Probability and Variance Score: an Efficient Supervised Feature Selection Method for Text Classification

Heyong Wang*, Ming Hong1
1Department of E-Business, South China University of Technology, Guangzhou, China, 510006
*wanghey@scut.edu.cn

Abstract

This paper proposes a new supervised feature selection method termed probability and variance score (PVS) for text classification. PVS aims to improve variance score (VS), a simple unsupervised feature selection method because VS only evaluates quantity of information of terms but it is not able to evaluate relationships between terms and classes of text documents. PVS not only evaluates quantity of information of terms by measuring variances of terms, the same method as VS, but also considers relationships between terms and classes by measuring posterior probabilities that terms occur given classes based on Bayesian Theory. Terms selected by PVS tends to be information-rich and highly class-related. Experimental results on three datasets indicate that PVS is efficient in selecting discriminative terms and outperforms VS.

Keywords
Text Mining; Text Classification; Feature Selection

1. Introduction

Text mining is also known as text data mining or knowledge discovery from textual databases and it is able to extract interesting and non-trivial patterns or knowledge from unstructured text documents [1]. Text mining has captured increasing interests in recent years because text data has a high commercial potential [1] and a large amount of text data is available as information-centric applications become popular [2]. Text mining has been applied to some areas such as understanding genetics of diseases [3], supporting the education evidence portal [4] and identifying biologically relevant entities in text data as well as constructing literature-based networks of protein-protein interactions [5].

Text data is always high-dimensional, so it is important to have an efficient method to remove indiscriminative terms so as to reduce dimensions of term space. Feature selection is such an ideal technique. Feature selection is a kind of dimension reduction method [6] to reduce term space from a high dimensional and indiscriminative one to a low dimensional and discriminative one, that is, feature selection is capable of keeping discriminative terms when the term space is reduced. Feature selection methods can be supervised or unsupervised. Supervised feature selection methods include information gain (IG), mutual information (MI), Gini index (GI) and expected cross-entropy (ECE). Unsupervised feature selection methods include document frequency (DF), variance score (VS) and Laplacian Score (LS).

VS is a simple unsupervised feature selection method. VS evaluates quantity of information of terms by measuring variances of terms. However, VS is not able to evaluate relationships between terms and classes. The relationships between terms and classes help to improve performances of classifiers in text classification because they offer more class-related information which is needed for construction classifiers. To improve VS, this paper proposes supervised PVS. Formula of PVS consists of two parts: variance of terms and impurity of terms. The same as VS, PVS evaluates quantity of information of terms by measuring variances. Moreover, PVS introduces impurity of
terms to evaluate relationships between terms and classes. The impurity is calculated based on Bayesian Theory and it offers more information for text classification.

The rest of this paper is organized as follows. Section 2 discusses several typical feature selection methods proposed in current studies. Section 3 formally introduces PVS and analyzes details of PVS. In Section 4, experiments are performed on two datasets from UCI machine learning repository and one dataset collected from Tencent microblog to validate efficiency of PVS in text classification. More discussions between efficiency of PVS and VS in text classification are also covered in Section 4. Finally, Section 5 gives conclusions.

2. Related Work

$n$ text documents are often represented as a document term matrix $DTM = (f_{ij})_{n \times m}$ (shown in Fig.1). $n$ denotes the number of text documents and $m$ denotes the total number of terms of the $n$ text documents. The $i$th row $D_i = [f_{i1}, f_{i2}, \ldots, f_{im}] (i = 1, 2, \ldots, n)$ denotes the $i$th text document among the $n$ text documents. The $j$th column $f_j = [f_{1j}, f_{2j}, \ldots, f_{nj}] (j = 1, 2, \ldots, m)$ corresponds to the $j$th term (denoted by $w_j$) and $f_{ij}$ is term frequency of $w_j$ in $D_i$.

![Fig.1 Document Term Matrix](image)

A lot of feature selection methods have been proposed currently. This section discusses several typical supervised and unsupervised feature selection methods proposed in current studies.

2.1. Supervised Feature Selection Methods

$P(c_k)$ denotes the prior probability that a text document $D_i$ belongs to class $c_k$. $P(f_j)$ denotes the prior probability that the $j$th term (denoted by $w_j$) occurs. $P(c_k|f_j)$ denotes the posterior probability of $c_k$ given $w_j$. $P(f_j|c_k)$ denotes the posterior probability that $w_j$ occurs given $c_k$.

