Akademie věd České republiky
Ústav teorie informace a automatizace

Academy of Sciences of the Czech Republic
Institute of Information Theory and Automation

RESEARCH REPORT

Jana Novovičová and Antonín Malík
Academy of Sciences of the Czech Republic
Institute of Information Theory and Automation

TEXT DOCUMENT CLASSIFICATION USING FINITE MIXTURES

No. 2063 December 2002

ÚTIA AV ČR, P.O. Box 18, 182 08 Prague, Czech Republic
E-mail: novovic@utia.cas.cz
Fax: (+420)(2)688 3031

Contents

1 Introduction **3**
1.1 Model for learning document classification systems 3
1.2 Feature generation and dimensionality reduction 4
1.3 Classifier design 5

2 Probabilistic Framework for Text Classification **5**

3 Standard Models with Naive Bayes Classifier **6**
3.1 Multivariate Bernoulli model 8
3.2 Multinomial model 9
3.3 Classification 10
3.4 Feature selection 10
3.5 Summary 12

4 Finite Mixture Models **12**
4.1 Bernoulli mixture model 14
4.2 Mixture of multinomials 14
4.3 Model fitting with the EM algorithm 15

5 Experimental Results **16**
5.1 Reuters data set 16
5.2 Comparison of standard models 17
5.3 Standard models versus mixture models 22
5.4 Comparison feature selection methods 22

6 Conclusions **23**

A Derivation of the EM algorithm **24**
A.1 Multivariate Bernoulli mixture model 26
A.2 Mixture of multinomials 27

B Bhattacharyya Distance Criterion **29**
B.1 Multivariate Bernoulli distribution 29
B.2 Multinomial distribution 30

References **33**

Abstract

Mixture modelling of class-conditional densities is a standard method in statistical pattern recognition. This study focuses on the application of two class-conditional finite mixture models for text document classification, when the vector document representations using the *bag-of-words* approach are employed. Particularly, the *mixtures of multivariate Bernoulli* distributions as well as *mixtures of multinomial distributions* are used for the task of text documents classification. Proposed approach is a generalization of *naive Bayes* that tries to properly model significant class-conditional dependencies by spreading them over different mixture components. Preliminary experimental results on Reuters-21578 data set indicate the effectiveness of proposed mixture models in a text classification task.

Furthermore, the *multiclass Bhattacharyya distance* feature selection criterion for reducing dimensionality of feature vector document representation that takes into consideration the relationship between features, has been proposed. When compared the multiclass Bhattacharyya distance based feature selection method with *average mutual information* based method, an increase in classification accuracy was achieved.

1 Introduction

The goal of document analysis is to transform printed documents into an equivalent electronic representation. One of typical problems of document analysis is the selective extraction of information such as the type of text document - *text document classification*. The objective of text document classification is to assign a free document into one or more predefined set of *classes* or *categories* based on its contents. It is a straightforward concept from supervised pattern recognition or machine learning. It implies

(1) the existence of a labeled training data set,

(2) a way to represent the documents, and

(3) a statistical classifier trained using the chosen representation of the training set.

The classification of documents into different types allows the automatic routing of documents. While more and more textual information is available online, effective retrieval is difficult without good indexing and summarization of document contents. The automated document classification problem is of great practical importance given the massive volume online text available through the World Wide Web, Internet news feeds, electronic mail, medical patient records, digital libraries and so on.

1.1 Model for learning document classification systems

To carry out text classification, each document should be represented by a appropriate set of functions called *features* corresponding to the most indicative words that appear in the document.

In our model for learning document classification systems we distinguish between the two phases [11]. The first phase has the following tasks:

1. Construction of an appropriate set of features: A set of functions $\{f_1, ..., f_n\}$ has to be defined by using the training set of pre-classified documents.
   Many of approaches are based on the set of features, which could map each word occurring in the training set to its frequency in the given document.

2. Transformation of the documents into their *feature vector representation*: applying $\{f_1, ..., f_n\}$ the corresponding n-dimensional feature vector representation of each document is derived.

3. Construction of the classifier based on the feature vector representation of all documents.

   During its second phase, the document classification system has two primary tasks:

4. Transformation of a new document into its feature representation applying $\{f_1, ..., f_n\}$.

5. Classification of the document using the above feature vector representation.

Feature generation and *dimensionality reduction* as well as the classification construction comprise the kernel of a learning document classification system.

1.2 Feature generation and dimensionality reduction

Usually, features for document classification are associated with words that occurred in the training documents. This can easily result in several thousands or tens of thousands of features. In addition to single words, sometimes word combinations (word co-occurrence) and phrases are also considered, enlarging the number of relevant features enormously. For conventional classification methods this dimensionality might be to high. Theoretical considerations and experimental observations show that an intelligent reduction of a dimensionality of the feature vector can improve the effectiveness of classifier construction.

For dimensionality reduction *domain dependent* and *domain independent* methods can be used. The domain dependent methods relay on text. Typical domain dependent methods include:

- *Stop words elimination*: Non-content bearing words like prepositions, articles, are removed.

- *Stripping*: Simple reductions such as stripping of special characters at the beginning and the end of each word (periods, commas, parentheses, etc.).

- *Stemming*: Each word is replaced by its morphological root.

The high dimensionality of the feature space can be reduced dramatically by the reduction procedures mentioned above. However, the remaining number is still too high for conventional classification methods. For a further reduction, statistically motivated domain independent methods can be used.

Two possibilities for dimension reduction exist, either selecting a best set of original features - *feature selection, (feature subset selection)*, or transforming the features into new ones, that is, computing new features as some functions of the old ones - *feature extraction*. In document classification the dominant approach is feature selection using various criteria [27, 15].

1.3 Classifier design

A growing number of statistical classification methods and machine learning algorithms have been explored to build automatically a classifier by learning from previously labeled documents including decision trees [12], Bayesian probabilistic methods [23, 12], k-nearest neighbor classifiers [5], support vector machines [8] and neural networks [25].

2 Probabilistic Framework for Text Classification

This section presents probabilistic framework for characterizing the nature of documents and classifiers. We approach classification of the text document into one of a set of classes $C = \{c_1, ..., c_{|C|}\}$ in a Bayesian learning framework with *bag-of-words* representation of the documents. The framework defines a probabilistic generative model for the data and embodies two assumptions about the generation process:

A1) the data are produced by a mixture model,

A2) there is a one to one correspondence between mixture components and classes.

In this formulation, every document is generated according to a probability distribution defined by a set of parameters, denoted Θ. The probability distribution consists of a mixture of components $c_j \in C = \{c_1, ..., c_{|C|}\}$. Each mixture component is parameterized by a disjoint subset of Θ. The data generation procedure for a document d_i can be understood as

(1) selecting a mixture component (a class) according to the mixture weights - *class prior probabilities* (also called "between-class" mixing proportions), $P(c_j|\Theta)$, then

(2) having the corresponding mixture component generate a document according to its own parameters with *class-conditional probability* $P(d_i|c_j;\theta_j)$ (probability of a document given its class).

The *unconditional probability* of generating document d_i independent of its class is given by

$$P(d_i; \Theta) = \sum_{j=1}^{|C|} P(c_j|\Theta) P(d_i|c_j; \theta_j) \tag{1}$$

where the mixture weights (class prior probabilities), written $\theta_{c_j} = P(c_j|\Theta)$, $0 \leq \theta_{c_j} \leq 1$, $\sum_{j=1}^{|C|} \theta_{c_j} = 1$ indicate the probabilities of selecting the different mixture components, $P(d_i|c_j; \theta_j)$. Clearly, $\Theta = \{(\theta_{c_j}, \theta_j) : j = 1, ..., |C|\}$ is an unknown parameter set. Each document has a class label. We assume there is a one-to-one correspondence between classes and mixture model components, and thus the use c_j to indicate both the jth mixture component and the jth class.

Given a labeled training data we can calculate Bayes optimal estimates of the model parameters. Then equipped with these estimates, we classify new test documents by using Bayes rule to turn the generative model around and calculate the posterior probability that the class would generate the test document in question. Classification The class predicted for a test document is simply the class with the highest posterior probability.

3 Standard Models with Naive Bayes Classifier

Text classification is not only a task of practical significance, but is also a domain in which obtaining labeled data is expensive, since the human effort of reading and assigning documents to categories is almost required. Hence, the large number of parameters often must be estimated from a small amount of labeled data. When little training data is being used to estimate the parameters for a large number of features, it is often best to use a simple learning methods. In such cases, there is not enough data to estimate of feature correlations and other complex interactions. One such classification method that performs surprisingly well given its simplicity is *naive Bayes* classifier based on so-called "naive Bayes assumption": The words of the document are generated independently of the word's context, that is independently of the other words in the same document given the class label and furthermore, independent on its position in the document. It means the probabilistic models ignore the sequence in which the words occur. Naive Bayes is not always the best performing classification algorithm for text classification ([16, 8]), but it continues to be widely used for purpose because it is efficient and simple to implement.

First let us introduce some notation to describe text document. A document d_i is an ordered sequence of words $\{w_{d_{i,1}}, w_{d_{i,2}}, ...\}$. We write $w_{d_{i,k}}$ for the word

w_t in position k of the document d_i, where w_t is a word in the vocabulary $V = \{w_1, ..., w_{|V|}\}$. In *bag-of-words* or *unigram* representation, each document d_i is described by a feature vector consisting of one feature variable d_{it} for each word w_t in the document, $t \in \{1, ..., |V|\}$. It means $d_i = (d_{i1}, ..., d_{i|V|})$ is the $|V|$-dimensional feature vector representing the document. The feature variables d_{it} can indicate either the presence or absence of the word w_t or can indicate some measure of the frequency of the word w_t in the document d_i.

When the document is to be generated by a particular mixture component c_j, a document length, $|d_i|$ is chosen independently of the mixture component. Then the selected mixture component generates a word sequence of the specified length. We furthermore assume, it generates each words independently of the length.

