t-Test feature selection approach based on term frequency for text categorization

Deqing Wang, Hui Zhang, Rui Liu, Weifeng Lv, Datao Wang

Aimed at solving the above drawbacks, the paper focuses on how to construct a feature selection function based on term frequency, and proposes a new approach using student t-test. The t-test function is used to measure the diversity of the distributions of a term frequency between the specific category and the entire corpus. Extensive comparative experiments on two text corpora using three classifiers show that the proposed approach is comparable to the state-of-the-art feature selection methods in terms of macro-F1 and micro-F1. Especially on micro-F1, our method achieves slightly better performance on Reuters with ANNs and SVMs classifiers, compared to $\chi^2$ and IG.

1. Introduction

Text categorization/classification (TC) is to assign new unlabeled documents to the predefined thematic categories [1], which has been widely applied in many real-world applications, such as online news classification [2], large-scale web documents categorization [3], and software bug classification [4] etc. Many classification algorithms have been proposed for TC, e.g., decision trees [3], k-nearest neighbors [6], centroid-based classifier [7,8], and support vector machines (SVMs) [9] etc.

TC is different from other classification tasks because text feature space is often sparse and high-dimensional. For instance, the dimensionality of a moderate-sized text corpus can reach up to tens or hundreds of thousands. The high dimensionality of feature space will cause the “curse of dimensionality”, increase the training time, and affect the accuracy of classifiers [1,10,6,11–13]. Therefore, feature selection (FS) techniques are proposed to reduce the dimensionality under the premise of guaranteeing the performance of classifiers.

Existing feature selection methods include the removal of non-informative terms according to corpus statistics, and the construction of new features which combines lower level features into higher-level orthogonal dimensions [6,10]. In this paper, however, we mainly focus on the former, which is based on statistical theory or information theory. In this field, many classical approaches, such as chi-square statistic ($\chi^2$) [6,10], information gain (IG) [6,10], mutual information (MI) [6,10], and expected cross-entropy (ECE) [14], are widely used in TC tasks. The theoretical basis of these methods is sound, but the performances on TC tasks are different. According to the comparative experiments in Ref. [6], both $\chi^2$ and IG often achieved better accuracy than MI and document frequency (DF), and the four methods are all defined on DF. However, other authors suspected the performance of IG on skewed text corpora [15], and term frequency-based MI could achieve comparable performance to $\chi^2$ [15,16].
2. Related work

To deal with the large scale documents corpus, many feature selection approaches have been proposed. The FS methods can select more informative words, and improve the accuracy of classifiers. Besides the classical DF-based methods, many TF-based variations were also proposed [15,23,24,16] for problem specifics. Here we only give detailed definitions on four classical methods because they have been widely used and achieved better performance [6,25,10] in TC tasks. They are: $\chi^2$, IG, MI, and ECE. Formally, for the specific category $C_i$, we give the following definition as used in Refs [6,14]:

- $a$ is the number of documents $t_i$ and $C_j$ co-occur.
- $b$ is the number of documents $t_i$ occurs without $C_j$.
- $c$ is the number of documents $C_j$ occurs without $t_i$.
- $d$ is the number of documents neither $C_j$ or $t_i$ occurs.

Thus, $N$ is the number of documents in training set, and it is the sum of $a, b, c,$ and $d$.

2.1. Chi-square statistic

The $\chi^2$ statistic is used to measure the lack of independence between $t_i$ and $C_j$, and can be regarded as the $\chi^2$ distribution with one degree of freedom. It is defined to be:

$$\chi^2(t_i, C_j) = \frac{N \times (ad - bc)^2}{(a + c) \times (b + d) \times (a + b) \times (c + d)}$$ (1)

Generally, we combine the category-specific scores of a term into two alternate ways:

$$\chi^2_{avg}(t_i) = \sum_{k=1}^{m} p(C_k) \chi^2(t_i, C_k)$$ (2)

$$\chi^2_{max}(t_i) = \max_{k=1}^{m} \{\chi^2(t_i, C_k)\}$$ (3)

In real-world corpus, $\chi^2$ statistic is based, however, on several assumptions that do not hold for most textual analysis [26]. For instance, if term $t_1$ occurs in 50% documents of a specific category $C_i$ and term $t_2$ occurs in 49% documents, but the frequency of $t_2$ is much higher than that of $t_1$. Experts often think term $t_2$ should have more discriminating power than $t_1$ in the specific category $C_i$. $\chi^2$, however, will be prone to select term $t_1$ as feature, rather than $t_2$. The problem is that $\chi^2$ is not reliable for low-frequency terms [26].

