Character Variable Numeralization Based on Dimension Expanding and its Application on Text Classification

Li-xun Xu[1], Xu Yu[2(✉)], Yong Wang[3], and Yun-xia Feng[2]

[1] Sino-German Faculty, Qingdao University of Science and Technology, Qingdao 266061, China
[2] School of Information Science and Technology, Qingdao University of Science and Technology, Qingdao 266061, China
yuxu0532@163.com
[3] College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China

Abstract. The character variable discrete numeralization destroyed the disorder of character variables. As text classification problem contains more character variable, discrete numeralization approach affects the classification performance of classifiers. In this paper, we propose a character variable numeralization algorithm based on dimension expanding. Firstly, the algorithm computes the number of different values which the character variable takes. Then it replaces the original values with the natural bases in the *m*-dimensional Euclidean space. Though the algorithm causes a dimension expanding, it reserves the disorder of character variables because the natural bases are no difference in size, so this algorithm is a better character variable numerical processing algorithm. Experiments on text classification data sets show that though the proposed algorithm costs a little more running time, its classification performance is better.

Keywords: Character variable · Natural bases · Dimension expanding · Text classification

1 Introduction

Text classification [1, 2] is a very important direction of research in pattern recognition. With the development of Internet technology, text recognition is playing an increasingly crucial role. By text recognition, we can conduct public opinion analysis, which enables government to understanding aspirations of people and adjust measures in a timely manner. Text recognition can also help owners of online shopping sites to know the attitudes of consumers so that improve the service quality of their website.

Text classification typically includes the expression of texts, selection and training of classifiers, evaluation and feedback process of classification results. As in essence text classification problems belong to the scope of text classification, so a lot of typical pattern classification algorithms can be applied to text classification problems. As the effect of text classification algorithm based on statistical learning method is better, so statistical learning method is studied widely by scholars home and abroad. Statistical

W. Che et al. (Eds.): ICYCSEE 2016, Part I, CCIS 623, pp. 238–248, 2016.
DOI: 10.1007/978-981-10-2053-7_22

learning methods require a number of documents, which were accurately classified by human, as learning materials, and then computers mine rules from these documents. This process is called the training process, and the set of rules it summed up often are called classifiers. After training, the documents that computers have never seen before can be classified by the trained classifiers. Typical statistical learning methods contain Bayesian analysis method [3], KNN method [4], support vector machine method [5], artificial neural network method, the decision tree method [6] and etc. For example, Wajeed classified the textual data. In the process of classification the effects of the features distributed across the document is explored. KNN algorithm is employed and the results obtained are encouraging [7]. Sun et al. give a comparative study on text classification using SVM [8].

As mature classification methods, those methods have achieved better learning results on the text classification problems. However, text classification is a high-dimensional classification problem, and the feature vector contains a lot of character variables [9, 10]. Traditional text classification methods [11, 12] used discrete processing methods. By assigning the different values of properties to different nature numbers, the disorder character variables are undermined, and the recognition performance of text classification are affected to a certain extent.

For feature vectors in text classification problems contains a lot of character variables, this paper proposes a character variable numeralization algorithm based on dimension expanding. Firstly, the algorithm computes the number of different values which the character variable takes. Then it replaces the original values with the natural bases in the p-dimensional Euclidean space. Though the algorithm causes a dimension expanding, it reserves the disorder of character variables because the natural bases are no difference in size. Therefore, the data processing method can help classifiers to achieve a better performance. In general, the proposed data preprocessing algorithm is not limited to a particular classifier. In order to fully verify the algorithm, this paper selects KNN and support vector machine learning algorithm as the experimental classifiers. Experiments on a news text data set show that the proposed preprocessing algorithm allows the classifier to obtain a higher classification accuracy compared to a discrete numerical processing method.

This paper is organized as follows. A brief introduction to KNN algorithm and SVM algorithm is given in Sect. 2. In Sect. 3, a character variable numeralization algorithm based on dimension expanding is proposed. The KNN algorithm and SVM algorithm are used to conduct text classification experiments in Sect. 4, and the experimental results and a detailed analysis are also given in this part. Section 5 concludes the whole paper.

2 Review of the KNN Algorithm and the SVM Algorithm

2.1 The KNN Algorithm

K-nearest-neighbor is an lazy learning classification algorithm, usually denoted by KNN. For the KNN algorithm, training samples are represented by n-dimensional numerical attributes. For a label unknown sample, the KNN algorithm firstly finds the

nearest k training samples from the training set. The distance between two samples can be computed by Euclidean distance. The formula is as below.

$$d(X,Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{1}$$

where $X = (x_1, x_2, ..., x_n)$ and $Y = (y_1, y_2, ..., y_n)$ denote two samples in the n-dimensional Euclidean space.

