Naive Bayesian classifiers for multinomial features: a theoretical analysis

Ewald van Dyk[1], *Etienne Barnard*[2]

[1,2]School of Electrical, Electronic and Computer Engineering, University of North-West, South Africa
[1,2]Human Language Technologies Research Group, Meraka Institute, Pretoria, South Africa
evdyk@csir.co.za, ebarnard@csir.co.za

Abstract

We investigate the use of naive Bayesian classifiers for multinomial feature spaces and derive error estimates for these classifiers. The error analysis is done by developing a mathematical model to estimate the probability density functions for all multinomial likelihood functions describing different classes. We also develop a simplified method to account for the correlation between multinomial variables. With accurate estimates for the distributions of all the likelihood functions, we are able to calculate classification error estimates for any such multinomial likelihood classifier. This error estimate can be used for feature selection, since it is easy to predict the effect that different features have on the error rate performance.

1. Introduction

Recent years have seen a resurgence in the use of naive Bayesian (NB) classifiers [1, 2]. 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 [2], 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.

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 the present contribution we therefore develop a theoretical model for the specific case of frequency counts, which can be used to address some of these issues.

To be precise, we assume that each feature has a fixed class-conditional probability of occurring, but only one feature can occur in any given observation; these observations are repeated m times to create a feature vector. This is known as a Bernoulli process [3]; for any given vector, m Bernoulli trials are drawn, and each feature contains the frequency of occurrence of the corresponding entity. If all samples are drawn independently, the class-conditional density functions take on a multinomial distribution [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 [4]. Other search methods such as the Best individual N method may not be useful since, as we will show later on, the different features are correlated.

The class separability measure that we will use assumes that features are generated from a multinomial distribution with known feature 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.

There are many different ways in which we could approximate the likelihood function. One approach is to note that all features are individually binomial. Then, if we apply the naive Bayesian philosophy and define the likelihood function as the product of all binomial features, we get the likelihood function of class c_r

$$p(\bar{x}|c_r) = \prod_{d=1}^{D} \frac{m!}{x_d!(m-x_d)!} p_{dc_r}^{x_d} q_{dc_r}^{m-x_d} \qquad (1)$$

where $\bar{x}$ is the input vector, x_d is the frequency count for feature d, m is the number of Bernoulli trials done, p_{dc_r} is the probability of feature d occurring in a Bernoulli trial for class c_r and $q_{dc_r} = 1 - p_{dc_r}$.

The advantage of using eq. (1) is that features are easily added to or deleted from a given set by including or excluding the relevant terms in the product. Further, for a given input vector, the factors $m!$, $x_d!$ and $(m - x_d)!$ remain constant over all classes. Therefore they do not contribute to the error rate of the classifiers and may be omitted from the analysis. Therefore we define the modified log likelihood as follows

$$L(\bar{x}|c_r) = \sum_{d=1}^{D} x_d \log(p_{dc_r}) + (m - x_d) \log(q_{dc_r}) \qquad (2)$$

It is clear from eq. (2) that the distribution of $L(\bar{x}|c_r)$ is a linear combination of the binomial elements x_d. If we assume that all x_ds are uncorrelated, we can use the central limit theorem [5] to deduce that $L(\bar{x}|c_r)$ is approximately Gaussian. In this way, we can calculate a modified log likelihood distribution for all classes and calculate the overlap to estimate the error performance of the classifier. Unfortunately, we will see that it is a

poor assumption that all x_ds are uncorrelated, because $\sum x_d$ is constrained to equal m.

The technique that we will use for error analysis in this paper therefore uses the full multinomial distribution, rather that a product of binomial distributions, and thus accounts for the correlation between the x_ds. It should also be noticed that even though correlation is accounted for, the different Bernoulli trials are still assumed to be independent.

