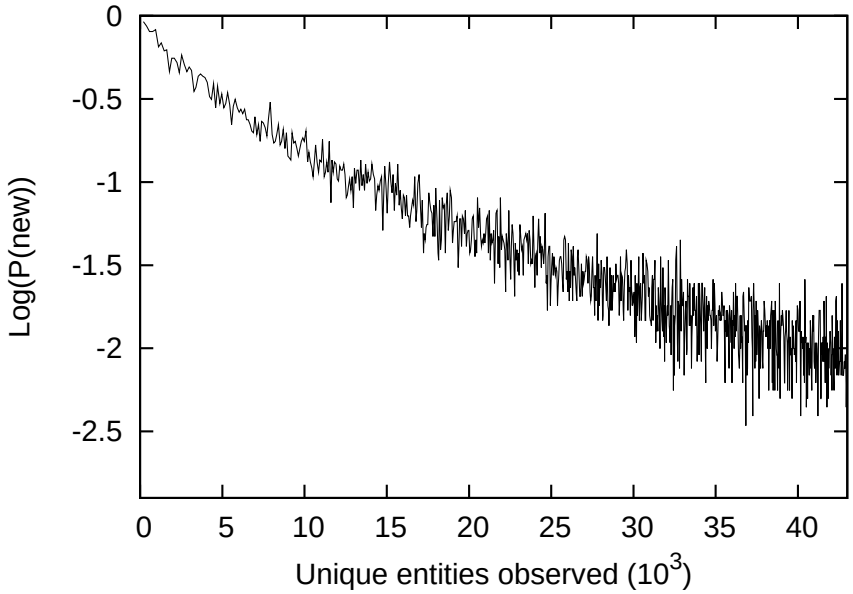

Figure A.6: *The probability of observing a new unique entity in Sepedi, while increasing the training set size*

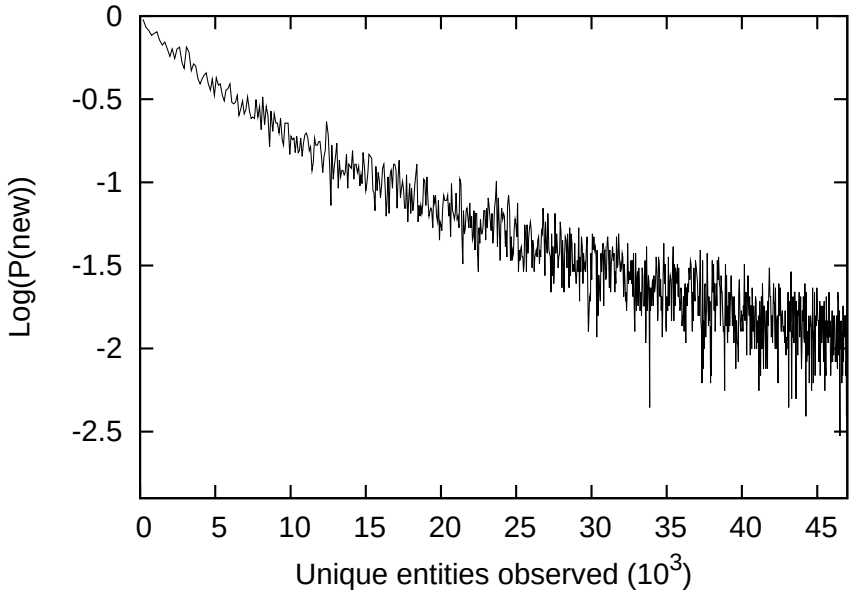

Figure A.7: *The probability of observing a new unique entity in Sesotho, while increasing the training set size*

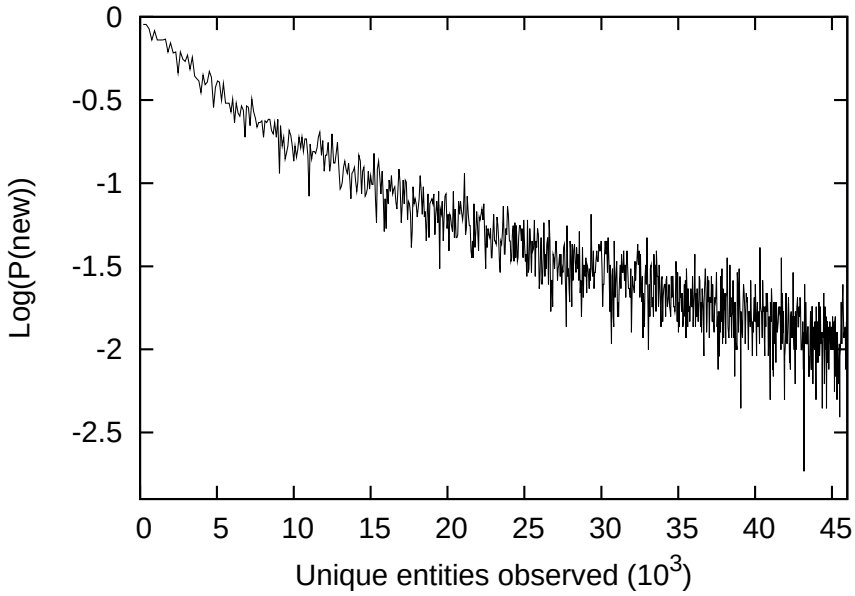

Figure A.8: *The probability of observing a new unique entity in Setswana, while increasing the training set size*

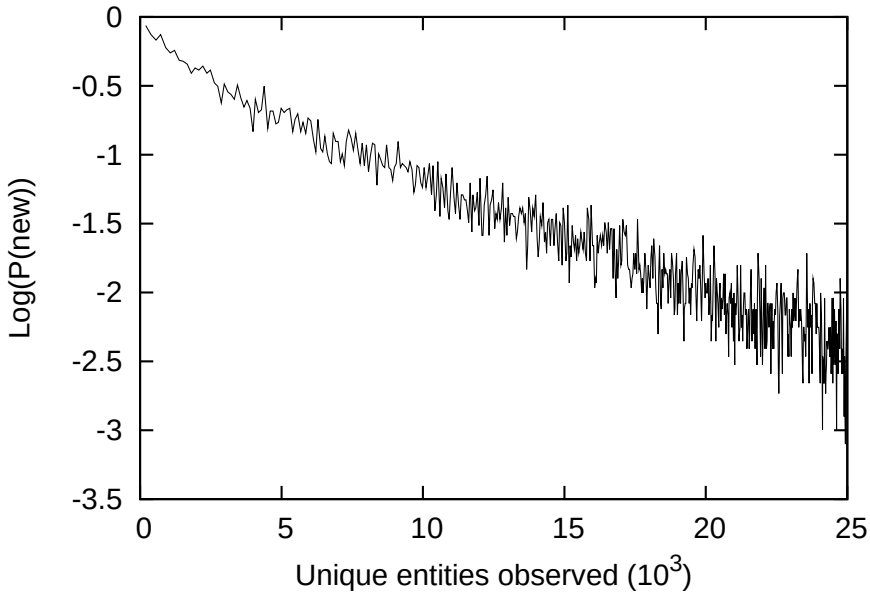

Figure A.9: *The probability of observing a new unique entity in siSwati, while increasing the training set size*

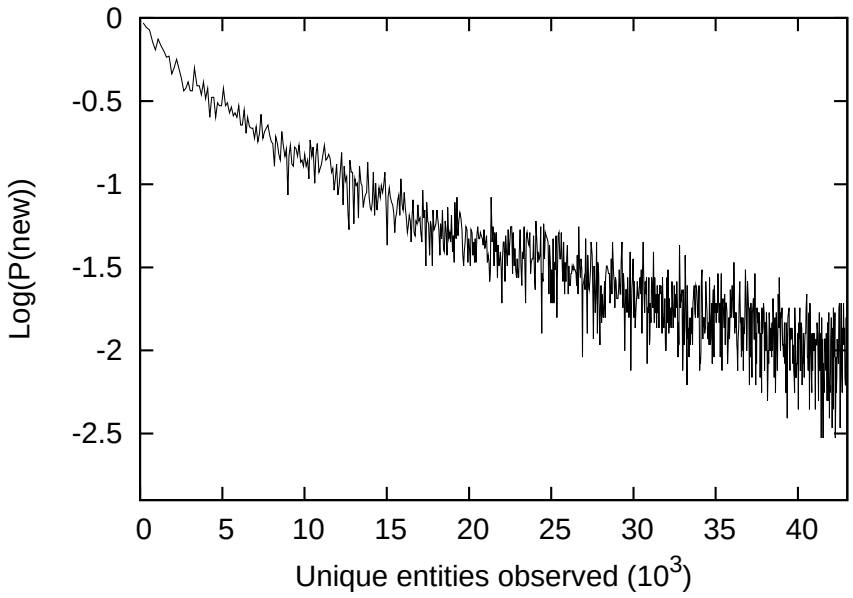

Figure A.10: *The probability of observing a new unique entity in Tshivenda, while increasing the training set size*

