Classification of Small Datasets: Why Using Class-Based Weighting Measures?

Flavien Bouillot[1,2], Pascal Poncelet[1], and Mathieu Roche[1,3]

[1] LIRMM, Univ. Montpellier 2, CNRS – France
[2] ITESOFT, Aimargues – France
[3] TETIS, Cirad, Irstea, AgroParisTech – France
{firstname.lastname}@lirmm.fr
http://www.lirmm.fr/

Abstract. In text classification, providing an efficient classifier even if the number of documents involved in the learning step is small remains an important issue. In this paper we evaluate the performance of traditional classification methods to better evaluate their limitation in the learning phase when dealing with small amount of documents. We thus propose a new way for weighting features which are used for classifying. These features have been integrated in two well known classifiers: Class-Feature-Centroid and Naïve Bayes, and evaluations have been performed on two real datasets. We have also investigated the influence on parameters such as number of classes, documents or words in the classification. Experiments have shown the efficiency of our proposal relatively to state of the art classification methods. Either with a very few amount of data or with a small number of features that can be extracted from poor content documents, we show that our approach performs well.

1 Introduction

Classification of documents is a topic addressed for a long time. Basically the problem can be summarized as follows: *How to efficiently assign a document to one or more classes according to its content?* Usually best results are obtained with an important number of examples (i.e. documents) and features (i.e. words) in order to build an efficient classification model.

However, more and more we need to provide a classifier even if the number of features is quite small [1]. For example, with the development of social networks, we need to use tools that classify tweets exchanged every days or every hours. Here we have to deal not only with the rapid rate but also with the poor content of exchanged texts (i.e. 140 characters). In this context the extraction of relevant and discriminative features represents a challenging issue. Another quite opportunity of applying classification on small number of documents is when the number of labeled documents is itself small. Basically labeling documents is a very time consuming process, requiring lots of efforts.

Recently, new approaches based on semi-supervised or active learning methods try to start a first classification over the small number of labeled documents and ask to the user to validate or not the model in a dynamic way (e.g. [2, 3]). For instance, in *Semi-Supervised Learning and Active Learning* approaches, they require a few number

of labeled documents and a huge number of unlabeled documents in order to improve the model. They must deal with the problem of small amount of data in the first steps of the classification. In the same way, *Real Time learning*, i.e. in a data stream context, have only a few training examples to start building a classifier.

For text classification algorithms, the more the number of learning data is, the better classification results are. And obviously, the classification performance decreases with a reduced training data set. The main contribution of this paper consists in proposing a new way for weighting features which are used for classifying. These features have been integrated in two well known classifiers: Class-Feature-Centroid and Naïve Bayes.

The remainder of this paper is organized as follows. In Section 2 we present some weighting methods and discuss about their efficiency in our context. The new way for weighting features which are TF-IDF-based is presented in Section 3. We present how they have been integrated in two classification approaches in Section 4. Conducted experiments on two real datasets have been compared with traditional classification approaches and are described in Section 5. Finally, Section 6 concludes and presents future work.

2 Weighting Measures

The main goal of the classification is to assign a document to one or more classes according to the terms used in the document. Let $C = C_1, C_2, ..., C_n$ be a set of n classes and $D = d_1, d_2, ..., d_m$ a set of m documents. In the learning phase, each document is attached to one class and we note $D_{i,j}$ the i^{th} document of the class j and $D_j = d_{1,j}, d_{2,j}, d_{3,j}$ the set of documents of the class j.
Texts are commonly represented in the *bag-of-words* model where all documents form a lexicon. $L = t_1, t_2, ..., t_{|L|}$ is a lexicon containing $|L|$ terms where t_i ($1 \leq i \leq |L|$) is a unique term in the lexicon. Each document is then represented by a weighted vector of terms without considering their position in the document. $\overrightarrow{D_j} = \{w_{1j}, w_{2j}, ..., w_{|L|j}\}$ is a vector representation of the document j where w_{ij} is a weighting factor (e.g. Frequency, Boolean, TF-IDF...) of the term t_i for the document j.

Traditionally, the TF-IDF measure gives greater weight to specific terms of a document [4]. TF-IDF is a weighting measure which has been proved to be well appropriate for text classification. It is obtained as follows. In a first step, the frequency of a term (*Term Frequency*) corresponding to its occurrences in the document is computed. It is called the *inner-document weight* of a term. Thus, for the document d_j and the term t_i, the frequency of t_i in d_j is given by the following equation:

$$TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

where $n_{i,j}$ stands for the number of occurrences of the term t_i in d_j. The denominator is the number of occurrences of all terms in the document d_j.