Therefore, we can expand the second term from (1), and express the probability of a document length and the words in the document. Thus, the probability of a word must be conditioned on all the words that precede it:

$$P(d_i|c_j;\theta_j) = P(\{w_{d_{i,1}}, ..., w_{d_{i,|d_i|}}\}|c_j;\theta_j) =$$
$$= P(|d_i|) \prod_{k=1}^{|d_i|} P(w_{d_{i,k}}|c_j;\theta_j;w_{d_{i,l}}, l < k). \tag{2}$$

The naive Bayes assumption we can express as:

$$P(w_{d_{i,k}}|c_j;\theta_j;w_{d_{i,l}}, l < k) = P(w_{d_{i,k}}|c_j;\theta_j)\ . \tag{3}$$

Combining Equations (2) and (3) gives the naive Bayes expression for the class-conditional probability of a document:

$$P(d_i|c_j;\theta_j) = P(|d_i|) \prod_{k=1}^{|d_i|} P(w_{d_{i,k}}|c_j;\theta_j)\ . \tag{4}$$

Thus the parameters of jth mixture component of mixture (1) are the collection of word probabilities, each written $\theta_{t|j} = P(w_t|c_j;\theta_j)$, $0 \leq \theta_{t|j} \leq 1$, where $t = 1, ..., |V|$. Since we assume, that for all classes, document length is identically distributed, it does not need to be parameterized for classification. The only other parameters of the model are the mixture weights, written θ_{c_j}, which indicate the probabilities of selecting the different mixture components. Thus the complete collection of model parameters is a set $\Theta = \{(\theta_{c_j}, \theta_j) : j = 1, ..., |C|\}$, $\theta_j = (\theta_{1|j}, ..., \theta_{|V||j})$.

Naive Bayes assumption allows the generative model to be characterized with a greatly reduced number of parameters (see e.g.[13, 16]).

There are two common approaches to naive Bayes text classification [13]. One, the *multivariate Bernoulli* model, is a model with no dependencies between words and binary word's counts; the document is represented by a feature vector and each feature is variable indicating the presence or absence of a particular word in that document. The other approach, the *multinomial* model, is a model with integer word counts; the document is represented by a feature vector and each feature is variable indicating the number of a particular word in that document.

Now we expand our notion by of how a document is generated by an individual mixture component $P(d_i|c_j;\theta_j)$ in (1).

3.1 Multivariate Bernoulli model

In the multivariate Bernoulli model a document is a binary vector over the space of words. Dimension t, $t \in \{1, ..., |V|\}$ of the feature vector for document d_i is written B_{it}, and is either 0 or 1, indicating whether word w_t occurs at least once in the document. With such a document representation and naive Bayes assumption the probability of a document given its class from (1) is the product of the form

$$P(d_i|c_j;\theta_j) = \prod_{t=1}^{|V|} P(w_t|c_j;\theta_j)^{B_{it}}(1 - P(w_t|c_j;\theta_j))^{(1-B_{it})} . \tag{5}$$

In this model a document can be therefore seen as a collection of multiple independent Bernoulli experiments, one for each word in the vocabulary V with the probabilities for each of this words defined by each generative component, $\theta_{t|j} = P(w_t|c_j;\theta_j),\ 0 \le \theta_{t|j} \le 1$. The complete collection of model parameters is a set $\Theta = \{(\theta_{c_j}, \theta_j),\ j = 1, ..., |C|\},\ \ \theta_j = (\theta_{1|j}, ..., \theta_{|V||j})$.

Given a set of labeled training data, $\mathcal{D} = \{d_1, ..., d_{|D|}\}$, learning the parameters of a probabilistic classification model corresponds to estimating each of the class-conditional word probabilities $\theta_{t|j}$ and class priors θ_{c_j}. Define $P(c_j|d_i) \in \{0, 1\}$ as given by the documents class label. Then, using Laplacean prior, priming each word's absence and occurrence count with a count of one to avoid probabilities of zero or one, we estimate $\theta_{t|j}$ with:

$$\hat{\theta}_{t|j} = \frac{1 + \sum_{i=1}^{|\mathcal{D}|} B_{it} P(c_j|d_i)}{2 + \sum_{i=1}^{|\mathcal{D}|} P(c_j|d_i)}. \tag{6}$$

The class prior parameters, θ_{c_j} are estimated by maximum-likelihood estimate:

$$\hat{\theta}_{c_j} = \frac{\sum_{i=1}^{|\mathcal{D}|} P(c_j|d_i)}{|\mathcal{D}|}. \tag{7}$$

3.2 Multinomial model

In the multinomial model the document d_i is represented by a feature vector, each dimension t of this is the count of the number of times word w_t occurs in document d_i written by N_{it}. In this model each document d_i is drawn from a multinomial distribution over the set of all words in the vocabulary V with as many independent trials as the length $|d_i|$. The probability of a document d_i given its class from (1) is the multinomial distribution:

$$P(d_i|c_j;\theta_j) = P(|d_i|)\frac{|d_i|!}{\prod_{t=1}^{|V|} N_{it}!}\prod_{t=1}^{|V|} P(w_t|c_j;\theta_j)^{N_{it}} \tag{8}$$

where the length of the document d_i is $|d_i| = \sum_{t=1}^{|V|} N_{it}$. Again, the parameters of the generative component for each class are the probabilities for each word, written $\theta_{t|j} = P(w_t|c_j;\theta_j)$, $0 \leq \theta_{t|j} \leq 1$, $\sum_{t=1}^{|V|} \theta_{t|j} = 1$. The complete collection of model parameters is a set $\Theta = \{(\theta_{c_j}, \theta_j) : j = 1, ..., |C|\}$, $\theta_j = (\theta_{1|j}, ..., \theta_{|V||j})$. In contrast to the multivariate Bernoulli model, the multinomial model captures word frequency information in documents.

The word probability estimates $\hat{\theta}_{t|j}$ are using Laplacean prior:

$$\hat{\theta}_{t|j} = \frac{1 + \sum_{i=1}^{|\mathcal{D}|} N_{it} P(c_j|d_i)}{|V| + \sum_{r=1}^{|V|}\sum_{i=1}^{|\mathcal{D}|} N_{ir} P(c_j|d_i)} \tag{9}$$

(i.e. adding one to all the word counts for a class).

Witten-Bell smoothing [26] sets $\hat{\theta}_{t|j}$ as follows:

$$\hat{\theta}_{t|j} = \begin{cases} \frac{N(w_t,c_j)}{|V_j| + \sum_{r=1}^{|V|} N(w_r,c_j)} & \text{if } N(w_t,c_j) \neq 0 \\ \\ \frac{|V|}{|V_j| + \sum_{r=1}^{|V_j|} N(w_r,c_j)} \frac{1}{|V| - |V_j|} & \text{if } N(w_t,c_j) = 0 \end{cases} \tag{10}$$

where $N(w_t, c_j)$ is the count of the number of times word w_t occurs in the training data for class c_j ($N(w_t, c_j) = \sum_{i=1}^{|\mathcal{D}|} N_{it} P(c_j|d_i)$), $|V_j|$ is the total number of unique words in class c_j. Witten-Bell differs from Laplace smoothing in that it strength of the prior depends on the relationship between the number of unique words and the total number of word occurrences in the training data for the class; if most of the words are often repeated, the prior is weaker.

The class prior probability parameters are calculate as before according to (7).

3.3 Classification

Given estimates $\hat{\Theta}$ of the parameters Θ of the model (1) calculated from the training documents, classification can be performed on test document d by calculating the posterior probability of each class by applying Bayes rule

$$P(c_j|d;\hat{\Theta}) = \frac{P(c_j|\hat{\Theta})P(d|c_j;\hat{\theta}_j)}{P(d;\hat{\Theta})}, \quad \sum_{j=1}^{|C|} P(c_j|d;\hat{\Theta}) = 1. \tag{11}$$

If the task is to classify a test document into a single class, simply select the class $c^\star$ with the highest posterior probability

$$c^\star = \operatorname*{argmax}_{j=1,\ldots,|C|} P\left(c_j|d;\hat{\Theta}\right) = \operatorname*{argmax}_{j=1,\ldots,|C|} \left[P\left(c_j|\hat{\Theta}\right) P\left(d|c_j;\hat{\theta}_j\right)\right]. \tag{12}$$

The right hand side of (11) may be expanded by first substituting using equations (1) and (7). Then the expansion of the individual terms for this equation are dependent on the model. For the multi-variate Bernoulli model we use equations (5) and (6). For the multinomial model we use equations (8) and (9).

3.4 Feature selection

Feature selection is a very important step in text classification, because irrelevant and redundant features often degrade the performance of classification algorithms both in speed and prediction accuracy. Methods for feature subset selection for text document classification task use some evaluation function that is applied to a single feature. All features are independently evaluated, a score is assigned to each of them and the features are sorted according to the assigned score. Then, a predefined number of the best features is taken to form the best feature subset.

Scoring of individual features can be performed using some of the measures, for instance, *document frequency*, *term frequency*, *mutual information*, *information gain (average mutual information)*, χ^2 *statistic*, and *term strength*. Yang and Pedersen [27], Mladenic [15] give experimental comparison of the above mentioned measures in text classification. The average mutual information and a very simple frequency measures were reported to work well on text data.

For the experiments reported in the Sections 5.2 and 5.3 the *average mutual information* (AMI) of the word w_t has been used for reducing the vocabulary size. Using a multivariate Bernoulli model, the AMI between 1) the class of a document and 2) the absence or presence of a word w_t in the document has been calculating. Let C_j is random variable over all classes and F_t is a random variable over the absence or presence of word w_t in the document. The random variable F_t takes on values $f_t \in \{0,1\}$, $f_t = 0$ indicates the absence of the word w_t and $f_t = 1$ indicates the presence of w_t. The AMI is defined to be:

$$J(w_t) = \sum_{j=1}^{|C|} \sum_{f_t \in \{0,1\}} P(c_j, f_t) \log \frac{P(c_j, f_t)}{P(c_j)P(f_t)} \tag{13}$$

where
- $P(c_j)$ is the number of training documents in class c_j divided by the total number of documents;
- $P(f_t)$ is the number of training documents containing one or more occurrence of word w_t divided by the total number of documents;
- $P(c_j, f_t)$ is the number of training documents in class c_j that also contain the word w_t divided by the total number of documents.

In the case of the multinomial model, the AMI between 1) the class of the document from which the word occurrence is drawn, and 2) a random variable over all word occurrences has been used. In this case Eq. (13) is also used, but
- $P(c_j)$ is the number of word occurrences appearing in training documents in class c_j divided by the total number of word occurrences;
- $P(f_t)$ is the number of occurrences of word w_t in all documents, divided by the total number of word occurrences;
- $P(c_j, f_t)$ is the number of occurrences of word w_t that also appear in documents in class c_j, divided by the total number of word occurrences.

