A new term-weighting scheme for naïve Bayes text categorization

Marcelo Mendoza

Computer Science Department, Universidad Técnica Federico Santa María, Santiago, Chile

Abstract

Purpose – Automatic text categorization has applications in several domains, for example e-mail spam detection, sexual content filtering, directory maintenance, and focused crawling, among others. Most information retrieval systems contain several components which use text categorization methods. One of the first text categorization methods was designed using a naïve Bayes representation of the text. Currently, a number of variations of naïve Bayes have been discussed. The purpose of this paper is to evaluate naïve Bayes approaches on text categorization introducing new competitive extensions to previous approaches.

Design/methodology/approach – The paper focuses on introducing a new Bayesian text categorization method based on an extension of the naïve Bayes approach. Some modifications to document representations are introduced based on the well-known BM25 text information retrieval method. The performance of the method is compared to several extensions of naïve Bayes using benchmark datasets designed for this purpose. The method is compared also to training-based methods such as support vector machines and logistic regression.

Findings – The proposed text categorizer outperforms state-of-the-art methods without introducing new computational costs. It also achieves performance results very similar to more complex methods based on criterion function optimization as support vector machines or logistic regression.

Practical implications – The proposed method scales well regarding the size of the collection involved. The presented results demonstrate the efficiency and effectiveness of the approach.

Originality/value – The paper introduces a novel naïve Bayes text categorization approach based on the well-known BM25 information retrieval model, which offers a set of good properties for this problem.

Keywords Text retrieval, Information searches, Classification, Text categorization, Bayesian models, Information retrieval

Paper type Research paper

1. Introduction

The explosive growth of the available information contained in digital documents has pushed to the limit, the development of new large-scale search technologies. To facilitate information search, documents are frequently categorized into a set of predefined topics, according to their contents. Typically, the set of topics follows a hierarchical structure of relationships also known as a taxonomy, in which a fraction of the knowledge is represented. These knowledge structures are consolidated by human experts of each area.

Many tasks related to text categorization have been addressed by human experts. For example, the administration of a library catalogue is usually conducted by humans without considering automatic methods for the maintenance of the collection.

The author was supported by the projects FONDEF DO1155 and UTFSM-DGIP 24.11.19.
However, some of these tasks, as text categorization, are very expensive regarding human efforts. Due to this fact, the administration of large-scale documentary collections involves the participation of many experts, limiting its maintainability.

The emergence of the WWW at the beginning of the 1990s pushed the growth of more and more text collections. In this context, the design of strategies to allow automatic categorization of documents into pre-defined topics is a very relevant task. Text categorization methods previously discussed in natural language processing and statistical learning communities have been successfully applied to this task.

Automatic text categorization methods use algorithms capable of identifying patterns or classification rules with good generalization properties. This task is known as training. In a training process a number of examples for each category is provided. Thus, human experts are needed to label a significant number of training examples. This is the reason why this process is known as supervised learning.

Automatic text categorization has applications in several domains, as for example e-mail Spam detection, sexual content filtering, directory maintenance, and focused crawling, among others. Currently, most information retrieval systems contain several components which use text categorization methods.

One of the first text categorization methods was designed using a naïve Bayes representation of the text (Maron and Kuhns, 1960). Naïve Bayes text representation assumes statistical independence in term co-occurrence and use each term as a feature for the description of the content of each document. Later, McCallum and Nigam (1998) studied alternative representations based on Bernoulli and multinomial approaches for naïve Bayes. Currently, a number of variations of naïve Bayes have been discussed achieving good performance results without introducing high computational costs.

A number of techniques based on non-Bayesian approaches have been studied in text categorization. Among these techniques k-nearest neighbors, support vector machines (SVMs) and logistic regression (LR) have achieved good performance results. A main drawback of these methods relies in the introduction of high computational costs in training steps, limiting their application to large-scale web information systems.

We propose a new variation to naïve Bayes which allows us improving the performance without introducing new computational costs. To do this, we work at document representation level, studying the effect of term frequency and inverse document frequency in the performance. We introduce some modifications to document representations based on the well known BM25 text information retrieval method. In particular, we studied in our evaluation the performance of several extensions to naïve Bayes, showing that our proposal improves performance. We show also that our proposal does not introduce new computational costs and achieves very similar performance results as more complex methods based on criterion function optimization as SVMs or LR.

The paper organization is the following: in Section 2 we review state-of-the-art methods. In Section 3 we discuss background and extensions to the original naïve Bayes method. Our naïve Bayes extensions are introduced in Section 4. Section 5 shows experimental results. Finally, we conclude in Section 6.