Then the label of the unknown sample is assigned with the most common class of the k neighbors. Specially, if $k = 1$, the unknown sample is assigned with the same class as its nearest neighbor.

2.2 The SVM Algorithm

Let the training sample set be $T = \{(x_1, y_1), \ldots, (x_l, y_l)\}$, where $x_i \in R^n$, $y_i \in \{-1, 1\}$, $i = 1, \ldots, l$. Assuming that the training sample set is linear separable, the SVM algorithm obtains the classification hyperplane by solving the following quadratic optimization problem.

$$\begin{aligned} \min_{a} \quad & \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y_i y_j (x_i \cdot x_j) a_i a_j - \sum_{i=1}^{l} a_i \\ \text{s.t.} \quad & \sum_{i=1}^{l} a_i y_i = 0 \\ & 0 \le a_i \le C, \quad i = 1, \ldots, l \end{aligned} \tag{2}$$

where a_i is Lagrange multipliers, the parameter $C > 0$ controls the trade-off between the slack variable penalty and the margin.

If the original training set is non-linear separable, the SVM algorithm converts it into a linear separable problem, and then compute the classification hyper-plane by solving the following quadratic optimization problem.

$$\begin{aligned} \min_{a} \quad & \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y_i y_j K(x_i \cdot x_j) a_i a_j - \sum_{i=1}^{l} a_i \\ \text{s.t.} \quad & \sum_{i=1}^{l} a_i y_i = 0 \\ & 0 \le a_i \le C, \quad i = 1, \ldots, l \end{aligned} \tag{3}$$

The decision functions corresponding to linear separable and non-linear separable are listed as below.

$$f(x) = \mathrm{sgn}(\sum_{i=1}^{l} a_i^* y_i (x_i \cdot x) + y_i - \sum_{i=1}^{l} a_i^* y_i (x_i \cdot x_j)) \tag{4}$$

$$f(x) = \mathrm{sgn}(\sum_{i=1}^{l} a_i^* y_i K(x_i \cdot x) + y_j - \sum_{i=1}^{l} y_i a_i^* K(x_i, x_j)) \tag{5}$$

where a_i^* is the optimizing solution of the corresponding optimizing problem.

3 A New Character Variable Numeralization Method

Feature vectors in text classification problems contain plenty of character variables, and most current statistical learning algorithms require the input vector must be numeric vectors. Thus the data set of text classification problems must be preprocessed. Traditional processing method for character variable is as follows. Let a character variable take m different character values, then the usual approach represents the m values of characters variables with 1,2,…,m respectively. In this paper, the above method is referred to as a character variable discrete numerical approach.

As there are no large character variable or small character variable in essence, the shortcomings of this approach lies in that it undermines the disorder of the character variables, and degrades the performance of classifiers. For this problem, this paper proposes a character variable numeralization algorithm based on dimension expanding. The proposed method replaces the p values of a character variable with the m natural bases (0,0,…,0,1),…, (1,0,…,0,0). The natural base refers to a m-dimensional unit vector of which only one component is 1 and the others are 0.

Figure 1 shows the results of data processing with the proposed method when the variable has two different values.

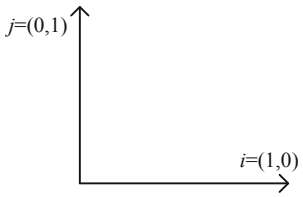

Fig. 1. Illustration of a character variable numeralization algorithm based on dimension expanding

This method replaces the variable 0 and 1 in the traditional method with two linear independent natural base i and j in the two-dimensional Euclidean space. Although it increases the dimensions of the original data, it maintains well the disorder of feature variables.

The text classification algorithm based on character variable numeralization by expanding dimensions, denoted as TCABCVNED, is given below.