It has been proven that choosing features individually is not optimal [21]. For the experiments reported in the Section 5.4, the *multiclass Bhattacharrya distance* [2, 24] for multinomial class-conditional probabilities was used as a feature selection criterion to measure the ability of feature subsets in discriminating between classes and to take into consideration how features work together. Multiclass Bhattacharyya distance, as criterion function for sequential methods of feature selection, is defined by

$$B = \sum_{j=1}^{|C|-1} \sum_{k=j+1}^{|C|} \theta_{c_j} \theta_{c_k} B_{jk} \tag{14}$$

where θ_{c_j} and θ_{c_k} are class prior probabilities from equation (7) and B_{jk} is Bhattacharyya distance between two class-conditional probability functions $P(d|c_j;\theta_j)$ and $P(d|c_k;\theta_k)$. The explicit formula of (14) for multivariate Bernoulli and multinomial distributions is derived in Appendix B. For Bernoulli model is

$$B_{jk} = \sum_{t=1}^{|V|} \log \left(\sqrt{\theta_{t|j}\theta_{t|k}} + \sqrt{(1-\theta_{t|j})(1-\theta_{t|k})} \right) \tag{15}$$

and for mutltinomial model is

$$B_{jk} = -|d| \log \sum_{t=1}^{|V|} \sqrt{\theta_{t|j} \theta_{t|k}} \,. \tag{16}$$

We estimate $\theta_{t|j}$ and $\theta_{t|k}$ by variables from equation (6) for multivariate Bernoulli model and by (9) for multinomial model. $|V|$ is size of selected subsets of words and $j, k = 1, \ldots, |C|$, $j \neq k$. After substitution we obtain Bhattacharyya distance criterion function for Bernoulli model

$$B = \sum_{j=1}^{|C|-1} \sum_{k=j+1}^{|C|} \theta_{c_j} \theta_{c_k} \sum_{t=1}^{|V|} \log \left(\sqrt{\theta_{t|j} \theta_{t|k}} + \sqrt{(1 - \theta_{t|j})(1 - \theta_{t|k})} \right) \tag{17}$$

and for multinomial model

$$B = -|d| \sum_{j=1}^{|C|-1} \sum_{k=j+1}^{|C|} \theta_{c_j} \theta_{c_k} \log \sum_{t=1}^{|V|} \sqrt{\theta_{t|j} \theta_{t|k}} \,. \tag{18}$$

3.5 Summary

While the naive Bayes assumption is clearly false in the most real-word tasks, naive Bayes classifier often performs classification very well (see e.g. [13, 27, 8, 16]). McCallum and Nigam in the paper [13] has compared the theory and practice of two different probabilistic classifiers, both of which makes the naive Bayes assumption. The authors found that the multivariate Bernoulli model performs well with small vocabulary sizes, but that multinomial model usually performs better at larger vocabulary sizes, providing on average a 27% reduction in error over the multivariate Bernoulli model at any vocabulary size.

4 Finite Mixture Models

In a simple text document classification problem, there is a fixed, known number of classes and each class is modelled by a single component. A collection of class-labelled training data is used to estimate the class priors and class-conditional probabilities, following the conventional supervised statistical learning approach. Then tested new document is classified by maximum posterior probability.

While the naive Bayes classifier is still considered to be among the most accurate text classifiers, it is commonly accepted that better classifiers can be designed by simply relaxing the class-conditional independence assumption. The usage of finite mixtures for class-conditional probability functions is a useful method in pattern recognition, because mixture models are able to represent arbitrarily complex probability functions. Mixtures are flexible enough for finding an appropriate tradeoff between model complexity and the amount of the training data available. Usually, model complexity is controlled by varying the number of mixture components while keeping the same parametric form for all components. If each class is to be modelled by more then one component, we face an unsupervised learning problem. Maximum-likelihood (ML) estimation of mixture parameters can be reliably accomplished by *expectation-maximization* (EM) algorithm.

Although mixture models are important for continuous data, some research shows, that these models perform, with discrete date in text classification, very well too [9, 10].

Our approach to learning on text document is based on the fact that documents are often mixtures of multiple topics. Words within the document are not independent of each other. In mixture approach to text document classification the jth class-conditional probability $P(d_i|c_j;\Theta_j)$ is modeled as a finite mixture of the same M probabilities with its own parameters. That is it can be expressed as

$$P(d_i|c_j;\Theta_j) = \sum_{m=1}^{M} \alpha_{jm} P_m(d_i|s_{jm};\theta_{jm}) \; . \tag{19}$$

The unknown parameter set is $\Theta_j = \{(\alpha_{jm}, \theta_{jm}) : m = 1, ..., M\}$, $j = 1, ..., |C|$. Here M denotes the number of latent subclasses, say s_{jm}, in each class c_j, α_{jm} is the mixing proportion of the mth model component (or model component s_{jm}) in jth class (or in class c_j), (also called "within-class" mixing proportion), $\alpha_{jm} \geq 0$, $\sum_{m=1}^{M} \alpha_{jm} = 1$, $j = 1, ..., |C|$. $P_m(d_i|s_{jm};\theta_{jm})$ denotes the probability of the document d_i in the mth subclass within of the class c_j. Note, that in proposed approach each class c_j is divided into M artificial subclasses s_{jm}.

The unconditional probability of document d_i independent of its class is now given by

$$P(d_i;\Theta) = \sum_{j=1}^{|C|} P(c_j|\Theta) P(d_i|c_j;\Theta_j) \tag{20}$$

where $\Theta = \{(\theta_{c_j}, \Theta_j) : j = 1, ..., |C|\}$.

4.1 Bernoulli mixture model

A Bernoulli mixture model is a mixture of the form (19) such that each component has a multivariate Bernoulli probability function

$$P_m(d_i|s_{jm};\theta_{jm}) = \prod_{t=1}^{|V|} \theta_{t|jm}^{B_{it}} (1-\theta_{t|jm})^{(1-B_{it})}, \quad m = 1, ..., M. \tag{21}$$

Here associated with each individual subclass s_{jm} is a word probability written $\theta_{t|jm} = P(w_t|s_{jm};\theta_{jm})$ for all words in the vocabulary $|V|$, $0 \leq \theta_{t|jm} \leq 1$.

Therefore, the probability of the document given its class is approximated by multivariate Bernoulli mixture

$$P(d_i|c_j;\Theta_j) = \sum_{m=1}^{M} \alpha_{jm} \prod_{t=1}^{|V|} \theta_{t|jm}^{B_{it}} (1-\theta_{t|jm})^{(1-B_{it})} \ . \tag{22}$$

The parameter set $\Theta_j = \{(\alpha_{jm}, \theta_{jm}) : m = 1, ..., M\}$, $j = 1, ..., |C|$ with $\theta_{jm} = (\theta_{1|jm}, ..., \theta_{|V||jm})$ is unknown.

4.2 Mixture of multinomials

We proposed to use the mixture of multinomial distributions as a model for the text document classification task. It means that text document in jth class has the form

$$P(d_i|c_j;\Theta_j) = \sum_{m=1}^{M} \alpha_{jm} P_m(d_i|s_{jm}, \theta_{jm}) \tag{23}$$

where

$$P_m(d_i|s_{jm};\theta_{jm}) = \frac{|d_i|!}{\prod_{t=1}^{|V|} N_{it}!} \prod_{t=1}^{|V|} \theta_{t|jm}^{N_{it}}, \quad m = 1, ..., M \ . \tag{24}$$

The parameter set $\Theta_j = \{(\alpha_{jm}, \theta_{jm}) : m = 1, ..., M\}$, $j = 1, ..., |C|$ with $\theta_{jm} = (\theta_{1|jm}, ..., \theta_{|V||jm})$ is unknown, $\sum_{t=1}^{|V|} \theta_{t|jm} = 1$.

To our knowledge, multinomial mixture model have not been applied earlier to document classification tasks.

4.3 Model fitting with the EM algorithm

Let $\mathcal{D}_j = \{d_1, ..., d_{|\mathcal{D}_j|}\}$ be a set of $|\mathcal{D}_j|$ independent and identically distributed training documents from class $c_j \in C$. We find out the parameters of class-conditional probability by maximizing the log-likelihood function

$$L_j = \sum_{i=1}^{|\mathcal{D}_j|} \log \left[\sum_{m=1}^{M} \alpha_{jm} P_m(d_i|s_{jm}; \theta_{jm}) \right] \tag{25}$$

where $\Theta_j = \{(\alpha_{jm}, \theta_{jm}) : m = 1, ..., M\}$, $j = 1, ..., |C|$ stands for the model parameters to be estimated (notice that α_{jM} is redundant as $\alpha_{jM} = 1 - \sum_{m=1}^{M-1} \alpha_{jm}$). We are excluding the number of class-conditional components from the estimation problem, it is we assume the number of components is known. The estimate $\hat{\Theta}_j$ cannot be found analytically. A possible approach is the *Expectation Maximization* (EM) algorithm which is a general framework to incomplete data problems [1]. From this point of view, an observed document d_i can be regarded as being incomplete where the missing part is the true subclass labeling.

The EM algorithm alternates two steps: E-step (an expectation step) where posterior probabilities are computed for the latent subclass variables, based on current estimates of the parameters, (2) M-step (maximization step) where parameters are updated based on so called expected complete data log-likelihood which depends on the posterior probabilities computed in the E-step.

The EM algorithm derived for models (22) and (23) in the Appendix A, takes the following form:

E-Step: $j = 1, ...|C|,\ m = 1, ..., M,\ i = 1, ..., |\mathcal{D}_j|,\ k = 0, 1...$

$$p_B^{(k)}(s_{jm}|d_i) = \frac{\alpha_{jm}^{(k)} \prod_{t=1}^{|V|} \left(\theta_{t|jm}^{(k)}\right)^{B_{it}} \left(1 - \theta_{t|jm}^{(k)}\right)^{1-B_{it}}}{\sum_{r=1}^{M} \alpha_{jr}^{(k)} \prod_{t=1}^{|V|} \left(\theta_{t|jr}^{(k)}\right)^{B_{it}} \left(1 - \theta_{t|jr}^{(k)}\right)^{1-B_{it}}} \tag{26}$$

in the case of model (22) or

$$p_M^{(k)}(s_{jm}|d_i) = \frac{\alpha_{jm}^{(k)} \prod_{t=1}^{|V|} \left(\theta_{t|jm}^{(k)}\right)^{N_{it}}}{\sum_{r=1}^{M} \alpha_{jr}^{(k)} \prod_{t=1}^{|V|} \left(\theta_{t|jr}^{(k)}\right)^{N_{it}}} \tag{27}$$

in the case of model (23).

