IMPROVING CENTROID-BASED TEXT CLASSIFICATION USING TERM-DISTRIBUTION-BASED WEIGHTING SYSTEM AND CLUSTERING

Thanaruk Theeramunkong and Verayuth Lertnattee

Information Technology Program
Sirindhorn International Institute of Technology
Thammasart University, Rangsit Campus
Pathumthani, 12121, Thailand
Phone:+66-2-980-9009, Fax:+66-2 -986-9112
Email:ping@siit.tu.ac.th

Information Technology Program
Sirindhorn International Institute of Technology
Thammasart University, Rangsit Campus
Pathumthani, 12121, Thailand
Phone:+66-2-980-9009, Fax:+66-2 -986-9112
Email:verayuth@siit.tu.ac.th

ABSTRACT

Centroid-based text classification is one of the most popular supervised approaches to classify texts into a set of pre-defined classes with relatively low computation. Based on the vector-space model, the performance of this classification particularly depends on the way to weigh terms in documents in order to construct a representative class vector for each class and degree of spherical shape in class. In this paper, we propose a method to improve classification accuracy by considering a number of statistical term weighting systems based on term-distribution, including factors of *intra-class*, *inter-class, overall* term frequency distribution and *term-length* normalization. An improvement using a clustering technique called hierarchical EM is also investigated. A number of experiments using drug information web pages and newsgroups data set, are made. The results show that our method outperforms standard *tf-idf* centroid-based, k-nearest neighbor and naïve Bayesian classifiers to some extent.

1. INTRODUCTION

Among fast growth of online text information, there has been extreme need to find and organize relevant materials. For this purpose, text categorization is an important tool. Given a training set of labeled documents, text categorization is a method to assign a category label to each new document. In the past, there were a number of categorization methods proposed based on k-nearest neighbor [9], Bayesian model [1,2,5,9], support vector machine (SVM) [6]. Centroid-based document classification algorithm was explored in [2]. Compared with other methods, centroid-based classification has less computation and word dependency. Another important issue in centroid-based is that each category is assumed to be spherical. This may be not true in all cases.

Centroid-based classification constructs a class document vector (a representative vector for all documents in a class) instead of treating individual document vector. Given a test document, each class vector is calculate similarity function (such as cosine function). The test document is assigned to a class which highest similarity to centroid-based vectors. It has shown good performance both accuracy and time complexity. However, there were still few researches on studying term weighting with its term distribution approach. Another issue is that, centroid-based approach assumes that each class is spherical, that is not true for all case. In this paper, we purpose term weighting system to centroid-based vector using its statistical with word term frequency distribution in *intra-class*, *inter-class*, *overall* and also *term-length normalization.* We also show a method to cluster a class with poor accuracy into a set of small clusters and then perform centroid-based classification can improve accuracy in classification.

2. PREVIOUS WORKS

This section shows some previous works on classification. There are several types of supervised classification methods. The popular are k-nearest neighbors, naïve Bayesian, support vector machine and centroid-based classification.

Nearest neighbor classifier is an instance-based algorithm based on learning by analogy. It has been applied to text categorization since the early days of research and has been shown to produce better results when compared against other machine learning algorithm such as C4.5 and RIPPER. There are some data sets, it gets poor classification and there are many enhancements to improve its accuracy. Naïve Bayesian algorithm has been widely used for document classification, and has been shown to produce very good performance. The algorithm assumes that the effect of an attribute value on a given class is independent of values of the other attributes. A support vector machine (SVM) algorithm was used as the classifier, because it has been shown in previous work to be both very fast and effective for text classification problems [6]. Centroid-based classifier, a simple linear-time centroid-based document classification has shown that consistently and substantially outperforms other algorithms such as k-nearest neighbors, C4.5 and naïve Bayes [2]. Chuang et. al., also have prototype vector that is very close to centroid-based vector and shown that it is a fast algorithm for hierarchical text classification and also good performance in accuracy [7].

Unsupervised learning can be applied to improve text categorization. Automatic cluster detection is a useful method that is searching for groups of document that are similar to one another. The major clustering methods including partitioning methods, hierarchical methods, density-based method, grid-based method and model-based method [4]. K. Nigam et. al. [5], use hierarchical Expectation-Maximization (EM) for text classification enhanced with supervised learning.

Many term weighting systems have been purposed by G. Salton [3]. They suggested that weighting system is composed of 3 parts: term frequency, collection frequency and document length normalization. They suggested weighting system for both document and query. These weighing system can be applied in centroid-based [2] and prototype based [7] classifications.

3. TERM WEIGHTING AND CLUSTERING

This section shows term weighting system. Standard *tf, idf* including our term distributions *csd, icsd, sd,* tf_{rms} and clustering methods applied in our framework.