$IG$ [7, 9, 11] selects terms according to amounts of information that a term uses for classification [8]. The IG of $f_j$ in text classification is given by formula (1).

$$IG(f_j) = - \sum_{k=1}^{c} P(c_k) \log(P(c_k)) + P(f_j) \sum_{k=1}^{c} P(c_k|f_j) \log P(c_k|f_j) + (1 - P(f_j)) \sum_{k=1}^{c} (1 - P(c_k|f_j)) \log(1 - P(c_k|f_j))$$ (1)

Some studies have used $IG$ for text classification. Thorsten Joachims studies performances of SVM in text classification and $IG$ is used to select a term subset [11]. Yunhui Bai introduces a covering algorithm based on $IG$. Good performances of this algorithm indicate effectiveness of $IG$ in feature selection [12]. Kaiqi Li et al. uses $IG$ to improve $TF-IDF$ in calculating weights of terms and experimental results indicate good performances of $IG$ [13]. Jing Shen applies $IG$ to improve performances of Latent Dirichlet Allocation (LDA). $IG$ is proved effective in selecting terms in experiments [14]. Harun Ug˘uz ranks importance of terms in the studies. Experiments on several classifiers indicate effectiveness of $IG$ in selecting important terms [15]. Other Studies about $IG$ are also proposed. Considering $IG$ is weak in deal with the problem of ignoring the unbalance of between-class or inside-class distribution of a term. Some studies have proposed solutions for this issue [16 - 23]. Zhixiong Chen et al. argue that $IG$ is not able to show class attributes of a term and introduce FoilGain to improve $IG$ [24]. Some studies pay attention to redundancy between terms and give their improvements using $IG$ [25-26].

$MI$ [27, 28] observes interesting relationship between two variables. The $MI$ of $f_j$ in text classification is given by
formula (2).

\[ MI(f_j) = \sum_{k=1}^{c} P(c_k) \log \frac{P(f_j | c_k)}{P(f_j)} \]  

(2)

\( MI \) has been used in text classification. Both Ge Zhou and Zhili Pei et al. develop an algorithm for text classification by \( MI \). Experimental results indicate good performances of \( MI \) in this algorithm [29,30]. Wen Li et al. apply \( MI \) to classification of Chinese legal text and \( MI \) is proved effective in experiments [31]. More studies focus on improvements of \( MI \). Some studies pay attention to the problem that \( MI \) tends to select rare terms. Their improvements on \( MI \) are proved effective in experiments [32–34]. Schneider KM argues that \( MI \) gives long documents higher weights in term scores and proposes an improved method which has better performances [35]. Arguing the problem that \( MI \) ignores that a term could be negative, some studies have proposed improved \( MI \) to distinguish negative and positive terms. Experimental results demonstrate effectiveness of these improvements [36–39]. Jianjun Wu et al. propose an improved \( MI \) to solve the problem that \( MI \) does not consider term frequency [40]. The same as \( IG \), some studies are proposed to solve the problem \( MI \) ignoring the unbalance of between-class or inside-class distribution of a feature [41–43]. Moreover, there are many other improvements of \( MI \)[44–48].

\( GI \) [8] evaluates impurity of \( f_j \). The \( GI \) of \( f_j \) is given by formula (3).

\[ GI(f_j) = \sum_{k=1}^{c} P_k^2(f_j | c_k) \]  

(3)

\( GI \) is firstly introduced in CART, a typical decision tree algorithm [8] and it can be used in feature selection for text classification and text clustering [49, 50]. Improvements of \( GI \) are covered in some studies and experimental results demonstrate their effectiveness [51–53].

\( ECE \) [54–57] is a measure from information theory. \( ECE \) is given by formula (4).

\[ ECE(f_j) = P(f_j) \sum_{k=1}^{c} P(c_k | f_j) \log \frac{P(c_k | f_j)}{P(f_j)} \]  

(4)

Some studies apply \( ECE \) to select terms from term space of \( DTM \) for text classification [58–62]. Few studies focus on improving \( ECE \) currently. Lili Shan et al. analyze \( ECE \) and propose an improved \( ECE \) to guarantee the balance of the numbers of terms contributing to different classes [63].

2.2. Unsupervised feature selection methods

\( DF \) [64] assumes that frequent terms are more informative than non frequent terms. \( DF \) is the total number of documents in which \( w_j \) occurs. \( DF \) is usually used as a criterion to evaluate efficiency of other feature selection methods. \( DF \) of \( f_j \) is given by formula (5).

\[ DF(f_j) = \text{num}(f_{ij} > 0) \]  

(5)

\( \text{num}(f_{ij} > 0) \) denotes the total number of documents in which \( w_j \) occurs. A discriminative term gets high value of \( DF \). Some studies propose improvements of \( DF \). Xu Y. et al. introduce term frequency (\( TF \)) to improve \( DF \) and experimental results demonstrate the new method is better than the original \( DF \) [65]. Some studies argue the problem that \( DF \) does not catch the relationships between terms and classes and propose their improvements and these improved methods are demonstrated better in experiments [66, 67].