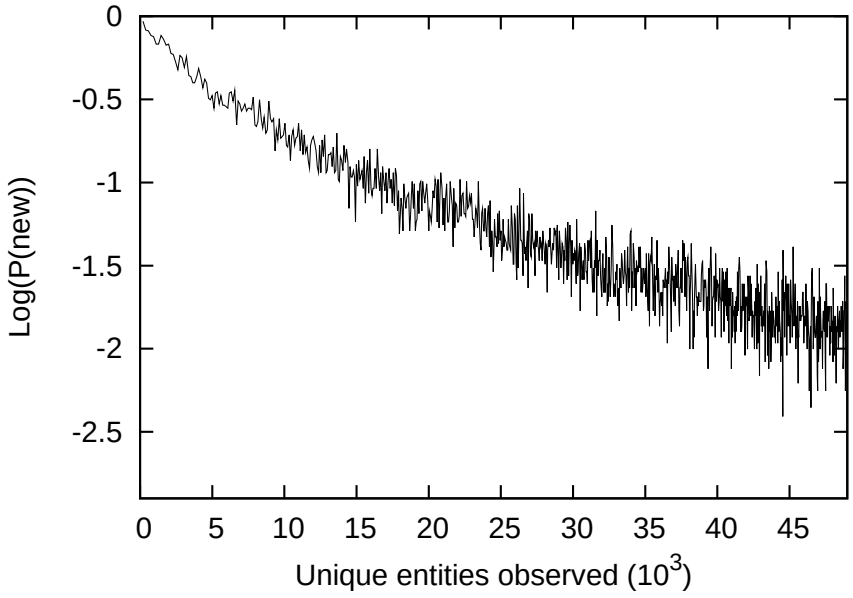

Figure A.11: *The probability of observing a new unique entity in Xitsonga, while increasing the training set size*

A.2 NUMBER OF UNIQUE 6-GRAM ENTITIES IN LANGUAGES WITH A 200K CHARACTER TRAINING SET

In this section we plot the number of unique 6-gram entities found with respect to the total number of multinomial trials observed in the training set, $F_{c_i}(x)$, for the 11 official languages of South Africa and also the analytical log functions that fits these curves best. These figures represent only 200K character training sets. Refer to Section 6.2.3.3 for more detail on the significance of these plots.

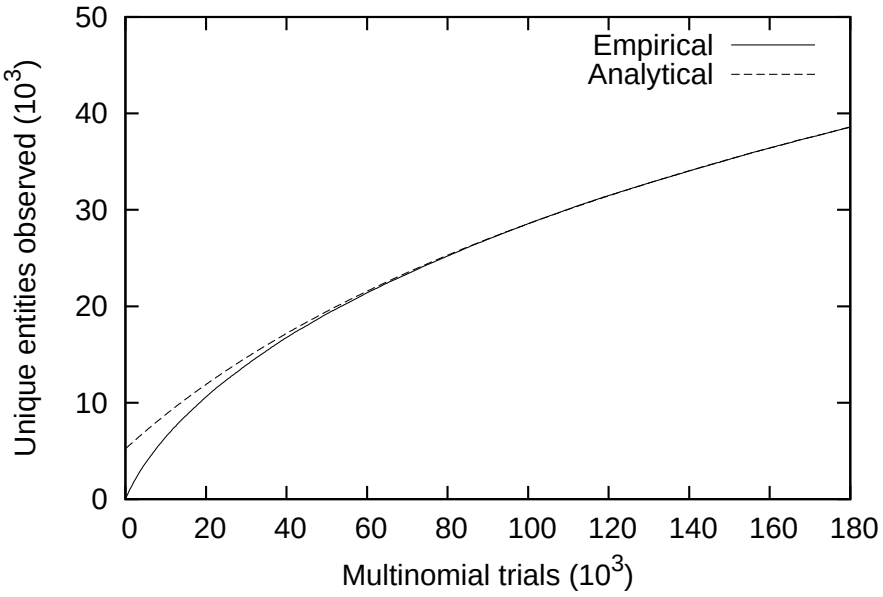

Figure A.12: *The cumulative number of unique entities in Afrikaans, while increasing the training set size*

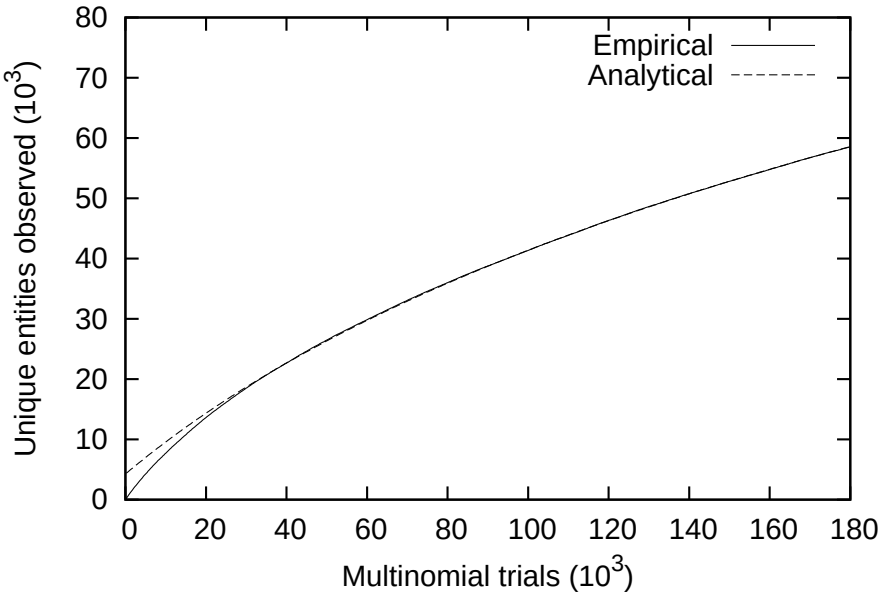

Figure A.13: *The cumulative number of unique entities in English, while increasing the training set size*

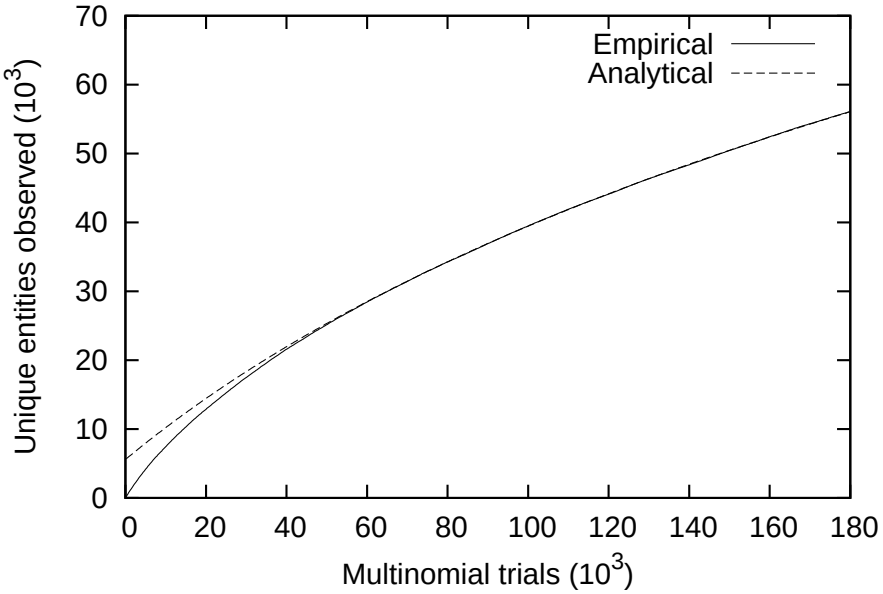

Figure A.14: *The cumulative number of unique entities in isiNdebele, while increasing the training set size*