The *Inverse Document Frequency* (IDF) measures the importance of the term in the corpus. It is the *inter-documents weight* of a term, obtained by computing the logarithm

of the inverse of the proportion of documents in the corpus containing the term. It is defined as follows:

$$IDF_i = log_2 \frac{|D|}{|\{d_j : t_i \in d_j\}|}$$

where $|D|$ stands for the total number of documents in the corpus and $|\{d_j : t_i \in d_j\}|$ is the number of documents having the term t_i. Finally, the $TD\text{-}IDF$ is obtained by multiplying inner-document weight and inter-documents weight as follows:

$$TF - IDF_{i,j} = TF_{i,j} \times IDF_i$$

Traditionally the $TF\text{-}IDF$ measure is used to evaluate the weight of a term within a document and does not take into account the weight of a term by considering the class of the document rather than the document itself. The weight of a term *i* for the class *j*, called w_{ij}, depends both on both inner-class term weight and inter-classes term weight. The *inner-class weight* measures the importance of a term within a class (Is this term representative of the class *j*?) while the *inter-classes weight* measures is used to evaluate if a term is discriminative relatively to other classes. Such weighting measures are, for instance, presented in [5] and [6] but they suffer the following drawbacks:

- When two classes are semantically close, they usually consider a term as representative even if it is used only once. In other words, the number of occurrences of a term in a class is not considered.
- When the class is composed of unbalanced documents (i.e. long vs. short documents for instance), they tend to give a higher weight to terms occurring in the longest documents rather than the shortest ones.

3 New Measures for Inner-class and Inter-classes Weighting

In order to weight terms in a class, we propose new inner-class and new inter-classes weighting measures which are based on $TF\text{-}IDF$.

3.1 Inner-class Measures: $inner\text{-}weight^{Tf}$ and $inner\text{-}weight^{Df}$

First we propose a weighting measure based on the term-frequency as described in the previous section. This measure is called $inner\text{-}weight^{Tf}$ and, for a term t_i in class j, $inner\text{-}weight_{ij}^{Tf}$ is obtained as follows:

$$inner\text{-}weight_{ij}^{Tf} = \frac{TF_{ti}^{j}}{|n_j|} \tag{1}$$

where TF_{ti}^{j} stands for the number of terms t_i in C_j and $|n_j|$ is the number of terms in C_j.

$Inner\text{-}weight^{Tf}$ has the same limits than those previously described when considering unbalanced documents within the class. We thus propose another weighting measure that focuses on the document.

In the following, we assume that: the most frequent term in a class is not the most representative term for this class. Now, we consider Document Frequency instead of Term Frequency. The $inner\text{-}weight^{Df}$ for a term t_i in class j is thus defined as follows:

$$inner\text{-}weight_{ij}^{Df} = \frac{DF_{ti}^{j}}{|d_j|} \tag{2}$$

where DF_{ti}^{j} is the number of documents containing t_i in C_j and $|d_j|$ is the number of documents in C_j.

Example 1. Let C_0, C_1 and C_2 be three classes. Each class is composed by three documents called respectively $d_{j,1}$, $d_{j,2}$ and $d_{j,3}$ where j refer to the class (i.e. $d_{0,1}$ refers to the document d_1 in class C_0). Each document is composed by several terms called respectively Term A, Term B and $Others$.

Class	Document	Terms A	Terms B	Others
C_0	$d_{0,1}$	4	1	10
	$d_{0,2}$	2	2	10
	$d_{0,3}$	0	1	10
C_1	$d_{1,1}$	1	3	10
	$d_{1,2}$	3	0	10
	$d_{1,3}$	2	0	10
C_2	$d_{2,1}$	0	0	10
	$d_{2,2}$	0	3	10
	$d_{2,3}$	0	0	10

In the following, we focus on the weighting factor of the terms A and B only for the class C_0. So first we compute $inner\text{-}weight_{i0}^{Tf}$ and $inner\text{-}weight_{i0}^{Df}$ for terms A and B.