3.1 Term weighting system

Centroid-based classifier is represented by vector model. Let the number of classes be *NC*, so we have *NC* centroid-based vectors $\{\vec{C}_1, \vec{C}_2, ..., \vec{C}_{NC}\}$, where each $\vec{C}_j$ is the centroid-based vector of the j^{th} class. In each class, it contains with $|C_j|$ documents $\{\vec{d}_1, \vec{d}_2, ..., \vec{d}_{|C_j|}\}$ where each $\vec{d}_k$ is the k^{th} document in class C_j

Class term frequency $tf(w_i,C_j)$: is the average number of times w_i occurs in class.

$$tf(w_i,C_j) \quad = \quad \frac{\sum\limits_k w_{ijk}}{|C_j|} \qquad (1)$$

Inverse document frequency $idf(w_i)$: is the log of the ratio between total number of document and the number of document that word w_i occurs.

$$idf(w_i) \quad = \quad \frac{\log \sum\limits_j |C_j|}{\sum\limits_j \sum\limits_k \delta\ (w_i, d_{jk})} \quad \delta \in \{0,1\} \qquad (2)$$

Class-standard deviation $csd(w_i,Cj)$: is the standard deviation of $tf(w_i,C_j)$ of the class C_j.

$$csd(w_i,C_j) = \sqrt{\frac{\sum\limits_k [w_{ijk} - tf(w_i,C_j)]^2}{|C_j|}} \qquad (3)$$

Inter-class standard deviation $icsd(w_i)$: is the standard deviation of word w_i calculate from $tf(w_i,C_j)$.

$$icsd(w_i) = \sqrt{\frac{\sum\limits_k \left[tf(w_i,C_j - \frac{\sum\limits_k tf(w_i,C_j)}{NC} \right]^2}{NC}} \qquad (4)$$

Standard deviation $sd(w_i)$: is the standard deviation calculate from term frequency of w_i in training document.

$$sd(w_i) = \sqrt{\frac{\sum\limits_j \sum\limits_k \left[w_{ijk} - \frac{\sum\limits_j \sum\limits_k w_{ijk}}{\sum\limits_j |C_j|} \right]^2}{\sum\limits_j |C_j|}} \qquad (5)$$

Root mean square of document term frequency in class $tf_{rms}(w_i,C_j)$: is defined as

$$tf_{rms}(w_i,C_j) \quad = \quad \sqrt{\frac{\sum\limits_k w_{ijk}^2}{|C_j|}} \qquad (6)$$

In centroid-based, we consider 4 factors:

(1) intra-class factors: *tf(wi,Cj), csd(wi,Cj).*

(2) inter-class factor: *icsd(wi).*

(3) overall documents factors: *sd(wi), idf(wi).*

(4) normalization factors: $tf_{rms}(wi,Cj)$.

Our concept is that not only $tf(w_i,C_j)$, *idf(wi)* and document length normalizing factor have effect to weighting system in centroid-based vector but also term distribution. The different between each class depends on word terms and/or term frequency. The terms that occur in few class has lower distribution should be applied more weight than higher distribution.

The $tf_{rms}(w_i,C_j)$ is surprisingly. The upper part from the equation (6) is considered as normalization of term length. The lower part can cutoff when normalizing the vector with its document length.

In the vector model, word weight before normalization define as:

$$W(wi,C_j) = tf(w_j,C_j) * idf(w_i) * Modifiers \qquad (7)$$

The *Modifiers* are *csd, icsd, sd* and tf_{rms}. These are modified only in centroid-based vector, after that we normalize with vector length. The test document weight is only standard *tf*idf* with normalization.

The similarity between test documents ($\vec{x}_i$) and centroid-based ($\vec{C}_j$) is measured using the cosine function. The class of a new document is determined as the most similar of the document with the centroid-based vector.

3.2 Clustering

Some classes that get poor performance after centroid-based classification due to non-uniformness of documents in classes, clustering technique may take advantage to produce the smaller and more uniform clusters.

There was a large number of clustering algorithms in the literature. The choice of clustering algorithm depends on the type of data available and on the particular purpose and application. We select exist hierarchical EM algorithm to perform clustering. Hierarchical EM is a divisive hierarchical clustering, that is starting with all objects in one clusters and subdivides the cluster into smaller and smaller pieces, until each object forms a cluster on its own or until it satisfies termination conditions.

4. EXPERIMENTAL RESULT

Two data sets are used for experiment. (1) Drug Information (DI) is the web documents that we have been collected from www.rxlist.com. There are 3,149 documents with 7 categories: adverse drug reaction, clinical pharmacology, description, indications, overdose, patient information, and warning. (2) Newsgroups data set was collected by K. Lang, contains about 20,000 articles divided among 20 UseNet discussion groups. Several classes are quite confusable. The main groups are computer, recreation, science, politics, religion, forsale and vehicle.