2.2. Mutual information

Mutual information [6] measures the mutual dependence of the two random variables. The formula of MI is defined to be:

$$I(t_i, C_j) = \log \frac{p(t_i, C_j)}{p(t_i) \times p(C_j)}$$ (4)

where $p(t_i, C_j)$ is estimated by $a/N$, and $p(t_i) = (a + b)/N$, $p(C_j) = (a + c)/N$. Then we substitute them into Eq. 4, and we obtain:

$$I(t_i, C_j) = \log \frac{a \times N}{(a + c) \times (a + b)}$$ (5)

if term $t_i$ is independent to category $C_j$, then $I(t_i, C_j) = 0$.

Just like $\chi^2$, The score of term $t_i$ in text collection is obtained by averaged or maximized the category-specific scores.

$$I_{avg}(t_i) = \sum_{k=1}^{m} p(C_k) I(t_i, C_k)$$ (6)
\[ I_{\text{max}}(t_i) = \max_{k=1}^{m} \{ I(t_i, C_k) \} \]  \hspace{1cm} (7)

The weakness of MI is that the score is strongly influenced by the marginal probabilities of terms, because rare terms will have a higher score than common terms. Therefore, the scores are not comparable across terms of widely differing frequency [6]. Previous work shows that DF-based MI has relatively poor performance in TC tasks [6,23,25]. But the improved TF-based MI can achieve comparable performance to \( \chi^2 \) [27,16].

2.3. Information gain

IG was firstly used as attribute selection measure in decision tree ID3 [5]. This measure is from entropy in information theory, which studies the value or “information content” of messages. The IG of term \( t_i \) in multi-class text corpus [6] is given by

\[ IG(t_i) = -\sum_{k=1}^{m} p(C_k) \log p(C_k) + p(t_i) \sum_{k=1}^{m} p(C_k | t_i) \log \frac{p(C_k | t_i)}{p(C_k)} + p(t_i) \sum_{k=1}^{m} p(C_k | t_i) \log p(C_k | t_i) \]  \hspace{1cm} (8)

where \( p(C_k) = (a + c)/N \), \( p(t_i) = (a + b)/N \), \( p(C_k | t_i) = (c + d)/N \), \( p(C_k | t_i) = a/(a + b) \), \( p(C_k | t_i) = c/(c + d) \).

IG is defined as the difference between the original information requirement (i.e., based on just the proportion of classes) and the new requirement (i.e., obtained after partitioning on term \( t_i \)). IG is also called average mutual information [28]. The weakness of IG method is that it prefers to select terms distributed in many categories, but these terms have less discriminating power in TC tasks.

2.4. Expected cross-entropy

ECE only considers the terms occurred in a document and ignores the absent terms [14]. It can be defined to be:

\[ ECE(t_i) = p(t_i) \sum_{k=1}^{m} p(C_k | t_i) \log \frac{p(C_k | t_i)}{p(C_k)} \]  \hspace{1cm} (9)

where \( p(C_k | t_i) = a/(a + b) \), \( p(C_k) = (a + c)/N \). We can observe that Eq. 9 can be rewritten as follows:

\[ ECE(t_i) = \sum_{k=1}^{m} p(t_i, C_k) I(t_i, C_k) \]  \hspace{1cm} (10)

the factor \( p(t_i, C_k) \) is introduced to decrease the influence by the marginal probabilities of terms in MI method.