2. Multinomial likelihood distribution estimation

Let us assume that we have a collection of D features and that a test sample consist of m independent Bernoulli trials, where each feature 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 feature (over m trials) represent a D-dimensional feature space. Assuming that all test samples are independent and that all Bernoulli trials are independent, we can calculate the likelihood of any given test vector, given class c_r:

$$p(\bar{x}|c_r) = \frac{m!}{x_1!x_2!...x_D!} p_{1c_r}^{x_1} p_{2c_r}^{x_2} ... p_{Dc_r}^{x_D} \tag{3}$$

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

$$L(\bar{x}|c_r) = \sum_{d=1}^{D} \alpha_{dr} x_d, \tag{4}$$

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 experimentally (see section 4.1) that $L(\bar{x}|c_r)$ is approximately Gaussian for high D. For a theoretical treatment on the central limit theorem for linear combinations of correlated multinomial variables, refer to [5]. 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 and use it 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

$$\mu = E[L(\bar{x}|c_r)] \tag{5}$$

$$\sigma^2 = E[L^2(\bar{x}|c_r)] - E^2[L(\bar{x}|c_r)] \tag{6}$$

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

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

$$\mu = E[\sum_{d=1}^{D} \alpha_{dr} x_d] = m \sum_{d=1}^{D} \alpha_{dr} p_{dc_t}. \tag{7}$$

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

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, and also the exact expression.

2.1. 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

$$\begin{aligned} \sigma_u^2 &= Var[\sum_{d=1}^{D} \alpha_{dr} x_d] \\ &= \sum_{d=1}^{D} \alpha_{dr}^2 Var(x_d) \\ &= m \sum_{d=1}^{D} p_{dc_t} q_{dc_t} \alpha_{dr}^2 \end{aligned} \tag{8}$$

where σ_u^2 represents the uncorrelated variance. This equation shows that σ_u^2 can be calculated in $O(D)$ computations.

2.2. Variance with correlation

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

$$\sigma^2 = E[(\sum_{d=1}^{D} \alpha_{dr} x_d)^2] - E^2[\sum_{d=1}^{D} \alpha_{dr} x_d] \tag{9}$$

From this equation we can rewrite the variance in terms of the multinomial covariance matrix $\mathbf{\Sigma}$:

$$\sigma^2 = \bar{\alpha}_{\mathbf{r}}^{\mathbf{T}} \mathbf{\Sigma} \bar{\alpha}_{\mathbf{r}} \tag{10}$$

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. (10) in terms of the uncorrelated variance as follow:

$$\sigma^2 = \sigma_u^2 - m \sum_{d} \sum_{e;e \neq d} \alpha_{dr} \alpha_{er} p_{dc_t} p_{ec_t} \tag{11}$$

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

2.3. Compensating for correlation

It is not surprising (see results) that the uncorrelated assumption gives inaccurate results, since the features are constrained to sum to a constant value. We now propose a method of compensation to correct 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 and add or subtract the necessary Δm to compensate.

We can express eq. (4) as

$$L(\bar{x}|c_r) = \sum_{d=1}^{D} L_d(x_d|c_r) \tag{12}$$

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

$$L_d(x_d + \Delta m_d|c_r) = L_d(x_d|c_r) + \Delta m_d \alpha_{dr} \quad (13)$$

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

$$\sigma^2 = E[(\sum_{d=1}^{D} L_d(x_d|c_r) - \mu)^2] \quad (14)$$

From eq. (14) we can also write an approximate expression for the true variance $\sigma^2_{\Delta m}$ when $\sum x_d = m + \Delta m$:

$$\sigma^2_{\Delta m} = E[(\sum_{d=1}^{D} L_d(x_d|c_r) + \Delta m \sum_{d=1}^{D} \alpha_{dr} p_{dc_t} - \mu)^2], \quad (15)$$

where we used eq. (13) and made the approximation $\Delta m_d = \Delta m p_{dc_t}$. By expanding the square in eq. (15) we obtain

$$\sigma^2_{\Delta m} = \sigma^2 + \Delta^2 m [\sum_{d=1}^{D} \alpha_{dr} p_{dc_t}]^2 \quad (16)$$

