J Internet Serv Appl (2011) 1: 183–200
DOI 10.1007/s13174-010-0014-7

ORIGINAL PAPER

Spam filtering: how the dimensionality reduction affects the accuracy of Naive Bayes classifiers

Tiago A. Almeida · Jurandy Almeida · Akebo Yamakami

Received: 1 March 2010 / Accepted: 17 November 2010 / Published online: 2 December 2010

Abstract E-mail spam has become an increasingly important problem with a big economic impact in society. Fortunately, there are different approaches allowing to automatically detect and remove most of those messages, and the best-known techniques are based on Bayesian decision theory. However, such probabilistic approaches often suffer from a well-known difficulty: the high dimensionality of the feature space. Many term-selection methods have been proposed for avoiding the curse of dimensionality. Nevertheless, it is still unclear how the performance of Naive Bayes spam filters depends on the scheme applied for reducing the dimensionality of the feature space. In this paper, we study the performance of many term-selection techniques with several different models of Naive Bayes spam filters. Our experiments were diligently designed to ensure statistically sound results. Moreover, we perform an analysis concerning the measurements usually employed to evaluate the quality of spam filters. Finally, we also investigate the benefits of using the Matthews correlation coefficient as a measure of performance.

Keywords Dimensionality reduction · Spam filter · Text categorization · Classification · Machine learning

T.A. Almeida (✉) · A. Yamakami
School of Electrical and Computer Engineering,
University of Campinas, UNICAMP, 13083–970, Campinas, SP, Brazil
e-mail: tiago@dt.fee.unicamp.br

A. Yamakami
e-mail: akebo@dt.fee.unicamp.br

J. Almeida
Institute of Computing, University of Campinas, UNICAMP, 13083–852, Campinas, SP, Brazil
e-mail: jurandy.almeida@ic.unicamp.br

1 Introduction

Electronic mail, commonly called e-mail, is a way of exchanging digital messages across the Internet or other computer networks. It is one of the most popular, fastest and cheapest means of communication which has become a part of everyday life for millions of people, changing the way we work and collaborate. The downside of such a success is the constantly growing volume of e-mail spam we receive.

The term *spam* is generally used to denote an unsolicited commercial e-mail. Spam messages are annoying to most users because they clutter their mailboxes. It can be quantified in economical terms since many hours are wasted everyday by workers. It is not just the time they waste reading the spam but also the time they spend removing those messages.

According to annual reports, the amount of spam is frightfully increasing. In absolute numbers, the average of spams sent per day increased from 2.4 billion in 2002[1] to 300 billion in 2010.[2] The same report indicates that more than 90% of incoming e-mail traffic is spam. According to the 2004 US Technology Readiness Survey,[3] the cost of spam in terms of lost productivity in the United States has reached US$ 21.58 billion per year, while the worldwide productivity cost of spam is estimated to be US$ 50 billion. On a worldwide basis, the total cost in dealing with spam was estimated to rise from US$ 20.5 billion in 2003, to US$ 198 billion in 2009.

Many methods have been proposed to automatic classify messages as spams or legitimates. Among all proposed techniques, machine learning algorithms have achieved more

[1] See http://www.spamlaws.com/spam-stats.html.

[2] See http://www.cisco.com/en/US/prod/collateral/vpndevc/cisco_2009_asr.pdf.

[3] See http://www.rockresearch.com/news_020305.php.

success [14]. Those methods include approaches that are considered top-performers in text categorization, like support vector machines and the well-known Naive Bayes classifiers.

A major difficulty in dealing with text categorization using approaches based on Bayesian probability is the high dimensionality of the feature space [7]. The native feature space consists of unique terms from e-mail messages, which can be tens or hundreds of thousands even for a moderate-sized e-mail collection. This is prohibitively high for most of learning algorithms (an exception is SVM [21]). Thus, it is highly desirable to develop automatic techniques for reducing the native space without sacrificing categorization accuracy [20].

In this paper, we present a comparative study of the most popular term-selection techniques with respect to different variants of the Naive Bayes algorithm for the context of spam filtering. Despite the existence of other successful text categorization methods, this paper aims to examine how the term-selection techniques affect the categorization accuracy of different filters based on the Bayesian decision theory. The choice of the Naive Bayes classifiers is due to the fact that they are the most employed filters for classifying spams nowadays [4, 25, 38, 46]. They are used in several free web-mail servers and open-source systems [25, 35, 45]. In spite of that, it is still unclear how their performance depends on the dimensionality reduction techniques. Here, we carry out a comprehensive performance evaluation with the specific and practical purpose of filtering e-mail spams using Naive Bayes classifiers in order to improve the filters accuracies. Furthermore, we investigate the performance measurements applied for comparing the quality of the anti-spam filters. In this sense, we also analyze the advantages of using the Matthews correlation coefficient to assess the quality of spam classifiers.

A preliminary version of this work was presented at ICMLA 2009 [2]. Here, we significantly extend the performance evaluation. First, we almost double the number of Naive Bayes filters and term-selection techniques. Second, and the most important, instead of using a fixed number of terms, we vary the number of selected terms from 10 to 100%. Additionally, we analyze the performance measurements applied for comparing the quality of the spam classifiers. Finally, we perform a carefully statistical analysis of the results.

The remainder of this paper is organized as follows. Section 2 presents the related work. Section 3 describes the most popular term-selection techniques in the literature. The Naive Bayes spam filters are introduced in Sect. 4. In Sect. 5, we discuss the benefits of using the Matthews correlation coefficient as a measure of quality for spam classifiers. Section 6 presents the methodology employed in our experiments. Experimental results are shown in Sect. 7. Finally, Sect. 8 offers conclusions and outlines for future work.

2 Related work

The problem of classifying e-mails as spams or legitimates has attracted the attention of many researchers. Different approaches have been proposed for filtering spams, such as rule-based methods, white and black lists, collaborative spam filtering, challenge-response systems, and many others [14].

Among all available techniques, machine learning approaches have been standing out [14]. Such methods are considered the top-performers in text categorization, like rule-induction algorithm [12, 13], Rocchio [27, 41], Boosting [11], compression algorithms [3], Support Vector Machines [1, 18, 21, 26, 31], memory-based learning [6], and Bayesian classifiers [5, 22, 38, 40, 46].

In this work, we are interested in anti-spam filters based on the Bayesian decision theory. Further details about other techniques used for anti-spam filtering and applications that employ Bayesian classifiers are available in Bratko et al. [9], Seewald [45], Koprinska et al. [32], Cormack [14], Song et al. [46], Marsono et al. [35] and Guzella and Caminhas [25].

The Bayesian classifiers are the most employed filters for classifying spams nowadays. They currently appear to be very popular in proprietary and open-source spam filters, including several free web-mail servers and open-source systems [25, 35, 45]. This is probably due to their simplicity, computational complexity and accuracy rate, which are comparable to more elaborate learning algorithms [35, 38, 46].

A well-known difficulty in dealing with text categorization using Bayesian techniques is the high dimensionality of the feature space [7]. To overcome the curse of dimensionality, many works perform a step of dimensionality reduction before applying the anti-spam filter to classify new messages.

Sahami et al. [40] proposed the first academic Naive Bayes spam filter. The authors employed the information gain to select the 500 “best” terms for applying to the classifier.

Androutsopoulos et al. [6] compared the performance of the Sahami’s scheme and memory-based approaches when the number of terms varies from 50 to 700 attributes. In the experiments, they used Ling-Spam corpus, 10-fold cross-validation and *TCR* as the performance measure. The authors claimed that the accuracy rate of the Sahami’s filter is better when the number of terms is close to 100. In Androutsopoulos et al. [5], they showed that word-processing techniques (e.g., lemmatization and stop-lists) are not recommended in spam filtering tasks.

Schneider [42] evaluated different Bayesian filters, such as multivariate Bernoulli, multinomial Boolean and multinomial term frequency. Metsis et al. [38] extended the Schneider’s analysis [42] to include flexible Bayes and multivariate Gauss Naive Bayes spam filters.

Androutsopoulos et al. [7] compared flexible Bayes, linear SVM and LogitBoost. Their results indicate that flexible Bayes and SVM have a similar performance.

It is important to emphasize that all the previous works have employed the information gain to reduce the dimensionality of the term space. Although several works in spam literature and commercial filters use term-selection techniques with Bayesian classifiers, there is no comparative study for verifying how the dimensionality reduction affects the accuracy of different Naive Bayes spam filters. In this work, we aim to fill this important gap.

3 Dimensionality reduction

In text categorization the high dimensionality of the term space ($\mathcal{S}$) may be problematic. In fact, typical learning algorithms used for classifiers cannot scale to high values of $|\mathcal{S}|$ [44]. As a consequence, a step of dimensionality reduction is often applied before the classifier, whose effect is to reduce the size of the vector space from $|\mathcal{S}|$ to $|\mathcal{S}'| \ll |\mathcal{S}|$; the set $\mathcal{S}'$ is called the reduced term set.

Dimensionality reduction is beneficial since it tends to reduce overfitting [48]. Classifiers that overfit the training data are good at reclassifying the data they have been trained on, but much worse at classifying previously unseen data. Moreover, many algorithms perform very poorly when they work with a large amount of attributes. Thus, a process to reduce the number of elements used to represent documents is needed.

Techniques for term-selection attempt to select, from the original set $\mathcal{S}$, the subset $\mathcal{S}'$ of terms (with $|\mathcal{S}'| \ll |\mathcal{S}|$) that, when used for document indexing, yields the highest effectiveness [19, 20, 48]. For selecting the best terms, we have to use a function that selects and ranks terms according to how "important" they are (those that offer relevant information in order to assist the probability estimation, and consequently improving the classifier accuracy). A computationally easy alternative is the filtering approach [29], that is, keeping the $|\mathcal{S}'| \ll |\mathcal{S}|$ terms that receive the highest score according to a function that measures the "importance" of the term for the text categorization task.

3.1 Representation

Considering that each message m is composed of a set of terms (or tokens) $m = \{t_1, \ldots, t_n\}$, where each term t_k corresponds to a word ("adult," for example), a set of words ("to be removed"), or a single character ("$"), we can represent each message by a vector $\mathbf{x} = \langle x_1, \ldots, x_n \rangle$, where x_k corresponds to the value of the attribute X_k associated with the term t_k. In the simplest case, each term represents a single word and all attributes are Boolean: $X_k = 1$ if the message contains t_k or $X_k = 0$, otherwise.