2. Related work
2.1 Text categorization based on Bayesian approaches
One of the first text categorizers was introduced by Maron and Kuhns (1960). That first text categorizer was based on naïve Bayes, assuming statistical independence
for term co-occurrence. Each document was represented by the terms which compound its content. Using a graphic modeling approach they represented each term by a node in a directed graph. Additional nodes were introduced to represent categories. Each node which represents a term was linked to a category node. Then each arc was weighted using an estimation of the probability associated to the event “the term occurs in the document given that the document belongs to the category”. Then for each category it is possible to estimate how probable is to observe the full document given that the document belongs to the category. Finally, the document is classified into the category which maximizes this probability.

McCallum and Nigam (1998) studied two formulations for naïve Bayes text categorization. One of them is based on a Bernoulli model, estimating the weight of each arc in the naïve Bayes model as 1 if the term occurs in the document, 0 in other case. A second formulation is based on a multinomial model, estimating the weight of each arc by the frequency of the term in the document. The study shows that the multinomial approach outperforms Bernoulli model in accuracy. However, the performance of the multinomial model varies depending on several factors. One of them corresponds to the length of the document (number of term occurrences in its content), factor that was studied by Bennett (2000). Another important factor which explains the diversity of results obtained using naïve Bayes is that some terms concentrate several occurrences in a very few documents. This phenomenon is known as term burstiness (Church and Gale, 1995).

Rennie et al. (2003) proposed a variation for the multinomial naïve Bayes method introducing a smoothing function over term frequency, reducing the effect of term burstiness. Experimental results showed that the smoothed version of naïve Bayes outperforms naïve Bayes. Schneider (2005) proposed to eliminate term frequency from the model applying also a feature selection procedure based on mutual information, allowing representing each document with its top best features. In the same sense, Kolcz and Yih (2007) proposed another variation for the naïve Bayes categorizer proposed by Rennie et al. (2003), achieving improvements on some benchmark collections. Another extension to naïve Bayes was introduced by Kim et al. (2006), whose proposal was based on the study of normalization at document level. Some of these normalization strategies achieve significant performance improvements. Recently, Qiang (2010) introduced a new function for frequency smoothing which also exhibits improvements.

Wilbur and Kim (2009) studied the effect of vocabulary size on naïve Bayes performance. They concluded that in large-scale text collections the use of term frequency affects naïve Bayes performance due to term burstiness. The article also argues that there is insufficient evidence to establish the same conclusion at small scale.

Feature selection for naïve Bayes text classification was explored by Liu et al. (2009), introducing a new text selection criteria based on term weighting. In the same research line, Chen et al. (2009) proposed text selection functions based on odd ratios. Recently, Altincay and Erenel (2010) presented an extensive evaluation of naïve Bayes text categorizers based on term weighting schemes, studying how different feature selection methods perform. These articles allow illustrating the impact of term weighting schemes in performance and also how useful are feature selection techniques to improve naïve Bayes efficiency.

2.2 Text categorization based on non-Bayesian approaches

Hastie et al. (2001) presented a complete description of categorization methods based on \( k \)-nearest neighbors. Using a distance function, for each new object the \( k \)-nearest
neighbors are determined, classifying the object according to the majority label in its neighborhood. Perkins et al. (2003) showed that in several texts benchmark collections k-nearest neighbors outperforms naïve Bayes. However, Indyk (2004) showed that the use of k-nearest neighbors introduces high computational costs in high dimensional spaces. Unfortunately this is the case of text. To address this limitation, Datar and Indyk (2004) studied how to reduce these computational costs keeping in main memory data structures for text based on hashing.

SVMs (Vapnik, 1998) were also studied in the context of text categorization. SVMs search for an optimum hyperplane to separate objects in vector spaces. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training data points of any class. One of the limitations of this approach is that in some vector spaces the objects to be discriminated are not linearly separable. It has been proposed that in these cases the original space be mapped into a much higher dimensional space, using kernel functions, making the separation easier in this new space. Another limitation is related to the tuning process of each machine. This process introduces high computational costs in the training step. Joachims (2006) studied these limitations introducing some modifications to SVMs. Lewis et al. (2004) determined that SVMs outperforms k-nearest neighbors in some benchmark collections such as RCV1. Voorhees and Harman (2005) show similar results on TREC. Ault and Yang (2002) show something similar on TDT using LR. Further details on text categorization were summarized by Sebastiani (2002).

3. Background
3.1 Multinomial naïve Bayes text classification
Document categorization is the task of assigning a Boolean value to each pair \((d, c) \in D \times C\), where \(D\) represents a document collection, \(C\) represents a set of categories, \(d\) is a document that belongs to \(D\), and \(c\) is a category which belongs to \(C\). A Boolean value equals to 1 indicates that \(d\) was categorized into \(c\), 0 in other case. This kind of categorization is known as hard categorization. There are also soft methods which associate a score to each pair \((d, c)\), allowing that eventually a document could belong to more than one category. Following Sebastiani (2002), a document categorization task corresponds to the approximation of an unknown target function \(\phi : D \times C \rightarrow \{0, 1\}\) when we consider hard categorization, and \(\phi : D \times C \rightarrow [0, 1]\) when soft categorization is considered.