Besides the classical methods, many improved methods have been proposed. For example, Schneider [16] proposed the Weighted Average Pointwise Mutual Information (WAPMI) to conquer the drawbacks of MI. However, WAPMI destroyed the theoretical basis of MI, and its “optimal” performance was obtained by heuristic experiments because of the introduced parameter \( \lambda \). Yang et al. [23] only considered the terms whose relative term frequency was larger than a predefined threshold \( \lambda \), and then modified the IG formula to select features. Forman [25] proposed the Bi-Normal Separation (BNS) method, which used the standard Normal distribution’s inverse cumulative probability function to construct feature selection function. According to the authors’ experiments on binary classification tasks, the BNS performed worse precision than IG, and it is difficult to apply into multi-class TC tasks. Uguz [29] proposed a two-stage FS method by combining IG, principal component analysis and genetic algorithm. More and more new FS methods have been generated, such as, mr2PSO [30] etc.

In next section, we will propose a new approach based on term frequency, and it can capture the information of high-frequency terms.

3. New approach based on term frequency and \( t \)-test

The \( t \)-test, namely the student \( t \)-test, is often used to assess whether the means of two classes are statistically different from each other by calculating a ratio between the difference of two class means and the variability of the two classes [31,19]. In this section, we explain why the averaged term frequency within a single category or within the whole corpus is approximately normal using Lindeberg–Levy central limit theorems, and then how the \( t \)-test is constructed based on the averaged term frequencies.

Let us consider the term frequency in a corpus consisting of \( N \) documents. Given a vocabulary \( V \), the term frequency (\( tf_{ij} \)) of a term \( t_i \) in the \( j \)th document \((1 \leq j \leq N)\) is considered as a random variable, which subjects to some distribution, e.g., multinomial model [27]. In the multinomial model, a document is a sequence of word events drawn from the same vocabulary \( V \), and the probability of each word event in a document is independent of the word’s context and position in the document. Therefore, each document \( d_j \) is drawn from a multinomial distribution of words with as many independent trials [27]. That is, the occurrence of one term in each document is dominated by a multinomial function. Then,

- For a specific term \( t_i \), let \( \{ tf_{i1}, \ldots, tf_{in} \} \) be a random sample of size \( N \), where \( N \) is the number of documents in the collection \( D \), and \( tf_{ij}(1 \leq j \leq N) \) is the term frequency of \( t_i \) in document \( d_j \). That is, a sequence of independent random variables.
- Let
  \[ \overline{tf_i} = \frac{1}{N} \sum_{j=1}^{N} tf_{ij} \]  \hspace{1cm} (11)
  be the sample average of \( t_i \) in the collection \( D \).
- And let
  \[ \overline{\overline{tf}_{ik}} = \frac{1}{N_k} \sum_{j=1}^{N_k} tf_{ijl} | d_j \in C_k, \]  \hspace{1cm} (12)
  be the average of term \( t_i \) within the single category \( C_k \), where \( l(d_j, C_k) \) is an indicator to discriminate whether document \( d_j \) belongs to \( C_k \), and \( N_k \) is the total number of documents in class \( k \).

According to Lindeberg–Feller central limit theorems (CLT) [32–34], \( \overline{tf_i} \) is approximately normal with mean \( \mu_{t_i} \) and variance \( \frac{1}{N} \sigma^2_i \), denoted as \( N(\mu_{t_i}, \frac{1}{N} \sigma^2_i) \); And \( \overline{\overline{tf}_{ik}} \) is approximately normal with mean \( \mu_{t_i} \) and variance \( \frac{1}{N_k} \sigma^2_i \), denoted as \( N(\mu_{t_i}, \frac{1}{N_k} \sigma^2_i) \). It is worthy noting that we assume the deviation of each variable is the same. Then we know that \( \overline{\overline{tf}_{ik}} - \overline{tf_i} \) is also approximately normal distributed with mean \( \mu_{t_i} - \mu_{t_j} \) and variance \( \sigma^2_i \). The variance is induced as follows:

\[ \text{Var}(\overline{\overline{tf}_{ik}} - \overline{tf_i}) = \frac{(N-N_k)^2 \times N_k \times \sigma^2_i}{N^2} + \frac{(N-N_k) \times \sigma^2_i}{N_k} \]
\[ = \left(1 + \frac{1}{N_k} \right) \times \sigma^2_i \]  \hspace{1cm} (13)

Besides, we define the pooled variance as follows:

\[ s^2 = \frac{1}{N-K} \sum_{j=1}^{K} \sum_{k \in C_k} (tf_{ij} - \overline{tf}_{ik})^2 \]  \hspace{1cm} (14)
According to the definition of the t-test [35], we construct the following formula:

$$t - \text{test}(t, C_k) = \frac{\bar{t}_k - \bar{t}_l}{m_k \cdot s_i}$$  \hspace{1cm} \text{(15)}$$

where $s_i$ is pooled standard deviation, and $m_k = \sqrt{\frac{1}{m_k} - 1}\pi$.