M-Step: $j = 1, ...|C|,\ m = 1, ..., M,\ t = 1, ..., |V|$

$$\alpha_{jm}^{(k+1)} = \frac{1}{|\mathcal{D}_j|} \sum_{i=1}^{|\mathcal{D}_j|} p_B^{(k)}(s_{jm}|d_i) \tag{28}$$

and

$$\theta_{t|jm}^{(k+1)} = \frac{\sum_{i=1}^{|\mathcal{D}_j|} B_{it}\; p_B^{(k)}(s_{jm}|d_i)}{\sum_{i=1}^{|\mathcal{D}_j|}\; p_B^{(k)}(s_{jm}|d_i)} \tag{29}$$

in the case of model (22).
For mixture model (23) we obtain

$$\alpha_{jm}^{(k+1)} = \frac{1}{|\mathcal{D}_j|} \sum_{i=1}^{|\mathcal{D}_j|} p_M^{(k)}(s_{jm}|d_i) \tag{30}$$

and

$$\theta_{t|jm}^{(k+1)} = \frac{\sum_{i=1}^{|\mathcal{D}_j|} N_{it}\; p_M^{(k)}(s_{jm}|d_i)}{\sum_{r=1}^{|V|} \sum_{i=1}^{|\mathcal{D}_j|} N_{ir}\; p_M^{(k)}(s_{jm}|d_i)}\;. \tag{31}$$

The class priors can be estimated as

$$\hat{\theta}_{c_j} = \frac{|\mathcal{D}_j|}{|\mathcal{D}|}. \tag{32}$$

An initial value for the parameters is required to start the EM algorithm. For the experiments reported in Section 5.3 we used random initialization for parameters $\theta_{t|jm}$ and $\alpha_{jm}^{(0)} = 1/M$ for both Bernoulli and multinomial model.

Remark: The model (22) has been investigated by Juan and Vidal in [9, 10]. The authors focused on the use of mixtures of multivariate Bernoulli distributions for a text classification task aimed at improving language modelling for machine translation. Their experimental results have been reported showing that significant improvements can be obtained by using Bernoulli mixtures as a models for the class-conditional probability functions.

5 Experimental Results

This section provides empirical evidence that the mixture models (multivariate Bernoulli mixture and the mixture of multinomials) proposed in Section 4, perform better than the corresponding standard models introduced in Section 3.

5.1 Reuters data set

For performance evaluation, we carried out the experiments in Sections 5.2 and 5.3 with the *Reuters*-21578 data collection[1], whose documents were collected

[1] http://www.daviddlewis.com/resources/testcollections/reuters21578.

from the newspaper articles. The Reuters data have been used by various authors (e.g. [27, 28, 13, 16, 4, 22]) as a test deck for evaluating differing methods of text classification. Since the aim is classification, in which each document has an exclusive category, we discarded documents with no label or with multiple labels. Furthermore, the classes with less than twenty documents. The resulting data set had 9159 documents with 33 document classes. We randomly split data set into two-third training set and one-third testing set. We repeated this random split twenty times. Each from twenty training sets contains two thirds of all data. The vocabulary was constructed by removing of stop words, too infrequent words (words that had less than four occurrences per document) and the Porter stemming algorithm[2] was used. This resulted in a vocabulary of 5935 words.

Following a traditional feature selection technique for text classification, non-discriminative words were removed in accordance to the *average mutual information* criterion defined in (13). It was computed from the training set for each word, and the v most informative words were selected. Then each document was represented as a v-dimensional feature vector. As the dimension v has an important effect in classification performance, several values of v were considered in the experiments.

The effectiveness of a text classifier is measured by classification accuracy estimated as

$$accuracy = \frac{N_c}{N_{test}} 100\% \tag{33}$$

where N_c is the number of correctly classified documents from testing set and N_{test} is the total number of documents in the testing set (percentage of the test documents that were correctly classified). The average classification accuracy is computed over all twenty testing sets.

5.2 Comparison of standard models

Related work: Juan and Vidal [9] compared standard Bernoulli model to mixture Bernoulli model with various number of components and they reached a similar results.

Figure 1 shows classification accuracy results for Bernoulli and multinomial model with the varying dimension of document representation. The horizontal axis indicates the number of features in feature vector document representation - *vocabulary size* that were used in the classification model. The classification accuracy is represented by the vertical axis.

[2] http://www.tartarus.org/˜martin/PorterStemmer.

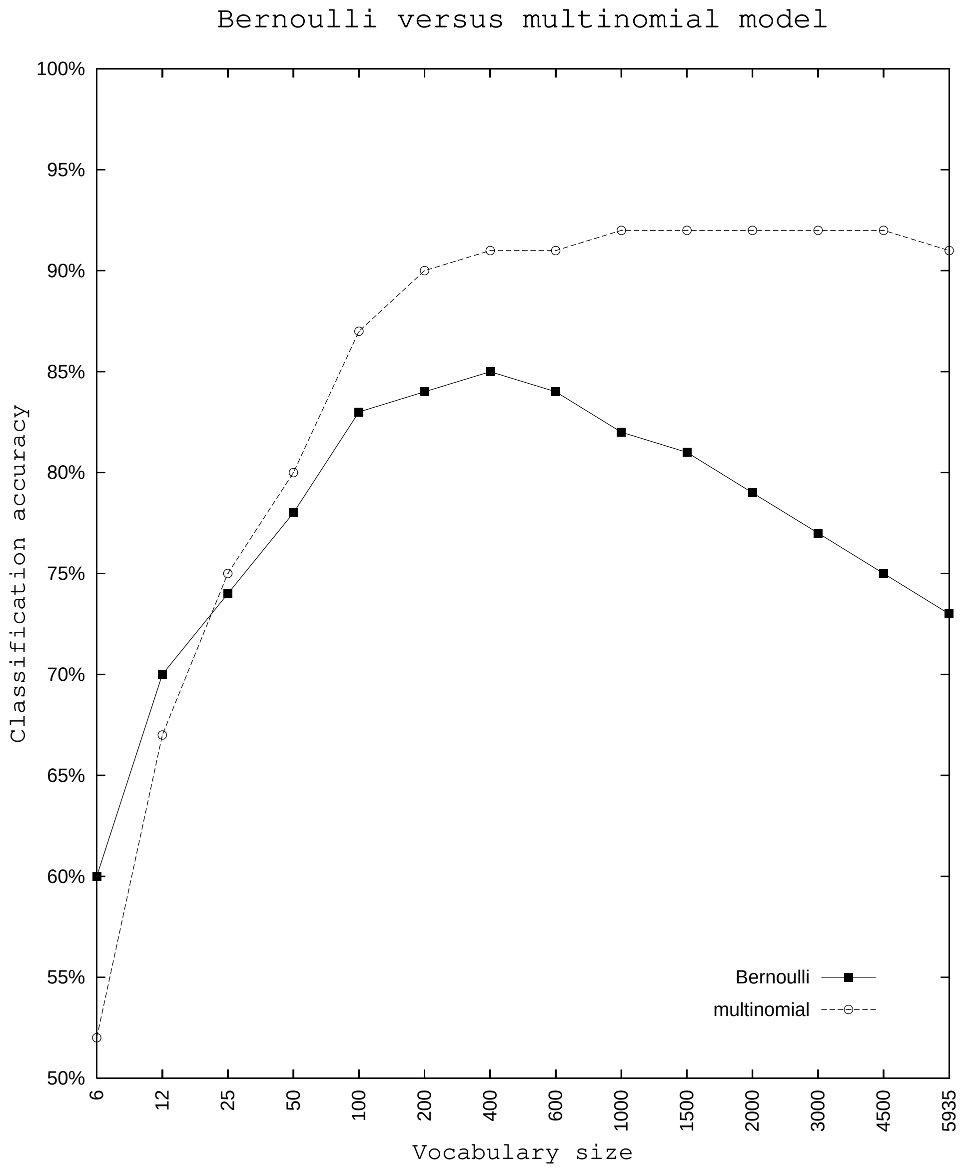

Figure 1: Classification accuracy using Bernoulli and multinomial model for several vocabulary sizes.