Class	Document	Terms A	Terms B	Others
C_0	$d_{0,1}$	4	1	10
	$d_{0,2}$	2	2	10
	$d_{0,3}$	0	1	10
$inner\text{-}weightsComputation$				
$inner\text{-}weight_{i0}^{Tf}$	$\frac{TF_{ti}^{j}}{\lvert n_0 \rvert}$	$\frac{6}{40} = 0,15$	$\frac{4}{40} = 0,10$	
$inner\text{-}weight_{i0}^{Df}$	$\frac{DF_{ti}^{j}}{\lvert d_0 \rvert}$	$\frac{2}{3} = 0,66$	$\frac{3}{3} = 1$	

3.2 Inter-classes Measures: $inter\text{-}weight^{class}$ and $inter\text{-}weight^{doc}$

The $inter\text{-}weight^{class}$ measures consider the number of classes containing a term. It is different from traditional approaches that focus on the number of documents and is obtained by:

$$inter\text{-}weight_{ij}^{class} = log_2(1 + \frac{|C|}{C_{ti}}) \tag{3}$$

where $|C|$ is the number of classes and C_{ti} is the number of classes containing the term t_i.

Considering only the presence or the absence of a term in a class might be too restrictive when:

- there are very few classes. The less the number of classes, the more important the inter-classes influence in the global weighting is.
- there are semantically close classes. Semantically close classes result in a high number of common terms between classes.
- there are a huge number of terms in classes (due to very long documents or a large number of documents). The higher the number of terms by class, the higher the probability to have a term appearing at least once in a class is.

We thus propose, as in the inner-class measure, to consider documents instead of classes. However here documents are documents within other classes. Otherwise, by taking into account all the documents, terms which are very frequent and discriminative of one class are underestimated.

So we define $inter\text{-}weight^{doc}$ as follows:

$$inter\text{-}weight_{ij}^{doc} = log_2(\frac{|d| - |d \in C_j| + 1}{|d : t_i| - |d : t_i \in C_j| + 1}) \tag{4}$$

where $|d|$ stands for the number of documents in all classes; $|d \in C_j|$ the number of documents in C_j; $|d : t_i|$: the number of documents in all classes containing the term t_i and $|d : t_i \in C_j|$ is the number of documents in C_j containing the term t_i. Adding the number of documents in all classes prevent the case where t_i is only used in C_j (when $|d : t_i| - |d : t_i \in C_j| = 0$).

Example 2. Let us compute $inter\text{-}weight_{i0}^{Class}$ and $inter\text{-}weight_{i0}^{Doc}$ for terms A and B.

Class	Document	Terms A	Terms B	Others
C_0	$d_{0,1}$	4	1	10
	$d_{0,2}$	2	2	10
	$d_{0,3}$	0	1	10
C_1	$d_{1,1}$	1	3	10
	$d_{1,2}$	3	0	10
	$d_{1,3}$	2	0	10
C_2	$d_{2,1}$	0	0	10
	$d_{2,2}$	0	3	10
	$d_{2,3}$	0	0	10
$inter\text{-}weightsComputation$				
$inter\text{-}weight_{i0}^{class}$	$\log_2(\frac{\lvert C\rvert}{C_{t_i}})$	$\log_2(\frac{3}{2}) = 0,58$	$\log_2(\frac{3}{3}) = 0$	
$inter\text{-}weight_{i0}^{doc}$	$\log_2(\frac{\lvert d\rvert - \lvert d \in C_0\rvert + 1}{\lvert d:t_i\rvert - \lvert d:t_i \in C_0\rvert + 1})$	$\log_2(\frac{7}{4}) = 0,81$	$\log_2(\frac{7}{3}) = 1,22$	

4 Integration of the New Measures in Classification Algorithms

Usually SVM (Support Vector Machine) and Naïve Bayes are recognized as two of the most effective text classification methods. Nevertheless they are not well adapted to

small learning datasets [7]. Sometimes, for instance, they require complicated adaptation such as the definition of a new kernel for SVM [8]. Our new weighting measures can be used in Naïve Bayes [9] and Class-Feature-Centroid [5] approaches which offer the following advantages: (i) they have been recognized as very efficient for text classification; (ii) based on weighted features, they can easily be modified; (iii) finally the obtained models are easy to interpret for users.