Alternatively, attributes may be integer values computed by term frequency (TF) indicating how many times each term occurs in the message. This kind of representation offers more information than the Boolean one [38]. A third alternative is to associate each attribute X_k to a normalized TF, $x_k = \frac{t_k(m)}{|m|}$, where $t_k(m)$ is the number of occurrences of the term represented by X_k in m, and $|m|$ is the length of m measured in term occurrences. Normalized TF takes into account the term repetition versus the size of the message. It is similar to the TF-IDF (term frequency-inverse document frequency) scores commonly used in information retrieval; an IDF component could also be added to denote terms that are common across the messages of the training collection.

3.2 Term-selection techniques

In the following, we briefly review the eight most popular *Term Space Reduction* (TSR) techniques. Probabilities are estimated by counting occurrences in the training set $\mathcal{M}$ and they are interpreted on an event space of documents, for example: $P(\bar{t}_k, c_i)$ denotes the probability that, for a random message m, term t_k does not occur in m and m belongs to category c_i.

Since there are only two categories $\mathcal{C} = \{\text{spam}(c_s), \text{legitimate}(c_l)\}$ in spam filtering, some functions are specified "locally" to a specific category. To assess the value of a term t_k in a "global" category-independent sense, either the sum $f_{\text{sum}}(t_k) = \sum_{i=1}^{|\mathcal{C}|} f(t_k, c_i)$, the weighted sum $f_{\text{wsum}}(t_k) = \sum_{i=1}^{|\mathcal{C}|} P(c_i) \cdot f(t_k, c_i)$ or the maximum $f_{\max}(t_k) = \max_{i=1}^{|\mathcal{C}|} f(t_k, c_i)$ of their category-specific values $f(t_k, c_i)$[4] are usually computed [44].

3.2.1 Document frequency

A simple and effective global TSR function is the document frequency (*DF*). It is given by the frequency of messages with a term t_k in the training database $\mathcal{M}$, that is, only the terms that occur in the highest number of documents are retained. The basic assumption is that rare terms are either non-informative for category prediction, or not influential in global performance. In either case, removal of rare terms reduces the dimensionality of the feature space. Improvement in categorization accuracy is also possible if rare terms happen to be noise terms. We calculate the *DF* of term t_k using

$$DF(t_k) = \frac{t_k(\mathcal{M})}{|\mathcal{M}|},$$

where $t_k(\mathcal{M})$ represents the number of messages containing the term t_k in the training database $\mathcal{M}$ and $|\mathcal{M}|$ is the amount of available messages [48].

[4] $f(t_k, c_i)$ corresponds to the score received by the term t_k of class c_i for a category-specific term-selection technique f.

3.2.2 DIA association factor

The DIA association factor of a term t_k for a class c_i measures the probability of finding messages of class c_i given the term t_k. The probabilities are calculated by term frequencies in the training database $\mathcal{M}$ [23] as

$$DIA(t_k, c_i) = P(c_i|t_k).$$

We can combine category-specific scores using function f_{sum} or $f_{\max}$ to measure the goodness of a term in a global feature selection.

3.2.3 Information gain

Information gain (*IG*) is frequently employed as a term-goodness criterion in the field of machine learning [39]. It measures the number of bits of information obtained for category prediction by knowing the presence or absence of a term in a message [48]. The *IG* of a term t_k is computed by

$$IG(t_k) = \sum_{c\in[c_i,\bar{c}_i]} \sum_{t\in[t_k,\bar{t}_k]} P(t,c) \cdot \log \frac{P(t,c)}{P(t)\cdot P(c)}.$$

3.2.4 Mutual information

Mutual information (*MI*) (also called pointwise mutual information) is a criterion commonly used in statistical language modeling of words' associations and related applications [48]. The mutual information criterion between t_k and c_i is defined as

$$MI(t_k, c_i) = \log \frac{P(t_k, c_i)}{P(t_k)\cdot P(c_i)}.$$

$MI(t_k, c_i)$ has a natural value of zero if t_k and c_i are independent. To measure the goodness of a term in a global feature selection, we can combine category-specific scores of a term into the three alternate ways: f_{sum}, f_{wsum} or $f_{\max}$, as previously presented.

IG is sometimes called mutual information, which causes confusion. It is probably because *IG* is the weighted average of the $MI(t_k, c_i)$ and $MI(\bar{t}_k, c_i)$, where the weights are the joint probabilities $P(t_k, c_i)$ and $P(\bar{t}_k, c_i)$, respectively. Therefore, information gain is also called average mutual information. However, there are two fundamental differences between *IG* and *MI*: first, *IG* makes a use of information about term absence, while *MI* ignores such information and *IG* normalizes the *MI* scores using the joint probabilities while *MI* uses the non-normalized score [48].

3.2.5 χ^2 statistic

χ^2 statistic measures the lack of independence between the term t_k and the class c_i. It can be compared to the χ^2 distribution with one degree of freedom to judge extremeness. χ^2 statistic has a natural value of zero if t_k and c_i are independent. We can calculate the χ^2 statistic for the term t_k in the class c_i by

$$\chi^2(t_k) = \frac{|\mathcal{M}| \cdot [P(t_k, c_i)\cdot P(\bar{t}_k, \bar{c}_i) - P(t_k, \bar{c}_i)\cdot P(\bar{t}_k, c_i)]^2}{P(t_k)\cdot P(\bar{t}_k)\cdot P(c_i)\cdot P(\bar{c}_i)}.$$

The computation of χ^2 scores has a quadratic complexity, similarly to *MI* and *IG* [48]. The major difference between χ^2 and *MI* is that χ^2 is a normalized value, and hence χ^2 values are comparable across terms for the same category.

3.2.6 Relevance score

First introduced by Kira and Rendell [30], the relevance score (*RS*) of a term t_k measures the relation between the presence of t_k in a class c_i and the absence of t_k in the opposite class $\bar{c}_i$:

$$RS(t_k, c_i) = \log \frac{P(t_k, c_i) + d}{P(\bar{t}_k, \bar{c}_i) + d},$$

where d is a constant damping factor.

Functions f_{sum}, f_{wsum} or $f_{\max}$ can be used to combine category-specific scores.

3.2.7 Odds ratio

Odds ratio (*OR*) was proposed by Van Rijsbergen [47] to select terms for relevance feedback. *OR* is a measure of effect size particularly important in Bayesian statistics and logistic regression. It measures the ratio between the odds of the term appearing in a relevant document and the odds of it appearing in a non-relevant one. In other words, *OR* allows to find terms commonly included in messages belonging to a certain category [16]. The odds ratio between t_k and c_i is given by

$$OR(t_k, c_i) = \frac{P(t_k, c_i)\cdot(1 - P(t_k, \bar{c}_i))}{(1 - P(t_k, c_i))\cdot P(t_k, \bar{c}_i)}.$$

An *OR* of 1 indicates that the term t_k is equally likely in both classes c_i and $\bar{c}_i$. If the *OR* is greater than 1, it indicates that t_k is more likely in c_i. On the other hand, *OR* less than 1 indicates that t_k is less likely in c_i. However, the *OR* must be greater than or equal to zero. As the odds of the c_i approaches zero, *OR* also approaches zero. As the odds of the $\bar{c}_i$ approaches zero, *OR* approaches positive infinity. We can combine category-specific scores using functions f_{sum}, f_{wsum} or $f_{\max}$.

Table 1 The most popular term-selection techniques

Technique	Denotation	Equation
Document frequency	$DF(t_k)$	$\frac{t_k(\mathcal{M})}{\lvert\mathcal{M}\rvert}$
DIA association factor	$DIA(t_k, c_i)$	$P(c_i \lvert t_k)$
Information gain	$IG(t_k)$	$\sum_{c\in[c_i,\bar{c}_i]}\sum_{t\in[t_k,\bar{t}_k]} P(t,c)\cdot\log\frac{P(t,c)}{P(t)\cdot P(c)}$
Mutual information	$MI(t_k, c_i)$	$\log\frac{P(t_k,c_i)}{P(t_k)\cdot P(c_i)}$
χ^2 statistic	$\chi^2(t_k)$	$\frac{\lvert\mathcal{M}\rvert\cdot[P(t_k,c_i)\cdot P(\bar{t}_k,\bar{c}_i)-P(t_k,\bar{c}_i)\cdot P(\bar{t}_k,c_i)]^2}{P(t_k)\cdot P(\bar{t}_k)\cdot P(c_i)\cdot P(\bar{c}_i)}$
Relevance score	$RS(t_k, c_i)$	$\log\frac{P(t_k,c_i)+d}{P(\bar{t}_k,\bar{c}_i)+d}$
Odds ratio	$OR(t_k, c_i)$	$\frac{P(t_k,c_i)\cdot(1-P(t_k,\bar{c}_i))}{(1-P(t_k,c_i))\cdot P(t_k,\bar{c}_i)}$
GSS coefficient	$GSS(t_k)$	$P(t_k,c_i)\cdot P(\bar{t}_k,\bar{c}_i) - P(t_k,\bar{c}_i)\cdot P(\bar{t}_k,c_i)$

3.2.8 GSS coefficient

GSS coefficient is a simplified variant of χ^2 statistic proposed by Galavotti et al. [24], which is defined as

$$GSS(t_k) = P(t_k, c_i)\cdot P\left(\bar{t}_k, \bar{c}_i\right) - P\left(t_k, \bar{c}_i\right)\cdot P\left(\bar{t}_k, c_i\right).$$

The greater (smaller) the positive (negative) values, the stronger the terms t_k will be to indicate the membership (non-membership) of class c_i.

For convenience, the mathematical equations of all presented techniques are summarized in Table 1.[5]

4 Naive Bayes spam filters

Probabilistic classifiers are historically the first proposed filters. These approaches are the most employed in proprietary and open-source systems proposed for spam filtering because of their simplicity and high performance [35, 38, 46].

Given a set of messages $\mathcal{M} = \{m_1, m_2, \ldots, m_j, \ldots, m_{|\mathcal{M}|}\}$ and category set $\mathcal{C} = \{\text{spam}(c_s), \text{legitimate}(c_l)\}$, where m_j is the jth mail in $\mathcal{M}$ and $\mathcal{C}$ is the possible label set, the task of automated spam filtering consists in building a Boolean categorization function $\Phi(m_j, c_i) : \mathcal{M}\times\mathcal{C} \to \{\texttt{True}, \texttt{False}\}$. When $\Phi(m_j, c_i)$ is `True`, it indicates that message m_j belongs to category c_i; otherwise, m_j does not belong to c_i.