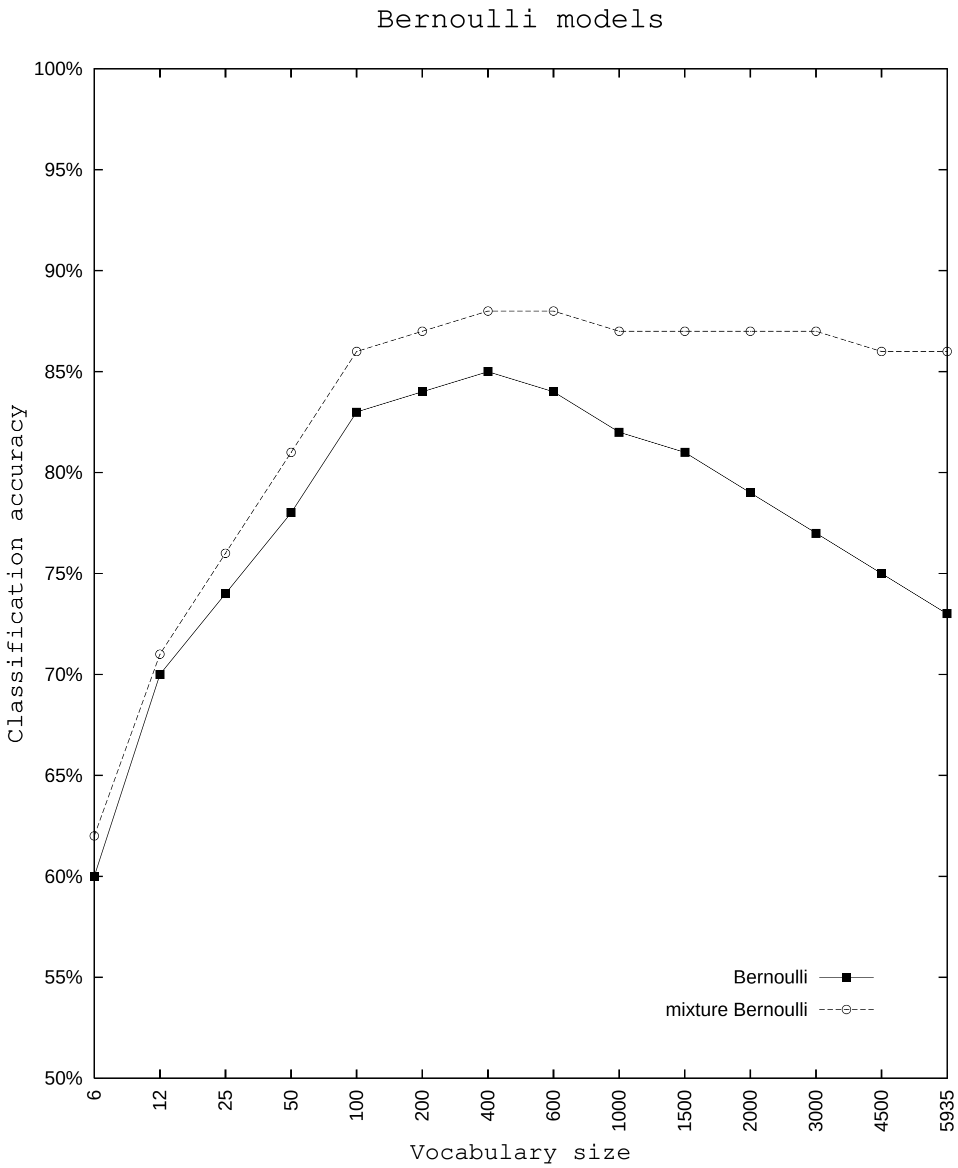

Figure 2: Classification accuracy of Bernoulli and mixture Bernoulli model for several vocabulary sizes.

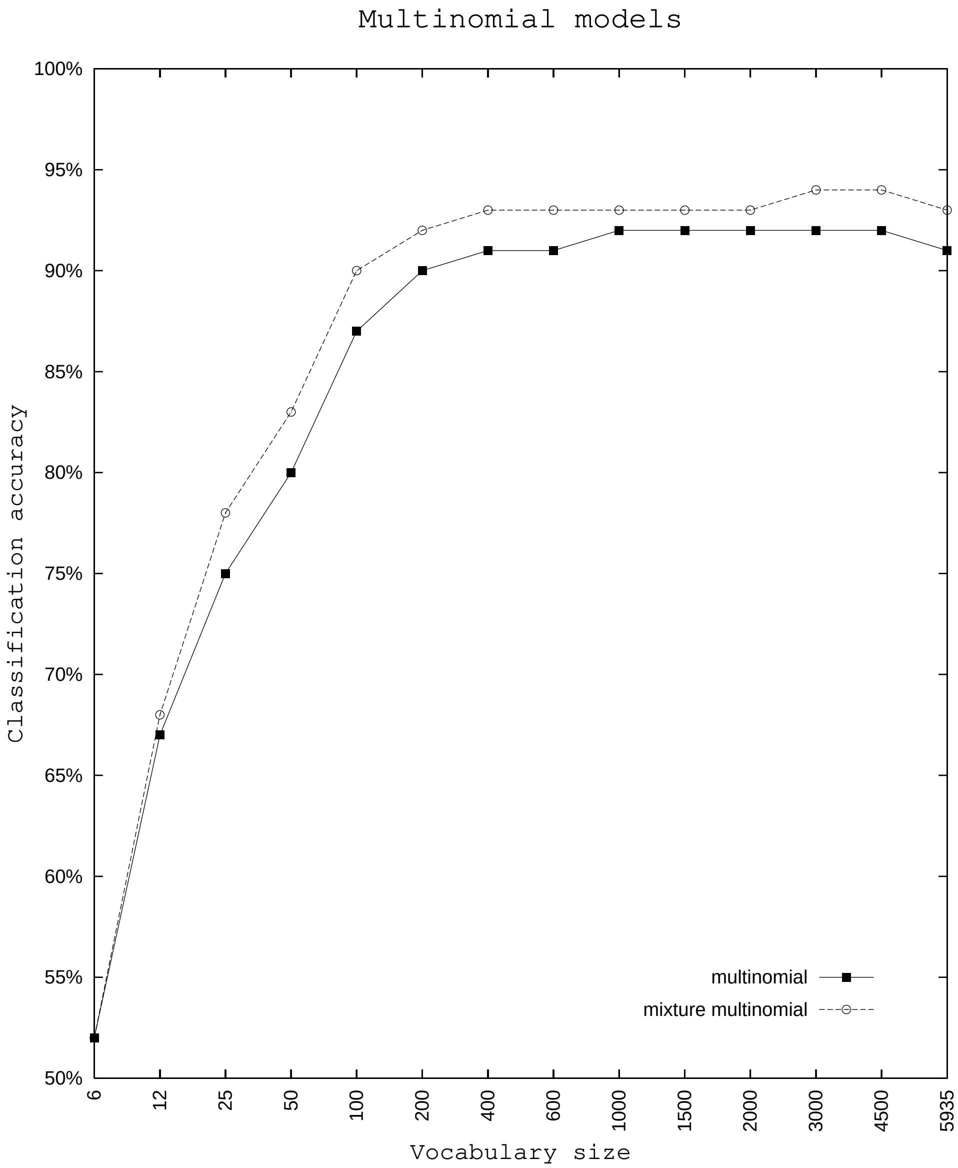

Figure 3: Classification accuracy of multinomial and mixture of multinomials for several vocabulary sizes.

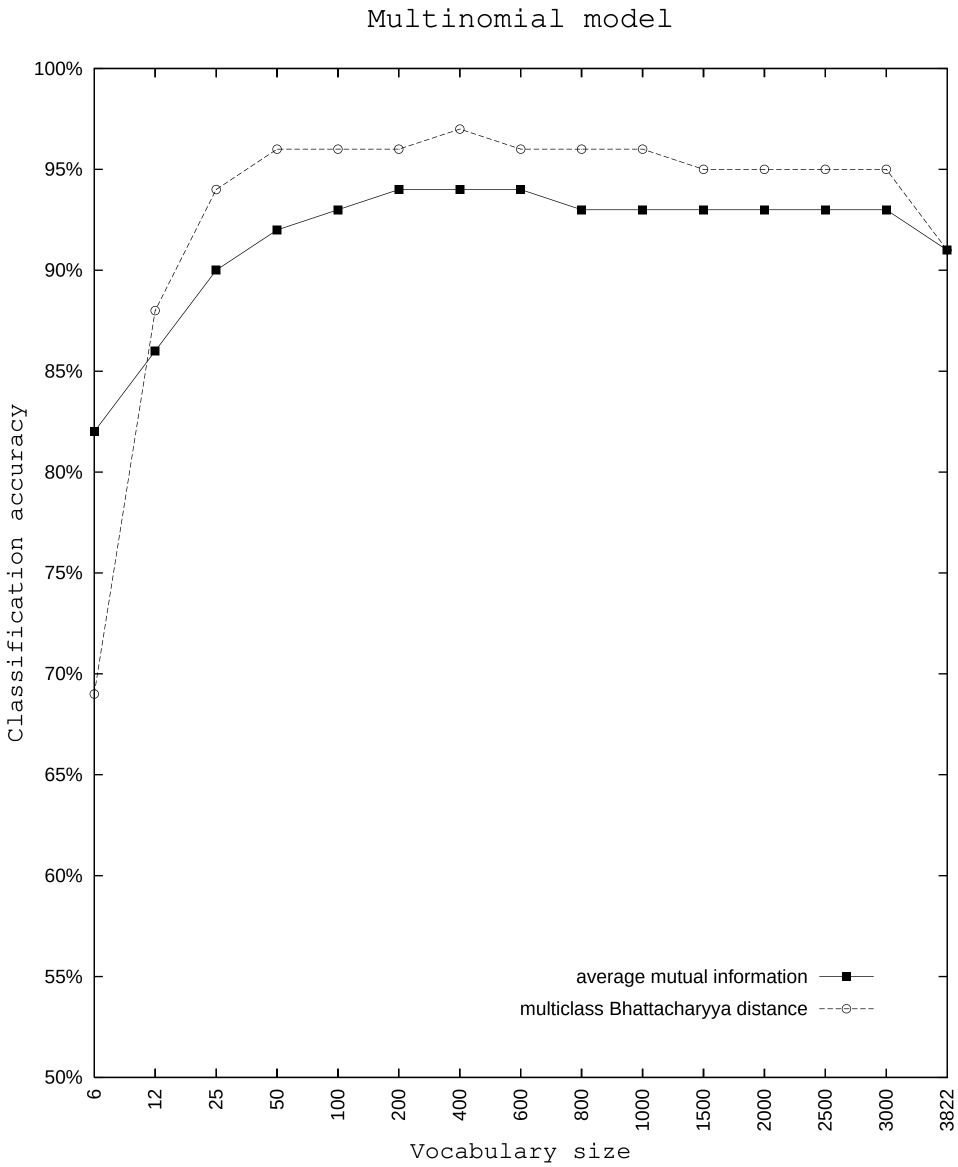

Figure 4: Classification accuracy using AMI and MBD.

We can see that the multivariate Bernoulli model performs well with extremely small dimensions of document representation (vocabulary sizes), but the multinomial model performs better at larger vocabulary sizes. Accuracy of multinomial model improves monotonically with the size of vocabulary, though is a point (for the size larger than 200) after which the improvements become small or some worsening is observed. Good behaviour of the Bernoulli model is observed for dimensions equal or smaller than 400, after this point the performance of the classifier degrades with an increase in the number of dimensions ("peaking phenomenon") [7].

5.3 Standard models versus mixture models

The behaviour of the standard and mixture models for text classification has been compared. Experiment A: Figure 2 shows the performance of both standard Bernoulli and mixture Bernoulli model for varying number of dimensions trained with three components for each class-conditional mixture. We can see that the accuracy of Bernoulli mixture model improves monotonically with the size of vocabulary, though is a point (for the size equal or larger than 100) after which the improvements become small or some worsening is observed. Bernoulli mixture model is found to be better than Bernoulli model. It increases the accuracy by an average of 5%. accuracy of the multinomial and the mixture of multinomials has been compared in Figure 3. Good behaviour was observed for dimensions equal or larger than 250 dimensions. The mixture of multinomials has achieved the highest accuracy 94% and is on average 2% better than in the case of standard multinomial model. The mixture of multinomials performed better results over all vocabulary sizes, except on very small number of words (6 and 12).