Bayesian categorization methods are based on generative models for document representation, using a parametric mixture approach, where each category represents one of the components to mix. The model has the following expression:

\[
p(d) = \sum_{j=1}^{c} p(c_j)p(d|c_j)
\]

(1)

where \(d \in D\) and \(c_j \in C\). Using the Bayes rule we can obtain the probability that represents the event “\(d\) generates \(c\)”, as follows:

\[
p(c|d) = \frac{p(c)p(d|c)}{p(d)}
\]

(2)
To classify a document, a maximum a posteriori probability selection process is conducted, where $p(d)$ is constant. Thus, the categorization is defined only by the product $p(c)p(d|c)$, also known as the discriminator. The $p(c)$ probabilities are a priori probabilities. These a priori probabilities can be estimated by counting the number of documents that belong to $c$. The $p(d|c)$ probabilities can be estimated using the terms that compound the content of $d$.

Several extensions to the basic Bayes method have been discussed. These extensions claim for different assumptions regarding the generative process of construction of each document. However, the vast majority of these extensions adopt a statistical independence assumption for term co-occurrence. These methods are known as naive Bayes methods.

Multinomial naive Bayes methods assume that term occurrences in a document follows a multinomial distribution. Thus, a document corresponds to a term sequence assuming that the position of each term is generated independently to other terms. Let $V = \{t_1, \ldots, t_n\}$ be a vocabulary where the content of the collection $D$ is represented. Each class $c \in C$ has an association with a set of parameters $\hat{\theta}_c = \{\theta_{c,1}, \ldots, \theta_{c,n}\}$, where each of them corresponds to a $p(t_i|c)$ probability, which represents the event “$t_i$ occurs in $c$”. It holds that:

$$\sum_{i=1}^{n} p(t_i|c) = 1, \quad \forall c \in C.$$  

Then, the likelihood of a document $d$ with respect to a class $c$ is given by the following expression:

$$p(d|c) = \frac{L_d!}{\prod_{i=1}^{n} T_{f_{i,d}}!} \prod_{i=1}^{n} p(t_i|c) T_{f_{i,d}}$$  

where $T_{f_{i,d}}$ represents the number of occurrences of $t_i$ in $d$ and $L_d$ represents the length of $d$, which corresponds to $\sum_{i=1}^{n} T_{f_{i,d}}$. Notice that:

$$\frac{L_d!}{\prod_{i=1}^{n} T_{f_{i,d}}!}$$  

is independent of $c$, reason why for the same document it holds that:

$$p(d|c) \approx \prod_{i=1}^{n} p(t_i|c) T_{f_{i,d}}$$  

Finally, the discriminator of the naive Bayes categorizer can be expressed as follows:

$$p(c|d) = p(c) \prod_{i=1}^{n} p(t_i|c) T_{f_{i,d}}$$  

3.2 State-of-the-art extensions to naive Bayes text classification

Due to the fact that many of the probabilities could be very close to zero (Bennett, 2000), naive Bayes implementations use smoothed versions of the posterior
probabilities, as is shown in the following expression:

$$\log(p(c|d)) \approx \log p(c) + \sum_{i=1}^{n} T_{f,i,d} \log(p(t_i|c))$$

(6)

The $p(t_i|c)$ probabilities can be estimated using a maximum likelihood estimator as follows:

$$\hat{p}(t_i|c) = \frac{T_{f,i,c}}{T_f}$$

(7)

where $T_{f,i,c}$ represents the number of occurrences of $t_i$ in $c$ and $T_f$ represents the sum of the number of occurrences of the terms of $V$ in $c$. Several articles (Rennie et al., 2003; Schneider, 2005; Qiang, 2010) also used a smoothed version of the maximum likelihood estimator, as is shown in the following expression:

$$\hat{p}(t_i|c) = \frac{1 + T_{f,i,c}}{n + T_f}$$

(8)

where $n$ represents the vocabulary size (the number of terms belonging to $V$).

Rennie et al. (2003) proposed several variations to the multinomial naive Bayes text categorizer. First, they observed that empirical term frequency distributions differ from multinomial. In fact, empirical term frequency distributions follow a power-law behavior. In addition, sometimes rare terms appear many times in a few documents. This phenomenon is known as term burstiness (Church and Gale, 1995). To mitigate this effect, they proposed the following transformations to the $T_{f,i,d}$ factor:

$$T_{f,i,d} = \log(1 + T_{f,i,d})$$

(9)

$$T_{f,i,d} = T_{f,i,d} \log \left( \frac{N}{n_i} \right)$$

(10)

$$T_{f,i,d} = \frac{T_{f,i,d}}{\sqrt{\sum_{i=1}^{n} T_{f,i,d}^2}}$$

(11)