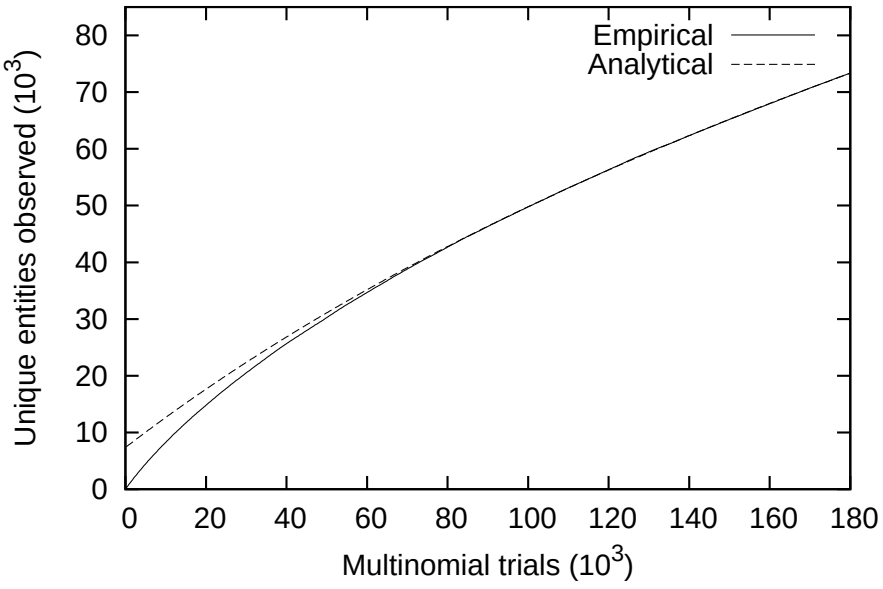

Figure A.15: *The cumulative number of unique entities in isiXhosa, while increasing the training set size*

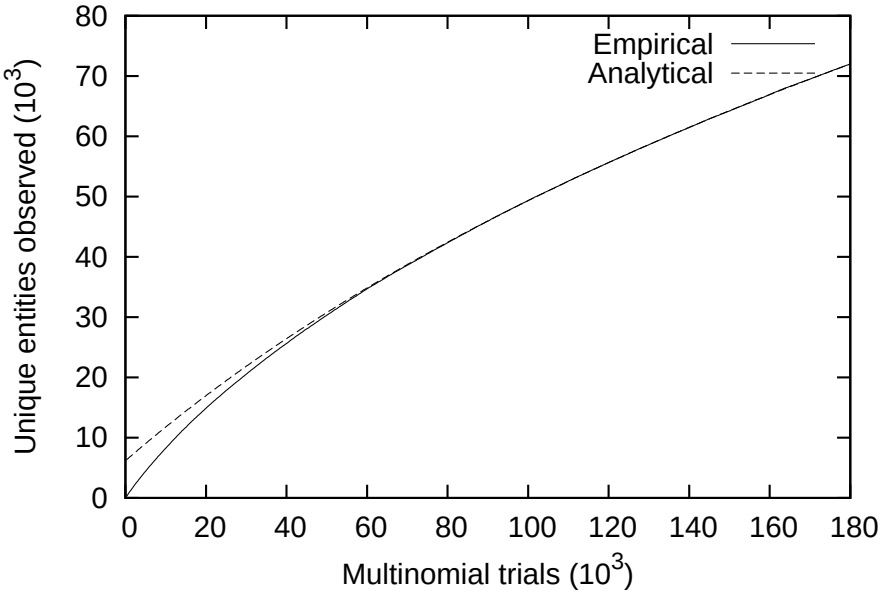

Figure A.16: *The cumulative number of unique entities in isiZulu, while increasing the training set size*

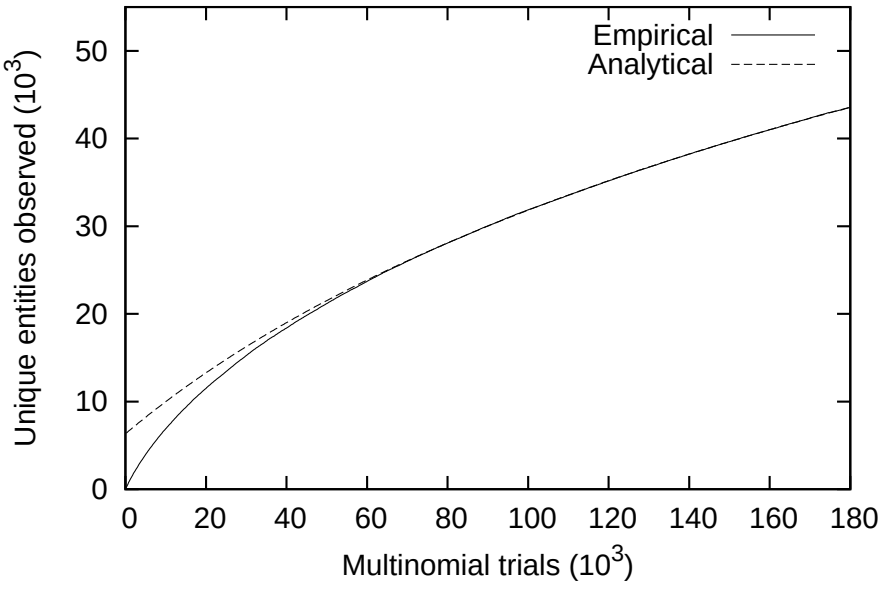

Figure A.17: *The cumulative number of unique entities in Sepedi, while increasing the training set size*

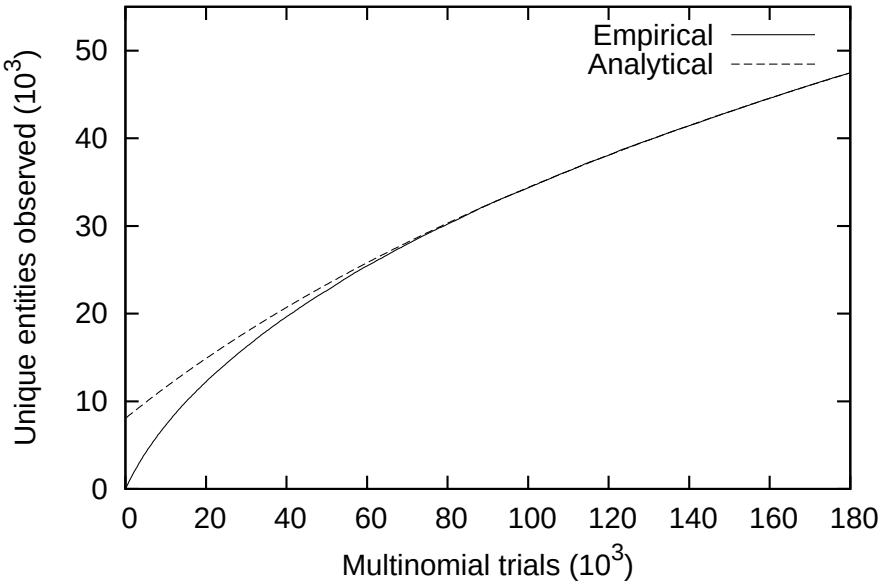

Figure A.18: *The cumulative number of unique entities in Sesotho, while increasing the training set size*