Algorithm 1 The TCABCVNED algorithm

Input: text classification data set D; statistical classification algorithm A
Output: text classification rule F
Method:

```
 for i=1to n {
                          /* n denotes the number of feature attributes in text classification
                             data set D /*
if(Is_Character_Variable(f_i))
                          /*The Is_Character_Variable function is to judge whether the attrib-
                             ute is a character attribute /*

{
 m=Dimension_compute(f_i)
                          /*The Dimension_compute function is to compute the number
                             of different values which a character attributes takes /*
  Base_generation(f_i, m);
   }
  i++;
  }
  for i=1to n{
       for j=1to t        /* t denotes the number of records in text classification data set D /*
  {
    if(Is_Character_Variable(f_i))
    Character_Variable_Numeralization (f_i(j));
                          /*Numerlization treatment on feature f_i by expanding dimensions /*
  }
  }
  F=A(D^new)
                          /* Learning on the new training set with statistical classification
                          algorithm A /*
```

Algorithm 1 shows that the proposed data preprocessing method can effectively reserve the disorder of character variables, which provides a possibility for classifiers to achieve a better performance. In Sect. 3, we will test the performance of the proposed method by several experiments.

4 Experiments

4.1 The Experimental Data Set Introduction

This paper selects a web page data set to test the performance of CABCVNED algorithm. As it is a high-dimensional text classification, containing many character attributes, it can test the performance of the proposed algorithm more precisely.

The web page data set comes from sohu news website and we extract four types news topic, including military, diplomatic, technology, and entainment, to test the classification algorithms. For each news topic, we choose randomly 600 samples for training and 300 samples for testing.

In this experiment, the data preprocessing method in reference [13] are used to obtain the training samples and the testing samples.

4.2 Classification Performances Metric

For a better performances evaluation of different classification algorithms, we choose precision and recall as the classification performances metrics. The computation formulas are as follows.

$$p = \frac{Number\ of\ correct\ predictions\ from\ one\ class}{Total\ number\ of\ samples\ predicted\ as\ one\ class} \tag{6}$$

$$r = \frac{Number\ of\ correct\ predictions\ from\ one\ class}{Total\ number\ of\ samples\ from\ one\ class} \tag{7}$$

Text classification system often needs to trade off recall for precision or vice versa. One commonly used trade-off is the F-score, which is defined as the harmonic mean of recall and precision:

$$F - score = \frac{p \times r}{(p + r)/2} \tag{8}$$

where p denotes precision, and r denotes recall.

Obviously, the algorithm can achieve a better performance, while both p and r are higher.

4.3 Detailed Experimental Method

We select KNN and SVM classification methods to conduct this experiment, and for SVM classification method, we choose the C-SVM algorithm and use Gaussian kernel functions. The formula of Gaussian kernel functions is as below,

$$K(x, y) = \exp(-g\|x - y\|^2) \tag{9}$$

where g denotes the width parameter. In this experiment, we use 10-folds cross-validation [14] to compute the best values of parameters.

As it is a multi-classification problem in this experiment, we choose the one against all (1-v-r) approach [15], which is to transform a k-class classification problem into k two-class classification problems.

4.4 The Experimental Results Analysis

We test the performance between the Character Variable Numeralization by Expanding Dimensions algorithm (CVNED) and the Character Variable Numeralization by Discreting algorithm (CVND). We first use the two algorithms to process the selected experiment data set, and then train classifiers with KNN and C-SVM algorithms. The average results about precision, recall, F-score and running time are shown in Figs. 2, 3, 4, 5, 6 and 7, and Table 1, where Training Set 1 is obtained by CVNED algorithm, and Training Set 2 is obtained by CVND algorithm.

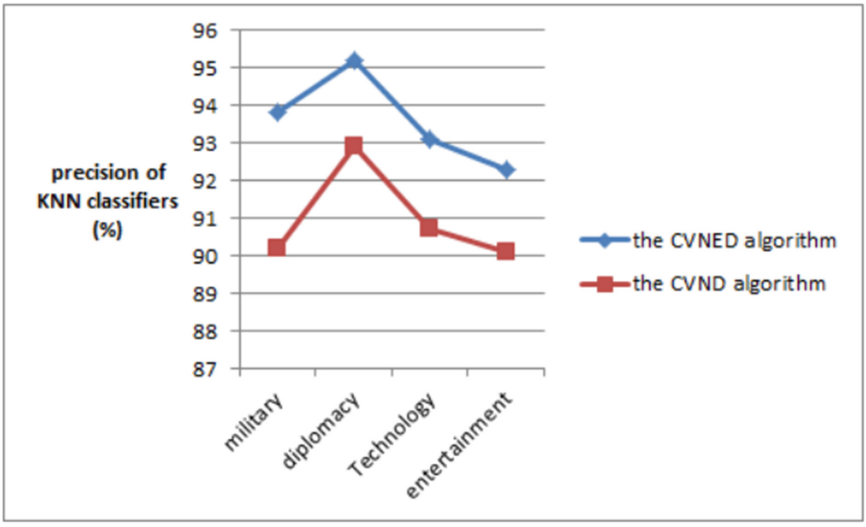

Fig. 2. Precision comparison of KNN classifiers between two data preprocessing methods