In the setting of spam filtering there exist only two category labels: `spam` and `legitimate`. Each message $m_j \in \mathcal{M}$ can only be assigned to one of them, but not to both. Therefore, we can use a simplified categorization function $\Phi_{\texttt{spam}}(m_j) : \mathcal{M} \to \{\texttt{True}, \texttt{False}\}$. Hence, a message is classified as spam when $\Phi_{\texttt{spam}}(m_j)$ is `True`, and legitimate otherwise.

[5]Table 1 shows all term-selection techniques presented in this section in terms of subjective probability. The equations refer to the "local" forms of the functions.

The application of supervised machine learning algorithms for spam filtering consists of two stages:

1. *Training*. A set of labeled messages ($\mathcal{M}$) must be provided as training data, which are first transformed into a representation that can be understood by the learning algorithms. The most commonly used representation for spam filtering is the vector space model, in which each document $m_j \in \mathcal{M}$ is transformed into a real vector $\mathbf{x}_j \in \Re^{|\mathcal{S}|}$, where $\mathcal{S}$ is the vocabulary (feature set) and the coordinates of $\mathbf{x}_j$ represent the weight of each feature in $\mathcal{S}$. Then, we can run a learning algorithm over the training data to create a classifier $\Phi_{\texttt{spam}}(\mathbf{x_j}) \to \{\texttt{True}, \texttt{False}\}$.
2. *Classification*. The classifier $\Phi_{\texttt{spam}}(\mathbf{x_j})$ is applied to the vector representation of a message $\mathbf{x}$ to produce a prediction whether $\mathbf{x}$ is spam or not.

From Bayes' theorem and the theorem of the total probability, the probability that a message with vector $\mathbf{x} = \langle x_1, \ldots, x_n\rangle$ belongs to a category $c_i \in \{c_s, c_l\}$ is:

$$P(c_i|\mathbf{x}) = \frac{P(c_i)\cdot P(\mathbf{x}|c_i)}{P(\mathbf{x})}.$$

Since the denominator does not depend on the category, Naive Bayes (NB) filter classifies each message in the category that maximizes $P(c_i)\cdot P(\mathbf{x}|c_i)$. In the spam filtering domain it is equivalent to classify a message as spam (c_s) whenever

$$\frac{P(c_s)\cdot P(\mathbf{x}|c_s)}{P(c_s)\cdot P(\mathbf{x}|c_s) + P(c_l)\cdot P(\mathbf{x}|c_l)} > T,$$

with $T = 0.5$. By varying T, we can opt for more true negatives (legitimate messages correctly classified) at the expense of fewer true positives (spam messages correctly classified), or vice versa. The *a priori* probabilities $P(c_i)$ can be estimated as frequency of occurrences of documents belonging to the category c_i in the training set $\mathcal{M}$, whereas $P(\mathbf{x}|c_i)$ is practically impossible to estimate directly because we would need in $\mathcal{M}$ some messages identical to the one we want to classify. However, the NB classifier makes

a simple assumption that the terms in a message are conditionally independent and the order they appear is irrelevant. The probabilities $P(\mathbf{x}|c_i)$ are estimated differently in each NB model.

Despite the fact that its independence assumption is usually over-simplistic, several studies have found the NB classifier to be surprisingly effective in the spam filtering task [7, 35].

In the following, we describe the seven most studied models of NB spam filter available in the literature.

4.1 Basic Naive Bayes

We call Basic NB the first NB spam filter proposed by Sahami et al. [40]. Let $\mathcal{S}' = \{t_1, \ldots, t_n\}$ be the set of terms after the term selection; each message m is represented as a binary vector $\mathbf{x} = \langle x_1, \ldots, x_n \rangle$, where each x_k shows whether or not t_k will occur in m. The probabilities $P(\mathbf{x}|c_i)$ are calculated by

$$P(\mathbf{x}|c_i) = \prod_{k=1}^{n} P(t_k|c_i),$$

and the criterion for classifying a message as spam is

$$\frac{P(c_s) \cdot \prod_{k=1}^{n} P(t_k|c_s)}{\sum_{c_i \in \{c_s, c_l\}} P(c_i) \cdot \prod_{k=1}^{n} P(t_k|c_i)} > T.$$

Here, probabilities $P(t_k|c_i)$ are estimated by

$$P(t_k|c_i) = \frac{|\mathcal{M}_{t_k,c_i}|}{|\mathcal{M}_{c_i}|},$$

where $|\mathcal{M}_{t_k,c_i}|$ is the number of training messages of category c_i that contain the term t_k, and $|\mathcal{M}_{c_i}|$ is the total number of training messages that belong to the category c_i.

4.2 Multinomial term frequency Naive Bayes

The multinomial term frequency NB (MN TF NB) represents each message as a set of terms $m = \{t_1, \ldots, t_n\}$, computing each one of t_k as many times as it appears in m. In this sense, m can be represented by a vector $\mathbf{x} = \langle x_1, \ldots, x_n \rangle$, where each x_k corresponds to the number of occurrences of t_k in m. Moreover, each message m of category c_i can be interpreted as the result of picking independently $|m|$ terms from $\mathcal{S}'$ with replacement and probability $P(t_k|c_i)$ for each t_k [37]. Hence, $P(\mathbf{x}|c_i)$ is the multinomial distribution:

$$P(\mathbf{x}|c_i) = P(|m|) \cdot |m|! \cdot \prod_{k=1}^{n} \frac{P(t_k|c_i)^{x_k}}{x_k!}.$$

Thus, the criterion for classifying a message as spam becomes

$$\frac{P(c_s) \cdot \prod_{k=1}^{n} P(t_k|c_s)^{x_k}}{\sum_{c_i \in \{c_s, c_l\}} P(c_i) \cdot \prod_{k=1}^{n} P(t_k|c_i)^{x_k}} > T,$$

and the probabilities $P(t_k|c_i)$ are estimated as a Laplacian prior

$$P(t_k|c_i) = \frac{1 + N_{t_k,c_i}}{n + N_{c_i}},$$

where N_{t_k,c_i} is the number of occurrences of term t_k in the training messages of category c_i, and $N_{c_i} = \sum_{k=i}^{n} N_{t_k,c_i}$.

4.3 Multinomial Boolean Naive Bayes

The multinomial Boolean NB (MN Boolean NB) is similar to the MN TF NB, including the estimates of $P(t_k|c_i)$, except that each attribute x_k is Boolean. Note that these approaches do not take into account the absence of terms ($x_k = 0$) from the messages.

Schneider [43] demonstrates that MN Boolean NB may perform better than MN TF NB. This is because the multinomial NB with term frequency attributes is equivalent to an NB version with the attributes modeled as following Poisson distributions in each category, assuming that the message length is independent of the category. Therefore, the multinomial NB may achieve better performance with Boolean attributes if the term frequencies attributes do not follow Poisson distributions.

4.4 Multivariate Bernoulli Naive Bayes

Let $\mathcal{S}' = \{t_1, \ldots, t_n\}$ be the result set of terms after the term selection. The multivariate Bernoulli NB (MV Bernoulli NB) represents each message m by computing the presence and absence of each term. Therefore, m can be represented as a binary vector $\mathbf{x} = \langle x_1, \ldots, x_n \rangle$, where each x_k shows whether or not t_k will occur in m. Moreover, each message m of category c_i is seen as the result of n Bernoulli trials, where at each trial we decide whether or not t_k will appear in m. The probability of a positive outcome at trial k is $P(t_k|c_i)$. Then, the probabilities $P(\mathbf{x}|c_i)$ are computed by

$$P(\mathbf{x}|c_i) = \prod_{k=1}^{n} P(t_k|c_i)^{x_k} \cdot \big(1 - P(t_k|c_i)\big)^{(1-x_k)}.$$

The criterion for classifying a message as spam becomes

$$\frac{P(c_s) \cdot \prod_{k=1}^{n} P(t_k|c_s)^{x_k} \cdot (1 - P(t_k|c_s))^{(1-x_k)}}{\sum_{c_i \in \{c_s, c_l\}} P(c_i) \cdot \prod_{k=1}^{n} P(t_k|c_i)^{x_k} \cdot (1 - P(t_k|c_i))^{(1-x_k)}} > T,$$

and probabilities $P(t_k|c_i)$ are estimated as a Laplacian prior

$$P(t_k|c_i) = \frac{1 + |\mathcal{M}_{t_k,c_i}|}{2 + |\mathcal{M}_{c_i}|},$$

where $|\mathcal{M}_{t_k,c_i}|$ is the number of training messages of category c_i that comprise the term t_k, and $|\mathcal{M}_{c_i}|$ is the total

Table 2 Naive Bayes spam filters

NB Classifier	$P(\mathbf{x}\|c_i)$	Complexity on	
		Training	Classification
Basic NB	$\prod_{k=1}^{n} P(t_k\|c_i)$	$O(n \cdot \|\mathcal{M}\|)$	$O(n)$
MN TF NB	$\prod_{k=1}^{n} P(t_k\|c_i)^{x_k}$	$O(n \cdot \|\mathcal{M}\|)$	$O(n)$
MN Boolean NB	$\prod_{k=1}^{n} P(t_k\|c_i)^{x_k}$	$O(n \cdot \|\mathcal{M}\|)$	$O(n)$
MV Bernoulli NB	$\prod_{k=1}^{n} P(t_k\|c_i)^{x_k} \cdot (1 - P(t_k\|c_i))^{(1-x_k)}$	$O(n \cdot \|\mathcal{M}\|)$	$O(n)$
Boolean NB	$\prod_{k=1}^{n} P(t_k\|c_i)$	$O(n \cdot \|\mathcal{M}\|)$	$O(n)$
MV Gauss NB	$\prod_{k=1}^{n} g(x_k; \mu_{k,c_i}, \sigma_{k,c_i})$	$O(n \cdot \|\mathcal{M}\|)$	$O(n)$
Flexible Bayes	$\prod_{k=1}^{n} \frac{1}{L_{k,c_i}} \sum_{l=1}^{L_{k,c_i}} g(x_k; \mu_{k,c_i,l}, \sigma_{c_i})$	$O(n \cdot \|\mathcal{M}\|)$	$O(n \cdot \|\mathcal{M}\|)$

number of training messages of category c_i. For more theoretical explanation, consult Metsis et al. [38] and Losada and Azzopardi [34].