The number of Bernoulli or multinomial components for each class conditional mixture also has an important impact in performance. Good behaviour has been observed with the three or more components per class.

5.4 Comparison feature selection methods

Two criteria has been used in obtaining subset of features representing the document using the multinomial model. The first criterion is *average mutual information* defined in (13) and is used with *Best Individual Features* search procedure [7]. The second evaluation function is the *multiclass Bhattacharyya* distance (MBD) defined in (18) and is used with *Sequential Forward Selection* search algorithm [7]. In Figure 4 the comparison of classification accuracy of Bayes classifier on the same Reuters data as in [4] using AMI criterion that does not take into consideration the relationship between features, with MBD criterion is depicted. When compared the MBD with AMI feature selection method an increase in classification accuracy of about 2-5% was achieved.

6 Conclusions

The following conclusions are reached from this study:

- The mixtures of multivariate Bernoulli distributions and mixtures of multinomial distributions has been used as class-conditional models for the task of text document classification to relax the naive Bayes class-conditional independence assumption. This generalization of naive Bayes tries to properly model significant class-conditional dependencies by spreading them over different mixture components.

- An observation of the performance of Bayes classifier for text classification on Reuters data suggests that learning methods based on mixture models for class-conditional probabilities of the documents (mixtures of multivariate Bernoulli distributions and mixtures of multinomials) performed better than the corresponding standard models.

- The mixture of multinomials is promising model for text classification. The experimental results are reported showing that the mixture of multinomials for class-conditional distributions performs better than the mixture of multivariate Bernoulli model.

- The experimental results on data used in [4] suggest that the multiclass Bhattacharyya distance can be a suitable feature selection criterion for the text document classification task.

A Derivation of the EM algorithm

In this appendix, we briefly state the mixture model problem and its Expectation-Maximization (EM) solution.

We consider a finite mixture of probability functions

$$P(d|c_j;\Theta_j) = \sum_{m=1}^{M} \alpha_{jm} P_m(d|s_{jm},\theta_{jm}) \ , \tag{34}$$

where d is $|V|$-dimensional vector representing a document d, M is the number of mixture components. The complete set of parameters for mixture $P(d|c_j;\Theta_j)$ defined in (34) is $\Theta_j = \{(\alpha_{jm},\theta_{jm}) : m = 1,...,M\}$, where θ_{jm} is a parameter vector specifying the m-th mixture component $P_m(d|s_{jm},\theta_{jm})$ in the class c_j, α_{jm} denotes the mixing weight of the m-th mixture component satisfying $0 < \alpha_{jm} < 1$, $\sum_{m=1}^{M} \alpha_{jm} = 1$. We assume a fixed number of mixture components M.

Maximum likelihood (ML) parameter estimation leads to the optimization of the log-likelihood function of the parameters Θ_j given a training data set $\mathcal{D}_j = \{d_1,...,d_{|\mathcal{D}_j|}\}$

$$L(\Theta_j) = \log \prod_{i=1}^{|\mathcal{D}_j|} P(d_i|c_j;\Theta_j) = \sum_{i=1}^{|\mathcal{D}_j|} \log \left[\sum_{m=1}^{M} \alpha_{jm} P_m(d_i|s_{jm};\theta_{jm}) \right] . \tag{35}$$

The ML estimate $\hat{\Theta}_j = \operatorname{argmax}_{\Theta_j} L(\Theta_j)$ cannot be found analytically in the case of mixture models.

The EM algorithm [1] maximizes $L(\Theta_j)$ iteratively maximizing the function

$$Q(\Theta_j,\Theta_j^{(k)}) = \sum_{m=1}^{M} \sum_{i=1}^{|\mathcal{D}_j|} p^{(k)}(s_{jm}|d_i) \Big[\log \alpha_{jm} P_m(d_i|s_{jm};\theta_{jm}) \Big] \tag{36}$$

with respect to the parameter set Θ_j given the previous estimate $\Theta_j^{(k)}$.
Here,

$$p^{(k)}(s_{jm}|d_i) = \frac{\alpha_{jm}^{(k)} P_m(d_i|s_{jm};\theta_{jm}^{(k)})}{\sum_{r=1}^{M} \alpha_{jr}^{(k)} P_m(d_i|s_{jr};\theta_{jr}^{(k)})} \tag{37}$$

denotes the a posteriori probability, given $\mathcal{D}_j$ and $\Theta_j^{(k)}$, that d_i belongs to the mixture component $P_m(d_i|s_{jm};\theta_{jm})$. In [1], it is proven that maximizing $Q(\Theta_j,\Theta_j^{(k)})$ is equivalent to maximizing $L(\Theta_j)$. This maximization problem is solved either by differentiating Q with respect to each parameter in the set Θ_j, setting the resulting partial derivatives equal to zero, and then solving for each parameter, or by method of Lagrange multipliers if parameter constraints are necessary. The expression for Q in equation (36) may be rewritten as

$$Q(\Theta_j, \Theta_j^{(k)}) = Q_\alpha + Q_\theta = \tag{38}$$
$$\sum_{m=1}^{M} \sum_{i=1}^{|\mathcal{D}_j|} p^{(k)}(s_{jm}|d_i) \log \alpha_{jm} + \sum_{m=1}^{M} \sum_{i=1}^{|\mathcal{D}_j|} p^{(k)}(s_{jm}|d_i) \log P_m(d_i|s_{jm};\theta_{jm})$$

where Q_α depends only on $\{\alpha_{jm}, m = 1, ..., M\}$, and Q_θ depends only on $\{\theta_{jm}, m = 1, ..., M\}$. Consequently, we can maximize the term containing α_{jm} and the term containing θ_{jm} independently since they are not related. If Q_α and Q_θ are each maximized separately, then Q is also maximized.

The EM algorithm starts with some initial guess at the ML parameters, $\Theta_j^{(0)}$ and then proceeds iteratively generate estimates $\Theta_j^{(1)}, \Theta_j^{(2)}, ...$ by repeatedly applying the following two steps until some convergence criterion is met:

- *E-step:* For $m = 1, 2, ..., M$ and $i = 1, 2, ..., |\mathcal{D}_j|$ compute a posteriori probabilities $p^{(k)}(s_{jm}|d_i)$ using equation (37) from the current parameters $(\alpha_{jm}^{(k)}, \theta_{jm}^{(k)})$ at iteration k.
- *M-step:* Updates the parameter estimates according to

  $$\Theta_j^{(k+1)} = \underset{\Theta_j}{\text{argmax}}\ Q(\Theta_j, \Theta_j^{(k)}) \tag{39}$$

  under the constraints $\sum_{m=1}^{M} \alpha_{jm} = 1$.

It leads to

$$\alpha_{jm}^{(k+1)} = \frac{1}{|\mathcal{D}_j|} \sum_{i=1}^{|\mathcal{D}_j|} p^{(k)}(s_{jm}|d_i) \tag{40}$$

and

$$\theta_{jm}^{(k+1)} = \underset{\theta_{jm}}{\text{argmax}} \sum_{i=1}^{|\mathcal{D}_j|} p^{(k)}(s_{jm}|d_i) \log P_m(d_i|s_{jm};\theta_{jm}) \ . \tag{41}$$

The exact formulas for the $p^{(k)}(s_{jm}|d_i)$, $\alpha_{jm}^{(k+1)}$ and $\theta_{jm}^{(k+1)}$ depend on the involved parametric family of distribution probabilities.

A.1 Multivariate Bernoulli mixture model

Now, we consider a finite mixture of multivariate Bernoulli distributions, that is, the j-th class-conditional probability function has the following form:

$$P(d|c_j;\Theta_j) = \sum_{m=1}^{M} \alpha_{jm} \prod_{t=1}^{|V|} \theta_{t|jm}^{B_{it}} (1-\theta_{t|jm})^{(1-B_{it})} \tag{42}$$

with parameters $\theta_{jm} = (\theta_{1|jm}, ..., \theta_{|V||jm}),\ B_{it} \in \{0,1\},\ t = 1, ..., |V|$.
The log-likelihood function of the parameters α_{jm}, θ_{jm} given a training set $\mathcal{D}_j$ is

$$L(\Theta_j) = \sum_{i=1}^{|\mathcal{D}_j|} \log \left[\sum_{m=1}^{M} \alpha_{jm} \prod_{t=1}^{|V|} \theta_{t|jm}^{B_{it}} (1-\theta_{t|jm})^{(1-B_{it})} \right]. \tag{43}$$

The function $Q(\Theta_j, \Theta_j^{(k)})$ is now

$$\begin{aligned} & Q(\Theta_j, \Theta_j^{(k)}) = \\ & = \sum_{m=1}^{M} \sum_{i=1}^{|\mathcal{D}_j|} p_B^{(k)}(s_{jm}|d_i) \left(\log \alpha_{jm} + \log \prod_{t=1}^{|V|} \theta_{t|jm}^{B_{it}} (1-\theta_{t|jm})^{(1-B_{it})} \right) \end{aligned} \tag{44}$$

It is easily seen that

$$\frac{\partial Q_\alpha}{\partial \alpha_{jm}} = \frac{1}{\alpha_{jm}} \sum_{i=1}^{|\mathcal{D}_j|} p_B^{(k)}(s_{jm}|d_i) - |\mathcal{D}_j|, \quad m = 1, ..., M \tag{45}$$

$$\begin{aligned} \frac{\partial Q_\theta}{\partial \theta_{t|jm}} &= \sum_{i=1}^{|\mathcal{D}_j|} p_B^{(k)}(s_{jm}|d_i) \left(\frac{B_{it}}{\theta_{t|jm}} - \frac{1-B_{it}}{1-\theta_{t|jm}} \right) \\ &= \frac{1}{\theta_{t|jm}(1-\theta_{t|jm})} \sum_{i=1}^{|\mathcal{D}_j|} (B_{it} - \theta_{t|jm}) p_B^{(k)}(s_{jm}|d_i) \end{aligned} \tag{46}$$

$$m = 1, ..., M,\ t = 1, ...|V|,$$

where

$$p_B^{(k)}(s_{jm}|d_i) = \frac{\alpha_{jm}^{(k)} \prod_{t=1}^{|V|} \left(\theta_{t|jm}^{(k)}\right)^{B_{it}} \left(1-\theta_{t|jm}^{(k)}\right)^{(1-B_{it})}}{\sum_{r=1}^{M} \alpha_{jr}^{(k)} \prod_{t=1}^{|V|} \left(\theta_{t|jr}^{(k)}\right)^{B_{it}} \left(1-\theta_{t|jr}^{(k)}\right)^{(1-B_{it})}}. \tag{47}$$

The term $-|\mathcal{D}_j|$ in equation (45) results from the constraint $\sum_{m=1}^{M} \alpha_{jm} = 1$.