Equation (9) represents a smoothed version of the $T_{f,i,d}$ factor. Equation (10) represents a TF-Idf weighted schema, usually discussed in information retrieval and introduced by Salton and Buckley (1988). In equation (10), $N$ represents the number of documents in the collection, and $n_i$ the number of documents where the term occurs. Finally, equation (11) represents a normalized version of the $T_{f,i,d}$ factor, using the $L_2$ norm. Another variation was proposed by Schneider (2005), who proposed to eliminate the effect of the most frequent terms using the following expression for the $T_{f,i,d}$ factor:

$$T_{f,i,d} = \text{Min}\{\log(1 + T_{f,i,d}), 1\}$$

(12)

Another variation was proposed by Kolcz and Yih (2007), who changed equation (11) by using the $L_1$ norm as follows:

$$T_{f,i,d} = \frac{T_{f,i,d}}{\sum_{i=1}^{n} T_{f,i,d}}$$

(13)
Finally, a last modification was proposed by Qiang (2010), who introduced the following modification:

\[ Tf_{i,d} = 1 + \log Tf_{i,d} \]  

Regarding text normalization at document level, Kim et al. (2006) explored how different normalization methods perform for naive Bayes text categorization. Their results showed that the use of strategies for document length normalization is useful for this problem, showing benefits when dataset are imbalanced, when documents register length differences and/or a significant number of categories has an insufficient number of training examples. Among the discussed normalization strategies, they propose to estimate term frequencies by including a normalization factor as follows:

\[ Tf_{i,d} = \frac{Tf}{avdl + (1 - \alpha) \cdot dl_i} \]  

where \( \alpha \) is a normalization parameter, \( avdl \) indicates the average number of tokens of the documents of the dataset and \( dl_i \) represents the number of tokens of \( d \). The factor of equation (15) is known as the relative frequency (RF) of \( t_i \) in \( d \).

A summary of the extensions to the naive Bayes text categorization method is presented in Table I. The first column indicates the name of the method, the second shows the weighting scheme formula and the last column shows the bibliographic reference. We will compare these extensions with our proposed extensions in Section 5.

As a technique for text preprocessing the impact of term selection in naive Bayes text categorization has been widely explored. Feature selection criteria such as mutual information, \( \chi^2 \) statistics, and odd ratios have shown a diverse range of performance results (Chen et al., 2009; Liu et al., 2009). We will study the impact of feature selection on performance in Section 5.

### 4. New extensions to the naive Bayes approach

The extensions discussed in Section 3.2 to the naive Bayes approach are inspired in extensions proposed to the well known vector space model used in information retrieval (Salton and Buckley, 1988). However, subsequent studies have made important variations on the vector space model. One of these efforts was conducted by Robertson and Walker (1994), who introduced the BM25 model, which is currently considered state-of-the-art in information retrieval.

<table>
<thead>
<tr>
<th>Method</th>
<th>Formula</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>NBM-U</td>
<td>( Tf_{i,d} / Tf_c )</td>
<td>McCallum and Nigam (1998)</td>
</tr>
<tr>
<td>Smoothing 1</td>
<td>( 1 + Tf_{i,d} / (\alpha + Tf_i) )</td>
<td>Rennie et al. (2003)</td>
</tr>
<tr>
<td>Smoothing 2</td>
<td>( \log(1 + Tf_{i,d}) )</td>
<td>Rennie et al. (2003)</td>
</tr>
<tr>
<td>Tf-Ldf</td>
<td>( Tf_{i,d} \log(N/n_i) )</td>
<td>Salton and Buckley (1988)</td>
</tr>
<tr>
<td>Tf-LZ</td>
<td>( Tf_{i,d} / \sqrt{\sum_{i=1}^{n} Tf_{i,d}^2} )</td>
<td>Salton and Buckley (1988)</td>
</tr>
<tr>
<td>Smoothing 3</td>
<td>( \min { \log(1 + Tf_{i,d}), 1 } )</td>
<td>Schneider (2005)</td>
</tr>
<tr>
<td>Tf-L1</td>
<td>( Tf_{i,d} / \sum_{i=1}^{n} Tf_{i,d} )</td>
<td>Kolcz and Yih (2007)</td>
</tr>
<tr>
<td>Smoothing 4</td>
<td>( 1 + \log Tf_{i,d} )</td>
<td>Qiang (2010)</td>
</tr>
<tr>
<td>RF</td>
<td>( Tf / (avdl + (1 - \alpha) \cdot dl_i) )</td>
<td>Kim et al. (2006)</td>
</tr>
</tbody>
</table>

Table I. Extensions to naive Bayes text categorization
The BM25 model introduced some modifications over Tf and Idf factors. In this article, we considered some of these modifications to propose new extensions to the multinomial naive Bayes approach for text categorization.