\( VS \) [68] is a simple feature selection method which evaluates quantity of information of \( w_j \) by calculating variance of \( f_j \) in \( DTM \). The \( VS \) of \( f_j \) is given by formula (6).

\[ VS(f_j) = \text{Var}(f_j) = \frac{1}{n} \sum_{i=1}^{n} (f_{ij} - \bar{f}_j)^2 \]  

(6)

\( \bar{f}_j \) is the mean of \( f_j \) and it is given by formula (7).
\[ \bar{f}_j = \frac{1}{n} \sum_{i=1}^{n} f_{ij} \]  

Terms with high indicate that they contain more information. A term with high VS is considered as discriminative for text classification. LS \[69\] can be treated as an improvement of VS. Besides evaluating variances of features, LS introduces a similarity measure to evaluate similarity between text documents and similarity between features.

3. Probability and Variance Score

PVS evaluates quantity of information of a term by measuring variance of the term and evaluates relationships between terms and classes by measuring impurity of the term. PVS is mainly based on Bayesian Theory. Section 3.1 introduces the concepts of PVS to evaluate how likely a term belongs to a particular class based on Bayesian Theory. Section 3.2 introduces construction of impurity of a term. Illustration of both PVS and feature selection using PVS.

3.1. Relationships between Terms and Classes

In practice, there are relationships between terms and classes. For example, terms “CPU” and “software” may belong to class “computer science”, instead of class “literature”. PVS aims to evaluate these relationships in order to improve VS. Bayesian Theory is useful to provide a way to calculate posterior probability \[23\]. In PVS, relationship between \( w_j \) and class \( c_k \) is described by \( P(c_k|f_j) \) which denotes the posterior probability that \( w_j \) belongs to \( c_k \), that is, how likely \( w_j \) belongs to \( c_k \). Large value of \( P(c_k|f_j) \) indicates a strong relationship between \( w_j \) and \( c_k \). Our idea comes from the concept of Naïve Bayes classifier \[8\] which predicts how likely an instance \( X \) belongs to \( c_k \) by measuring \( P(c_k|X) \). This paper extends the concept of Naïve Bayes classifier and uses \( P(c_k|f_j) \) to predict how likely \( w_j \) belongs to \( c_k \). Moreover, \( X \) can only belong to a specific class but \( w_j \) possibly belongs to several classes. \( P(c_k|f_j) \) is given by formula (8) based on Bayesian Formula.

\[ P(c_k|f_j) = \frac{P(f_j|c_k) \times P(c_k)}{P(f_j)} \]  

(8)

\( P(c_k) \) denotes the prior probability that \( c_k \) occurs and \( P(c_k) \) is given by formula (9).

\[ P(c_k) = \frac{n_{c_k}}{n} \]  

(9)

\( n_{c_k} \) denotes the total number of text documents which belong to \( c_k \). \( P(f_j|c_k) \) denotes the posterior probability that \( w_j \) occurs given \( c_k \). Actually, authors write text documents independently and they are not affected by each other, which indicates elements of \( f_j = [f_{j1}, f_{j2}, ..., f_{jn}] \) are independent of each other given \( c_k \). Based on the independent assumption, \( P(f_j|c_k) \) is given by formula (10).

\[ P(f_j|c_k) = \prod_{i=1}^{n} P(f_{ji} > 0|c_k) \]  

(10)

\( P(f_j) \) denotes the prior probability that \( w_j \) occurs. According to total probability theorem, \( P(f_j) \) is given by formula (11).

\[ P(f_j) = \sum_{k=1}^{k} P(f_j|c_k) = \sum_{k=1}^{k} P(f_j|c_k) \times P(c_k) \]  

(11)

Moreover, we have

\[ \sum_{k=1}^{k} P(c_k|f_j) = \frac{\sum_{k=1}^{k} P(f_j|c_k) \times P(c_k)}{P(f_j)} = \frac{\sum_{k=1}^{k} P(f_j|c_k) \times P(c_k)}{P(f_j)} = 1 \]  

(12)

3.2. Impurity of Term

Section 3.1 introduces \( P(c_k|f_j) \) to evaluate relationship between \( w_j \) and \( c_k \). This section describes the solution to combine the relationships between \( w_j \) and each class so as to evaluate the importance of \( w_j \) among all the classes.
The impurity of $w_j$ is introduced as the solution. The idea of impurity comes from the concept of $GI$. Impurity of $w_j$ is constructed using $P(c_k|f_j)$ and given by formula (13):

$$IP(f_j) = \sum_{k=1}^{r} P^2(c_k|f_j)$$ (13)