First, for each class of the learning set, we compute $C_j = \{w_{1j}, w_{2j}, ..., w_{|L|j}\}$ for different combinations of inner- and inter-classes weights:

- $w_{ij}^{Tf-Class}$=$inner\text{-}weight^{Tf} \times inter\text{-}weight^{class}$
- $w_{ij}^{Df-Class}$=$inner\text{-}weight^{Df} \times inter\text{-}weight^{class}$
- w_{ij}^{Tf-Doc}=$inner\text{-}weight^{Tf} \times inter\text{-}weight^{doc}$
- w_{ij}^{Df-Doc}=$inner\text{-}weight^{Df} \times inter\text{-}weight^{doc}$

Example 3. For example with our running example, for term A and class C_0, we have: $w_{A0}^{Tf-Class}$=$inner\text{-}weight_{A0}^{Tf} \times inter\text{-}weight_{A0}^{class} = 0,15 \times 0,58 = 0,09$

The integration of our measures in Naïve Bayes is done as follows. After computing $C_j = \{w_{1j}, w_{2j}, ..., w_{|L|j}\}$ where $w_{i,j}$ is the weight of the i^{th} term in the class C_j, we estimate the probability that an unlabeled document d belongs to a class C_j : $P(d \in C_j) = P(C_j) \prod_i (w_{i,j})$. Experiments combining $w_{ij}^{Tf-Class}$ weighting method and Naïve Bayes approach is called $Nb^{Tf-Class}$ in the following (resp $Nb^{Df-Class}$, Nb^{Tf-Doc}, and Nb^{Df-Doc}).

Class-Feature-Centroid is a recent learning model presented in [5]. In Class-Feature-Centroid, each class is considered as a Vector Space Model [4] which is based on the *bags of words* representation. Each class is represented as a term-vector and a class-vector is a centroid. $\overrightarrow{C_j} = \{w_{1j}, w_{2j}, ..., w_{|L|j}\}$ is the centroid representation of the class j where w_{ij} is the weight of the term t_i for the class j. For classifying an unlabeled document d, the document is also considered as a term-vector ($\overrightarrow{d} = \{w_{1j}, w_{2j}, ..., w_{|L|j}\}$) and the distance (e.g. cosine) between document-vector $\overrightarrow{d}$ and all centroids $\overrightarrow{C_j}$ is compared. Then, we compute $Class\text{-}Feature\text{-}Centroid$ approach with each w_{ij}. Experiments are called $Cfc^{Tf-Class}$, $Cfc^{Df-Class}$, Cfc^{Tf-Doc}, and Cfc^{Df-Doc}.

5 Experiments

5.1 Experimental protocol and datasets

In order to evaluate our proposal we selected two different datasets:

- The *Reuter* dataset is the Reuters-21578[1] frequently used for evaluating classification and information retrieval algorithms. It contains news from Reuters newswire and proposes different categories such as sugar, gold, soybean, etc. Documents have been manually classified.

[1] http://trec.nist.gov/data/reuters/reuters.html/

– The *Tweet* dataset is composed of French tweets that have been collected for the Polop project[2] during the French Presidential and Legislative elections in 2012. It is composed of more than 2 122 012 tweets from 213 005 users. We consider the set of tweets of a user as one document rather than one document per tweet.

In order to evaluate the efficiency of our proposal, experiments have been done by comparing the results of different supervised classification methods[3] with features weighted by the classical $TF\text{-}IDF$: (i) Two different versions of SVM: *SMO*, using a polynomial kernel [10] and *LibSVM*, using a linear kernel [11]; (ii) One original Naïve Bayes approach known as very efficient for text classification: *DMNB* [12] and one Decision Tree: LadTree [13]. Other comparisons have been performed including Naive-Bayes [14], NaiveBayes Multinomial [15], LibSVM with Radial Basis Function and Polynomial kernel [11], J48 and RepTree [16]. As they clearly performed very badly with our datasets, results are not presented in this paper[4]. For each dataset, we remove stop words and words having less than 3 characters. We choose to not apply lemmatization tools since they are not really well adapted to Tweets.

On first experiments for each series, we did a classical 3 fold cross validation. We consider the test dataset as represented in Figure 1. Three iterations have been chosen in order to have reliable results and we decide not to apply a classical cross validation in order to keep sufficient numbers of documents in the test datasets. Moreover, using the same test dataset over all experiments allows us to ensure that modifications on performance are only due to changes made on the learning dataset.

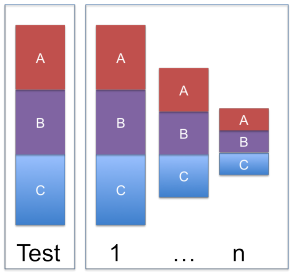

Fig. 1. Validation Process

In this paper, the quality of the classification is evaluated by using micro-averaged F-measure[5].