The **NULL hypothesis** of Eq. 15 is that the means of two samples are equal, i.e., $\mu_k = \mu_l$. So the Eq. 15 is used to measure whether the means of the two samples (i.e., $\bar{t}_k$ and $\bar{t}_l$) have the statistically significant difference. Therefore, the bigger the value of $t - \text{test}(t, C_k)$ is, the larger the difference of the means is. For some threshold $\theta$, if the $t - \text{test}(t, C_k) < \theta$, it implies that the averaged frequency of term $t$, within the specific category $C_k$ has the same or similar mean with that within the entire corpus; Otherwise, it implies the averaged frequency of term $t$ in the specific category $C_k$ is significantly different from that in the entire corpus, and the term has more discriminating power for the specific category $C_k$. Compared with the average of term frequency in the entire corpus, the term $t_i$ occurred many or few times in $C_k$ can be considered as the feature of category $C_k$.

We combine the category-specific scores of a term into two alternate ways:

$$t - \text{test}_{av}(t_i) = \sum_{k=1}^{K} t - \text{test}(t_i, C_k)$$  \hspace{1cm} \text{(16)}$$

$$t - \text{test}_{max}(t_i) = \max_{k=1}^{K}(t - \text{test}(t_i, C_k))$$  \hspace{1cm} \text{(17)}$$

The Pseudocode of t-test algorithm is as follows:

**Algorithm 1.** The student T-test Feature Selection Algorithm

**Input:**
- $D$: The training set, $m$: The number of selected features
- Boolean: true for “averaged” or false for “maximized”

**Output:** $F$: The top m features in $D$.

**Procedure:** t-test ($D, m$, Boolean)

1. **Init:** $\bar{t}_k = 0$, $\bar{t}_l = 0$, $t - \text{test}_{av}(t) = 0$ and $t - \text{test}_{max}(t) = 0$
2. for each document $d_j \in D$ do
   3. for each word $w_i \in d_j$ do
      4. $\bar{t}_k^2 = t_f_j/|D|$; $\bar{t}_l^2 = t_f_j + l(d_j, C_k)/N_k$; // Eqs. 11 & Eq. 12
   5. end for
3. end for
4. for each document $d_j \in D$ do
   5. for each word $w_i \in d_j$ do
      6. $s_i^2 = \frac{1}{|D|} l(d_j, C_k) \cdot (t_f_j - \bar{t}_k)^2$; // according to Eq. 14
   6. end for
7. end for
8. for each word $w_i$ in class $k$ do
   9. $t - \text{test}(w_i, C_k) = \frac{|\bar{t}_k - \bar{t}_l|}{m_k \cdot s_i}$; // according to Eq. 15
10. end for
11. if Boolean == true
12. $t - \text{test}_{av}(t_i) = t - \text{test}(t_i, C_k)/K$
13. else
14. $t - \text{test}_{max}(t_i) = \max(t - \text{test}_{max}(t_i), t - \text{test}(t_i, C_k))$
15. end if
16. $F = \text{select Top Features} (m)$; // sort $t - \text{test}(t)$ descendingly and select in terms
17. return $F$

### 4. Experimental setup

#### 4.1. Data sets

**Reuters-21578**: The Reuters corpus is a widely used benchmark collection [26,25,6,23,8]. According to the ModApte split, we get a collection of 52 categories after removing unlabeled documents and documents with more than one class label, which contains 6532 training documents and 2568 test documents, respectively. Note that Reuters-21578 is a very skewed data set, in which the majority category (earn) accounts for 43% of the whole training instances, whereas the top-ten minority categories only contain several training instances in each category. Altogether 319 stop words, punctuation and numbers are removed. All letters are converted into lowercase, and the word stemming is applied. The final vocabulary has 22,411 words.