4.5 Boolean Naive Bayes

We denote as Boolean NB the classifier similar to the MV Bernoulli NB with the difference that it does not take into account the absence of terms. Hence, the probabilities $P(\mathbf{x}|c_i)$ are estimated only by

$$P(\mathbf{x}|c_i) = \prod_{k=1}^{n} P(t_k|c_i),$$

and the criterion for classifying a message as spam becomes

$$\frac{P(c_s) \cdot \prod_{k=1}^{n} P(t_k|c_s)}{\sum_{c_i \in \{c_s, c_l\}} P(c_i) \cdot \prod_{k=1}^{n} P(t_k|c_i)} > T,$$

where probabilities $P(t_k|c_i)$ are estimated in the same way as used in the MV Bernoulli NB.

4.6 Multivariate Gauss Naive Bayes

Multivariate Gauss NB (MV Gauss NB) uses real-valued attributes by assuming that each attribute follows a Gaussian distribution $g(x_k; \mu_{k,c_i}, \sigma_{k,c_i})$ for each category c_i, where the μ_{k,c_i} and σ_{k,c_i} of each distribution are estimated from the training set $\mathcal{M}$.

The probabilities $P(\mathbf{x}|c_i)$ are calculated by

$$P(\mathbf{x}|c_i) = \prod_{k=1}^{n} g(x_k; \mu_{k,c_i}, \sigma_{k,c_i}),$$

and the criterion for classifying a message as spam becomes

$$\frac{P(c_s) \cdot \prod_{k=1}^{n} g(x_k; \mu_{k,c_s}, \sigma_{k,c_s})}{\sum_{c_i \in \{c_s, c_l\}} P(c_i) \cdot \prod_{k=1}^{n} g(x_k; \mu_{k,c_i}, \sigma_{k,c_i})} > T.$$

4.7 Flexible Bayes

Flexible Bayes (FB) works similarly to MV Gauss NB. However, instead of using a single normal distribution for each attribute X_k per category c_i, FB represents the probabilities $P(\mathbf{x}|c_i)$ as the average of L_{k,c_i} normal distributions with different values for μ_{k,c_i} but the same one for σ_{k,c_i}:

$$P(x_k|c_i) = \frac{1}{L_{k,c_i}} \sum_{l=1}^{L_{k,c_i}} g(x_k; \mu_{k,c_i,l}, \sigma_{c_i}),$$

where L_{k,c_i} is the quantity of different values that the attribute X_k has in the training set $\mathcal{M}$ of category c_i. Each of these values is used as $\mu_{k,c_i,l}$ of a normal distribution of the category c_i. However, all distributions of a category c_i are taken to have the same $\sigma_{c_i} = \frac{1}{\sqrt{|\mathcal{M}_{c_i}|}}$.

The distribution of each category becomes narrower as more training messages of that category are accumulated. By averaging several normal distributions, FB can approximate the true distributions of real-valued attributes more closely than the MV Gauss NB when the assumption that attributes follow normal distribution is violated. For further details, consult John and Langley [28] and Androutsopoulos et al. [7].

Table 2 summarizes all NB spam filters presented in this section.[6]

5 Performance measurements

According to Cormack [14], the filters should be judged along four dimensions: autonomy, immediacy, spam identification, and non-spam identification. However, it is not obvious how to measure any of these dimensions separately, nor how to combine these measurements into a single one for the purpose of comparing filters. Reasonable standard

[6]The computational complexities are according to Metsis et al. [38]. At classification time, the complexity of FB is $O(n \cdot |\mathcal{M}|)$ because it needs to sum the L_k distributions.

Table 3 All possible prediction results

Notation	Composition	Also known as
$\mathcal{TP}$	Set of spam messages correctly classified	True positives
$\mathcal{TN}$	Set of legitimate messages correctly classified	True negatives
$\mathcal{FN}$	Set of spam messages incorrectly classified as legitimate	False negatives
$\mathcal{FP}$	Set of legitimate messages incorrectly classified as spam	False positives

Table 4 Popular performance measurements used in the literature

Measurement	Equation
True positive rate (*Tpr*), spam caught (%) or sensitivity	$Tpr = \frac{\lvert\mathcal{TP}\rvert}{\lvert c_s\rvert}$
False positive rate (*Fpr*), blocked ham (%)	$Fpr = \frac{\lvert\mathcal{FP}\rvert}{\lvert c_l\rvert}$
True negative rate (*Tnr*), legitimate recall or specificity	$Tnr = \frac{\lvert\mathcal{TN}\rvert}{\lvert c_l\rvert}$
False negative rate (*Fnr*), spam misclassification rate	$Fnr = \frac{\lvert\mathcal{FN}\rvert}{\lvert c_s\rvert}$
Spam precision (*Spr*)	$Spr = \frac{\lvert\mathcal{TP}\rvert}{\lvert\mathcal{TP}\rvert+\lvert\mathcal{FP}\rvert}$
Legitimate precision (*Lpr*)	$Lpr = \frac{\lvert\mathcal{TN}\rvert}{\lvert\mathcal{TN}\rvert+\lvert\mathcal{FN}\rvert}$
Accuracy rate (*Acc*)	$Acc = \frac{\lvert\mathcal{TP}\rvert+\lvert\mathcal{TN}\rvert}{\lvert c_s\rvert+\lvert c_l\rvert}$
Error rate (*Err*)	$Err = \frac{\lvert\mathcal{FP}\rvert+\lvert\mathcal{FN}\rvert}{\lvert c_s\rvert+\lvert c_l\rvert}$

measures are useful to facilitate comparison, given that the goal of optimizing them does not replace that of finding the most suitable filter for the purpose of spam filtering.

Considering the category set $\mathcal{C} = \{\text{spam}(c_s), \text{legitimate}(c_l)\}$ and all possible prediction results presented in Table 3, some well-known evaluation measures are presented in Table 4.

All the measures presented in Table 4 consider a false negative as harmful as a false positive. Nevertheless, failures to identify legitimate and spam messages have different consequences [14, 15]. According to Cormack [14], misclassified legitimate messages increase the risk that the information contained in the message will be lost, or at least delayed. It is very difficult to measure the amount of risk and delay that can be supported, once the consequences depend on the relevance of the message content for a given user. On the other hand, failures to identify spam also vary in importance, but are generally less critical than failures to identify non-spam. Viruses, worms, and phishing messages may be an exception, as they pose significant risks to the user.

Whatever the measure adopted, an aspect to be considered is the asymmetry in the misclassification costs. A spam message incorrectly classified as legitimate is a significantly minor problem, as the user is simply required to remove it. On the other hand, a legitimate message mislabeled as spam can be unacceptable, as it implies the loss of potentially important information, particularly for those settings in which spam messages are automatically deleted.

To overcome the lack of symmetry, Androutsopoulos et al. [5] proposed a further refinement based on spam recall and precision in order to allow the performance evaluation based on a single measure. They consider a false positive as being λ times more costly than false negatives, where λ equals to 1 or 9. Thus, each false positive is accounted as λ mistakes.

In this case, the total cost ratio (*TCR*) can be calculated by

$$TCR = \frac{|c_s|}{\lambda|\mathcal{FP}| + |\mathcal{FN}|}.$$

TCR is an evaluation measurement commonly employed to compare the performances achieved by different spam filters. It offers an indication of the improvement provided by the filter. A bigger *TCR* indicates a better performance, and for $TCR < 1$, not using the filter is preferable.

The problem of using *TCR* is that it does not return a value inside a predefined range [10, 15]. For instance, consider two classifiers A and B employed to filter 600 messages (450 spams + 150 legitimates, $\lambda = 1$). Suppose that A attains a perfect prediction with $\mathcal{FP}_A = \mathcal{FN}_A = 0$, and B incorrectly classifies 3 spam messages as legitimate, thus $\mathcal{FP}_B = 0$ and $\mathcal{FN}_B = 3$.

In this way, $TCR_A = +\infty$ and $TCR_B = 150$. Intuitively, we can observe that both classifiers achieved a similar performance with a small advantage for A. However, if we analyze only the *TCR*, we may mistakenly claim that A is much better than B. Notice that *TCR* just returns the information about the improvement provided by the filter. However, it does not offer any information about how much the classifier can be improved. Thus, *TCR* is not a representative measure that can assist us to make assumptions about the performance of a single classifier.

To avoid those drawbacks, we propose the use of the Matthews correlation coefficient (*MCC*) [36] for assessing the performance of spam classifiers. *MCC* is used in machine learning as a quality measure of binary classifications, which provides much more information than *TCR*. It returns a real value between -1 and $+1$. A coefficient equal to $+1$ indicates a perfect prediction; 0, an average random prediction; and -1, an inverse prediction.

MCC provides a balanced evaluation of the prediction (i.e., the proportion of correct predictions for each class), especially if the two classes are of very different sizes [8]. It can be calculated by

$$MCC = \frac{(|\mathcal{TP}| \cdot |\mathcal{TN}|) - (|\mathcal{FP}| \cdot |\mathcal{FN}|)}{\sqrt{(|\mathcal{TP}| + |\mathcal{FP}|) \cdot (|\mathcal{TP}| + |\mathcal{FN}|) \cdot (|\mathcal{TN}| + |\mathcal{FP}|) \cdot (|\mathcal{TN}| + |\mathcal{FN}|)}}.$$

Using the previous example, the classifiers A and B achieve $MCC_A = 1.000$ and $MCC_B = 0.987$, respectively. Now, it is noteworthy that we can make correct assumptions for the prediction in-between the classifiers as well as for each individual performance.

As with *TCR*, we can define an independent rate $\lambda > 1$ to indicate how much a false positive is worse than a false negative. For that, the amount of false positives ($|\mathcal{FP}|$) in the *MCC* equation is simply multiplied by λ:

$$MCC = \frac{(|\mathcal{TP}| \cdot |\mathcal{TN}|) - (\lambda|\mathcal{FP}| \cdot |\mathcal{FN}|)}{\sqrt{(|\mathcal{TP}| + \lambda|\mathcal{FP}|) \cdot (|\mathcal{TP}| + |\mathcal{FN}|) \cdot (|\mathcal{TN}| + \lambda|\mathcal{FP}|) \cdot (|\mathcal{TN}| + |\mathcal{FN}|)}}.$$

Moreover, *MCC* can also be combined with other measures in order to guarantee a fairer comparison, such as precision $\times$ recall, blocked hams (false positive) and spam caught (true positive) rates.