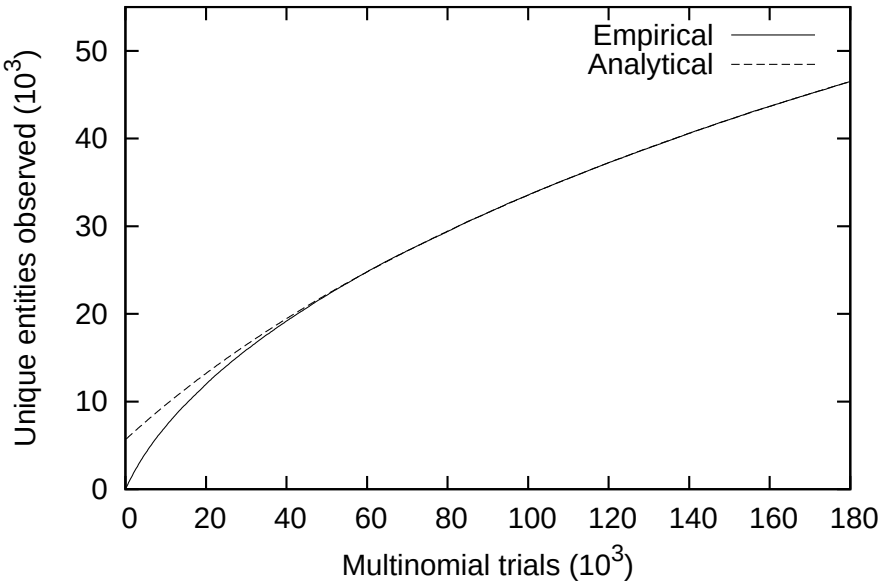

Figure A.19: *The cumulative number of unique entities in Setswana, while increasing the training set size*

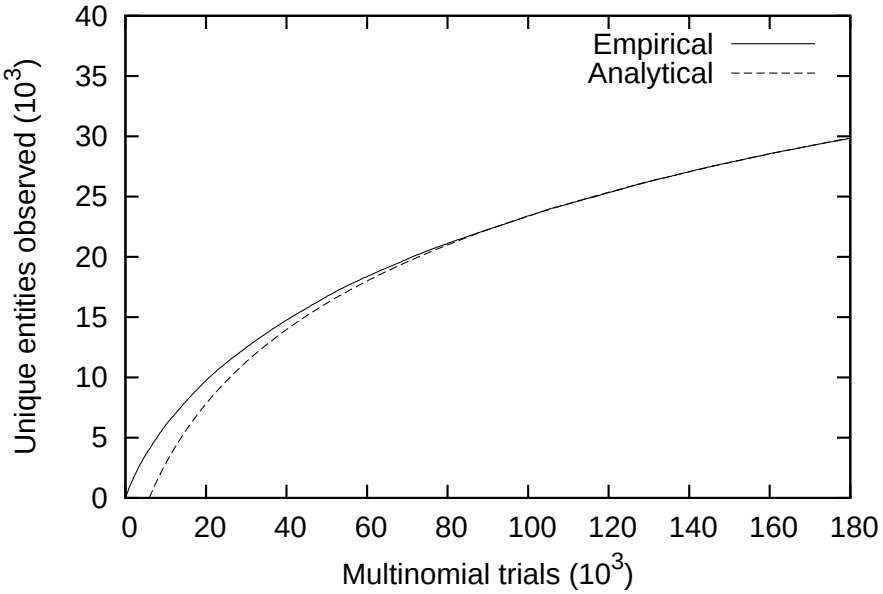

Figure A.20: *The cumulative number of unique entities in siSwati, while increasing the training set size*

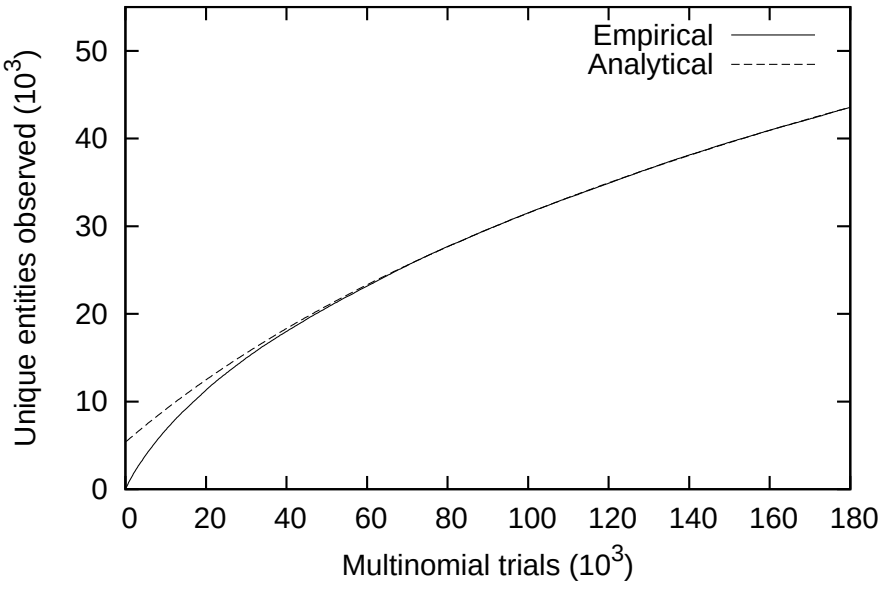

Figure A.21: *The cumulative number of unique entities in Tshivenda, while increasing the training set size*

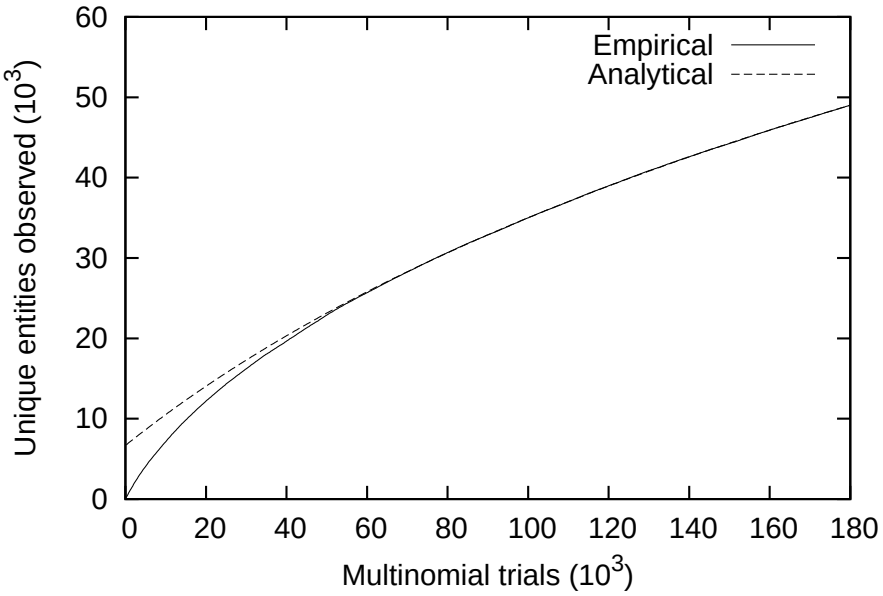

Figure A.22: *The cumulative number of unique entities in Xitsonga, while increasing the training set size*

A.3 NB CLASSIFIER PERFORMANCE WHILE VARYING THE PENALTY FACTORS FOR UNSEEN ENTITIES

In this section, we test the NB classifier's performance while varying the unseen entity penalty factors. We show figures for classifier performances for training set sizes of 200K, 400K, 800K, 1.6M and 2.0M characters (with window sizes of 15, 100 and 300 characters). These figures also show the performances for analytical penalty factors calculated in Table 6.1.