[2] http://www.lirmm.fr/%7Ebouillot/polop

[3] These methods are available on Weka. http://www.cs.waikato.ac.nz/ml/weka/

[4] Interested reader may found all the results at the following URL: www.lirmm.fr/%7Ebouillot/weipond

[5] Results with Macro and Micro-averaged Precision, Macro and Micro-averaged Recall and Macro-averaged F-measure are available in the web page

5.2 Results

In order to study the behavior of Naïve Bayes and Class-Feature-Centroid approaches and other supervised algorithms according to the reduction of the dataset's volume, we realized three series of experiments on *"Reuter"* and *"Tweet"* dataset. The first series of experiments evaluate the impact on classification results when the number of classes decreases, the second when the number of documents decreases, and the third, when the number of words decreases. As expected, the impact on the decreasing number of classes is consistent. Detailed results of the experiments can be found on the results webpage. Ultimately, we can say that LadTree is a bit more impacted by a large number of classes, and Class-Feature-Centroid and Naïve Bayes approaches used with new weighting method slightly outperform classical algorithms.

How the number of documents impacts classification? Our second series of experiments focuses on the number of documents per class. We set the number of classes (i.e. 10) and we reduce the number of documents per class from 50 to 3. Nine experiments have been performed (see Table 1).

Table 1. Micro-Averaged F-Measure when Decreasing the Number of Documents per Class

	Datasets								
Classes	10	10	10	10	10	10	10	10	10
Doc	500	450	390	330	270	210	150	90	30
Terms	62808	57336	47753	42219	33572	26040	17596	9641	3023
	Results								
DMNB	76%	76%	76%	74%	71%	68%	67%	54%	38%
LadTree	**80%**	**80%**	**81%**	**80%**	**79%**	**76%**	**78%**	51%	16%
LibSVM	69%	71%	66%	59%	54%	47%	45%	30%	21%
SMO	73%	72%	71%	68%	64%	59%	57%	41%	22%
$Cfc^{Df-Class}$	**78%**	**79%**	**76%**	73%	72%	72%	69%	56%	36%
Cfc^{Df-Doc}	75%	75%	73%	71%	70%	72%	67%	55%	36%
$Cfc^{Tf-Class}$	**77%**	**78%**	**78%**	**77%**	**78%**	**75%**	**72%**	**64%**	**45%**
Cfc^{Tf-Doc}	**77%**	**78%**	**77%**	**76%**	**77%**	**75%**	**70%**	**63%**	**45%**
$Nb^{Df-Class}$	**77%**	**77%**	**74%**	**72%**	70%	69%	66%	53%	36%
Nb^{Df-Doc}	73%	72%	71%	69%	67%	68%	65%	51%	36%
$Nb^{Tf-Class}$	**78%**	**78%**	**78%**	**77%**	**78%**	**75%**	**71%**	**65%**	**49%**
Nb^{Tf-Doc}	**77%**	**78%**	**77%**	**76%**	**77%**	**75%**	**71%**	**64%**	**49%**

From these experiments, we can conclude that our new weighting measures with Class-Feature-Centroid and Naïve Bayes approaches (1) slightly outperform other algorithms (except LadTree), (2) are more resistant than most of other algorithms when the number of documents decreases dramatically.

How the number of words impacts classification? On the *"Tweet"* dataset, we decide to set the number of classes (5) and documents (1 186) and to randomly remove

words in order to decrease the number of terms available per document. We assume that randomly removing terms may change the document nature. Seven experiments have been done (see Table 2).

Table 2. Micro-averaged F-Measure when Decreasing the Number of Words per Documents

Datasets							
Classes	5	5	5	5	5	5	5
Documents	1186	1186	1186	1186	1186	1186	1186
Terms	1579374	1322148	613777	264025	202166	157177	76851
Results							
DMNB	**93%**	**92%**	**87%**	**77%**	**78%**	70%	60%
LadTree	72%	72%	67%	56%	56%	53%	49%
LibSVM	67%	51%	50%	51%	51%	30%	22%
SMO	**91%**	**90%**	**82%**	**71%**	70%	61%	51%
$Cfc^{Df-Class}$	79%	79%	71%	57%	57%	57%	53%
Cfc^{Df-Doc}	38%	38%	37%	37%	37%	37%	38%
$Cfc^{Tf-Class}$	**86%**	**86%**	**82%**	**72%**	**72%**	**72%**	**67%**
Cfc^{Tf-Doc}	57%	56%	55%	52%	52%	52%	50%
$Nb^{Df-Class}$	80%	79%	71%	57%	55%	53%	47%
Nb^{Df-Doc}	37%	37%	37%	37%	37%	38%	38%
$Nb^{Tf-Class}$	**88%**	**87%**	**83%**	**74%**	**71%**	**68%**	**59%**
Nb^{Tf-Doc}	57%	56%	55%	52%	51%	50%	46%

Conclusions on these experiments are (1) Naïve Bayes and Class-Feature-Centroid with new weighting measures give results slightly better than SVM and DMNB when the number of terms is low (experiments 6 and 7), and slightly worse otherwise (experiments 1 and 2), (2) Naïve Bayes and Class-Feature-Centroid with new weighting measures outperform LadTree and LibSVM. It is interesting to underline that results are similar on an English corpus (i.e. *"Reuter"*) and on a French corpus (i.e. *"Tweet"*).