6 Experimental protocol

In this section, we present the experimental protocol designed for the empirical evaluation of the different term-selection methods presented in Sect. 3. They were applied for reducing the dimensionality of the term space before the classification task performed by the Bayesian filters presented in Sect. 4.

We carried out this study on the six well-known, large, real and public Enron[7] data sets. The corpora are composed of legitimate messages extracted from the mailboxes of six former employees of the Enron Corporation. For further details about the data set statistics and composition, refer to Metsis et al. [38].

For providing an aggressive dimensionality reduction, we performed the training stage using the first 90% of the received messages (training set). The remaining ones were separated for classifying (testing set).

After the training stage, we applied the term-selection techniques (TSTs) presented in Sect. 3 for reducing the dimensionality of the term space.[8] In order to perform a comprehensive performance evaluation, we varied the number of terms to be selected from 10 to 100% of all retained terms in the preprocessing stage.

Next, we classified the testing messages using the Naive Bayes spam filters presented in Sect. 4. We set the classification threshold $T = 0.5$ ($\lambda = 1$) as used in Metsis et al. [38]. By varying T, we can opt for more true negatives at the cost of fewer true positives, or vice versa.

We tested all possible combinations between NB spam filters and term-selection methods. In spite of using all the performance measurements presented in Table 4 for evaluating the classifiers, we selected the *MCC* to compare their results.

7 Experimental results

This section presents the results achieved for each corpus. In the remainder of this paper, consider the following abbreviations: Basic NB as Bas, Boolean NB as Bool, MN Boolean NB as MN Bool, MN term frequency NB as MN TF, MV Bernoulli NB as Bern, MV Gauss NB as Gauss, and flexible Bayes as FB.

7.1 Overall analysis

Due to space limitations, we present only the best combination (i.e., TST and % of $|\mathcal{S}|$) for each NB classifier.[9] We define "best result" the combination that obtained the highest *MCC*.

Tables 5, 7, 9, 11, 13, and 15 show the best combination for each filter and its corresponding *MCC*. Additionally, we present the complete set of performance measures for the best classifiers in Tables 6, 8, 10, 12, 14, and 16.

It can be seen from Table 6 that both Bern with $DIA_{\max}$ @50% and Basic with RS_{wsum}@80% obtained the same *TCR* but different *MCC* for Enron 1. This happens because the *MCC* offers a balanced evaluation of the prediction, particularly if the classes are of different sizes, as discussed in Sect. 5.

Table 11 shows another drawback of *TCR*. Bern with *IG*@10% achieved a perfect prediction ($|\mathcal{FP}| = |\mathcal{FN}| = 0$) for Enron 4, attaining $MCC = 1.000$ and $TCR = +\infty$. On the other hand, Bool with $DIA_{\max}$@40% incorrectly classified one spam as legitimate ($|\mathcal{FP}| = 0$, $|\mathcal{FN}| = 1$), accomplishing $MCC = 0.996$ and $TCR = 450$. If we analyze only

[7] The Enron data sets are available at http://www.iit.demokritos.gr/skel/i-config/.

[8] For relevance score, we used a damping factor $d = 0.1$ [44].

[9] The complete set of results is available at http://www.dt.fee.unicamp.br/~tiago/Research/Spam/spam.htm

Table 5 Enron 1: the best result achieved by each NB filter

Classifier	TST	% of $\|\mathcal{S}\|$	MCC
MV Bernoulli NB	$DIA_{\max}$	50	0.885
Basic NB	RS_{wsum}	80	0.872
Boolean NB	$DIA_{\max}$	50	0.867
MN Boolean NB	OR_{wsum}	50	0.861
MN TF NB	OR_{wsum}	50	0.844
MV Gauss NB	*IG*	70	0.839
Flexible Bayes	*IG*	70	0.833

Table 6 Enron 1: two classifiers that attained the best individual performance

Measurement	Bern & $DIA_{\max}$	Basic & RS_{wsum}
$\|\mathcal{S}'\|$(% of $\|\mathcal{S}\|$)	50	80
Blocked ham(%)	7.06	2.45
Spam caught(%)	99.33	88.00
Tpr(%) & *Spr*(%)	99.33 & 85.14	88.00 & 93.62
Tnr(%) & *Lpr*(%)	92.93 & 99.71	97.55 & 95.23
Acc(%)	94.79	94.79
TCR	5.556	5.556
MCC	0.885	0.872

Table 7 Enron 2: the best result achieved by each filter

Classifier	TST	% of $\|\mathcal{S}\|$	MCC
MV Bernoulli NB	$OR_{\max}/OR_{\text{sum}}$	40	0.952
Boolean NB	DIA_{sum}	50	0.915
Basic NB	$RS_{\max}$	30	0.909
MV Gauss NB	χ^2	20	0.896
MN TF NB	$OR_{\max}/OR_{\text{sum}}$	40	0.874
MN Boolean NB	$OR_{\max}/OR_{\text{sum}}$	40	0.861
Flexible Bayes	*IG*	10	0.855

Table 8 Enron 2: two classifiers that attained the best individual performance

Measurement	Bern & $OR_{\max}/OR_{\text{sum}}$	Bool & DIA_{sum}
$\|\mathcal{S}'\|$(% of $\|\mathcal{S}\|$)	40	50
Blocked ham(%)	2.29	0.23
Spam caught(%)	99.33	88.00
Tpr(%) & *Spr*(%)	99.33 & 93.71	88.00 & 99.25
Tnr(%) & *Lpr*(%)	97.71 & 99.77	99.77 & 96.04
Acc(%)	98.13	96.76
TCR	13.636	7.895
MCC	0.952	0.915

the *TCR*, we may wrongly claim that the first combination is much better than the second one.

Figure 1 shows the TSTs that attained the best average prediction (i.e., the highest area under the curve) for each NB classifier. In this figure, we present the individual results of each data set.

Note that the classifiers generally worsen their performance when the complete set of terms $|\mathcal{S}|$ is used for training, except for *MI*. There is a trade-off between 30 and 60% of $|\mathcal{S}|$ which usually achieves the best performance for the other TSTs. Even a set of selected terms composed by only 10–30% of $|\mathcal{S}|$ generally offers better results than a set with

Table 9 Enron 3: the best result achieved by each filter

Classifier	TST	% of $\|\mathcal{S}\|$	MCC
Boolean NB	*IG*	60	0.991
MV Bernoulli NB	*IG*	30	0.973
Basic NB	*IG*	10	0.950
MN Boolean NB	*IG*	10	0.936
MV Gauss NB	χ^2	10	0.917
MN TF NB	*IG*	10	0.884
Flexible Bayes	$MI_{\max}$	20	0.880

Table 10 Enron 3: two classifiers that attained the best individual performance

Measurement	Bool & *IG*	Bern & *IG*
$\|\mathcal{S}'\|$(% of $\|\mathcal{S}\|$)	60	30
Blocked ham(%)	0.00	1.24
Spam caught(%)	98.67	99.33
Tpr(%) & *Spr*(%)	98.67 & 100.00	99.33 & 96.75
Tnr(%) & *Lpr*(%)	100.00 & 99.50	98.76 & 99.75
Acc(%)	99.64	98.91
TCR	75.000	25.000
MCC	0.991	0.973

Table 11 Enron 4: the best result achieved by each filter

Classifier	TST	% of $\|\mathcal{S}\|$	MCC
MV Bernoulli NB	*IG*	10/20	1.000
Boolean NB	$DIA_{\max}/OR$	40	1.000
MN Boolean NB	$DIA_{\max}/OR$	40	0.996
MN TF NB	$DIA_{\max}/OR$	40	0.996
Basic NB	DIA_{sum}	40	0.978
Flexible Bayes	$DIA_{\max}/OR$	40	0.974
MV Gauss NB	$DIA_{\max}/OR$	40	0.970

Table 12 Enron 4: two classifiers that attained the best individual performance

Measurement	Bern & *IG*	Bool & $DIA_{\max}/OR$
$\|\mathcal{S}'\|$(% of $\|\mathcal{S}\|$)	10/20	40
Blocked ham(%)	0.00	0.00
Spam caught(%)	100.00	100.00
Tpr(%) & *Spr*(%)	100.00 & 100.00	100.00 & 100.00
Tnr(%) & *Lpr*(%)	100.00 & 100.00	100.00 & 100.00
Acc(%)	100.00	100.00
TCR	$+\infty$	$+\infty$
MCC	1.000	1.000

all the terms of $\mathcal{S}$. On the other hand, it is also noteworthy that *MI* often achieves better results when we employ the complete set of terms $|\mathcal{S}|$.

Regarding the TSTs, the results indicate that $\{IG, \chi^2, DF, OR, DIA\} > \{RS, GSS\} \gg MI$, where ">" means "performs better than." However, if we consider the average prediction, we can see that *IG* and χ^2 are less sensitive to the variation of $|\mathcal{S}'|$ and they usually offer better results than *OR* and *DIA*.