4.1 Investigating Term Weight Systems

Before constructing the centroid-based vector, we use stopwords to eliminate common words from the documents (a, an, the, …). We also removed word tags in web documents in DI and header in News.

In open-test, the data sets were splited into 90 % for training set and 10% randomly for test set. We performed 5 times for each experiment. In close-test experiment, all documents were selected as training and test set. The classification accuracy is defined as the ratio between the number of documents assigned with correct classes and total number of documents in test set. The results are shown in Table 1.

4.2 Improving Accuracy Using Clustering

If the result from closed test show that there are some classes that have poor accuracy. We perform the hierarchical EM to those classes to divide into small clusters. Suppose that there are *l* classes and *m* classes with low accuracy and we cluster each class into *n* clusters. We have total of *(l-m)+mn* classes that allow performing centroid-based vectors into next step. We define accuracy of the class from the sum of corrected classify among the clusters in the class. In DI, we perform with branching factor = 2 and in News with branching factor = 3. So we divide classes in DI into 4 and News into 9 sub clusters.

We perform learning phase again with *(l-m)+mn* centroid-based vectors and select four term weighting system that give better accuracy from Table 4.1 in both data sets. If a test document is classified in one of the clusters of the class, we classify the document belong to that class. The result was shown in Table 2. for DI and Table 3. for News, report in percent accuracy ± standard error.

Table 1. Weighting system with centroid-based classifier. *csd*, *icsd*, *sd* and tf_{rms} are modifiers to standard tf*idf in open test and closed test. The first column expresses the set number of our weighting system considers: (1) One factor of term distribution, (2) two factors and (3) using tf_{rms} with term distribution.

SET	Weighting Method	Open Test (%)		Closed Test (%)	
		DI	News	DI	News
0	k-NN (k=30, tf*idf)	82.74	79.26	88.98	84.16
	Naïve Bayes	89.81	82.29	88.47	82.79
	Tf	79.62	66.08	80.66	69.16
	Idf	43.25	75.45	76.09	90.81
	tf*idf	78.85	74.52	86.19	79.09
1	tf*idf*csd	57.26	53.42	72.56	57.46
	tf*idf/csd	51.40	71.42	94.09	77.76
	tf*idf*sd	65.41	56.92	76.56	59.70
	tf*idf/sd	**89.87**	**83.85**	**92.25**	**91.37**
	tf*idf*icsd	66.50	55.49	92.22	94.37
	tf*idf*/ icsd	68.66	82.55	83.26	59.68
2	tf*idf*csd*sd	45.92	40.40	64.81	45.62
	tf*idf*(csd/sd)	70.64	74.20	81.68	80.43
	tf*idf*(sd/csd)	55.99	35.40	94.57	39.36
	tf*idf/(csd*sd)	82.23	82.60	94.22	91.91
	tf*idf*csd*icsd	51.34	43.43	72.59	46.01
	tf*idf*(icsd/csd)	63.76	31.45	80.79	93.34
	tf*idf*(csd/icsd)	46.31	72.69	96.51	31.46
	tf*idf/(csd*icsd)	59.04	76.47	90.57	93.95
	tf*idf*icsd*sd	55.41	22.84	70.40	22.32
	tf*idf*(icsd/sd)	90.38	69.88	63.48	78.69
	tf*idf*(sd/icsd)	42.10	74.27	95.81	73.34
	tf*idf/(icsd*sd)	65.73	76.53	91.17	94.22
3	**tf*idf/tf$_{rms}$**	**88.79**	**83.23**	**96.22**	**91.51**
	tf*idf/ (tf$_{rms}$*sd)	80.25	80.10	94.19	93.95
	tf*idf / sqrt(tf$_{rms}$*sd)	**91.02**	**83.99**	**93.57**	**91.52**
	tf*idf*2/(tf$_{rms}$+sd)	**92.10**	**84.59**	**93.93**	**91.68**

The result showed that centroid-based classifier, with some modifier weighting systems can outperform, k-NN and naïve Bayes classifier.

The result of close test showed that the centroid-based classifier with our weighting system is very efficient. Some weighting systems can perform more than 90 % accuracy. However from the detail of close test, we notice that the class "patient information" has poor accuracy. Class "comp.os.ms-windows.misc" and "talk.religion. misc" in 20 Newsgroups have poor accuracy. The patient information has many topics about patient who takes those drugs. This information may overlap to the other classes, especially overdose. Therefore, many test documents (of patient information) are classifed as overdoses. In 20 newsgroups, there is no doubt in these two groups because their name ".misc" (for miscellaneous) . So many documents in comp.os.ms-windows.misc are classified as others group in computer. The “talk.religion.misc” are always classified as “alt.athseim”. From this result, we perform hierarchical EM to these classes. Patient information is divided into 4 clusters and comp.os.ms-windows.misc and talk.religion.misc, each of them is divided into 9 clusters.