A first modification introduced in the BM25 model was done on the Idf factor, which is calculated as follows:

\[
Idf(t_i) = \log \left( \frac{N - n_i}{n_i} \right)
\]  

Due to the fact that \( n_i \in \{1,N\} \), the \( Idf(t_i) \) factor is undefined when \( n_i = N \). Then a modified version of \( Idf(t_i) \) must be used:

\[
Idf(t_i) = \log \left( \frac{N - n_i + 0.5}{n_i + 0.5} \right)
\]  

With this modification, \( Idf(t_i) \) takes values in:

\[ \left[ \log \left( \frac{0.5}{N + 0.5} \right), \log \left( \frac{N - 0.5}{1.5} \right) \right] \].

Notice that a fraction of the \( Idf(t_i) \) range can take negative values. To avoid this situation we propose a modified version of the \( Idf(t_i) \) factor, defined as follows:

\[
Idf(t_i) = \log \left( \frac{2N - n_i + 1}{n_i} \right)
\]  

Notice that with this modification \( Idf(t_i) \) takes positive values in:

\[ \left[ \log \left( \frac{N + 1}{N} \right), \log(2N) \right] \].

The Tf factor is defined in the BM25 model by the following expression:

\[
Tf(t_i) = \frac{Tf_i, d(1 + k)}{Tf_i, d + k(1 - b) + b(L_d/avdl)}
\]  

where \( avdl \) is the average document length in the text collection from which documents are drawn. The parameters \( k \) and \( b \) are free parameters, usually chosen as \( k = 2 \) and \( b = 0.75 \).

We introduce some modifications to the \( Tf(t_i) \) factor to favor its use on Bayesian text categorization. First, we disregard the \( k \) and \( b \) parameters since they do not have a clear justification in the Bayesian approach. To do this we diminish the effect of the choice of these parameters in the weighting scheme, using \( k = 1 \) and \( b = 1 \). This design choice simplifies the denominator formulation of equation (18) to \( Tf_i, d + L_d/avdl \). We also eliminate the constant of the numerator.

We study also the impact of the use of \( avdl \). As \( avdl \) requires to be recalculated each time a new training document is added to the model, we evaluate in Section 5 a version of Tf which does not consider \( avdl \), reducing the denominator of equation (18) to \( Tf_i, d + L_d \). Notice also that as in general it holds that \( L_d \gg Tf_i, d \), so we can reformulate the \( Tf(t_i) \) denominator by eliminating the Tf factor.
Regarding the term \textit{burstiness} phenomenon, we studied also a version of $Tf(t_i)$, introducing a smoothing function as follows:

$$Tf(t_i) \approx \frac{\log(Tf_{i,d})}{L_d}$$

(20)

Notice that $Tf(t_i)$ is undefined for $Tf_{i,d} = 0$. To avoid this situation we introduce the following variation:

$$Tf(t_i) \approx \frac{\log(1 + Tf_{i,d})}{L_d}$$

(21)

Previous modification allows $Tf(t_i)$ taking only positive values.

Using equations (18) and (21), we calculate the Tf-Idf product that allows obtaining a new expression for $Tf_{i,d}$.

Finally, we introduce a modification to the multinomial naïve Bayes categorizer. To avoid the \textit{burstiness} effect introduced by equation (7) or (8) for the estimation of $\hat{p}(t_i|c)$, we use a smoothed version $\log \hat{p}(t_i|c)$ by using the \textit{Idf}$(t_i)$ factor calculated over each class of $C$. Let $cf_i$ be the number of classes of $C$ in which $t_i$ occurs. Using our modified version of \textit{Idf}$(t_i)$ given by equation (17), we propose the following estimation for $\log \hat{p}(t_i|c)$:

$$\log \hat{p}(t_i|c) \approx \log \left( \frac{2|C| - cf_i + 1}{cf_i} \right)$$

(22)

where $|C|$ represents the number of classes in $C$. Due to the fact that $cf_i$ takes values in $\{1, |C|\}$, the \textit{Idf}$(t_i)$ factor takes values in:

$$\left[ \log \left( \frac{|C| + 1}{|C|} \right), \log(2|C|) \right].$$

Normalizing by $\log(2|C|)$, $\hat{p}(t_i|c)$ takes values in $[0,1]$.

Finally, we obtain the following expression for the naïve Bayes discriminator:

$$\log(\hat{p}(c)) = \log \hat{p}(c)$$

$$+ \sum_{i=1}^{n} \frac{\log(1 + Tf_{i,d})}{\log(2|C|)L_d} \log \left( \frac{2N - n_i + 1}{n_i} \right) \log \left( \frac{2|C| - cf_i + 1}{cf_i} \right)$$

(23)

Schneider (2005) showed another phenomenon which affects the performance of naïve Bayes categorizers which is the bias introduced by the \textit{a priori} probabilities of the classes which exhibits numerous training examples. To illustrate this phenomenon, he compared the performance achieved by a naïve Bayes version which does not consider a priori odds, showing that by disregarding these probabilities it is possible to obtain better performance results.