The rest of this section discusses the effectiveness of IP in selecting discriminative terms. Let $K_j = \{c_k|P(c_k|f_j) > 0, k = 1, 2, \ldots, t\}$. A discriminative term is able to discover significant classes from $K_j$ so as to distinguish significant classes from other classes. For each $c_k \in K_j, P(c_k|f_j)$ is treated as an significance of $c_k$ for $w_j$. If value of $P(c_k|f_j)$ is big, then $c_k$ is very significant for $w_j$ because $w_j$ offers much information of $c_k$. For example, let $K_j = \{c_1, c_2\}$ consider the following cases of $c_j$:

Case 1: If $P(c_1|f_j) = P(c_2|f_j) = \frac{1}{2}$, $w_j$ is able to offer half of information of it to both $c_1$ and $c_2$ because $c_1$ and $c_2$ are the same. In this case, $w_j$ is considered not discriminative because $c_1$ and $c_2$ are the same significant for $w_j$, which means $w_j$ is not able to discover significant classes.

Case 2: If $P(c_1|f_j) = \frac{2}{5}$ and $P(c_2|f_j) = \frac{3}{5}$, $w_j$ is able to distinguish $c_1$ and $c_2$ because quantity of information it offers to $c_1$ and $c_2$ are quite different. $w_j$ offers lots of information to $c_1$ and little to $c_2$. In this case, $w_j$ is considered discriminative because $c_1$ can be distinguished from $c_2$ by $w_j$.

Let $P_j = \{P(c_k|f_j)|k = 1, 2, \ldots, t\}$ and $Var(P_j)$ denotes variance of $P_j$ given $f_j$. From the two cases described above, it is clear that $w_j$ is discriminative when variance of $P_j$ is large, that is $Var(P_j)$ can be used to evaluate the ability of $w_j$ to discover significant classes in $K_j$. Consider two cases of $Var(P_j)$ in Fig.2, each dot represents $P(c_k|f_j)$ of $c_k$ in $K_j$ ($K_j = \{c_1, c_2, \ldots, c_8\}$) and $\bar{P}$ denotes the mean of $P_j$. Fig.2 (a) shows that each $P(c_k|f_j)$ is near $\bar{P}$, that is, significances of classes ($P(c_k|f_j)$, $c_k \in K_j$ in $K_j$ are nearly the same, which means that $w_j$ is not able to discover significant classes in $K_j$. Fig.3 (b) shows that $P(c_3|f_j)$, $P(c_5|f_j)$ and $P(c_8|f_j)$ are obviously higher than $\bar{P}$, that is, significances of $c_3$, $c_5$ and $c_8$ are obviously higher than the rest classes in $K_j$, which means that $w_j$ is able to discover significant classes ($c_3$, $c_5$ and $c_8$) in $K_j$. To sum up, if $Var(P_j)$ is big, then $w_j$ is more capable of discovering significant classes in $K_j$. In other words, if $Var(P_j)$ is big, then $w_j$ uses its information efficiently because information showing distinction of classes is good for text classification. We have Rule 1.

**Rule 1:** $Var(P_j)$ evaluates the ability of $w_j$ to use its information efficiently. It is a requirement for $w_j$ to be discriminative that $Var(P_j)$ is large.

The following propositions help to understand $IP$. **Proposition 1 and 2** gives reasons why $IP(f_j)$ should be replace $Var(P_j)$ as the solution to distinguish significant classes from other classes in PVS.

**Proposition 1:** $IP(f_j) \sim Var(P_j)$

Proof 1:

Formula (12) indicates that $\sum_{k=1}^{r} P(c_k|f_j) = 1$. Let $\bar{P}$ denote the mean of $P_j$ then $\bar{P} = \frac{1}{n} \sum_{k=1}^{r} P(c_k|f_j) = \frac{1}{n}$.
Then, we have formula (14).

\[
V\text{ar}(P_j) = \frac{1}{n} \sum_{k=1}^{n} \left( P(c_k|f_j) - \bar{P} \right)^2 = \frac{1}{n} \sum_{k=1}^{n} \left( P(c_k|f_j) - \frac{1}{n} \right)^2 \\
= \frac{1}{n} \sum_{k=1}^{n} \left( P^2(c_k|f_j) - \frac{2}{n} P(c_k|f_j) + \frac{1}{n^2} \right) \\
= \frac{1}{n} \sum_{k=1}^{n} P^2(c_k|f_j) + \frac{2}{n^2} \sum_{k=1}^{n} P(c_k|f_j) + \frac{1}{n^2} \sum_{k=1}^{n} \frac{1}{n} \\
= \frac{1}{n} IP(f_j) + \frac{2n+t}{n^2} 
\]

(14)

Finally, \( V\text{ar}(P_j) = \frac{1}{n} IP(f_j) + \frac{2n+t}{n^2} \), so, \( V\text{ar}(P_j) \sim DF(f_j) \). Furthermore, \( IP(f_j) = n V\text{ar}(P_j) - \frac{2n+t}{n} \), so \( IP(f_j) \sim V\text{ar}(P_j) \).