A.3.1 200K CHARACTER TRAINING SET

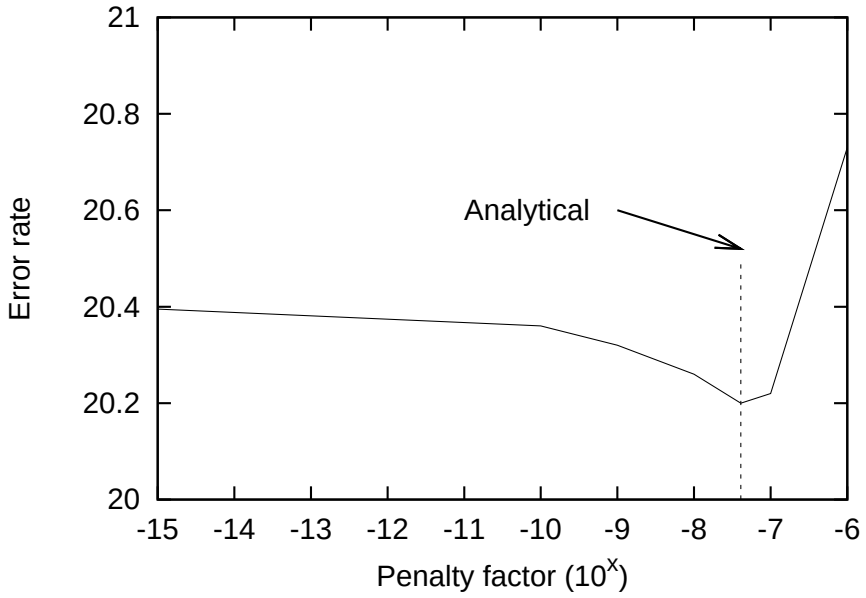

Figure A.23: *6-gram NB classifier error performance while varying the unseen entity penalty factor for a 15 character window size, 200K characters training set*

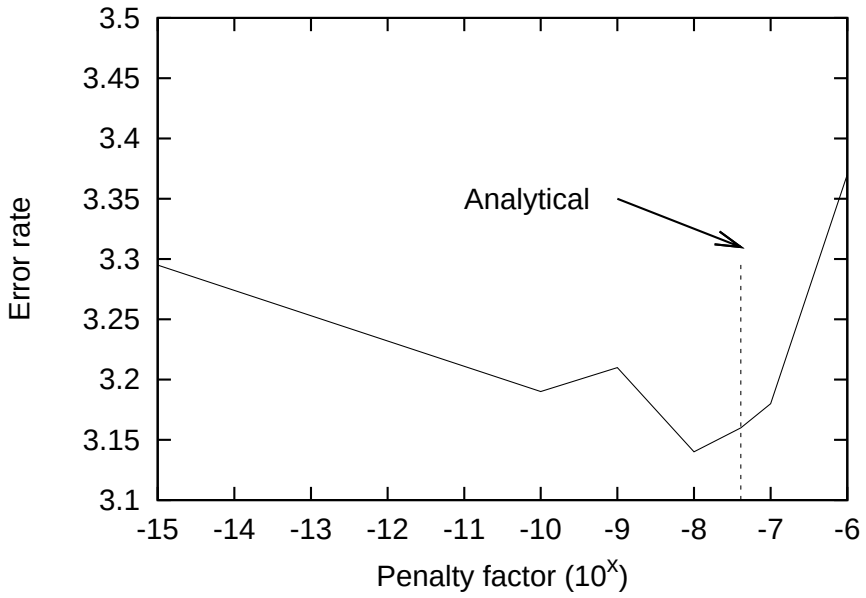

Figure A.24: *6-gram NB classifier error performance while varying the unseen entity penalty factor for a 100 character window size, 200K characters training set*

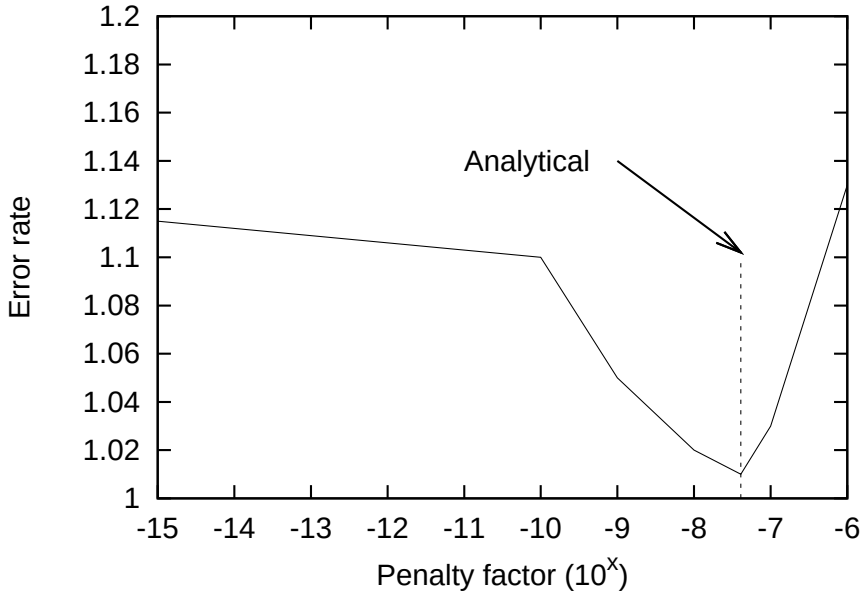

Figure A.25: *6-gram NB classifier error performance while varying the unseen entity penalty factor for a 300 character window size, 200K characters training set*

A.3.2 400K CHARACTER TRAINING SET

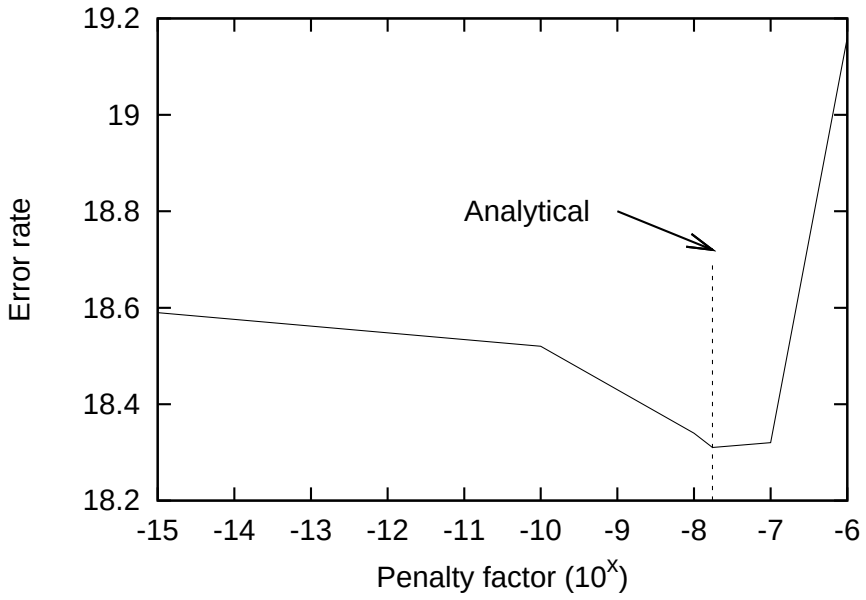

Figure A.26: *6-gram NB classifier error performance while varying the unseen entity penalty factor for a 15 character window size, 400K characters training set*

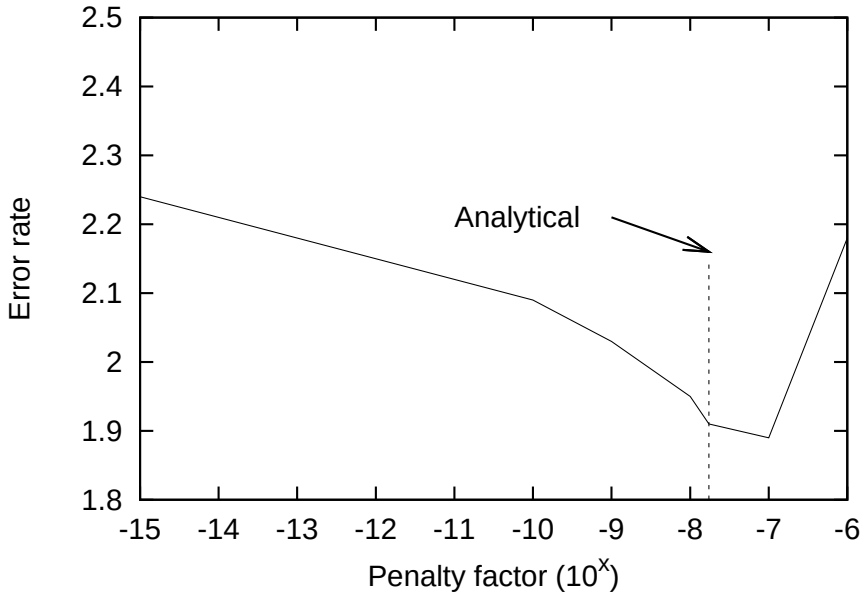

Figure A.27: *6-gram NB classifier error performance while varying the unseen entity penalty factor for a 100 character window size, 400K characters training set*