Two basic equations of the EM algorithm for finite mixtures of multivariate Bernoulli distributions become:

- E-step: Computation of the a posteriori probabilities $p_B^{(k)}(s_{jm}|d_i)$ using (47) from the current parameter estimates $\{\alpha_{jm}^{(k)}, \theta_{jm}^{(k)}\}$ at iteration k.
- M-step: The parameter re-estimation according to:

$$\alpha_{jm}^{(k+1)} = \frac{1}{|\mathcal{D}_j|} \sum_{i=1}^{|\mathcal{D}_j|} p_B^{(k)}(s_{jm}|d_i) \ , \tag{48}$$

$$\theta_{t|jm}^{(k+1)} = \frac{\sum_{i=1}^{|\mathcal{D}_j|} B_{it} p_B^{(k)}(s_{jm}|d_i)}{\sum_{i=1}^{|\mathcal{D}_j|} p_B^{(k)}(s_{jm}|d_i)} \tag{49}$$

A.2 Mixture of multinomials

The class-conditional probability function using mixture of multinomials is of the form

$$P(d|c_j;\Theta_j) = \sum_{m=1}^{M} \alpha_{jm} P(|d|) \frac{|d|!}{\prod_{t=1}^{|V|} N_t!} \prod_{t=1}^{|V|} \theta_{t|jm}^{N_t} \tag{50}$$

with parameters $\theta_{jm} = (\theta_{1|jm}, ..., \theta_{|V||jm})$ satisfying constraints

$$\sum_{t=1}^{|V|} \theta_{t|jm} = 1 \ . \tag{51}$$

In this case, the function $Q(\Theta_j, \Theta_j^{(k)})$ takes the form

$$Q(\Theta_j, \Theta_j^{(k)}) = \tag{52}$$
$$= \sum_{m=1}^{M} \sum_{i=1}^{|\mathcal{D}_j|} p_M^{(k)}(s_{jm}|d_i) \log \alpha_{jm} P(|d_i|) \frac{|d_i|!}{\prod_{t=1}^{|V|} N_{it}!} \prod_{t=1}^{|V|} \theta_{t|jm}^{N_{it}}$$

where

$$p_M^{(k)}(s_{jm}|d_i) = \frac{\alpha_{jm}^{(k)} \prod_{t=1}^{|V|} \left(\theta_{t|jm}^{(k)}\right)^{N_{it}}}{\sum_{r=1}^{M} \alpha_{jr}^{(k)} \prod_{t=1}^{|V|} \left(\theta_{t|jr}^{(k)}\right)^{N_{it}}} \ . \tag{53}$$

Starting with parameter set $\{\theta_{jm}, m = 1, ..., M\}$, Q_θ is maximized with respect subject to constraint (51). The maximization problem with respect to parameters $\theta_{1|jm}$,...,$\theta_{|V||jm}$ under constraint (51) is solved by the method of Lagrange multiplier. The appropriate Lagrangian for $\{\theta_{jm}\}$ is

$$Q_\lambda = Q_\theta + \lambda \left(\sum_{t=1}^{|V|} \theta_{t|jm} - 1 \right) \tag{54}$$

where λ is the Lagrange multiplier. Differentiating Q_λ with respect to θ_{jm} and setting it equal to zero yields

$$\sum_{i=1}^{|\mathcal{D}_j|} \frac{p^{(k)}(s_{jm}|d_i)N_{it}}{\theta_{t|jm}} + \lambda = 0. \tag{55}$$

Hence,

$$\theta_{t|jm} = \frac{-\sum_{i=1}^{|\mathcal{D}_j|} p^{(k)}(s_{jm}|d_i)N_{it}}{\lambda} . \tag{56}$$

Substituting into the constraint (51) results in

$$\sum_{t=1}^{|V|} \frac{-\sum_{i=1}^{|\mathcal{D}_j|} p^{(k)}(s_{jm}|d_i)N_{it}}{\lambda} = 1. \tag{57}$$

Therefore,

$$\lambda = -\sum_{t=1}^{|V|}\sum_{i=1}^{\mathcal{D}_j} p^{(k)}(s_{jm}|d_i)N_{it} . \tag{58}$$

Hence, the estimate of the parameters $\theta_{t|jm}$ is, from equation (55)

$$\hat{\theta}_{t|jm} = \frac{\sum_{i=1}^{|\mathcal{D}_j|} p^{(k)}(s_{jm}|d_i)N_{it}}{\sum_{t=1}^{|V|}\sum_{i=1}^{|\mathcal{D}_j|} p^{(k)}(s_{jm}|d_i)N_{it}} . \tag{59}$$

Now we give two basic equations of the EM algorithm for finite mixtures of multinomials are

- E-step: Computation of the a posteriori probabilities $p_M^{(k)}(s_{jm}|d_i)$ using (53) from the current parameter estimates $\{\alpha_{jm}^{(k)}, \theta_{jm}^{(k)}\}$ at iteration k.
- M-step: Updates $\{\alpha_{jm}^{(k+1)}, \theta_{jm}^{(k+1)}, m = 1, ..., M\}$ according to:

$$\alpha_{jm}^{(k+1)} = \frac{1}{|\mathcal{D}_j|}\sum_{i=1}^{|\mathcal{D}_j|} p_M^{(k)}(s_{jm}|d_i) , \tag{60}$$

$$\theta_{t|jm}^{(k+1)} = \frac{\sum_{i=1}^{|\mathcal{D}_j|} N_{it}\ p_M^{(k)}(s_{jm}|d_i)}{\sum_{r=1}^{|V|}\sum_{i=1}^{|\mathcal{D}_j|} N_{ir}\ p_M^{(k)}(s_{jm}|d_i)} . \tag{61}$$

B Bhattacharyya Distance Criterion

Let $\mathbf{X} = (X_1, ..., X_k)$ be a k-dimensional random vector. The *Bhattacharyya distance* between two density functions $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ is defined as follows:

$$B = -\log \rho = -\log \int_{\mathcal{X}} \sqrt{f_1(\mathbf{x}) f_2(\mathbf{x})} d\mathbf{x} \tag{62}$$

where ρ is the *Bhattacharyya coefficient.* In the following subsections, we calculate Bhattacharyya distance explicitly for the multivariate Bernoulli distribution and for the multinomial distribution.

B.1 Multivariate Bernoulli distribution

Consider the class-conditional probability of the document d

$$P(d|c_j; \Theta_j) = \prod_{t=1}^{v} \theta_{t|j}^{B_t} (1 - \theta_{t|j})^{(1-B_t)} \tag{63}$$

where B_t is either 0 or 1. For simplicity we denote by v the vocabulary size $|V|$.

First, we derive the Bhattacharyya coefficient ρ for two multivariate Bernoulli probability functions $P(d|c_j, \theta_j)$ and $P(d|c_k, \theta_k)$:

$$\begin{aligned}
\rho &= \sum_{B_1=0}^{1} \cdots \sum_{B_v=0}^{1} \sqrt{\prod_{t=1}^{v} \theta_{t|j}^{B_t} (1-\theta_{t|j})^{(1-B_t)} \prod_{t=1}^{v} \theta_{t|k}^{B_t} (1-\theta_{t|k})^{(1-B_t)}} \qquad (64)\\
&= \sum_{B_1=0}^{1} \cdots \sum_{B_v=0}^{1} \prod_{t=1}^{v} \left(\sqrt{\theta_{t|j}\theta_{t|k}}\right)^{B_t} \left(\sqrt{(1-\theta_{t|j})(1-\theta_{t|k})}\right)^{(1-B_t)} \\
&= \frac{\prod_{t=1}^{v} \psi_t}{\prod_{t=1}^{v} \psi_t} \sum_{B_1=0}^{1} \cdots \sum_{B_v=0}^{1} \prod_{t=1}^{v} \left(\sqrt{\theta_{t|j}\theta_{t|k}}\right)^{B_t} \left(\sqrt{(1-\theta_{t|j})(1-\theta_{t|k})}\right)^{(1-B_t)} \\
&= \left(\prod_{t=1}^{v} \psi_t\right) \sum_{B_1=0}^{1} \cdots \sum_{B_v=0}^{1} \prod_{t=1}^{v} \frac{\left(\sqrt{\theta_{t|j}\theta_{t|k}}\right)^{B_t} \left(\sqrt{(1-\theta_{t|j})(1-\theta_{t|k})}\right)^{(1-B_t)}}{\psi_t} \\
&= \left(\prod_{t=1}^{v} \psi_t\right) \sum_{B_1=0}^{1} \cdots \sum_{B_v=0}^{1} \prod_{t=1}^{v} \left(\frac{\sqrt{\theta_{t|j}\theta_{t|k}}}{\psi_t}\right)^{B_t} \left(\frac{\sqrt{(1-\theta_{t|j})(1-\theta_{t|k})}}{\psi_t}\right)^{(1-B_t)}
\end{aligned}$$

where

$$\psi_t = \sqrt{\theta_{t|j}\theta_{t|k}} + \sqrt{(1-\theta_{t|j})(1-\theta_{t|k})}\ . \tag{65}$$

Denoting

$$q_t = \frac{\sqrt{\theta_{t|j}\theta_{t|k}}}{\psi_t}\ , \quad 1 - q_t = 1 - \frac{\sqrt{\theta_{t|j}\theta_{t|k}}}{\psi_t} = \frac{\sqrt{(1-\theta_{t|j})(1-\theta_{t|k})}}{\psi_t} \tag{66}$$

$$0 < q_t < 1$$

we can write

$$\prod_{t=1}^{v}\left(\frac{\sqrt{\theta_{t|j}\theta_{t|k}}}{\psi_t}\right)^{B_t}\left(\frac{\sqrt{(1-\theta_{t|j})(1-\theta_{t|k})}}{\psi_t}\right)^{(1-B_t)} =$$

$$= \prod_{t=1}^{v} q_t^{B_t}(1-q_t)^{(1-B_t)} \tag{67}$$

It holds

$$\sum_{B_1=0}^{1} \dots \sum_{B_v=0}^{1} \prod_{t=1}^{v} q_t^{B_t}(1-q_t)^{(1-B_t)} = 1\ . \tag{68}$$

Therefore, the Bhattacharyya coefficient takes

$$\rho = \prod_{t=1}^{v} \psi_t = \prod_{t=1}^{v}\left(\sqrt{\theta_{t|j}\theta_{t|k}} + \sqrt{(1-\theta_{t|j})(1-\theta_{t|k})}\right) \tag{69}$$

Therefore, the Bhattacharyya distance between $P(d|c_j;\Theta_j)$ and $P(d|c_k;\Theta_k)$ becomes

$$B_{jk} \quad = \quad -\log\rho = -\sum_{t=1}^{v}\log\left(\sqrt{\theta_{t|j}\theta_{t|k}} + \sqrt{(1-\theta_{t|j})(1-\theta_{t|k})}\right)\ . \tag{70}$$

B.2 Multinomial distribution

Consider the class-conditional probability of the document d of the form:

$$P(d|c_j;\Theta_j) = \frac{n!}{N_1!...N_v!}\prod_{t=1}^{v}\theta_{t|j}^{N_t}\ . \tag{71}$$

Again, for simplicity $v = |V|$ is the vocabulary size and $n = |d|$ is the length of the document d. We can calculate Bhattacharyya distance explicitly for the multinomial distribution.