In Table II we summarize the new extensions proposed by us to the naïve Bayes text categorization method. The first column indicates the design criteria and the second column shows the weighting scheme formula.
5. Experimental results
In a first evaluation we consider a document collection with imbalance in the number of instances per class. To do this we crawled a set of documents from the Merlot[1] platform. Merlot is a platform to retrieve learning objects, providing useful resources for the preparation of teaching materials.

Each learning object has an abstract describing the content of the object. We consider abstracts and keywords of 300 learning objects in the same number of examples from the three main categories of “science and technology”: 100 objects from “computer science”, 100 objects from “chemistry” and 100 objects from “biology”. This collection considers 11,191 term occurrences over a vocabulary compounded by 4,484 different terms.

Each main category is composed by several sub categories. In particular, “biology” exhibits three sub categories with an almost even distribution of examples. The other two categories exhibit imbalance in the number of examples. Figure 1 shows how training examples are distributed over the sub categories.

We consider threefold cross validation as an evaluation strategy. This procedure is performed on the three possible permutations of the folds. The use of cross validation allows us to obtain more robust results.

We study how our method performs compared to classification methods based on learning. We consider for the evaluation a comparison with text classifiers based on SVM and LR. Table III shows results regarding accuracy, false positive rate and F-measure. The values obtained are averages of the assessments obtained in each fold.

<table>
<thead>
<tr>
<th>Method</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extended Idf</td>
<td>$\log((2N - n_i + 1)/n_i)$ (18)</td>
</tr>
<tr>
<td>Extended Tf 1</td>
<td>$\log(1 + T_{f_i,d}/L_d)$ (21)</td>
</tr>
<tr>
<td>Extended Tf 2</td>
<td>$\log(1 + T_{f_i,d}/L_d)/L_d$</td>
</tr>
<tr>
<td>Extended Tf-Idf 1</td>
<td>$(\log(1 + T_{f_i,d}/L_d)) (\log((2N - n_i + 1)/n_i)$</td>
</tr>
<tr>
<td>Extended Tf-Idf 2 (NBME-U)</td>
<td>$(\log(1 + T_{f_i,d}/L_d)) (\log((2N - n_i + 1)/n_i))$</td>
</tr>
<tr>
<td></td>
<td>$(\log(2</td>
</tr>
</tbody>
</table>

Table II. Proposed extensions to the naive Bayes text categorization method

Figure 1. Class distribution in Merlot
We compare these techniques with the basic naïve Bayes method (NBM-U) and with our proposed method (NBME-U). Both Bayesian methods consider uniform distribution for a priori probabilities.

Table III shows that the proposed Bayesian categorizer (NBME-U) obtains results that compare favorably to the obtained by LR and SVM. Notice that in average, NBME-U achieves the best results for accuracy and F-measure. Results obtained by the three methods are comparable, outperforming NBM-U by 8 to 10 points in accuracy and F-measure and by 5 to 7 points in false positive rate.

Figure 2 shows results obtained for F-measure by sub category. As in the above evaluation, we use threefold cross validation. Notice that this evaluation represents a difficult barrier because in some sub categories the collection exhibits a low number of examples. As Figure 2 shows, NBME-U obtains a performance comparable to SVM, outperforming LR by some points and achieving a significant difference with NBM-U by 4 and 6 points in micro and macro average measures, respectively. We can observe also that the most significant difference is obtained in those categories which exhibit fewer examples. Notice, for example, that in the case of “software engineering” and “databases”, which correspond to the subcategories with fewer examples in the “computer science” collection, NBME-U outperforms NBM-U by 8 and 21 points, respectively.

To illustrate the performance of the categorizers by the number of examples in each sub category, we measure the difference in F-measure between each method and NBM-U. Categories have been deployed in decreasing order according to the number of examples that each one registers in Merlot. Results for these experiments are shown in Figure 3.

Figure 3 shows in gray the difference between NBME-U and NBM-U, in dark gray the difference between LR and NBM-U and in black the difference between SVM and NBM-U. Notice that to the extent that the number of examples decreases the difference becomes more significant. This fact indicates to us that NBME-U and SVM maintain the performance when the number of examples decreases. This is not true for the NBM-U.

Now we evaluate the performance of our naïve Bayes extensions (Table II) with the state-of-the-art extensions of naïve Bayes of Table I. To conduct this evaluation we
consider two benchmark collections: 20 Newsgroups and Web-KB. These benchmark collections have been used previously in the evaluations performed by the methods of Table I. In particular, 20 Newsgroups is a benchmark collection of approximately 20,000 news distributed over 20 different topics.