**20Newsgroup**: The Newsgroup corpus is also a widely used benchmark [26,25,6,8], and consists of 18,828 documents, which are uniformly distributed in twenty categories. We randomly select 33% instances from each category as testing instances and the rest as training instances. We only keep the information of “Subject”, “Keyword” and “Content”. Other information, such as “Path”, “From”, “Message-ID”, “Sender”, “Organization”, “References”, “Date”, “Lines”, and email addresses, are filtered out. The stop words list has 823 words, and we filter words containing non-characters. All letters are converted into lowercase and word stemming is applied. The total number of unique terms is 203,279.

Documents are typically represented by vector space model. That is, the content of a text is represented by a vector in the term space, i.e., $d = \langle w_1, \ldots, w_m \rangle$, where $w_i$ is the weight of term $t_i$, which is calculated by tf-idf [36] and normalized to have one unit length, and $M$ is the size of the term set.

#### 4.2. Classifiers

In our experiments, we choose three well-established classifiers for the comparison purpose. They are: support vector machines (SVMs) [9,37,38,7], weighted nearest neighbor classifier (kNN) [6], and classic centroid-based classifier (CC) [7]. SVMs is not affected by different feature number distinctly, we select SVMs because we want to guarantee that our method will not reduce the performance of SVMs. In fact, our method improves the performance of SVMs significantly on macro-$F_1$ in the following experiments. The SVMs implementation we use is LIBSVM [39] with linear kernels. For $k$NN and CC, the feature number often affects their performances, and they achieve better performance with fewer features. The two classifiers are coded by ourselves in Java language. For $k$NN, we set $k = 10$ for each experiment [6]. The similarity measure we use for the classifiers is the cosine function. Unless otherwise specified, we use the default parameter values for each classifier in our experiments.

#### 4.3. Performance measures

We measure the effectiveness of classifiers in terms of the combination of precision ($p$) and recall ($r$) measures widely used for text categorization. That is, we use the well-known $F_1$ function [40,1] as follows:

$$F_1 = \frac{2pr}{p + r}$$  \hspace{1cm} \text{(18)}$$

1 Available on http://ronaldo.cs.tcd.ie/esslli07/sw/step01.tgz.
For multi-class text categorization, $F_1$ is usually estimated in two ways, i.e., the macro-averaged $F_1$ (macro-$F_1$) and the micro-averaged $F_1$ (micro-$F_1$) [1], as the following:

$$\text{macro-}F_1 = \frac{\sum_{i=1}^{K} F_1(i)}{K}$$  \hspace{1cm} (19)

$$\text{micro-}F_1 = \frac{2pr}{p+r}$$  \hspace{1cm} (20)

where $F_1(i)$ is the $F_1$ value of the predicted $i$th class, and $p$ and $r$ are the precision and recall values across all classes, respectively.

In general, macro-$F_1$ gives the same weight to all categories, thus can be greatly influenced by poor results of rare categories. In contrast, micro-$F_1$ gives the same weight to each instance, which can be dominated by the performance of common or majority categories. As a result, for text data sets of skewed categories, macro-$F_1$ and micro-$F_1$ may give quite different evaluations.

5. Results

Firstly, we show one case study of t-test in real-world corpus. And then we show the performance of t-test on two corpora with three classifiers. For Reuters-21,578 corpus, the number of feature space is all (22,411), 17,000, 15,000, 13,000, 11,000, 10,000, 8000, 6000, 4000, and 2000, respectively, accounting to ten groups of data sets. On 20 Newsgroup corpus, the original feature space reaches up to 210 thousand and we only select less terms as features in order to save the training time. The dimensionality of feature space is all (216,252), 2000, 1500, 1000, 500, and 200, respectively, accounting to six groups of data sets.