Proposition 2: \( \frac{1}{\text{Len}(K_j)} \leq IP(f_j) \leq 1 \) (\( \text{Len}(K_j) \) is the number of elements in \( K_j \)).

Proof 2:

The proposition is actually the solution of the following problem:

\[
\text{min} IP(f_j) = \sum_{k=1}^{\text{Len}(K_j)} P^2(c_k|f_j) \\
\text{s.t.} \sum_{k=1}^{\text{Len}(K_j)} P(c_k|f_j) = 1, \quad P(c_k|f_j) \in (0,1] 
\]

Cauchy inequality is used to solve the problem and it is given by formula (15).

\[
\sum_{i=1}^{n} a_i^2 \sum_{j=1}^{n} b_j^2 \geq \left( \sum_{i=1}^{n} a_i b_j \right)^2 
\]

(15)

If \( a_i > 0 \) and \( b_i > 0 \), then \( \sum_{i=1}^{n} a_i^2 \sum_{j=1}^{n} b_j^2 = \left( \sum_{i=1}^{n} a_i b_j \right)^2 \) only if \( \frac{a_1}{b_1} = \frac{a_2}{b_2} = \ldots = \frac{a_n}{b_n} \).

According to formula (15), we have formula (16).

\[
\sum_{k'=1}^{\text{Len}(K_j)} 1 \sum_{k=1}^{\text{Len}(K_j)} P^2(c_k|f_j) = \text{Len}(K_j) \sum_{k=1}^{\text{Len}(K_j)} P^2(c_k|f_j) \\
= \text{Len}(K_j) IP(f_j) \geq \sum_{k=1}^{\text{Len}(K_j)} P(c_k|f_j) = 1 
\]

(16)

Here, \( P(c_k|f_j) > 0 \) and furthermore, \( \text{Len}(K_j) IP(f_j) = 1 \) only if \( \frac{1}{P(c_1|f_j)} = \frac{1}{P(c_2|f_j)} = \ldots = \frac{1}{P(c_{\text{Len}(K_j)}|f_j)} \), that is

\[
P\left(c_1|f_j\right) = P\left(c_2|f_j\right) = \ldots = P\left(c_{\text{Len}(K_j)}|f_j\right) = \frac{1}{\text{Len}(K_j)} 
\]

so

\[
\text{min} IP(f_j) = \sum_{k=1}^{\text{Len}(K_j)} P^2(c_k|f_j) = \sum_{k=1}^{\text{Len}(K_j)} \left( \frac{1}{\text{Len}(K_j)} \right)^2 = \frac{1}{\text{Len}(K_j)}. 
\]

According to definition of probability, \( 0 \leq P\left(c_k|f_j\right) \leq 1 \), then \( P^2\left(c_k|f_j\right) \leq P\left(c_k|f_j\right) \). Moreover, according to formula (12), \( \sum_{k=1}^{\text{Len}(K_j)} P\left(c_k|f_j\right) = 1 \). Then \( IP(f_j) = \sum_{k=1}^{\text{Len}(K_j)} P^2\left(c_k|f_j\right) \leq \sum_{k=1}^{\text{Len}(K_j)} P\left(c_k|f_j\right) = 1 \). Finally, we have \( \frac{1}{\text{Len}(K_j)} \leq IP(f_j) \leq 1 \).

Proposition 1 demonstrates positive correlation between \( V\text{ar}(P_j) \) and \( IP(f_j) \), which indicates that \( IP(f_j) \) is also able to evaluate the ability of \( w_j \) to discover significant classes in \( K_j \). \( PVS \) chooses \( IP \), instead of variance, to evaluate the ability of a term to distinguish significant classes from other classes because formula of \( IP \) is simpler than that of variance, which is more suitable for computational operations. Moreover, Proposition 2 demonstrates that \( IP(f_j) \) is a positive numeric value not larger than 1, so \( IP(f_j) \) can be treated as a percentage to show how well \( f_j \) is able to discover significant classes. All possible values of \( IP(f_j) \) fall in \( \left[ \frac{1}{\text{Len}(K_j)}, 1 \right] \). However, values of
do not fall in any specific intervals, which means it is much easier to judge the ability of discovering significant classes by \( IP(f_j) \) than by \( Var(P_j) \). In summary, Rule 2 is considered more suitable than Rule 1.

**Rule 2:** \( IP(f_j) \) evaluates the ability of \( w_j \) to use its information efficiently. It is a requirement for \( w_j \) to be discriminative that \( IP(f_j) \) is large.