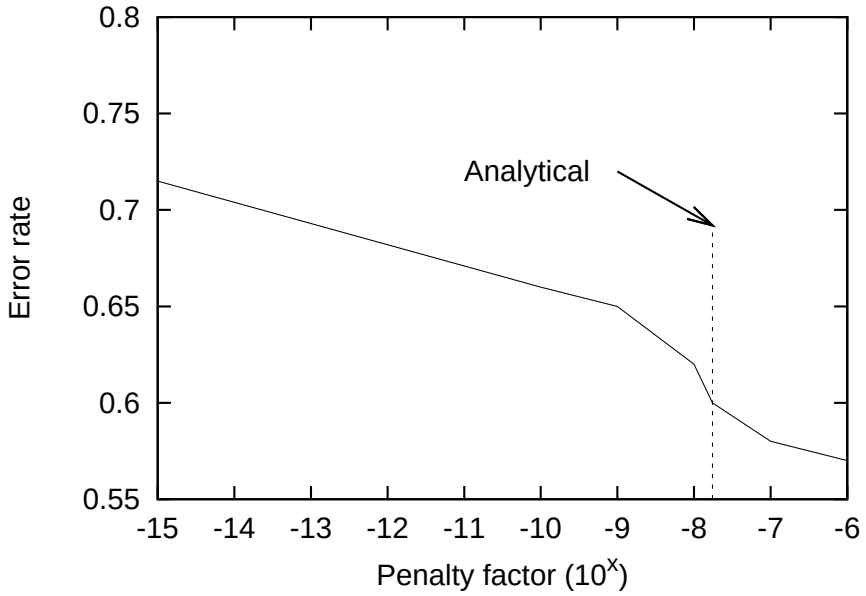

Figure A.28: *6-gram NB classifier error performance while varying the unseen entity penalty factor for a 300 character window size, 400K characters training set*

A.3.3 800K CHARACTER TRAINING SET

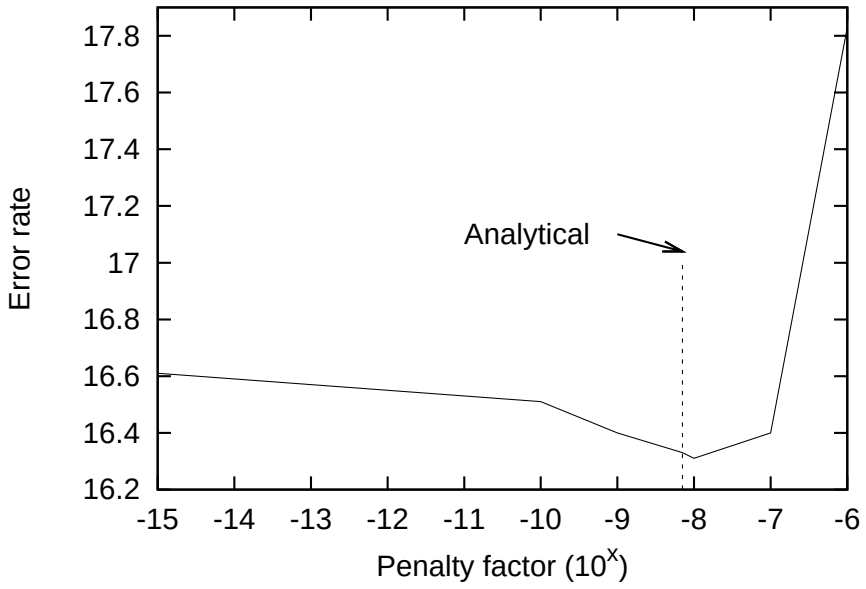

Figure A.29: *6-gram NB classifier error performance while varying the unseen entity penalty factor for a 15 character window size, 800K characters training set*

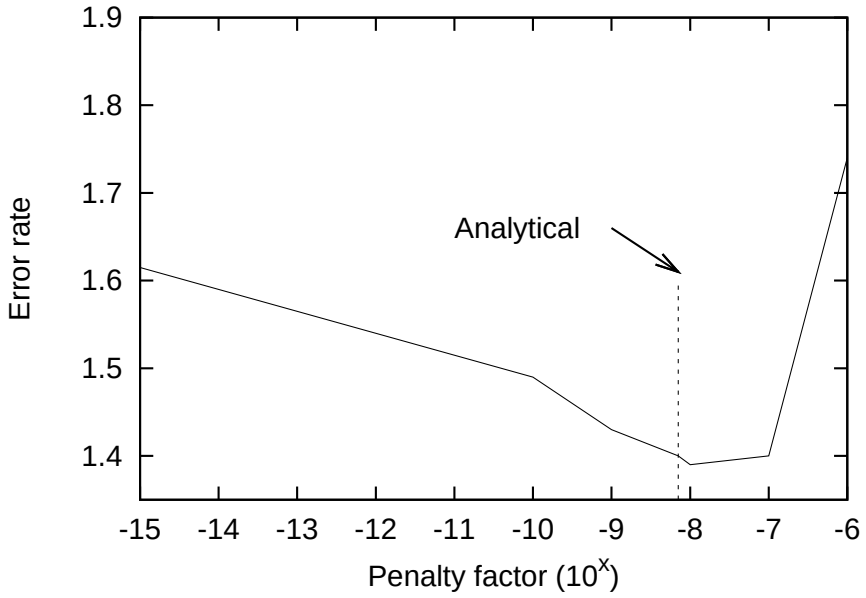

Figure A.30: *6-gram NB classifier error performance while varying the unseen entity penalty factor for a 100 character window size, 800K characters training set*

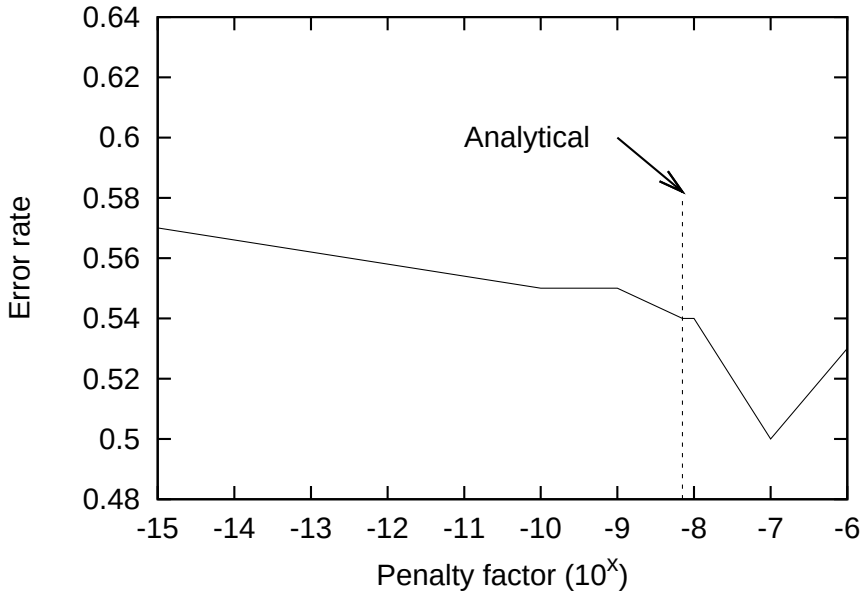

Figure A.31: *6-gram NB classifier error performance while varying the unseen entity penalty factor for a 300 character window size, 800K characters training set*