Next, the uncorrelated variance σ^2_u generates $\sum x_d = m + \Delta m$ with a distribution mean and variance for $\Delta m = \sum x_d - m$ given by

$$Mean(\Delta m) = m \sum_{d=1}^{D} p_{dc_t} - m = 0$$
$$Var(\Delta m) = \sum_{d=1}^{D} Var(x_d) = m \sum_{d=1}^{D} p_{dc_t} q_{dc_t} \quad (17)$$

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:

$$\sigma^2_u = \sum_{\Delta m} \sigma^2_{\Delta m} p(\Delta m) \quad (18)$$

Combining eq. (16) and (18) we get

$$\sigma^2_u = \sigma^2 + (\sum_{d=1}^{D} \alpha_{dr} p_{dc_t})^2 \sum_{\Delta m} \Delta^2 m p(\Delta m) \quad (19)$$

and we notice that:

$$\sum_{\Delta m} \Delta^2 m p(\Delta m) = Var(\Delta m) = m \sum_{d=1}^{D} p_{dc_t} q_{dc_t} \quad (20)$$

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

$$\sigma^2 = \sigma^2_u - m(\sum_{d=1}^{D} \alpha_{dr} p_{dc_t})^2 (\sum_{d=1}^{D} p_{dc_t} q_{dc_t}) \quad (21)$$

Notice the similarities between eq. (11) and (21) and also that eq. (21) can be calculated in $O(D)$ computations.

As we will see below, experimental evidence shows that eq. (21) is accurate for values of $m p_{dc_t}$ not too high. This is a reasonable condition for high dimensional applications such as text-based language identification.

2.4. 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 (eq. (3) to (21)) 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. Therefore we define the following parameters:

$$\begin{aligned} p_{Rc_r} &= 1 - \sum_{d \subset C} p_{dc_r} \\ p_{Rc_t} &= 1 - \sum_{d \subset C} p_{dc_t} \\ x_R &= m - \sum_{d \subset C} x_d \\ \alpha_{Rr} &= log(p_{Rc_r}) \end{aligned} \quad (22)$$

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. (21) might be inaccurate. Therefore we need to take correlation into consideration between features $d \subset C$ and R.

Eq. (23), (24) and (25) are the new formulas that are equivalent to eq. (4), (7) and (8) respectively:

$$L(\bar{x}|c_r) = \sum_{d \subset C} (\alpha_{dr} x_d) + \alpha_{Rr} x_R \quad (23)$$

$$\mu = m \sum_{d \subset C} p_{dc_t} \alpha_{dr} + m p_{Rc_t} \alpha_{Rr} \quad (24)$$

$$\sigma^2_u = m \sum_{d \subset C} [p_{dc_t} q_{dc_t} \alpha^2_{dr} - 2 p_{dc_t} p_{Rc_t} \alpha_{dr} \alpha_{Rr}] + m p_{rc_t} q_{rc_t} \alpha^2_{Rr} \quad (25)$$

Notice that eq. (23) to (25) can all be calculated in $O(D_C)$ calculations, where D_C is the number of features considered (length of subset C). It is also important to notice that σ^2_u ignores all correlation between features in subset C, but takes all the correlation into consideration with feature R. We can therefore use a modified version of eq. (21) to include all correlation. The new version of eq. (21) can be expressed as

$$\sigma^2 = \sigma^2_u - m(\sum_{d \subset C} \alpha_{dr} \frac{p_{dc_t}}{q_{Rc_t}})^2 (\sum_{d \subset C} p_{dc_t} q_{dc_t} - m p_{Rc_t} q_{Rc_t}) \quad (26)$$

Eq. (26) can also be calculated in $O(D_C)$ computations.

3. 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 [4]:

$$c = \max_{i=1,\ldots,C} L_i + log(p_i), \quad (27)$$

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 real class is j, is:

$$p_{i|j} = p(c = i|c_j) \tag{28}$$

We can combine eq. (27) and (28) to get:

$$\begin{aligned} p_{i|j} = \quad & p[(L_i + log(p_i) \geq L_1 + log(p_1)) \cap \\ & (L_i + log(p_i) \geq L_2 + log(p_2)) \cap \\ & \vdots \\ & (L_i + log(p_i) \geq L_C + log(p_C))|c_j] \end{aligned} \tag{29}$$

In addition, we can assume independence between all comparisons to get a pessimistic estimate:

$$p_{i|j} = \prod_{k=1}^{C} p_{ik|j} \tag{30}$$

where

$$p_{ik|j} = p[(L_{ik} \geq T_{ik})|c_j] \tag{31}$$

where $L_{ik} = L_i - L_k$ and $T_{ik} = log(p_k) - log(p_i)$. Notice that $p_{ik|j} = 1$ for $i = k$. Finally, for a binary (two class) classifier, the expression in eq. (30) is exact.