### 3.3. Feature Selection Using PVS

The PVS of \( f_j \) is formulated as formula

\[
PVS(f_j) = IP(f_j) \times Var(f_j) = \left[ \sum_{k=1}^{n} P^2(c_k|f_j) \right] \sum_{i=1}^{n} (f_{ij} - \bar{f}_j)^2
\]

where \( \bar{f}_j \) is the mean of \( f_j \). PVS is constructed by two parts: impurity of \( w_j \) (\( IP(f_j) \)) and standard variance of \( w_j \) (\( Var(f_j) \)). \( IP(f_j) \) evaluates the ability of \( w_j \) to use its information efficiently and \( Var(f_j) \) evaluates quantity of information offered by \( w_j \). If \( IP(f_j) \) is large, then \( w_j \) is able to use information of it efficiently. However, there is still a problem that we have to understand how much information \( w_j \) contains because \( IP(f_j) \) is not able to guarantee that \( w_j \) contains enough information for text classification. It is simple to evaluate information contained in a term by measuring its variance, so PVS uses \( Var(f_j) \) to evaluate information offered by \( w_j \). If \( w_j \) contains much information for text classification, then \( Var(f_j) \) tends to be large. To sum up, large \( IP(f_j) \) guarantees that \( w_j \) uses its information efficiently and large \( Var(f_j) \) guarantees that \( w_j \) is able to offer as much information as possible for text classification. If \( w_j \) is discriminative, then \( PVS(f_j) \) is large, so Rule 3 is the final rule for a discriminative term.

**Rule 3:** \( f_j \) is discriminative if \( PVS(f_j) \) is large.

VS only considers quantity of information of \( w_j \) by \( Var(f_j) \), ignoring the relationships between the term and classes. The improvement of PVS on VS is that PVS not only evaluates quantity of information of \( w_j \) by \( Var(f_j) \) but also evaluates relationship between \( w_j \) and classes and the relationships are good for text classification because the classification algorithm needs class-related information to build classifier.

Algorithm FSPVS in Fig.3 shows the algorithm of feature selection using PVS in text classification. The code from Line 2 to Line 7 calculates values of PVS for each term. The calculation of impurity of each term is covered in Line 4 to Line 6. In Line 7, a map is used to store \( w_j \) (key) and \( PVS(f_j) \) (value). Elements of the map is sorted decreasingly and the \( N \) terms with large PVS (the top \( N \) terms) are chosen for text classification. Value of \( N \) is determined manually.

---

**Procedure FSPVS**

**Input:**
- \( DTM \): document term matrix of \( m \) text documents, \( DTM = [f_1, f_2, ..., f_m] \)
- \( CL \): classes attached to each text documents
- \( N \): the number of terms to be selected

**Output:**
- \( MAPN \): the map storing names and PVS's of the top \( N \) terms

1. **Begin**
2. For \( j = 1 \) to \( m \)
3. \( IP(j) = 0 \)
4. For \( k = 1 \) to \( j \\
5. \( IP(j) = IP(j) + P^2(c_k|f_j) \)
6. End for
7. \( MAPN[w_j] = IP(j) \cdot Var(f_j) \)
8. End for
9. Return sort(\( MAPN. decreasing = True \))[1:N].

End

**Fig.3. Algorithm of Feature Selection Using PVS**
4. Experiments

4.1. Datasets

Three datasets are used in this paper for experiments. The first dataset CNAE-9 is taken from UCI machine learning repository. It is represented as a DTM containing 1080 documents of free text business descriptions of Brazilian companies categorized into a subset of 9 classes and 856 terms. The second one DBWorld is also taken from UCI machine learning repository and contains 64 e-mails of two classes manually collected from DBWorld mailing list. It is originally represented as a DTM containing 64 text documents and 3721 terms. The last dataset TM is collected from Tencent Microblog. It is represented as a DTM containing 2557 text documents belonging to one of the 3 topics (“education”, “house”, “transportation”) and 4569 terms. Details about the three datasets are shown in Tab.1.

<table>
<thead>
<tr>
<th></th>
<th>CNAE-9</th>
<th>DBWorld</th>
<th>TM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Documents</td>
<td>1080</td>
<td>64</td>
<td>2557</td>
</tr>
<tr>
<td>Number of Terms</td>
<td>856</td>
<td>3721</td>
<td>4568</td>
</tr>
<tr>
<td>Number of Classes</td>
<td>9</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

4.2. Discriminative Terms Experiments

PVS of each term in DTM is calculated according to formula (15) and VS of each term is calculated according to formula (6). The top N terms with large VS and PVS in DTM are selected separately to compare performance of PVS and VS in text classification as N rapidly decreases. Values of N are decided as follows: N = 656, 456, 256, 156, 56 for CNAE-9, N = 721, 521, 321, 121, 21 for DBWorld and N = 569, 369, 169, 69, 49, 29 for TM.