A.3.4 1.6M CHARACTER TRAINING SET

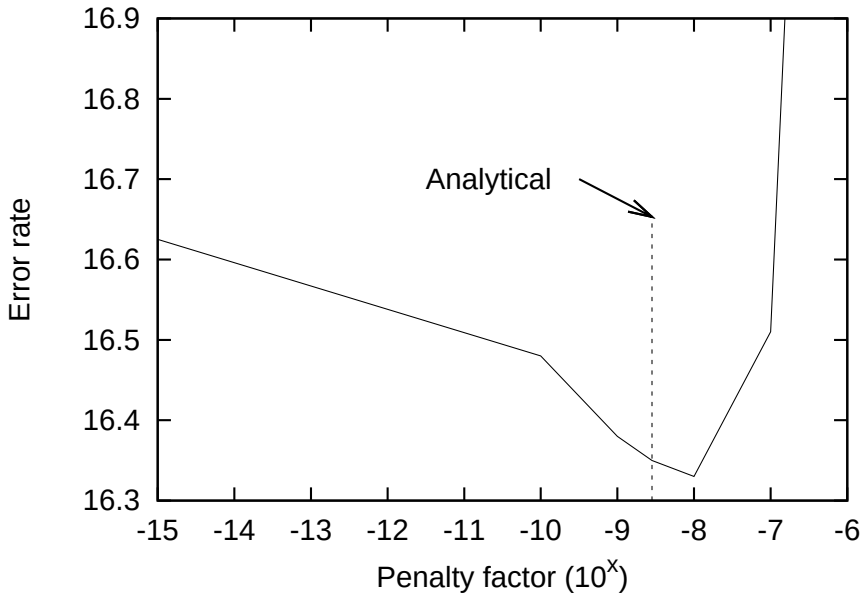

Figure A.32: *6-gram NB classifier error performance while varying the unseen entity penalty factor for a 15 character window size, 1.6M characters training set*

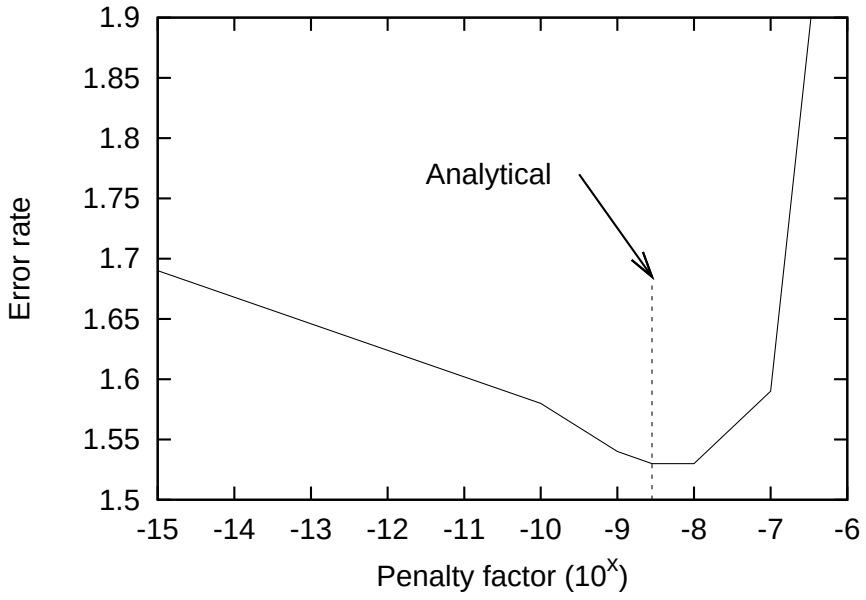

Figure A.33: *6-gram NB classifier error performance while varying the unseen entity penalty factor for a 100 character window size, 1.6M characters training set*

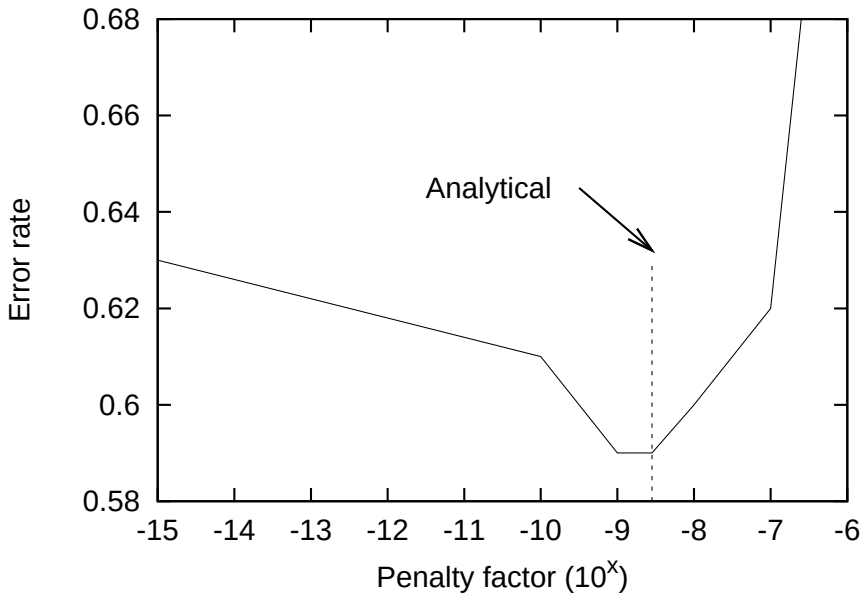

Figure A.34: *6-gram NB classifier error performance while varying the unseen entity penalty factor for a 300 character window size, 1.6M characters training set*

A.3.5 2.0M CHARACTER TRAINING SET

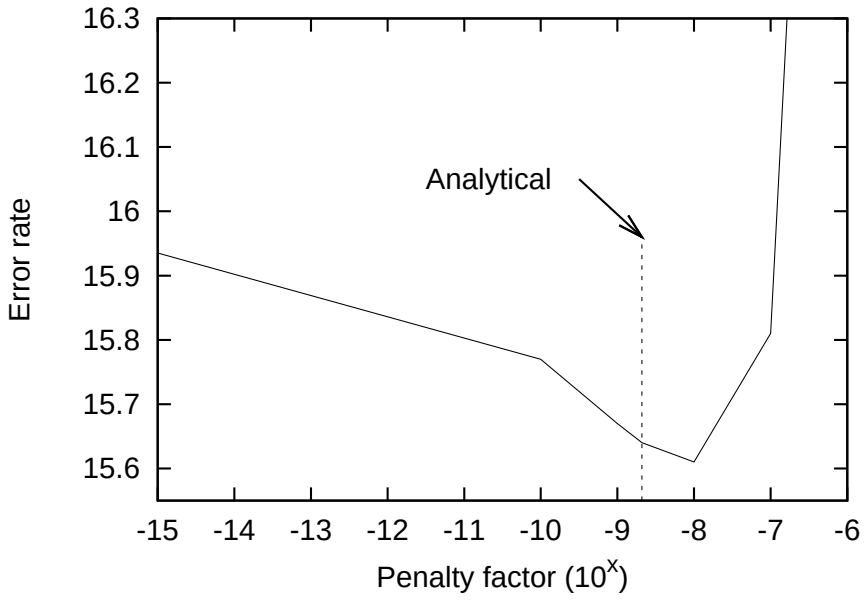

Figure A.35: *6-gram NB classifier error performance while varying the unseen entity penalty factor for a 15 character window size, 2.0M characters training set*

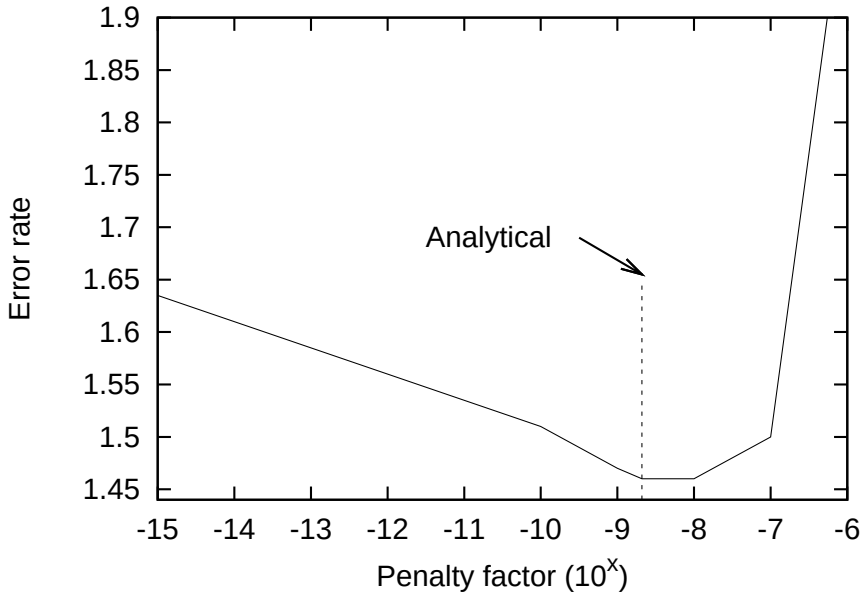

Figure A.36: *6-gram NB classifier error performance while varying the unseen entity penalty factor for a 100 character window size, 2.0M characters training set*