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

$$L_{ik} = L_i - L_k = \sum_{d=1}^{D} (\alpha_{di} - \alpha_{dk}) x_d \tag{32}$$

Notice that eq. (32) is similar to eq. (4). Therefore we can use eq. (7), (21) and (26) to estimate the distribution of L_{ik} by simply substituting α_{dr} with $(\alpha_{di} - \alpha_{dk})$.

Figure 1 shows how we can calculate $p_{ik|j}$ from the distribution of L_{ik}:

$$p_{ik|j} = \frac{1}{2} - \frac{1}{2} erf[\frac{T_{ik} - \mu_{ik|j}}{\sigma_{ik|j}\sqrt{2}}] \tag{33}$$

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:

$$\epsilon = 1 - \frac{\sum_{i=1}^{C} p_{i|i} p_i}{\sum_{i=1}^{C} \sum_{j=1}^{C} p_{i|j} p_j} \tag{34}$$

4. Results

In this section we will investigate the validity of the various approaches derived above with a simulated problem consisting of two multinomial classes with Bernoulli probabilities, as shown in figure 2.

These two classes are generated over a feature space of 500 dimensions and all tests are done both empirically and with the theoretical model given above. The empirical distributions are calculated by generating 10000 multinomial samples of each class and drawing a histogram for all the likelihood functions. The error analysis is done by simply testing the error rate of the 10000 samples of each class. All tests are done with a Bernoulli count $m = 10$.

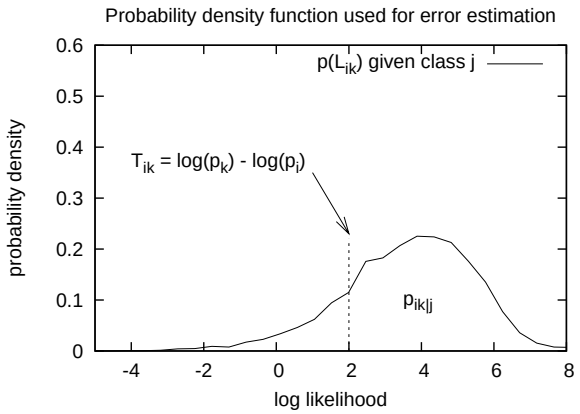

Figure 1: *Estimating $p_{ik|j}$ from the probability density function of L_{ik}.*

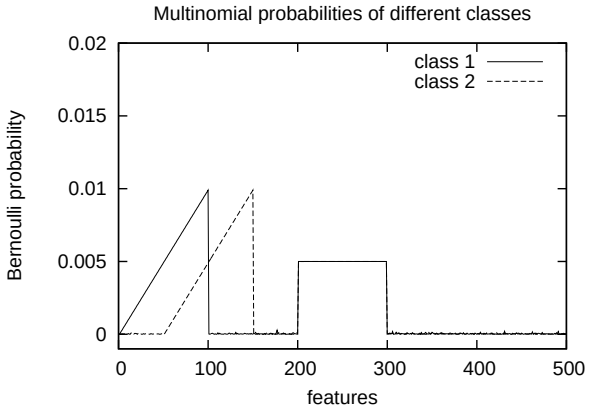

Figure 2: *Two classes generated with different Bernoulli feature probabilities.*

4.1. Probability distributions of Likelihood functions

In this section the theoretical distributions of two likelihood functions will be 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 3 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 shown in figure 2. 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.

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 3) 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. (32).