In order to compare performance of PVS and VS in text classification, terms selected by PVS and VS are tested by classifiers. The commonly used classifiers for text classification include Naïve Bayes [70, 71], k-nearest neighbor [72, 73], neural network [73, 74], support vector machine (SVM) [71] and decision tree (DT) [71, 74]. This paper applies DT and SVM to test terms selected by PVS and VS because DT has been used either as main classification tool or as baseline classifier and SVM offers two advantages for text classification, according to Fabrizio’s paper [74].

The three datasets for experiments are separated into training sets and testing sets. The training sets are 70% of the whole datasets. DT and SVM classifiers are constructed to test the N terms selected by PVS and VS. Meanwhile, DT and SVM classifiers constructed by total number of terms in DTM are set as baselines to compare performances of PVS and VS in text classification. Unsupervised DF is applied in experiments because it can be used as a criterion to evaluate performances of other feature selection methods. At each value of N, classification experiments are repeated 10 times for both DT and SVM using randomly separated training sets and the final classification accuracy at each value of N is average accuracies of the 10 times.

Experimental results are shown in Fig.4- 6. To simplify the statements, DT using terms selected by PVS (VS, DF) are short for DT-PVS (DT- VS, DT-DF) and SVM using terms selected by PVS (VS, DF) are short for SVM-PVS (SVM-VS, SVM-DF). Discussions of the experimental results are as follows.

1) Performance of PVS

It shows in Fig.4 - Fig.6 that at each value of N, accuracies of DT-PVS and SVM-PVS are above 84% in CNAE-9 and TM. In DBWorld, accuracies of DT-PVS are still above 74% but accuracies of SVM-PVS are only above 50% at all the values of N. However, as N is reduced to 121, accuracies of both DT-PVS and SVM-PVS are above 80%. Other feature selection methods do not perform well in DBWorld either. It is a possible explanation that number of documents of DBWorld is too small for SVM to construct efficient classifiers.
Anyway, PVS performs well at low values of $N$ in the three datasets, which means when, $PVS$ is able to keep relatively high accuracies term space is reduced.

2) Performance of PVS Compared with DF

It shows in Fig.4-Fig.6 that in the three datasets, accuracies of both DT-PVS and SVM-PVS are higher than those of both DT-DF and SVM-DF, which means $PVS$ performs better than $DF$.

Generally, $PVS$ is efficient feature selection method because it shows its efficiency compared with $DF$, the typical feature selection method in text domain.

3) Performance of PVS Compared with VS

For CNAE-9, it is indicated in Fig.4 that accuracies of DT-PVS (SVM-PVS) are generally higher than those of DT-VS (SVM-VS), which indicates that $PVS$ performs better than $VS$ in CNAE-9.

For DBWorld, it is indicated in Fig.5 that accuracies of DT-PVS (SVM-PVS) are similar to those of DT-VS (SVM-VS) when $N=721, 521, 321$ but accuracies of DT-PVS (SVM-PVS) are higher than those of DT-VS (SVM-VS) when $N=121$, which indicates that in DBWorld, $PVS$ performs better than $VS$ when at a relatively low value of $N$.

For TM, it is indicated in Fig.6 that accuracies of DT-VS are higher than those of DT-PVS, which indicates $VS$ performs better than $PVS$ in DT but accuracies of SVM-PVS are generally higher than those of SVM-VS, which indicates that $PVS$ performs just better than $VS$ in SVM.

4) Summary

In summary, $PVS$ performs much better than $VS$ in CNAE-9 and $PVS$ performs better than $VS$ in DBWorld. Moreover, $PVS$ performs better than $VS$ in SVM in TM. According to discussions above, experimental results on the three datasets indicate that $PVS$ has better performances than $VS$ in text classification performed by both DT and SVM classifiers. $PVS$ is able to keep classification accuracies at low values of $N$ to guarantee efficiencies of the small term spaces.

![Classification Accuracy Graphs](image)

**Fig.4. Classification Accuracy of CNAE-9 Dataset**

4.3. More Discussions of PVS and VS

This section gives one possible explanation for experimental results that $PVS$ outperforms $VS$. Let $F$ denote the whole term space and $F^{(N)}$ denote the term space of the top $N$ terms selected by a specific feature selection method. For a specific feature selection, $CV^{(N)}$ and $CPV^{(N)}$ are used to analyze experimental results, $CV^{(N)}$ and $CPV^{(N)}$ are given by formula (17) and formula (18). To simplify the statements in the following, $F^{(N)}(PVS)$ denotes the term space of the top $N$ terms selected by $PVS$. $CPV^{(N)}(PVS)(VS)$ denotes $CPV^{(N)}$ of $PVS(VS)$. $CV^{(N)}(PVS)(VS)$ denotes $CV^{(N)}$ of $F^{(N)}(PVS)(VS)$. 