Table 1

<table>
<thead>
<tr>
<th>Term</th>
<th>Acquire</th>
<th>Stake</th>
<th>Payout</th>
<th>Dividend</th>
</tr>
</thead>
<tbody>
<tr>
<td>t-test</td>
<td>28.053</td>
<td>22.567</td>
<td>3.272</td>
<td>17.796</td>
</tr>
<tr>
<td>$\chi^2$</td>
<td>479.482</td>
<td>270.484</td>
<td>131.104</td>
<td>344.045</td>
</tr>
<tr>
<td>MI</td>
<td>0.078</td>
<td>0.042</td>
<td>0.009</td>
<td>0.036</td>
</tr>
<tr>
<td>ECE</td>
<td>0.084</td>
<td>0.050</td>
<td>0.028</td>
<td>0.060</td>
</tr>
<tr>
<td>DF</td>
<td>576</td>
<td>359</td>
<td>224</td>
<td>563</td>
</tr>
<tr>
<td>TF</td>
<td>749</td>
<td>646</td>
<td>232</td>
<td>903</td>
</tr>
</tbody>
</table>

Fig. 1. The comparative curves of five feature selection methods with kNN on skewed Reuters-21578 corpus in terms of macro-$F_1$ (a) and micro-$F_1$ (b).
5.1. Case study

Table 1 lists the scores of seven different feature selection functions for the selected four terms in category "acq" from the real-life corpus, i.e., Reuters-21578. Based on the literal meaning, the first two terms, i.e., “acquir” and “stake”, are closely related to the content of category “acq”, while the last two terms, i.e., “payout” and “dividend”, belong to other category. However, according to the χ², ECE, DF and TF methods, we wrongly select “acquir” and “dividend” as the features of category “acq”, whereas t-test, IG and MI select the features correctly.

Then we compare the top 10 keywords selected by six methods from Reuters-21578. The results show that, χ², IG, and ECE select the similar keywords. Compared to χ² and DF, our t-test method only has 4 and 6 same keywords, respectively. It is worth noting that our method is prone to choosing the keywords whose distribution of term frequency among category is significantly different. For instance, the DF and TF of “degussa” is 2 and 8, respectively, and it only occurs in one category. The rest methods will not select the low-frequency term, however, our t-test select it as feature. On the contrary, the TF values of “price” and “billion” are 1987 and 2837, respectively, they are not chosen as features by the rest methods because of their DF values, which are 835 and 902, respectively. We also observe that IG and ECE select 8 same words, and the features of MI are different with that of the remaining methods, which may explain the reason why it achieves the worse performance.

5.2. Performance evaluations

For χ², MI, and t-test methods, we tested the two alternative combinations, i.e., averaged and maximized ways. We observed that the averaged way was always better than the maximized way for multi-classes problem. Thus we only report the best results of three methods.

5.2.1. Performance of t-test with kNN classifier

The macro-F1 and micro-F1 of five feature selection methods with kNN classifier on imbalanced Reuters-21578 corpus are shown in Fig. 1(a), Fig. 1(b), respectively. It is clear that t-test, χ², and ECE achieve evidently better performance than MI and IG in terms of macro-F1. However, the diversity among the three methods is small. As shown in Fig. 1(a), when the number of feature space is larger than 13,000, χ², and ECE is a little better than t-test; however, when the number of features falls in [8000, 13,000], t-test performs the best macro-F1.

The micro-F1 of five feature selection methods increases as the number of features decreases, as shown in Fig. 1(b). It demonstrates that kNN classifier often obtains better performance with less features. Our t-test method performs consistently the best in distinct feature dimensionality, and the highest micro-F1 of t-test is 89.8% when the number of features is 4000, which improves up to 4.2% than χ². IG method achieves the worst performance in the all experiments on skewed corpus with kNN classifier.

As shown in Fig. 1(a) and and Fig. 1(b), for unbalanced multi-class TC tasks, we find IG is inferior to MI in terms of both macro-F1 and micro-F1, whereas IG is superior to MI for binary classification tasks according to the comparative experiments of Yang et al. [6]. The conflict shows that feature selection methods depends on the practical classification problem.

Because macro-F1 on balanced corpus is very close to micro-F1, we only show the comparative curves in terms of micro-F1 on 20 Newsgroup corpus. As shown in Fig. 2, the micro-F1 of both χ² and IG are slightly better than our t-test method, and the four methods are obviously better than MI method. Especially, the performance of IG method is comparable to χ², and ECE methods on balanced corpus.