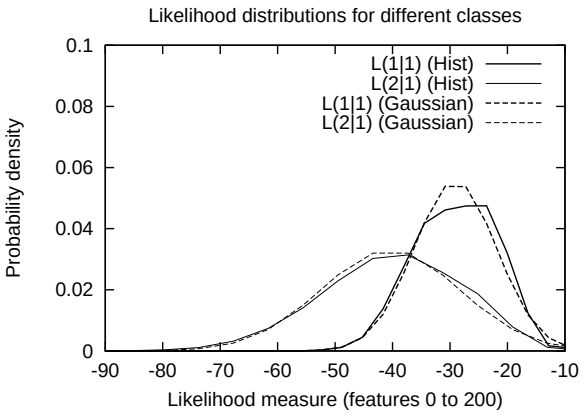

Figure 3: *Likelihood distributions of classes 1 and 2 for features 0 to 200*

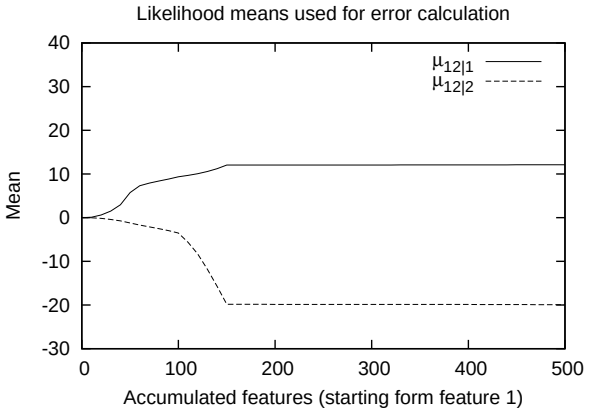

Figure 4: *Mean curves for the modified difference likelihood function L_{12} for input vectors from classes c_1 and c_2 while incrementally adding features*

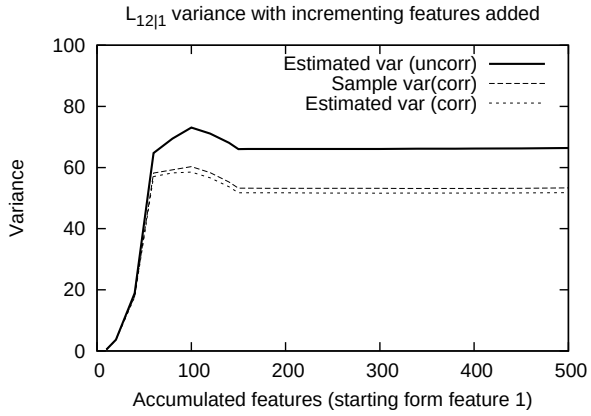

Figure 5: *Variance curves for L_{12}, given c_1, when incrementally adding features. Sampled values are compared to those computed from two different approximations.*

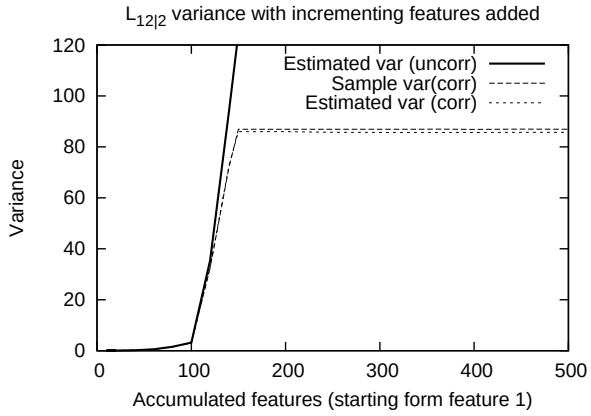

Figure 6: *Variance curves for L_{12}, given c_2, when incrementally adding features*

4.2. Error analysis

In this section, the empirical error rate (10000 test samples per class are used) of the naive Bayesian classifier will be compared with the theoretical error rate predicted earlier. For all the tests, we will use the two classes described in figure 2 with $m = 10$. We will assume equal priors for both classes.