We also verify that the performance of the NB filters is highly sensitive to the quality of terms selected by the TSTs and the number of selected terms $|\mathcal{S}'|$. For instance, MV Bernoulli NB achieved a perfect prediction ($MCC =$

Table 13 Enron 5: the best result achieved by each filter

Classifier	TST	% of $\|\mathcal{S}\|$	MCC
MV Bernoulli NB	$OR_{\max}/OR_{\text{sum}}$	50	0.972
MN Boolean NB	OR_{wsum}	50	0.967
Boolean NB	$OR_{\max}/OR_{\text{sum}}$	60	0.955
MN TF NB	$OR_{\max}/OR_{\text{sum}}$	50	0.954
Flexible Bayes	χ^2	10	0.931
Basic NB	*DF*	10	0.924
MV Gauss NB	*GSS*	20	0.895

Table 14 Enron 5: two classifiers that attained the best individual performance

Measurement	Bern & $OR_{\max}/OR_{\text{sum}}$	MN Bool & OR_{wsum}
$\|\mathcal{S}'\|$(% of $\|\mathcal{S}\|$)	50	50
Blocked ham(%)	2.67	2.67
Spam caught(%)	99.46	99.18
Tpr(%) & *Spr*(%)	99.46 & 98.92	99.18 & 98.92
Tnr(%) & *Lpr*(%)	97.33 & 98.65	97.33 & 97.99
Acc(%)	98.84	98.65
TCR	61.333	52.571
MCC	0.972	0.967

Table 15 Enron 6: the best result achieved by each filter

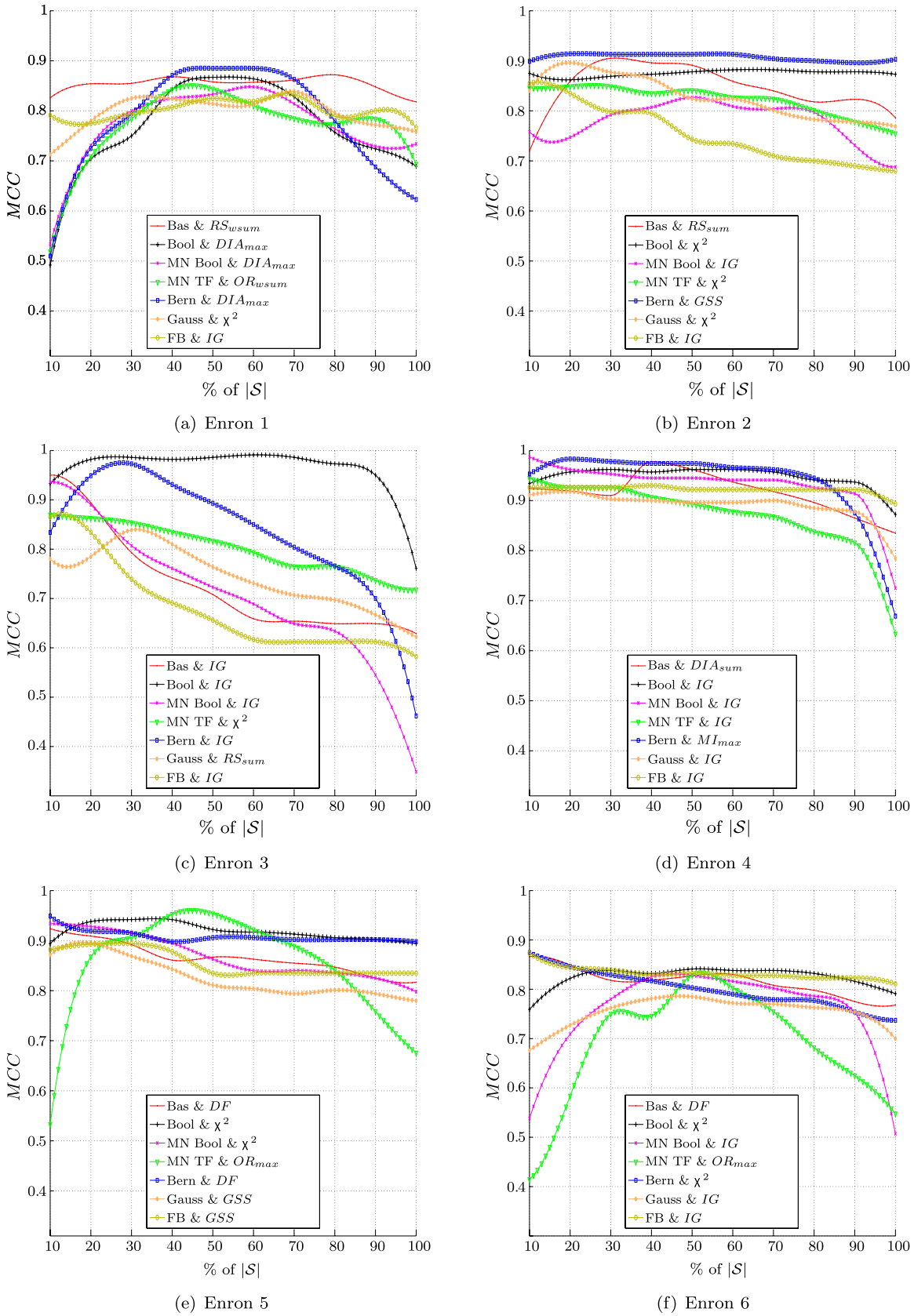
Classifier	TST	% of $\|\mathcal{S}\|$	MCC
Boolean NB	OR_{wsum}	60	0.929
MV Bernoulli NB	$OR_{\max}/OR_{\text{sum}}$	50	0.923
MN Boolean NB	$OR_{\max}/OR_{\text{sum}}$	60	0.897
Flexible Bayes	*IG*	10	0.873
Basic NB	*DF*	10	0.866
MN TF NB	$OR_{\max}/OR_{\text{sum}}$	50	0.829
MV Gauss NB	$OR_{\max}/OR_{\text{sum}}$	50	0.819

Table 16 Enron 6: two classifiers that attained the best individual performance

Measurement	Bool & OR_{wsum}	Bern & $OR_{\max}/OR_{\text{sum}}$
$\|\mathcal{S}'\|$(% of $\|\mathcal{S}\|$)	60	50
Blocked ham(%)	6.00	2.67
Spam caught(%)	98.45	96.88
Tpr(%) & *Spr*(%)	98.44 & 98.01	96.89 & 99.09
Tnr(%) & *Lpr*(%)	94.00 & 95.27	97.33 & 91.25
Acc(%)	97.33	97.00
TCR	28.125	25.000
MCC	0.929	0.923

1.000) for Enron 4 when we use 10% of $|\mathcal{S}|$ selected by *IG*, whereas it attained $MCC = -0.082$ when we employ MI_{sum}.

With respect to the filters, the individual and average results indicate that {Boolean NB, MV Bernoulli NB, Basic NB} > {MN Boolean NB, MN term frequency NB} > {flexible Bayes, MV Gauss NB}. MV Bernoulli NB and Boolean NB acquired the best individual performance for the most of the data sets. Further, MV Bernoulli NB is the only approach that takes into account the absence of terms in the messages. This feature provides more information, assisting the classifiers' prediction for those cases in which

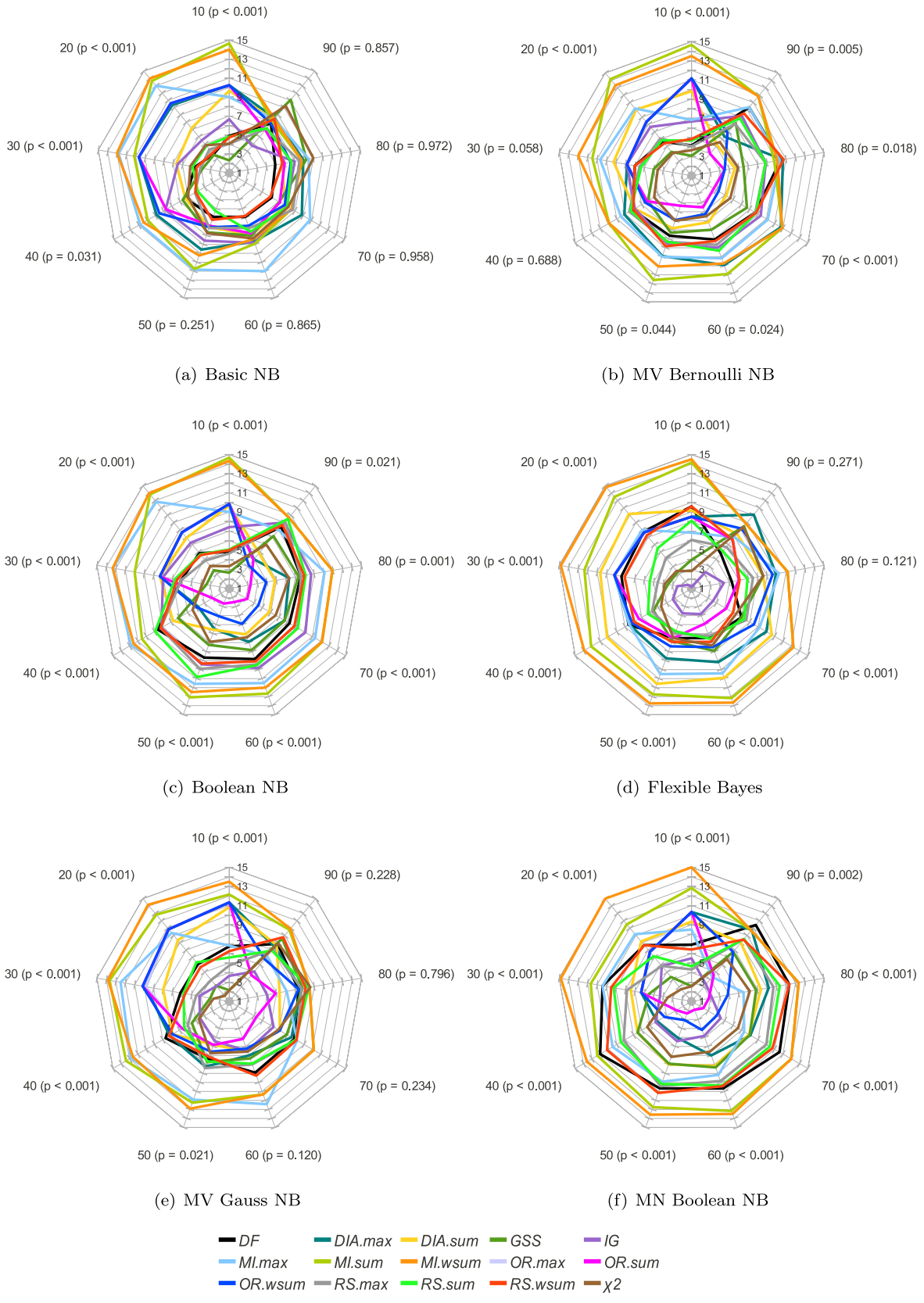


(a) Enron 1 (b) Enron 2

(c) Enron 3 (d) Enron 4

(e) Enron 5 (f) Enron 6

Fig. 1 TSTs that attained the best average prediction for each NB classifier by varying the number of selected terms