226
\[
CV_{P}^{(N)} = \frac{\sum_{j \in K} Var(P_j)}{\sum_{i=1}^{m} Var(P_i)}
\]

(17)

\[
CV_{V}^{(N)} = \frac{\sum_{j \in K} Var(f_j)}{\sum_{i=1}^{m} Var(f_i)}
\]

(18)

As mentioned in Section 4.2, \( Var(P_j) \) shows the ability of \( w_j \) to discover significant classes in \( K_j \). If \( Var(P_j) \) is large, then \( w_j \) is considered more capable of discovering significant classes. \( CPV_{V}^{(N)} \) is defined to evaluate the total ability of the top \( N \) terms to discover significant classes. Large value of \( CPV_{V}^{(N)} \) indicates high total ability of the top \( N \) terms to discover significant classes, that is, the top \( N \) terms are able to use their information efficiently. \( CV_{V}^{(N)} \) is defined to evaluate the total quantity of information of the top \( N \) terms. Large value of \( CV_{V}^{(N)} \) indicates large total quantity of information of the top \( N \) terms.

For CNAE-9 (shown in Fig.7), at each value of \( N \), \( CPV_{V}^{(N)}(PVS) > CPV_{V}^{(N)}(VS) \), which means that \( F^{(N)}(PVS) \) uses information of terms more efficiently than \( F^{(N)}(VS) \). Although \( CV_{V}^{(N)}(PVS) < CV_{V}^{(N)}(VS) \), \( F^{(N)}(VS) \) does not use the information of them as efficiently as \( F^{(N)}(PVS) \). \( CV_{V}^{(N)}(PVS) \) is relatively high and \( F^{(N)}(PVS) \) uses their information more efficiently than \( F^{(N)}(VS) \), which can possibly explain the experimental results that \( PVS \) performs much better than \( VS \) in CNAE-9.

In DBWorld (shown in Fig.8), at each value of \( N \), \( CPV_{V}^{(N)}(PVS) > CPV_{V}^{(N)}(VS) \), which indicates that \( F^{(N)}(PVS) \) uses the information of terms more efficiently than \( F^{(N)}(VS) \). \( CV_{V}^{(N)}(PVS) \ll CV_{V}^{(N)}(VS) \), but \( PVS \) still performs better than \( VS \) at a low value of \( N (N=121) \) because when \( N=121 \), \( CV_{V}^{(N)}(VS) - CV_{V}^{(N)}(PVS) \) is
relatively small, which means that effects of quantity of information is relatively low. $C^{(N)}(PVS)_{(PVS)}$ is very low but $F^{(N)}(PVS)$ still uses information of terms more efficiently than $F^{(N)}(VS)$, which can possibly explain the experimental results that $PVS$ performs just better than $VS$ in DBWorld.

In TM (shown in Fig.9), $CPV^{(N)}(PVS)$ is very small and similar to $CPV^{(N)}(VS)$ and moreover, $CV^{(N)}(VS)$ is a bit higher than $CV^{(N)}(PVS)$, which indicated that $PVS(f_j)$ is mainly determined by $Var(f_j)$. $CV^{(N)}(VS)$ is a bit higher than $CV^{(N)}(PVS)$ and $CPV^{(N)}(PVS)$ is very small, similar to $CPV^{(N)}(VS)$, which can possibly explain the experimental results that $PVS$ does not perform well in DT in TM. DT only uses part of the terms in $F^{(N)}$ to construct classifiers. Low $CV^{(N)}(PVS)$ limits the quantity of information of $PVS$ offers for DT.
5. Conclusion

This paper proposes a supervised feature selection method \( PVLS \) to improve \( VS \), for text classification. \( VS \) evaluates the quantity of information of terms by variances but it is not able show relationships between terms and classes, that is, \( VS \) is not able offer class-related information for the construction of classifiers. \( PVLS \) evaluates not only quantity of information of terms but also relationships between the term and classes. The same as \( VS \), \( PVLS \) evaluates quantity of information that terms are able to offer for text classification by variances. Bayesian Theory is the basic for \( PVLS \) to discover class-related information. Posterior probabilities are used to describe how likely a term belongs to a particular class and the ability of the term to discover significant classes which have strong relationships with the term is evaluated by impurity, the combination of posterior probabilities. Discussions in Section 3.2 demonstrate the efficiency of impurity. Experimental results indicates the efficiency of \( PVLS \) and generally, \( PVLS \) achieves better classification accuracies than \( VS \). The possible explanation of the results is that term space selected by \( PVLS \) keeps relatively large quantity of information and is able to distinguish significant classes because enough class-related information is important for construction of classifiers.

ACKNOWLEDGMENT

This research was supported by Project of National Social Sciences Foundation, Grant No. 13BTJ005, the Fundamental Research Funds for the Central Universities, Grant No. 2015GD04.

REFERENCES