4.2.1. Effects of feature addition on likelihood means

Figure 4 shows the predicted likelihood mean values of L_{12} for input vectors from classes c_1 and c_2 when features are incrementally added into the analysis (see eq. (32) and (33)). Notice that L_{ik} is symmetric to L_{ki} in eq. (32) 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 afterwards, even though classes one and two have a dense probability space between features 200 and 300 (see figure 2).

4.2.2. Effects of feature addition on likelihood variance

Figures 5 and 6 show the predicted likelihood variances for L_{12} given input vectors from classes c_1 and c_2 respectively. Notice again, that 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 5 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. (10)).

4.2.3. 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. (34)). Figure 7 shows the empirical error rate (on a test set of 10000 samples per class) and the estimated error rate, while incrementally adding features.

In figure 7, 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

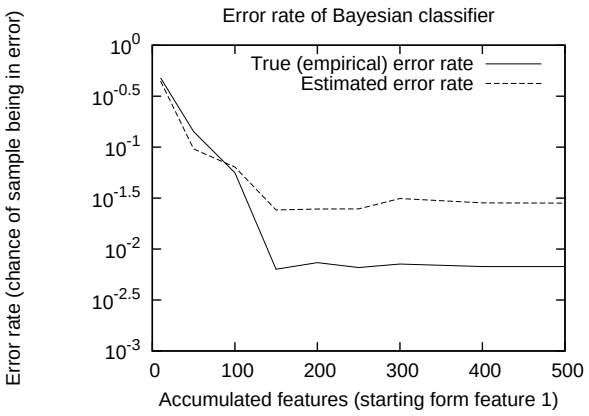

Figure 7: *Classification error rate ϵ of Bayesian classifier while incrementally adding features*

feature selection.

5. Conclusions

We have found that by using the Gaussian probability distribution approximations we can get a useful error estimate for feature selection in naive Bayesian classifiers. 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 mean and variance of all the likelihood distributions are calculated accurately, independently of the final distribution. The only assumption made on the distributions are that they are composed as a sum of multinomial variables.

Usually, the likelihood distributions become Gaussian (and therefore provide accurate error analysis) when many Bernoulli 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 isn't 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 this paper we used $m = 10$ and the effect of this can be seen by observing skewed Gaussian curves that would typically result in poor error estimates for low error rates (since the tail of the distribution is somewhat inaccurate). Future research will investigate whether more accurate estimates can be derived by correcting for this deviation from normality.

In the analysis given above, all the feature 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. To model this fact, one could follow a Bayesian approach and take expectations over the distributions of the Bernoulli probabilities. However, for low dimensional spaces one would expect the probability estimates on training samples to be accurate and therefore the Bayesian approach is expected to provide little benefit. On the other hand, when the problem expands into extremely high dimensional spaces the feature probability estimates become inaccurate. Even though these estimates are inaccurate, the distributions of different classes might become more mutually exclusive. The naive Bayesian classifier that is trained on the inaccurate probabilities might still perform very well (and often much better than a lower dimensional classifier on the same problem). 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 feature probability estimates become inaccurate. Hence, the full Bayesian analysis may be important in practice. A particular issue that will be addressed by such an analysis is the proper treatment of observations that occur in the test set, but not the training set. Eq. (4) would assign negative infinity to the log likelihood function, which is not a good choice in many practical situations. The theoretical basis developed in this paper will help us choose more suitable values for this penalty.

We also intend to use this analysis to gain a more intuitive understanding on the contribution of individual features, and to apply that understanding to improve the performance of our language-identification system.

6. References

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

[2] G. Botha, V. Zimu, and E. Barnard, "Text-based language identification for the South African languages," in *Proceedings of the 17th Annual Symposium of the Pattern Recognition Association of South Africa*, South Africa, 2006, pp. 46–52.

[3] J. A. Rice, *Mathematical Statistics and Data Analysis*, Wadsworth, Inc., California, USA, 1988.

[4] A. Webb, *Statistical Pattern Recognition*, John Wiley & Sons, Ltd., England, second edition, 2002.

[5] C. Morris, "Central limit theorems for multinomial sums," *The Annals of Statistics*, vol. 3, no. 1, pp. 165–188, January 1975.