Web-KB is a collection compounded by 8,282 Web pages crawled from different universities. These pages have been categorized into seven different categories. Regarding the 20 Newsgroups dataset, we have used only the content of each indexed page,
discarding headers, following the pre-processing procedure defined by Schneider (2005). The same process was considered for the processing of Web-KB, using only contents and discarding headers and HTML tags. These processes are different from the ones used by Kim et al. (2006), where the authors conducted random samples over the collections. In our evaluation we consider full datasets.

As in previous experiments, we follow a threefold cross validation strategy. We considered also LR and SVM. As several of these methods are extensions of the well known BM25 method, we also consider BM25 in the evaluation. We considered uniform distribution for a priori probabilities in naı̈ve Bayes methods. In particular, the RF method was evaluated using a normalization factor equals to 0.2 in 20 Newsgroups and 0.4 in Web-KB, values which achieve the best F-measures performance results in these datasets. Something similar was done for SVMs and LR, where a tuning process was conducted using in this evaluation the set of parameter values which achieve the best performance results. Table IV shows averaged results over each fold for accuracy, false positive rate and F-measure (micro average). As the number of documents is balanced across the classes in these collections, micro and macro averages are very similar. Thus, we discard from this section macro average results. According to Tables I and II, the first column of Table IV indicates the name of the evaluated method.

Table IV shows several interesting results. NBM-U is outperformed by its extensions in both datasets. The use of BM25 as weighting scheme allows achieving relevant improvements over NBM-U. In particular, BM25 outperforms Tf-Idf in both datasets. Regarding state-of-the-art extensions, the use of smoothing over term frequency performs well, being best results achieved by Smoothing 3 (Schneider, 2005). Tf extensions perform well, being best results achieved by Tf-L1 in both datasets (Kolcz and Yih, 2007). The inclusion of the Idf factor has a relevant performance impact, being Tf-Idf performance results very strong. Intuitively, the absence of the Idf factor reduces the discriminative capacity of each categorizer. The use of frequency normalization at document level in the RF scheme (Kim et al., 2006) has a positive impact in false positive

<table>
<thead>
<tr>
<th></th>
<th>20 Newsgroups</th>
<th></th>
<th>Web-KB</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accuracy</td>
<td>FP-rate</td>
<td>F</td>
<td>Accuracy</td>
</tr>
<tr>
<td>NBM-U</td>
<td>0.812</td>
<td>0.175</td>
<td>0.672</td>
<td>0.833</td>
</tr>
<tr>
<td>Smoothing 1</td>
<td>0.823</td>
<td>0.164</td>
<td>0.680</td>
<td>0.840</td>
</tr>
<tr>
<td>Smoothing 2</td>
<td>0.827</td>
<td>0.158</td>
<td>0.692</td>
<td>0.842</td>
</tr>
<tr>
<td>Tf-Idf</td>
<td>0.840</td>
<td>0.135</td>
<td>0.694</td>
<td>0.852</td>
</tr>
<tr>
<td>Tf-L2</td>
<td>0.820</td>
<td>0.152</td>
<td>0.686</td>
<td>0.836</td>
</tr>
<tr>
<td>Smoothing 3</td>
<td>0.838</td>
<td>0.174</td>
<td>0.692</td>
<td>0.846</td>
</tr>
<tr>
<td>Tf-L1</td>
<td>0.824</td>
<td>0.137</td>
<td>0.689</td>
<td>0.842</td>
</tr>
<tr>
<td>Smoothing 4</td>
<td>0.820</td>
<td>0.134</td>
<td>0.692</td>
<td>0.846</td>
</tr>
<tr>
<td>RF</td>
<td>0.833</td>
<td>0.129</td>
<td>0.691</td>
<td>0.841</td>
</tr>
<tr>
<td>LR</td>
<td>0.858</td>
<td>0.112</td>
<td>0.710</td>
<td>0.882</td>
</tr>
<tr>
<td>SVM</td>
<td>0.874</td>
<td>0.098</td>
<td>0.722</td>
<td>0.902</td>
</tr>
<tr>
<td>BM25</td>
<td>0.844</td>
<td>0.128</td>
<td>0.702</td>
<td>0.870</td>
</tr>
<tr>
<td>Extended Idf</td>
<td>0.832</td>
<td>0.117</td>
<td>0.685</td>
<td>0.849</td>
</tr>
<tr>
<td>Extended Tf 1</td>
<td>0.826</td>
<td>0.121</td>
<td>0.688</td>
<td>0.848</td>
</tr>
<tr>
<td>Extended Tf 2</td>
<td>0.830</td>
<td>0.128</td>
<td>0.690</td>
<td>0.851</td>
</tr>
<tr>
<td>Extended Tf-Idf 1</td>
<td>0.852</td>
<td>0.112</td>
<td>0.701</td>
<td>0.881</td>
</tr>
<tr>
<td>NBME-U</td>
<td>0.868</td>
<td>0.095</td>
<td>0.728</td>
<td>0.894</td>
</tr>
</tbody>
</table>