5.2.2. Performance of t-test with SVMs classifier

Fig. 3(a) and and Fig. 3(b) depict the macro-F1 and micro-F1 performance of different feature selection methods on the Reuters-21578 corpus using SVMs. The t-test, χ², and ECE methods perform similar performances, which are better than IG and MI methods. Meanwhile, the macro-F1 scores of three methods increase as the number of features reduces. It is worth noting that MI does better than other methods when the number of features is in [15,000, 22,411], and then MI falls dramatically.

The performance of these methods in terms of micro-F1 measure on Reuters-21578 corpus is shown in Fig. 3(b). The micro-F1 points of different feature selection methods show a tendency to increase as the number of the features decreases. However, these methods show consistent performance in micro-F1, and the t-test method is still the best among these methods.

Fig. 4 depicts the micro-F1 performance of different feature selection methods on the 20 Newsgroups corpus using SVM. The trends of the curves are similar to those in Fig. 2. The t-test, χ², IG, and ECE methods achieve similar performances, which are
Fig. 3. The macro-$F_1$ (a) and micro-$F_1$ (b) of different methods on Reuters-21578 using SVMs.

Fig. 4. The micro-$F_1$ of different methods on 20 Newsgroup using SVMs.
better than the MI method. Our t-test is the best among these methods, except the last point (200 features are used).

5.2.3. Performance of t-test with centroid-based classifier

For centroid-based classifier, the macro-$F_1$ of five feature selection methods is shown in Fig. 5(a). We can observe that $\chi^2$, ECE, and t-test do better than MI and IG methods, and $\chi^2$ is slightly better than ECE and t-test. The same conclusion can be done in terms of micro-$F_1$, as shown in Fig. 5(b).

Meanwhile, our t-test is comparable to the $\chi^2$, ECE, and IG methods on 20 Newsgroup corpus, and the four methods outperform the MI method significantly, as shown in Fig. 6.

5.3. Discussion

Our t-test method is based on term frequency, rather than document frequency, which can overcome “the drawback of low-frequency terms” of traditional feature selection methods. Meanwhile, it can capture the relation between term frequency and category topics. Through extensive experiments on two common text corpora with three well-established classifiers (i.e., kNN, SVMs and Centroid-based), their relationships with algorithms and data corpora can be summarized as follows:

- Our t-test method considers each term frequency as a random variable, that is, the occurrence of a term in a document subjects to some distribution. Using central limit theorems and within-class deviation, we construct a t-test between the distribution of a term frequency in specific category and that in the entire collection, and use it to select features.
- Through extensive experiments on two common text corpora with three well-established classifiers, we observe that our t-test method can achieve comparable performance to the state-of-the-art $\chi^2$ in terms of macro-$F_1$, and t-test is slightly better than $\chi^2$ and ECE in terms of micro-$F_1$ on the Reuters data set when kNN or SVMs is used. Meanwhile, it is significantly better than IG and MI methods on imbalanced corpus. On balanced corpus, our t-test method is comparable to the state-of-the-art $\chi^2$, IG, and ECE.
- Differing from the results in Ref. [6], the IG method achieves worse performance on skewed Reuters-21578 corpus, which is even inferior to the MI method. The same conclusion can be found in Ref. [15]. However, the performance of IG is compara-

![Fig. 5. The macro-$F_1$ (a) and micro-$F_1$ (b) of five methods on Reuters-21578 using centroid-based classifier.](image-url)
ble to t-test, \( \chi^2 \), and ECE on the balanced corpora. The reason could be that the performance of IG depends on data corpus, which has been explained in Ref. [6].

We should point out that the observations and conclusions above are made in combination with linear SVM and/or kNN algorithms in terms of micro-\( F_1 \) and/or macro-\( F_1 \) and other controlled experimental settings. It will be interesting to verify if we can observe the similar results on a more general learning algorithm, such as Naive bayes method, or by using other performance measure.

6. Conclusion and future work

Term frequency often impacts the topics of corpus, so we propose a t-test feature selection approach based on term frequency. The student t-test is used to assess whether the averaged term frequencies of a term between two classes are statistically different from each other by calculating a ratio between the difference of the two class means and the variability of the two classes. Then we will verify our approach on more text collections and classification algorithms.

Acknowledgement

This research was supported by the fund of the State Key Laboratory of Software Development Environment (No. SKLSDE-2013ZX-36).

References