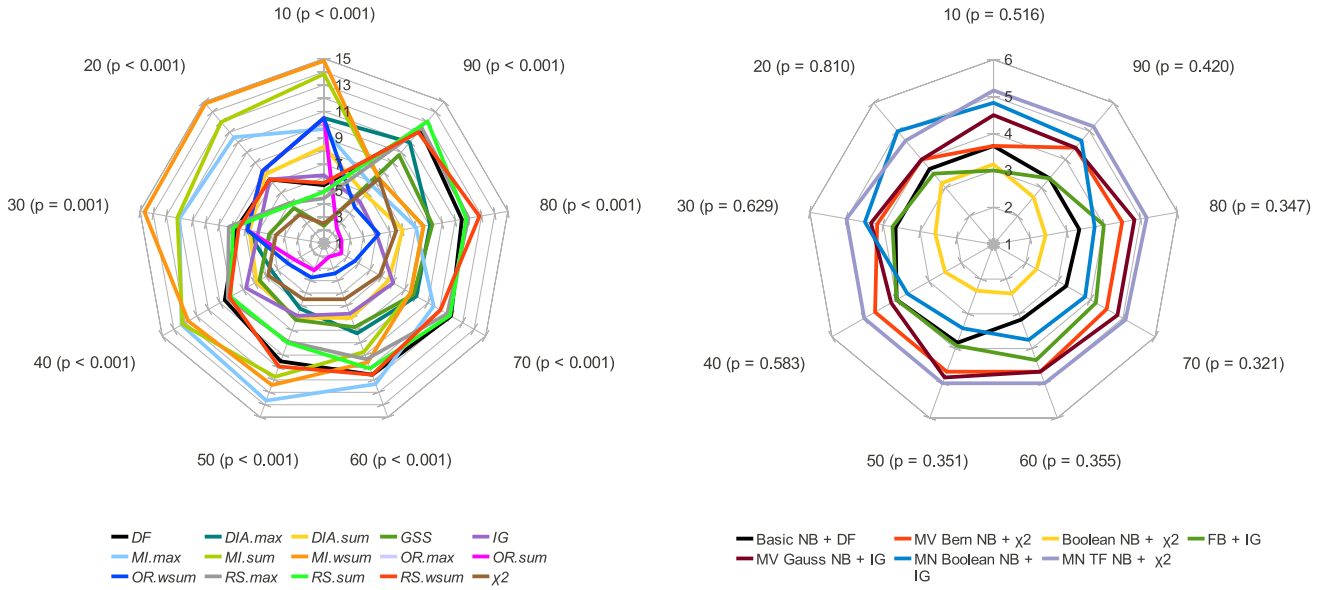


(a) Basic NB (b) MV Bernoulli NB

(c) Boolean NB (d) Flexible Bayes

(e) MV Gauss NB (f) MN Boolean NB

Fig. 2 Average rank achieved by TSTs for each spam filter



(a) MN Term Frequency NB (b) Best Combinations

Fig. 3 Average rank achieved by TSTs for MN Term Frequency NB spam filter (**a**) and the best combinations between filters and TSTs (**b**)

Table 17 Summary of the observed results

Classifier	% of $\|\mathcal{S}\|$	Highlights	
		Best	Worst
Basic NB	10–30	*DF*, *GSS*, *RS*, χ^2	*MI*
MV Bernoulli NB	10–20	χ^2, *GSS*, *RS*, *DF*	*MI*
MV Bernoulli NB	70	*OR*	$DIA_{\max}$, *MI*
Boolean NB	10–70	χ^2	*MI*
Flexible Bayes	10–70	*IG*	DIA_{sum}, *MI*
MV Gauss NB	10–40	*IG*, χ^2, *GSS*	*MI*
MN Boolean NB	10–80	*IG*	*MI*
MN Term Frequency NB	10–50	χ^2	*MI*
MN Term Frequency NB	50–100	*OR*	*DF*, *RS*

users generally receive messages with some specific terms, such as names or signatures.

7.2 Statistical analysis

In the following, we present a statistical analysis of the results. For that, we used a Friedman's test [17] for comparing the distribution of ranks among the analyzed algorithms across the six Enron data sets.

Figures 2 and 3(a) show the average rank achieved by each TST.[10] In those figures, we present the individual results of each spam classifier. The x axis shows the different values of $|\mathcal{S}|\%$ and the p-values at each level. It is important to note that the smaller the geometric area, the better the technique. According to Nemenyi test [17], the critical distance (CD) for pairwise comparisons between TSTs at $p = 0.01$ is 8.16.

Table 17 summarizes the analysis of the results. For each classifier, we present the % of $|\mathcal{S}|$ in which we observe statistical differences between the TSTs. In those cases, we identify three groups. Clearly, the best methods outperform the worst ones. However, the experimental data are not sufficient for assuming any conclusion to which group the remainder of techniques belong.

For instance, considering the Basic NB, Fig. 2(a) indicates that *DF* achieved the best average rank for all e-mail collections, regardless the amount of selected terms ($|\mathcal{S}|\%$). On the other hand, *MI* accomplished the worst average rank.

[10] All color pictures are available at http://www.dt.fee.unicamp.br/~tiago/Research/Spam/spam.htm.

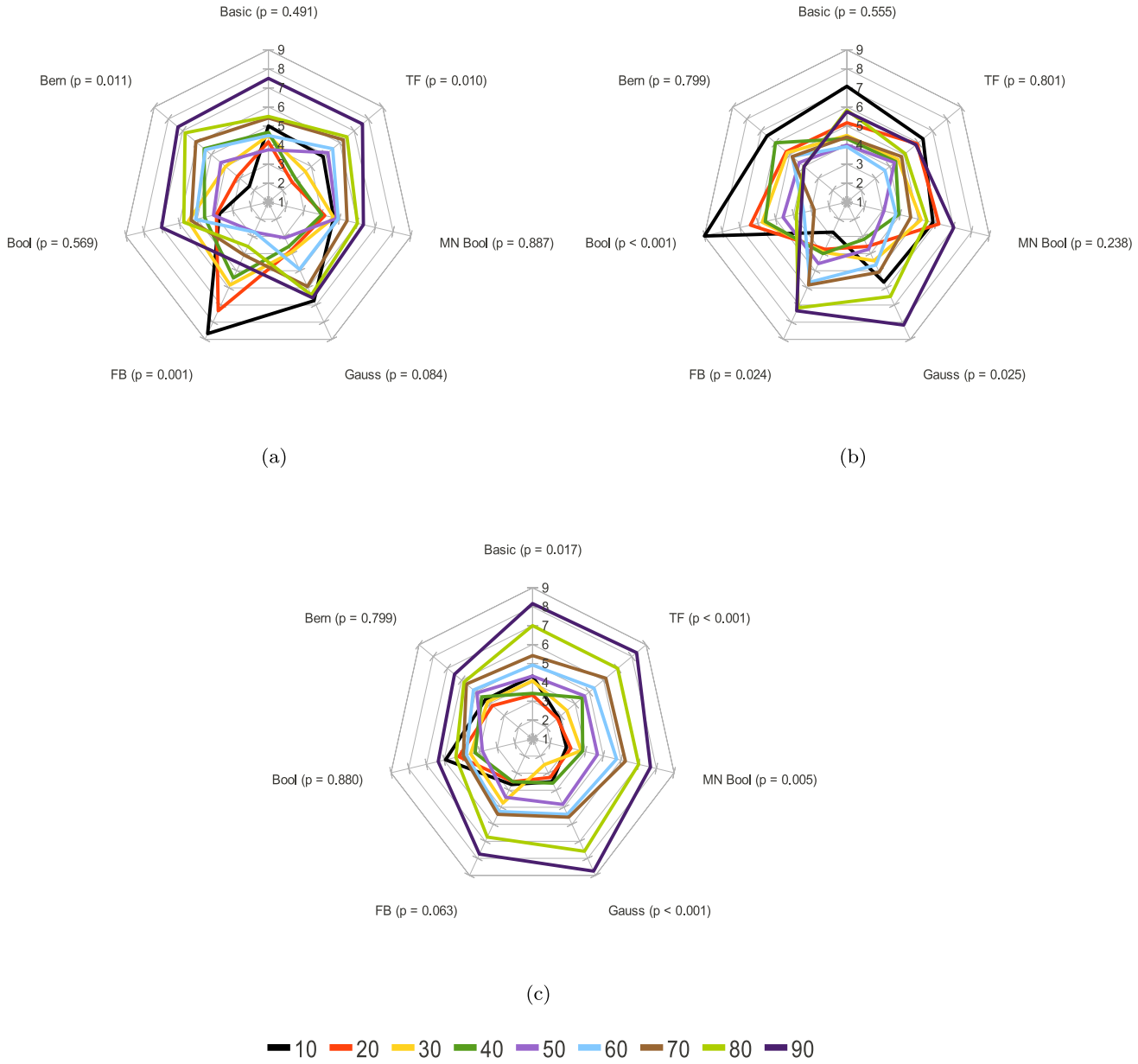


Fig. 4 Average rank achieved by NB filters when document frequency (**a**), information gain (**b**) and χ^2 statistic (**c**) are used

When $|\mathcal{S}|\%$ varies from 10 to 30%, there is a significant statistical difference between TSTs. Notice that the performance of MI_{sum} and MI_{wsum} is significantly worse than that of *DF*, *GSS*, *RS* and χ^2 for such an interval. However, we cannot reach any conclusion regarding the remainder of TSTs.

Another interesting result can be observed in Fig. 2(d). *IG* undoubtedly achieved the best average rank for Flexible Bayes. *MI* has attained again the worst average rank. The performance of MI_{sum} and MI_{wsum} is significantly worse than *IG* when 10–70% of $|\mathcal{S}|$ was selected.

Additionally, we have compared the average rank attained by the statistically best combinations (NB spam filter and TST), as illustrated in Fig. 3(b). Although the results indicate that Boolean NB with χ^2 statistic is better (in average rank) than any other evaluated combination, the experimental data is not sufficient to reach any conclusion.

Finally, we have also evaluated how the number of selected terms affects the average rank achieved by the statistically best TSTs (*DF*, *IG* and χ^2 statistic) for all the compared NB spam filters (Fig. 4).

The analysis suggests that there is a significant statistical difference between distinct values of $|\mathcal{S}|\%$ for the combinations of Boolean NB with *IG*, MV Gauss NB with χ^2 statistic and MN term frequency NB with χ^2 statistic.

In general, the statistical results are consistent with the individual best results (Sect. 7.1), with few exceptions. For instance, MV Bernoulli NB has presented good individual performance for each e-mail collection and, however, the statistical analysis indicates that such a filter is inferior to Basic NB and Boolean NB in average rank. Moreover, the Friedman's test also indicates that Flexible Bayes is not worse than other filters, as the individual results have presented.