6 Conclusions and Future work

Dealing with few data in text classification still remains an important issue which is little addressed in the literature. However, there are many applications where having a large amount of data is not possible or desirable. In this paper, we introduced new weighting measures that we apply in Class-Feature Centroid-based and Naïve Bayes approaches. Experiments show that these new measures are well adapted for dealing with small set of learning data. We compared its efficiency relatively to seven other supervised learning approaches and showed that it performed very well even for different languages. We also investigated the impact of different parameters by varying the number of classes, documents, and words. Conducted experiments have highlighted that best results are obtained when the number of features is reduced. Furthermore, generated models are easy to validate and interpret. This interpretation is important, for instance, when we

have to face with news or tweets evolving at a rapid rate and then obtain the top-k relevant terms for a class over time. The generated models could be used to automatically extract trends over time as the ones that have been proposed during the US elections. In future work, we plan to adapt our approach with $Okapi_{BM25}$ measure. We also want to better investigate the *semantically closeness properties* of classes in order to better evaluate among inner-class and inter class weights which one is the most appropriate.

References

1. Forman, G., Cohen, I.: Learning from little: comparison of classifiers given little training. In: Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases. PKDD '04, Springer-Verlag (2004) 161–172
2. Zeng, H.J., Wang, X.H., Chen, Z., Lu, H., Ma, W.Y.: Cbc: Clustering based text classification requiring minimal labeled data. In: Proceedings of the Third IEEE International Conference on Data Mining. ICDM '03 (2003) 443–
3. Lin, F., Cohen, W.W.: Semi-supervised classification of network data using very few labels. In: Proceedings of the 2010 International Conference on Advances in Social Networks Analysis and Mining. ASONAM '10 (2010) 192–199
4. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill, Inc., New York, NY, USA (1986)
5. Guan, H., Zhou, J., Guo, M.: A class-feature-centroid classifier for text categorization. In: Proceedings of the 18th international conference on World wide web. WWW '09, New York, NY, USA, ACM (2009) 201–210
6. Zhang, X., Wang, T., Liang, X., Ao, F., Li, Y.: A class-based feature weighting method for text classification. Journal of Computational Information Systems **8**(3) (2012) 965–972
7. Kim, S.B., Han, K.S., Rim, H.C., Myaeng, S.H.: Some effective techniques for naive bayes text classification. Knowledge and Data Engineering, IEEE Transactions on **18**(11) (2006) 1457–1466
8. Joachims, T.: A statistical learning learning model of text classification for support vector machines. In: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval. SIGIR '01, ACM (2001) 128–136
9. Lewis, D.D.: Naive (bayes) at forty: The independence assumption in information retrieval, Springer Verlag (1998) 4–15
10. Platt, J.C.: Advances in kernel methods. MIT Press, Cambridge, MA, USA (1999) 185–208
11. Chang, C.C., Lin, C.J.: Libsvm: A library for support vector machines. ACM Trans. Intell. Syst. Technol. **2**(3) (May 2011) 27:1–27:27
12. Su, J., Zhang, H., Ling, C.X., Matwin, S.: Discriminative parameter learning for bayesian networks. In: Proceedings of the 25th international conference on Machine learning, ACM (2008) 1016–1023
13. Holmes, G., Pfahringer, B., Kirkby, R., Frank, E., Hall, M.: Multiclass alternating decision trees. In: Proceedings of the 13th European Conference on Machine Learning. ECML '02, London, UK, UK, Springer-Verlag (2002) 161–172
14. John, G.H., Langley, P.: Estimating continuous distributions in bayesian classifiers. In: Proceedings of the Eleventh conference on Uncertainty in artificial intelligence, Morgan Kaufmann Publishers Inc. (1995) 338–345
15. McCallum, A., Nigam, K., et al.: A comparison of event models for naive bayes text classification. In: AAAI-98 workshop on learning for text categorization. (1998) 41–48
16. Quinlan, J.R.: C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1993)