As shown in Table 1, the traditional classification algorithms cost much more time on the data sets preprocessing by the proposed CVNED. The main reason is that the CVNED algorithm increases the dimension number of samples. From Figs. 2 3, 4, 5, 6 and 7, we can see that traditional methods, like SVM and KNN, can obtain better performances in both precision and recall indexes after preprocessing by the CVNED algorithm. That is because the CVNED algorithm can reserve the disorder of character variables. The experiment results also shown that the proposed CVNED algorithm in this paper is a more reasonable character variables numeralization method than previous methods.

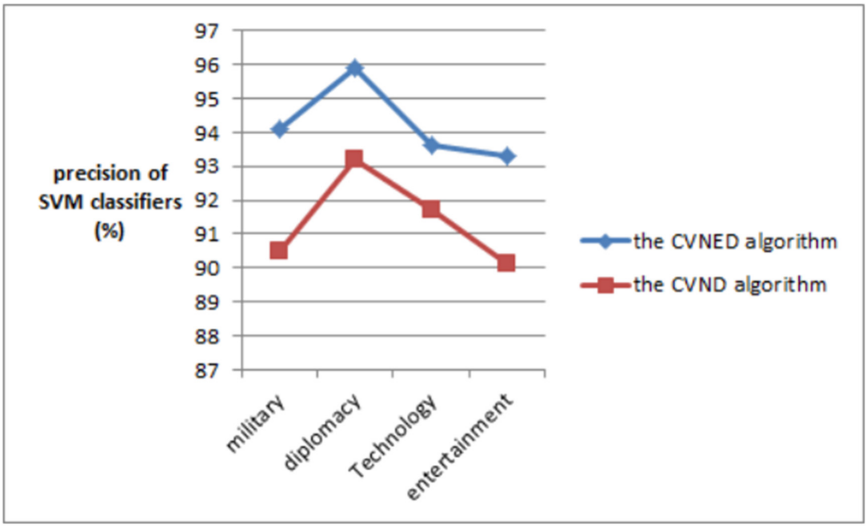

Fig. 3. Precision comparison of C-SVM classifiers between two data preprocessing methods

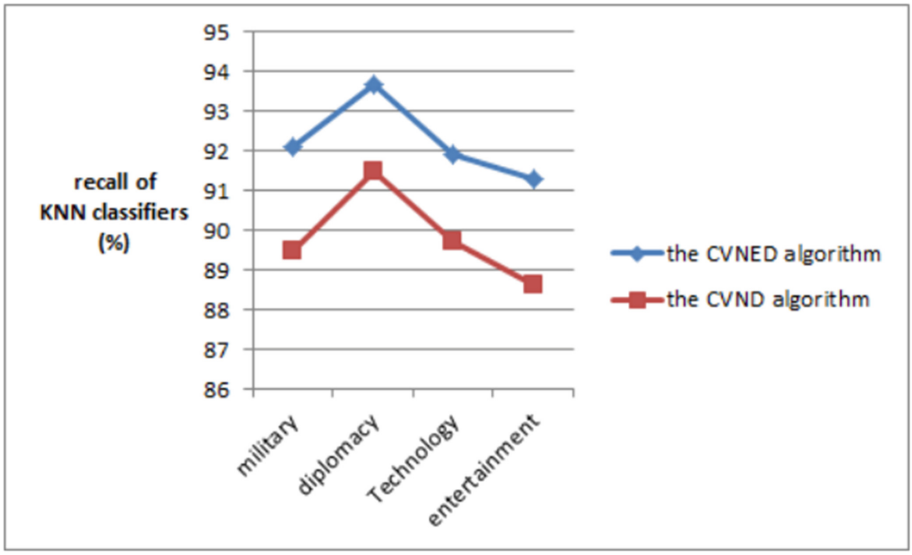

Fig. 4. Recall comparison of KNN classifiers between two data preprocessing methods