For the Bhattacharyya coefficient we obtain:

$$
\begin{aligned}
\rho &= \underbrace{\sum_{N_1=0}^{n} \cdots \sum_{N_v=0}^{n}}_{\sum_{t=1}^{v} N_t = n} \left(\sqrt{\frac{n!}{N_1! \ldots N_v!}}\right)^2 \left(\sqrt{\theta_{1|j}\theta_{1|k}}\right)^{N_1} \ldots \left(\sqrt{\theta_{v|j}\theta_{v|k}}\right)^{N_v} \\
&= \underbrace{\sum_{N_1=0}^{n} \cdots \sum_{N_v=0}^{n}}_{\sum_{t=1}^{v} N_t = n} \frac{n!}{N_1! \ldots N_v!} \prod_{t=1}^{v} \left(\sqrt{\theta_{t|j}\theta_{t|k}}\right)^{N_j} \\
&= \frac{\left(\sum_{r=1}^{v} \sqrt{\theta_{r|j}\theta_{r|k}}\right)^n}{\left(\sum_{r=1}^{v} \sqrt{\theta_{r|j}\theta_{r|k}}\right)^n} \underbrace{\sum_{N_1=0}^{n} \cdots \sum_{N_v=0}^{n}}_{\sum_{t=1}^{v} N_t = n} \frac{n!}{N_1! \ldots N_v!} \prod_{t=1}^{v} \left(\sqrt{\theta_{t|j}\theta_{t|k}}\right)^{N_j} \\
&= \left(\sum_{r=1}^{v} \sqrt{\theta_{r|j}\theta_{r|k}}\right)^n \underbrace{\sum_{N_1=0}^{n} \cdots \sum_{N_v=0}^{n}}_{\sum_{t=1}^{v} N_t = n} \frac{n!}{N_1! \ldots N_v!} \frac{\prod_{t=1}^{v} \left(\sqrt{\theta_{t|j}\theta_{t|k}}\right)^{N_j}}{\left(\sum_{r=1}^{v} \sqrt{\theta_{r|j}\theta_{r|k}}\right)^{\sum_{t=1}^{v} N_t}} \\
&= \left(\sum_{r=1}^{v} \sqrt{\theta_{r|j}\theta_{r|k}}\right)^n \underbrace{\sum_{N_1=0}^{n} \cdots \sum_{N_v=0}^{n}}_{\sum_{t=1}^{v} N_t = n} \frac{n!}{N_1! \ldots N_v!} \prod_{t=1}^{v} \left(\frac{\sqrt{\theta_{t|j}\theta_{t|k}}}{\sum_{r=1}^{v} \sqrt{\theta_{r|j}\theta_{r|k}}}\right)^{N_j} \qquad (72)
\end{aligned}
$$

Denoting

$$
q_t = \frac{\sqrt{\theta_{t|j}\theta_{t|k}}}{\sum_{r=1}^{v} \sqrt{\theta_{r|j}\theta_{r|k}}} , \tag{73}
$$

it is easily seen that the parameters $(q_1, \ldots, q_v)$ satisfy the conditions

$$
0 < q_t < 1, \quad \sum_{t=1}^{v} q_t = \sum_{t=1}^{v} \left(\frac{\sqrt{\theta_{t|j}\theta_{t|k}}}{\sum_{r=1}^{v} \sqrt{\theta_{r|j}\theta_{r|k}}}\right) = 1 . \tag{74}
$$

Since

$$
\frac{n!}{N_1! \ldots N_v!} \prod_{t=1}^{v} \left(\frac{\sqrt{\theta_{t|j}\theta_{t|k}}}{\sum_{r=1}^{v} \sqrt{\theta_{r|j}\theta_{r|k}}}\right)^{N_j} \tag{75}
$$

is the probability function of the multinomial distribution with the parameters $(n, q_1, ..., q_v)$, it holds

$$\underbrace{\sum_{N_1=0}^{n} ... \sum_{N_v=0}^{n}}_{\sum_{t=1}^{v} N_t = n} \frac{n!}{N_1!...N_v!} \prod_{t=1}^{v} \left(\frac{\sqrt{\theta_{t|j}\theta_{t|k}}}{\sum_{r=1}^{v} \sqrt{\theta_{r|j}\theta_{r|k}}} \right)^{N_t} = 1 \ . \tag{76}$$

Therefore, the Bhattacharyya coefficient is now

$$\rho = \left(\sum_{t=1}^{v} \sqrt{\theta_{r|j}\theta_{r|k}} \right)^{n} \tag{77}$$

and we obtain the following expression for the Bhattacharyya distance between $P(d|c_j; \Theta_j)$ and $P(d|c_k; \Theta_k)$:

$$B_{jk} = -\log \rho = -\log \left(\sum_{r=1}^{v} \sqrt{\theta_{r|j}\theta_{r|k}} \right)^{n} = -n \log \sum_{r=1}^{v} \sqrt{\theta_{r|j}\theta_{r|k}} \ . \tag{78}$$

References

[1] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm, *J. Royal Stat. Soc.*, 39, 1–38, 1977.

[2] P.A. Devijver, and J. Kittler. *Pattern Recognition: A Statistical Approach*, Prentice-Hall International, London, 1992.

[3] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers, *Machine Learning*, 29:131-163, 1997.

[4] K. Fuka and R. Hanka. Feature set reduction for document classification problems, *IJCAI-01 Workshop: Text Learning: Beyond Supervision*, Seattle, US, 2001.

[5] R.H. Greecy, B.M. Masand, S.J. Smith, and D.L. Waltz. Trading maps and memory for knowledge engineering: Classifying census returns on the connection machine, *Comm. ACM*, 35:48-63, 1992.

[6] J. Grim. Multivariate statistical pattern recognition with nonreduced dimensionality, *Kybernetika*, 22: 142–157, 1986.

[7] A.K. Jain, R.P.W. Duin, and J. Mao. Statistical pattern recognition: A review, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(2): 4–37, 2000.

[8] T. Joachims. Text categorization with support vector machines: Learning with many relevant features, *Proc. of the ECML'98*,137-142, 1998.

[9] A. Juan and E.Vidal. On the use of Bernoulli mixture models for text clasification, *Proc. of the Workshop on Pattern Recognition in Information Systems (PRIS 01)*, Setúbal, Portugal, 118–126, 2001.

[10] A. Juan and E. Vidal. On the use of Bernoulli mixture models for text clasification, *Pattern Recognition*, 35: 2705–2710, 2002.

[11] M. Junker and R. Hoch. An experimental evaluation of OCR test representations for learning document classifiers, *International Journal on Document Analysis nad Recognition*, 1: 116–122, 1998.

[12] D.D. Lewis and M. Ringuette. Comparison of two learning algorithms for text categorization, *Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval*, 1996, 298-306.

[13] A. McCallum and K. Nigam. A comparison of event models for naive Bayes text classification, *Proceedings of the AAAI-98 workshop on Learning for Text Categorization*, 1998.

[14] G.J. McLachlan and D. Peel. *Finite Mixture Models*, John Wiley, New York, 2000.

[15] D. Mladenic. Feature subset selection using in text learning, *In European Conference on Machine Learning*, 95–100, 1998.

[16] K. Nigam, A.K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM, *Machine Learning*, 39(2/3): 103-134, 2000.

[17] J. Novovičová, P. Pudil, and J. Kittler. Divergence based feature selection for multimodal class densities, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18: 218–223, 1996.

[18] P. Pudil, J. Novovičová, and J. Kittler. Floating search methods in feature selection. *Pattern Recognition Letters*, 15(11): 1119–1125, 1994.

[19] P. Pudil, J. Novovičová, and J. Kittler. Simultaneous learning of decision rules and important attributes for classification problems in image analysis, *Image and Vision Computing*, 12: 193–198, 1994.

[20] P. Pudil, J. Novovičová, and J. Kittler. Feature selection based on approximation of class densities by finite mixtures of special type, *Pattern Recognition*, 28: 1389–1398, 1995.

[21] B. Ripley. *Pattern Recognition and Neural Networks*, Cambridge, Mass., Cambridge University Press, 1996.

[22] K. Torkkola. Discriminative features for document classification, In *Proc. of the 16th International Conference on Pattern Recognition ICPR'2002*, Quebec City, Canada, 2002.

[23] K. Tzeras and S. Hartman. Automatic indexing based on Bayesian inference networks, *Proc 16th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'93)*, pages 22-34, 1993.

[24] A. Webb. *Statistical Pattern Recognition*, Arnold, London, Sydney, Auckland, 1999.

[25] E. Wiener, J.O. Pedersen, and A.S. Weigend. A neural network approach to topic spotting, *Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval (SDAIR'95)*, 1995.

[26] I.H. Witten and T.C. Bell. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory*, 37(4), July 1991.

[27] Y. Yang and J.O. Pedersen. A comparative study on feature selection in text categorization, *Proc. of Int'l Conf. on Machine Learning* (ICML-97), pp. 412-420, 1997.

[28] Y. Yang. An evaluation of statistical approaches to text categorization, *Journal of Information Retrieval*, 1(1/2):67-68, 1999.