8 Conclusions and further work

In this paper, we have presented a performance evaluation of several term-selection methods in dimensionality reduction for the spam filtering domain by classifiers based on the Bayesian decision theory. We have performed the comparison of the performance achieved by seven different Naive Bayes spam filters applied to classify messages from six well-known, real, public and large e-mail data sets, after a step of dimensionality reduction employed by eight popular term-selection techniques varying the number of selected terms.

Furthermore, we have proposed the Matthews correlation coefficient (*MCC*) as the evaluation measure instead of the total cost ratio (*TCR*). *MCC* provides a more balanced evaluation of the prediction than *TCR*, especially if the two classes are of different sizes. Moreover, it returns a value inside a predefined range, which provides more information about the classifiers' performance.

Regarding term-selection techniques, we have found that *DF*, *IG*, and χ^2 statistic are the most effective in aggressive term removal without losing categorization accuracy. *DIA*, *RS*, *GSS* coefficient and *OR* also provide an improvement on the filters' performance. On the other hand, *MI* generally offers poor results which frequently worsen the classifiers' performance.

Among of all presented classifiers, Boolean NB and Basic NB achieved best individual and average rank performance. The results also verify that Boolean attributes perform better than the term frequency ones as presented by Schneider [43].

We also have shown that the performance of Naive Bayes spam classifiers is highly sensitive to the selected attributes and the number of selected terms by the term-selection methods in the training stage. The better the term-selection technique, the better the filters' prediction.

Future works should take into consideration that spam filtering is a co-evolutionary problem, because while the filter tries to evolve its prediction capacity, the spammers try to evolve their spam messages in order to overreach the classifiers. Hence, an efficient approach should have an effective way to adjust its rules in order to detect the changes of spam features. In this way, collaborative filters [33] could be used to assist the classifier by accelerating the adaptation of the rules and increasing the classifiers' performance. Moreover, spammers generally insert a large amount of noise in spam messages in order to make the probability estimation more difficult. Thus, the filters should have a flexible way to compare the terms in the classifying task. Approaches based on fuzzy logic [49] could be employed to make the comparison and selection of terms more flexible.

Acknowledgement This work is partially supported by the Brazilian funding agencies CNPq, CAPES and FAPESP.

References

1. Almeida T, Yamakami A (2010) Content-based spam filtering. In: Proceedings of the 23rd IEEE international joint conference on neural networks, Spain, Barcelona, pp 1–7
2. Almeida T, Yamakami A, Almeida J (2009) Evaluation of approaches for dimensionality reduction applied with Naive Bayes anti-spam filters. In: Proceedings of the 8th IEEE international conference on machine learning and applications, Miami, FL, USA, pp 517–522
3. Almeida T, Yamakami A, Almeida J (2010) Filtering spams using the minimum description length principle. In: Proceedings of the 25th ACM symposium on applied computing, Sierre, Switzerland, pp 1856–1860
4. Almeida T, Yamakami A, Almeida J (2010) Probabilistic anti-spam filtering with dimensionality reduction. In: Proceedings of the 25th ACM symposium on applied computing, Sierre, Switzerland, pp 1802–1806
5. Androutsopoulos I, Koutsias J, Chandrinos K, Paliouras G, Spyropoulos C (2000) An evaluation of Naive Bayesian anti-spam filtering. In: Proceedings of the 11st European conference on machine learning, Barcelona, Spain, pp 9–17
6. Androutsopoulos I, Paliouras G, Karkaletsis V, Sakkis G, Spyropoulos C, Stamatopoulos P (2000) Learning to filter spam e-mail: a comparison of a Naive Bayesian and a memory-based approach. In: Proceedings of the 4th European conference on principles and practice of knowledge discovery in databases, Lyon, France, pp 1–13
7. Androutsopoulos I, Paliouras G, Michelakis E (2004) Learning to filter unsolicited commercial e-mail. Technical Report 2004/2, National Centre for Scientific, Research "Demokritos", Athens, Greece
8. Baldi P, Brunak S, Chauvin Y, Andersen C, Nielsen H (2000) Assessing the accuracy of prediction algorithms for classification: an overview. Bioinformatics 16(5):412–424
9. Bratko A, Cormack G, Filipic B, Lynam T, Zupan B (2006) Spam filtering using statistical data compression models. J Mach Learn Res 7:2673–2698
10. Carpinter J, Hunt R (2006) Tightening the Net: a review of current and next generation spam filtering tools. Comput Secur 25(8):566–578
11. Carreras X, Marquez L (2001) Boosting trees for anti-spam email filtering. In: Proceedings of the 4th international conference on recent advances in natural language processing, Tzigov Chark, Bulgaria, pp 58–64
12. Cohen W (1995) Fast effective rule induction. In: Proceedings of 12nd international conference on machine learning, Tahoe City, CA, USA, pp 115–123
13. Cohen W (1996) Learning rules that classify e-mail. In: Proceedings of the AAAI spring symposium on machine learning in information access, Stanford, CA, USA, pp 18–25

14. Cormack G (2008) Email spam filtering: a systematic review. Found Trends Inf Retr 1(4):335–455
15. Cormack G, Lynam T (2007) Online supervised spam filter evaluation. ACM Trans Inf Syst 25(3):1–11
16. Cunningham P, Nowlan N, Delany S, Haahr M (2003) A case-based approach to spam filtering that can track concept drift. In: Proceedings of the 5th international conference on case based reasoning. Trondheim, Norway, pp 115–123
17. Demsar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30
18. Drucker H, Wu D, Vapnik V (1999) Support vector machines for spam categorization. IEEE Trans Neural Netw 10(5):1048–1054
19. Forman G (2003) An extensive empirical study of feature selection metrics for text classification. J Mach Learn Res 3:1289–1305
20. Forman G, Kirshenbaum E (2008) Extremely fast text feature extraction for classification and indexing. In: Proceedings of 17th ACM conference on information and knowledge management, Napa Valley, CA, USA, pp 1221–1230
21. Forman G, Scholz M, Rajaram S (2000) Feature shaping for linear SVM classifiers. In: Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining. Paris, France, pp 299–308
22. Friedman N, Geiger D, Goldszmidt M (1997) Bayesian network classifiers. Mach Learn 29(3):131–163
23. Fuhr N, Buckley C (1991) A probabilistic learning approach for document indexing. ACM Trans Inf Syst 9(3):223–248
24. Galavotti L, Sebastiani F, Simi M (2000) Experiments on the use of feature selection and negative evidence in automated text categorization. In: Proceedings of 4th European conference on research and advanced technology for digital libraries, Lisbon, Portugal, pp 59–68
25. Guzella T, Caminhas W (2000) A review of machine learning approaches to spam filtering. Exp Syst Appl 36(7):10206–10222
26. Hidalgo J (2002) Evaluating cost-sensitive unsolicited bulk email categorization. In: Proceedings of the 17th ACM symposium on applied computing, Madrid, Spain, pp 615–620
27. Joachims T (1997) A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In: Proceedings of 14th international conference on machine learning, Nashville, TN, USA, pp 143–151
28. John G, Langley P (1995) Estimating continuous distributions in Bayesian classifiers. In: Proceedings of the 11st international conference on uncertainty in artificial intelligence, Montreal, Canada, pp 338–345
29. John G, Kohavi R, Pfleger K (1994) Irrelevant features and the subset selection problem. In: Proceedings of 11st international conference on machine learning, New Brunswick, NJ, USA, pp 121–129
30. Kira K, Rendell L (1992) A practical approach to feature selection. In: Proceedings of the 9th international workshop on machine learning, Aberdeen, Scotland, UK, pp 249–256
31. Kolcz A, Alspector J (2001) SVM-based filtering of e-mail spam with content-specific misclassification costs. In: Proceedings of the 1st international conference on data mining, San Jose, CA, USA, pp 1–14
32. Koprinska I, Poon J, Clark J, Chan J (2007) Learning to classify e-mail. Inf Sci 177(10):2167–2187
33. Lemire D (2005) Scale and translation invariant collaborative filtering systems. Inf Retr 8(1):129–150
34. Losada D, Azzopardi L (2008) Assessing multivariate Bernoulli models for information retrieval. ACM Trans Inf Syst 26(3):1–46
35. Marsono M, El-Kharashi N, Gebali F (2009) Targeting spam control on middleboxes: spam detection based on layer-3 e-mail content classification. Comput Netw 53(6):835–848
36. Matthews B (1975) Comparison of the predicted and observed secondary structure of *T4* phage lysozyme. Biochim Biophys Acta 405(2):442–451
37. McCallum A, Nigam K (1998) A comparison of event models for Naive Bayes text classification. In: Proceedings of the 15th AAAI workshop on learning for text categorization, Menlo Park, CA, USA, pp 41–48
38. Metsis V, Androutsopoulos I, Paliouras G (2006) Spam filtering with Naive Bayes—which Naive Bayes. In: Proceedings of the 3rd international conference on email and anti-spam, Mountain View, CA, USA, pp 1–5
39. Mitchell T (1997) Machine learning. McCraw-Hill, New York
40. Sahami M, Dumais S, Hecherman D, Horvitz E (1998) A Bayesian approach to filtering junk e-mail. In: Proceedings of the 15th national conference on artificial intelligence, Madison, WI, USA, pp 55–62
41. Schapire R, Singer Y, Singhal A (1998) Boosting and Rocchio applied to text filtering. In: Proceedings of the 21st annual international conference on information retrieval, Melbourne, Australia, pp 215–223
42. Schneider K (2003) A comparison of event models for Naive Bayes anti-spam e-mail filtering. In: Proceedings of the 10th conference of the European chapter of the association for computational linguistics, Budapest, Hungary, pp 307–314
43. Schneider K (2004) On word frequency information and negative evidence in Naive Bayes text classification. In: Proceedings of the 4th international conference on advances in natural language processing, Alicante, Spain, pp 474–485
44. Sebastiani F (2002) Machine learning in automated text categorization. ACM Comput Surv 34(1):1–47
45. Seewald A (2007) An evaluation of Naive Bayes variants in content-based learning for spam filtering. Int Data Anal 11(5):497–524
46. Song Y, Kolcz A, Gilez C (2009) Better Naive Bayes classification for high-precision spam detection. Softw Pract Exp 39(11):1003–1024
47. Van Rijsbergen C (1979) Information retrieval, 2nd edn. Butterworths, London
48. Yang Y, Pedersen J (1997) A comparative study on feature selection in text categorization. In: Proceedings of the 14th international conference on machine learning, Nashville, TN, USA, pp 412–420
49. Zadeh L (1965) Fuzzy sets. Inf Control 8(3):338–353