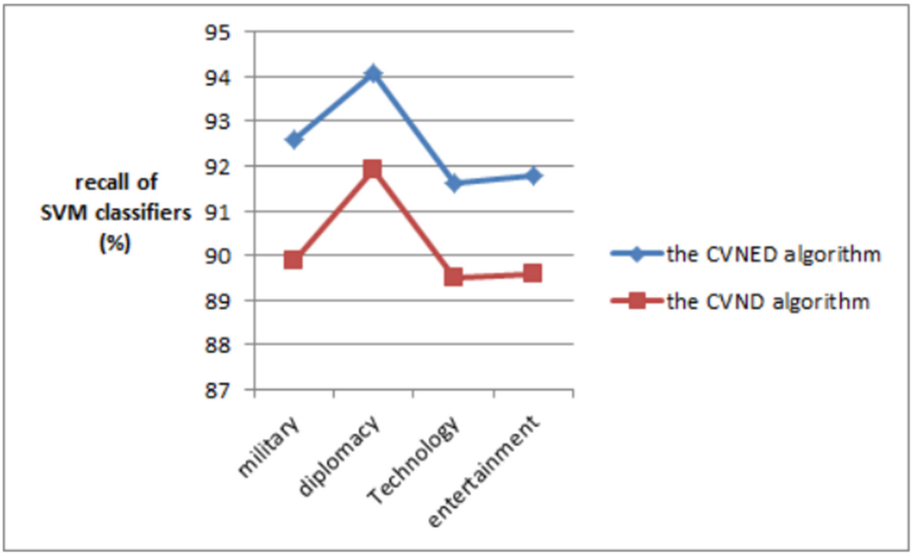

Fig. 5. Recall comparison of C-SVM classifiers between two data preprocessing methods

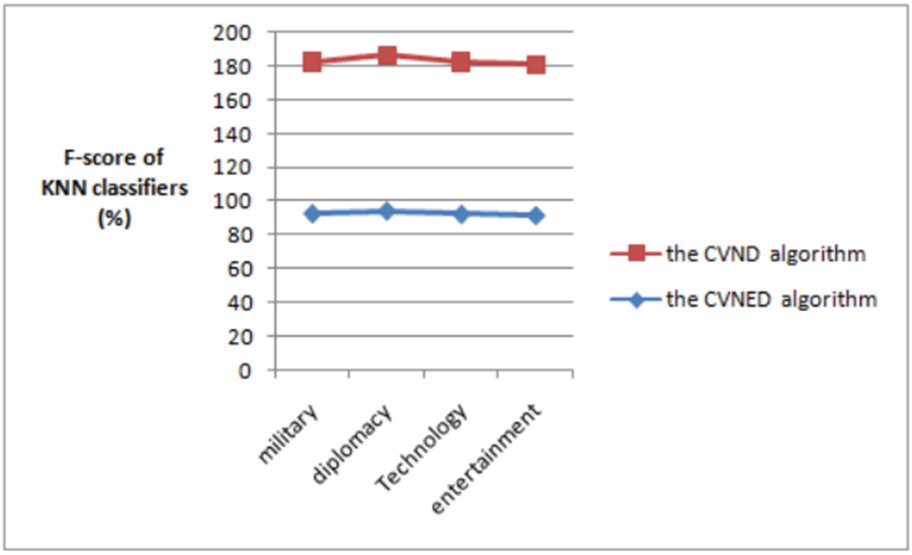

Fig. 6. F-score comparison of KNN classifiers between two data preprocessing methods

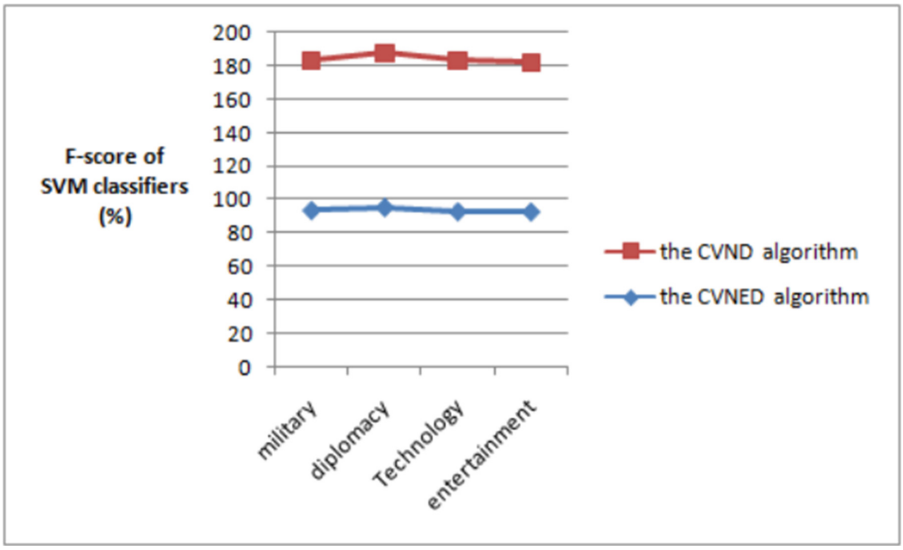

Fig. 7. F-score comparison of C-SVM classifiers between two data preprocessing methods