Table IV. Performance evaluation in 20 Newsgroups and Web-KB
rate reduction. Notice also that techniques based on supervised learning algorithms such as LR and SVMs achieve very strong results. However, the use of our extensions allows achieving competitive results with these techniques. Notice that the use of the extended Idf scheme outperforms several state-of-the-art extensions. Something similar occurs when we use our extensions to the Tf factor. In particular, the inclusion of the average document length (extended Tf 2) achieves only a small improvement over its alternate version (extended Tf 1), fact which indicates to us that the elimination of this variable does not impact the performance. The combination of both factors (extended Idf and extended Tf 1) is evaluated in the scheme extended Tf-Idf 1, model which outperforms BM25 and every extension previously proposed in the state-of-the-art in both datasets. Finally, the inclusion of the Idf factor proposed in equation (22) is evaluated in the scheme NBME-U, model which outperforms extension of naïve Bayes. Notice that NBME-U outperforms LR in all comparisons achieving similar results to the ones obtained by SVM. This is a very important result if we consider that the proposed scheme does not include significant computational costs as techniques based on supervised learning.

Now we illustrate how the size of the vocabulary affects our results. To do this, we use the mutual information coefficient as a criterion function for feature selection. The mutual information coefficient was studied in text categorization by Lewis and Ringuette (1994) allowing quantifying the descriptive capacity of each term. Using feature selection we evaluated the performance of each categorizer considering the top-n best features selected. Accuracy performance values are shown in Figure 4.

As Figure 4 shows, both benchmark collections show that the accuracy performance improves when the size of the vocabulary increases, result that coincides with the ones reported by Schneider (2005) and Kolcz and Yih (2007). In 20 Newsgroups, SVM outperforms NBME-U by a few points (2-3) when the size of the vocabulary is small. However, SVM and NBME-U achieve very similar results when the size of the vocabulary is equals or greater than 1,000. In Web-KB, NBME-U outperforms by few points SVM, however their performance is very similar for a vocabulary size equals to 20,000.

Finally, in Figure 5 we compare training times for each categorizer used in our evaluation, according to the vocabulary size. For the methods based on learning we do not consider the time involved in the tuning step, which consists of an expensive process of brute force search. Each categorizer was trained using best parameter values according to the tuning phase results. Training times are shown in Figure 5, in seconds, measured using an Intel® Centrino® – Intel® Pentium® M 730 (1.6 GHz, 2 MB L2 cache, 533 MHz FSB) computer.

Figure 5 shows that training times for the methods based on learning are higher than the ones involved in the construction of the naïve Bayes models. This difference increases when the size of the collection increases, registering for the complete collection a difference in seconds. However, as Table IV shows, the performance obtained by NBME-U is very similar to the one obtained by SVM and LR. This fact indicate to us that naïve Bayes methods are scalable, meaning that low computational times involved in model construction steps do not affect the performance of the categorizers.

6. Conclusion

We have introduced a new Bayesian text categorization method based on an extension of the naïve approach. Our categorizer introduces new factors inspired by the well known BM25 information retrieval method, allowing better text representation properties,
diminishing the burstiness effect and increasing the discriminative capacity of each term as a descriptive feature. Among these factors we consider inverse document frequency, smoothed versions of term frequency, and document length normalization. To avoid the effect introduced by unbalanced training examples, we consider uniformity for the a priori probability distribution. We study also the impact of a feature selection method based on mutual information.

Notes: (a) 20 Newsgroups; (b) Web-KB

Figure 4.
Accuracy performance by size of vocabulary
Our method does not introduce new computational costs. We evaluate our method performing a comparison with several state-of-the-art variations of the naïve Bayes approach, considering also more complex text categorization methods as SVMs and LR. Our method exhibits the best results regarding naïve Bayes variations and achieves very similar results to the ones obtained by SVMs and LR.

**Notes:** (a) 20 Newsgroups; (b) Web-KB
Our method scales well with the size of the collection, meaning that low computational times involved in model construction do not affect the performance of the proposed method.

Note

References


**About the author**

Marcelo Mendoza graduated as an Electric Engineer and obtained his MSc in Computer Science from Universidad Técnica Federico Santa María (UTFSM), in Chile. His PhD in Computer Science was obtained at Universidad de Chile. His current research interests are web mining and machine learning methods applied to information retrieval. Currently, he holds an academic position at the Computer Science Department, UTFSM, and is a research collaborator of Yahoo! Research. Marcelo Mendoza can be contacted at: marcelo.mendoza@usm.cl

To purchase reprints of this article please e-mail: reprints@emeraldinsight.com
Or visit our web site for further details: www.emeraldinsight.com/reprints