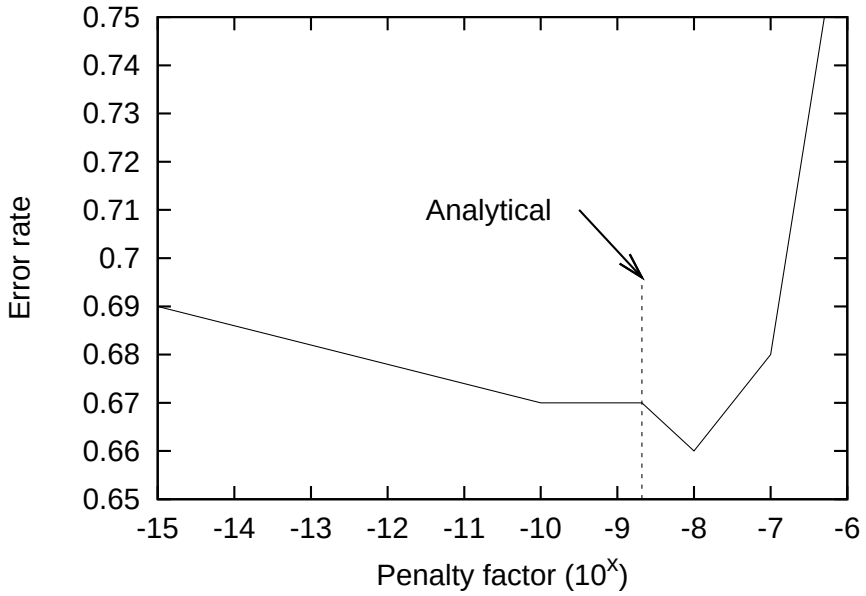

Figure A.37: *6-gram NB classifier error performance while varying the unseen entity penalty factor for a 300 character window size, 2.0M characters training set*

REFERENCES

Abramowitz, M. and Stegun, I.A. (1972). *Handbook of Mathematical Functions*. General Publishing Company, Ltd, Toronto, Ontario, Canada.

Avi-Itzhak, H. and Diep, T. (1996). Arbitrarily Tight Upper and Lower Bounds on the Bayesian Probability of Error. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 1, pp. 89–91.

Ayadi, M.E., Kamel, M. and Karray, F. (2008). Toward a Tight Upper Bound for the Error Probability of the Binary Gaussian Classification Problem. *Pattern Recognition*, vol. 41, pp. 2120–2132.

Bartlett, P.L., Jordan, M.I. and McAuliffe, J.D. (2006). Comments. *Statistical Science*, vol. 21, no. 3, pp. 341–346.

Bishop, C.M. (2006). *Pattern Recognition and Machine Learning*. Springer Science & Business Media, New York.

Botha, G. and Barnard, E. (2005). Two Approaches to Gathering Text Corpora from the World Wide Web. In: *Proceedings of the 16th Annual Symposium of the Pattern Recognition Association of South Africa*, p. 194. South Africa.

Botha, G. and Barnard, E. (2008). Factors that Effect the Accuracy of Text-based Language Identification. *Submitted to Computer Speech and Language*.

Botha, G., Zimu, V. and Barnard, E. (2006). Text-based Language Identification for the South African Languages. In: *Proceedings of the 17th Annual Symposium of the Pattern Recognition Association of South Africa*, pp. 46–52. South Africa.

Boullé, M. (2007). Compression-Based Averaging of Selective Naive Bayes Classifiers. *The Journal of Machine Learning Research*, vol. 8, pp. 1659–1685.

Burges, C.J.C. (1998). A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167.

Chapelle, O. (2007). Training a Support Vector Machine in the Primal. *Neural Computation*, vol. 19, no. 5, pp. 1155–1178.

Chernoff, H. (1952 December). A Measure for Asymptotic Efficiency of a Hypothesis Based on a sum of Observations. *The Annals of Mathematical Statistics*, vol. 23, no. 4, pp. 493–507.

Devijver, P. (1974). On a New Class of Bounds on Bayes Risk in Multihypothesis Pattern Recognition. *IEEE Transactions on Computers*, vol. 23, pp. 70–80.

Domingos, P. and Pazzani, M. (1997). On the Optimality of the Simple Bayesian Classifier under Zero-One Loss. *Machine Learning*, vol. 29, pp. 103–130.

Gradshteyn, I.S. and Ryzhik, I.M. (1980). *Tables of Integrals, Series, and Products*. Academic Press, Inc, Orlando, Florida, USA.

Hakkinen, J. and Tian, J. (2001 December). N-gram and Decision Tree Based Language Identification for Written Words. In: *IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 335–339.

Hand, D. and Yu, K. (2001 December). Idiot's Bayes: Not So Stupid after All? *International Statistical Review*, vol. 69, no. 3, pp. 385–399.

Hashlamoun, W., Varshney, P. and Samarasooriya, V. (1994). A Tight Upper Bound on the Bayesian Probability of Error. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 2, pp. 220–225.

Hellman, M. and Raviv, J. (1970). Probability of Error, Equivocation, and Chernoff Bound. *IEEE Transactions on Information Theory*, vol. 16, no. 4, pp. 368–372.

Ito, T. (1972). Approximate Error Bounds in Pattern Recognition. *Machine Intelligence*, vol. 7, pp. 369–372.

Jelinek, F. and Mercer, R. (1980 May). Interpolated Estimation of Markov Source Parameters from Sparse Data. In: *Proceedings of the Workshop on Pattern Recognition in Practice*. North-Holland, Amsterdam, The Netherlands.

Joachims, T. (2006). Training Linear SVMs in Linear Time. In: *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 217–226.

Moguerza, J.M. and Munoz, A. (2006). Support Vector Machines with Applications. *Statistical Science*, vol. 21, no. 3, pp. 322–336.

Morris, C. (1975 January). Central Limit Theorems for Multinomial Sums. *The Annals of Statistics*, vol. 3, no. 1, pp. 165–188.

Press, S. (1966). Linear Combinations of Non-Central Chi-Square Variates. *The Annals of Mathematical Statistics*, vol. 37, no. 2, pp. 480–487.

Raphaeli, D. (1996). Distribution of Noncentral Indefinite Quadratic Forms in Complex Normal Variables. *IEEE Transactions on Information Theory*, vol. 42, no. 3, pp. 1002–1007.

Rice, J.A. (1988). *Mathematical Statistics and Data Analysis*. Wadsworth, Inc., California, USA.

Rigouste, L., Cappé, O. and Yvon, F. (2005). Inference for Probabilistic Unsupervised Text Clustering. In: *Proceedings of the IEEE Workshop on Statistical Signal Processing (SSP'05)*. Bordeaux, France.

Ruben, H. (1962). Probability Content of Regions Under Spherical Normal Distributions, IV: The Distribution of Homogeneous and Non-Homogeneous Quadratic Functions of Normal Variables. *The Annals of Mathematical Statistics*, vol. 33, no. 2, pp. 542–570.

Russek, E., Kronmal, R. and Fisher, L. (1983). The Effect of Assuming Independence in Applying Bayes' Theorem to Risk Estimation and Classification in Diagnosis. *Computers and Biomedical Research*, vol. 16, pp. 537–552.

Russell, S.J. and Norvig, P. (1995). *Artificial Intelligence: a Modern Approach*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Shah, B.K. (1963). Distribution of Definite and of Indefinite Quadratic Forms from a Non-Central Normal Distribution. *The Annals of Mathematical Statistics*, vol. 34, no. 1, pp. 186–190.

Slatêr, L. (1960). *Confluent Hypergeometric Functions*. Cambridge University. Press, London.

Van Dyk, E. and Barnard, E. (2007). Naive Bayesian Classifiers for Multinomial Features: a Theoretical Analysis. In: *Proceedings of the 18th Annual Symposium of the Pattern Recognition Association of South Africa*, pp. 75–82. South Africa.

Webb, A. (2002). *Statistical Pattern Recognition*. 2nd edn. John Wiley & Sons, Ltd., England.