Table 1. Running time comparison between the two data preprocessing algorithms

Algorithms	Running time on Training Set 1 (s)	Running time on Training Set 2 (s)
The KNN algorithm	29.1	26.7
The C-SVM algorithm	30.2	27.9

5 Conclusion

For character attributes in high dimensional text classification data set, this paper proposed a character variable numeralization algorithm based on dimension expanding. The pretreated methods reserved the disorder of character variables and it is an effective data pretreated method independent of classifiers. After preprocessing, the classification performances of classifiers have been promoted largely. Experiments on text classification data sets show the effective of the proposed method.

Acknowledgement. This work is sponsored by the National Natural Science Foundation of China (Nos. 61402246, 61402126, 61370083, 61370086, 61303193, and 61572268), a Project of Shandong Province Higher Educational Science and Technology Program (No. J15LN38), Qingdao indigenous innovation program (No. 15-9-1-47-jch), the National Research Foundation for the Doctoral Program of Higher Education of China (No. 20122304110012), the Natural Science Foundation of Heilongjiang Province of China (No. F201101), the Science and Technology Research Project Foundation of Heilongjiang Province Education Department (No. 12531105), Heilongjiang Province Postdoctoral Research Start Foundation (No. LBH-Q13092), and the National Key Technology R&D Program of the Ministry of Science and Technology under Grant No. 2012BAH81F02.

References

1. Cheng, Y.C., Wang, P.C.: Packet classification using dynamically generated decision trees. IEEE Trans. Comput. **64**(2), 582–586 (2015)
2. Qiu, C., Jiang, L., Li, C.: Not always simple classification: learning SuperParent for class probability estimation. Expert Syst. Appl. **42**(13), 5433–5440 (2015)
3. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley, New York (2001)
4. Zhang, M.L., Zhou, Z.H.: ML-KNN: a lazy learning approach to multi-label learning. Pattern Recogn. **40**(7), 2038–2048 (2007)
5. Bai, L., Wang, Z., Shao, Y.H., et al.: A novel feature selection method for twin support vector machine. Knowl.-Based Syst. **59**(2), 1–8 (2014)
6. Quinlan, J.R.: Induction of decision trees. Mach. Learn. **1**(1), 81–106 (1986)
7. Wajeed, M.A., Adilakshmi, T.: Different vectors generation techniques with distributed features for text classification using KNN. In: 2012 1st International Conference on Recent Advances in Information Technology (RAIT), pp. 482–486. IEEE (2012)
8. Sun, A., Lim, E.P., Liu, Y.: On strategies for imbalanced text classification using SVM: a comparative study. Decis. Support Syst. **48**(1), 191–201 (2009)
9. Cai, Z., Zhang, T., Wan, X.: A computational framework for influenza antigenic cartography. PLoS Comput. Biol. **6**(10), e1000949 (2010)
10. Cai, Z., Ducatez, M.F., Yang, J., Zhang, T., Long, L.-P., Boon, A.C., Webby, R.J., Wan, X.-F.: Identifying antigenicity associated sites in highly pathogenic H5N1 influenza virus hemagglutinin by using sparse learning. J. Mol. Biol. **422**(1), 145–155 (2012)
11. Cai, Z., Goebel, R., Salavatipour, M., Lin, G.: Selecting genes with dissimilar discrimination strength for class prediction. BMC Bioinform. **8**, 206 (2007)
12. Yang, K., Cai, Z., Li, J., Lin, G.: A stable model-free gene selection in microarray data analysis. BMC Bioinform. **7**, 228 (2006)
13. Lan, J., Shi, H., Li, X., et al.: Associative web document classification based on word mixed weight. Comput. Sci. **38**(3), 187–190 (2011)
14. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proceedings of the 14th Joint International Conference Artificial Intelligence, pp. 1137–1145 (1995)
15. Hsu, C.W., Lin, C.J.: A comparison on methods for multi-class support vector machines. IEEE Trans. Neural Netw. **13**(2), 415–425 (2001)