Table 2. Experiments with Drug Information (DI) before/after divided “patient information” into 4 clusters (DI-1). (accuracy ± standard error)

Weighting Method	DI (%)	DI-1 (%)
Naïve Bayes	89.81±0.77	87.96±0.22
tf*idf/sd	89.87±0.67	**91.02±0.48**
tf*idf/tf_{rms}	88.79±0.66	88.09±0.20
tf*idf / sqrt(tf_{rms}*sd)	91.02±0.52	**93.31±0.45**
tf*idf*2/(tf_{rms}+sd)	92.10±0.93	**92.48±0.38**

Table 3. Experiments with 20 Newsgroups (News) before/after divided “comp.os.ms-windows.misc” into 9 clusters (News-1) and “talk.relegion.misc” into 9 clusters (News-2). (accuracy ± standard error)

Weighting Method	News (%)	News-1 (%)	News-2 (%)
Naïve Bayes	82.29±0.20	82.08±0.23	81.11±0.24
tf*idf/sd	83.85±0.55	**85.63±0.41**	**84.81±0.22**
tf*idf/tf_{rms}	83.23±0.21	**84.29±0.18**	**84.50±0.31**
tf*idf / sqrt(tf_{rms}*sd)	83.99±0.18	**85.55±0.22**	**86.55±0.27**
tf*idf*2/(tf_{rms}+sd)	84.59±0.36	**86.08±0.36**	**86.27±0.41**

5. DISCUSSION

After investigating several term weighting systems, we found out that the most effective ones are *sd* and tf_{rms}. The accuracy using these factors is higher than that of k-NN and Bayesian classifiers with nearly 8-10% and 1-3 % improvement, respectively. This means that the factors of overall term distribution and term-length normalization are important in improving classification accuracy. On the other hand, some *intra-class* and *inter-class* factors, that is *csd* and *icsd*, perform well in close-test experiments but are not useful for improving accuracy in the open-test experiments. This indicates that *csd* and *icsd* may make the learned class vector become too specific (over-specification problem).

For the classes with poor accuracy, clustering is a tool to improve accuracy. They can be broken down into some smaller classes before performing classification. We found a promising improvement using clustering techniques.

6. CONCLUSION

In this paper, we showed a method to improve classification accuracy using statistical term weighting systems based on term-distribution, including factors of *intra-class*, *inter-class, overall* term frequency distribution and *term-length* normalization. We also presented a method to improve classification accuracy by a clustering technique based on EM algorithm. By experiments using two datasets, the result showed that our method outperforms standard *tf-idf*-based centroid-based, k-nearest neighbor and naïve Bayesian classifiers with 3-12% accuracy improvement. As our further work, we will focus on the following topics: Some modifiers may work on one data set better than others, how can we manage these to get the best performance? Can we you these statistical values to feature selection? How to detect which class needs to be clustered and how many clusters which get the best accuracy?

ACKNOWLEDGEMENTS

This work has been supported by National Electronics and Computer Technology Center (NECTEC) under the project number (NT-B-06-4F-13-311 on year 2001)

REFERENCES

[1] A McCallum and K. Nigam: “ A Comparison of Event Models for Naïve Bayes Text Classification,” In AAAI-98 Workshop on Learning for Text Categorization. *http://www.cs.cmu.edu/~mccallum*.

[2] E. Han and G. Karypis: “Centroid-Based Document Classification: Analysis & Experimental Results,” In European Conference on Principles of Data Mining and Knowledge Discovery (PKDD), 2000. Also available on WWW at URL *http://www.cs.umn.edu/~karypis*.

[3] G. Salton: “Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer,” Addison Wesley, 1989.

[4] J. Han and M. Kamber: “Data Mining: Concepts and Techniques,” Morgan Kaufmann publishers, 2001.

[5] K. Nigam, A. McCallum, S. Thrun and T. Mitchell: “Text Classification from Labeled and Unlabelled Document using EM,” Machine Learning, Vol. 39, No. 2/3, pp. 103-134, 2000.

[6] T. Joachims: “Text Categorization with Support Vector Machines: Learning with Many Relevant Features,” In European Conference on Machine Learning (ECML), pp. 137-142, 1996.

[7] W. T. Chuang, et al: “A Fast Algorithm for Hierarchical Text Classification,” Data Warehousing and Knowledge Discovery, pp. 409-418, 2000.

[8] Y. Yang and J.P. Pedersen: “A Comparative Study on Feature Selection in Text Categorization,” The Fourteenth International Conference on Machine Learning, pp. 412-420, 1997.

[9] Y. Yang and X. Liu: “A Re-examination of Text Categorization Methods,” In Proceedings of the

22nd ACM SIGIR Conference, pp. 42-49, 1999.