International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.8, No.6 (2015), pp.235-244
http://dx.doi.org/10.14257/ijsip.2015.8.6.25

A Small Sample Text Classification Algorithm based on Kernel Density Estimation

Liu Yang^{1,2}, Han Liguo^{1*1} and Cai Xuesen^{2}

1 College of GeoExploration Science and Technology, Jilin University, Changchun 130026, China

2 College of Computer Science and Technology, Changchun Normal University, Changchun, 130032, China

han_lg@126.com

Abstract

This paper proposes a small sample text classification algorithm based on kernel density estimation. Firstly, the probability density of the text classification problem is estimated. Then we can construct auxiliary training samples by the estimated probability. Finally, the classification model is obtained with the help of auxiliary training samples. As the introduction of auxiliary training samples avoids over-fitting caused by small training samples, the proposed algorithm can effectively improve the performance of small sample text classification problems. The simulation experiments on the news text datasets fully verify the effectiveness of the proposed algorithm.

Keywords: *text classification; probability density estimation; small sample; news text dataset*

1. Introduction

Text classification [1, 2] is an extremely important research direction in pattern recognition. And with the development of Internet technology, the role of Text classification is becoming more and more important. For example, Good public opinion analyses can be made through text recognition, thus make the government timely understand of the people's demands, and meanwhile conducive to make timely adjustment measures for the government. Shopping site can grasp consumers' attitude very well through Text recognition, and timely improve their service quality. Currently, there have been a lot of technologies applied to text categorization, such as Bayesian analysis method [3], KNN method [4], Support vector machine (SVM) method [5], Neural network method [6], the decision tree method [7] and so on. As mature classification methods, these methods have achieved good learning results on text classification problems. However, in some cases, the number of training samples in text classification problem is very small, the traditional text classification are usually unable to obtain better classification effect, thus a new text classification algorithm which is suitable for small sample [8] text classification problem is required to be developed.

For small sample problem, a small sample text classification algorithm based on kernel density estimation was proposed. Firstly, the probability density of the text classification problem is estimated. Then auxiliary training samples are constructed by the estimated probability. Finally, the classification model is trained with the help of auxiliary training samples. As the introduction of auxiliary training samples avoids over-fitting caused by small training samples, the proposed algorithm can effectively improve the performance

[1] Corresponding author: Liguo Han

ISSN: 2005-4254 IJSIP

of small sample text classification problems. The simulation experiments on the news text datasets fully verify the effectiveness of the proposed algorithm.

The rest of this paper is organized as follows. Section 2 introduces the problem of text classification and the algorithm of kernel density estimation. Section 3 presents the small sample text classification algorithm based on kernel density estimation. In Section 4, we apply the proposed algorithm on news text dataset, give the main results, and make a full analysis. Section 5 summaries the main contribution of this paper.

2. Related Theories

2.1. Text Classification Problem and Systems

In text classification, we are given a description $d \in \mathbf{X}$ of a document, where $\mathbf{X}$ is the document space; and a fixed set of classes $\square = \{c_1, c_2, \cdots, c_J\}$. Classes are also called categories or labels. Typically, the document space $\mathbf{X}$ is some type of high-dimensional space, and the classes are human defined for the needs of an application, as in the examples China and documents that talk about multicore computer chips above. We are given a training set D of labeled documents $< d, c >$, where $< d, c > \in \mathbf{X} \times \square$. For example:

$$< d, c > = < \textit{Beijing joins the World Trade Organization}, \textit{China} >$$

for the one-sentence document Beijing joins the World Trade Organization and the class (or label) China[9].

Using a learning method or learning algorithm, we then wish to learn a classifier or classification function γ that maps documents to classes:

$$\gamma : \mathbf{X} \rightarrow \square \tag{1}$$

This type of learning is called supervised learning because a supervisor (the human who defines the classes and labels training documents) serves as a teacher directing the learning process. We denote the supervised learning method by Γ and write $\Gamma(D) = \gamma$. The learning method Γ takes the training set D as input and returns the learned classification function γ.

Figure 1 shows an example of text classification from the Reuters-RCV1 collection. There are six classes (UK, China, poultry, coffee, elections, sports), each with three training documents. We show a few mnemonic words for each document's content. The training set provides some typical examples for each class, so that we can learn the classification function γ. Once we have learned γ, we can apply it to the *test set* (or *test data*), for example, the new document first private Chinese airline whose class is unknown. In Figure 1, the classification function assigns the new document to class $\gamma(d) = \text{China}$, which is the correct assignment.

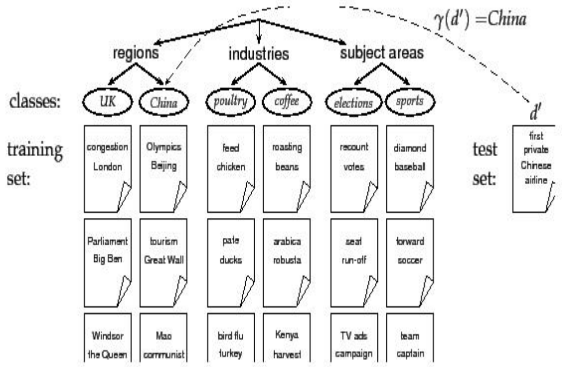

Figure 1. Classes, Training Set, and Test Set in Text Classification

Generally text classification includes the process of text expression, classifier selection and training, the evaluation and feedback of classification results. Text expression can be divided into some steps including text pre-processing, indexing, statistic and feature extraction ect. The overall function modules for text classification system are as follows[10]:

(1) Pre-treatment: format original corpus into the same format, in order to facilitating the subsequent unified treatment;
(2) Index: divide the document into basic processing units, while reducing the cost of subsequent processing;
(3) Statistics: Frequency statistics, the probability of items (words, concepts) and classification;
(4) Feature extraction: extract the features that reflect the theme of the document from the document;
(5) Classifier: the classifier training;
(6) Evaluation: Analysis of test results of classifiers.

2.2. Kernel Density Estimation

The idea of kernel density estimation comes from histogram methods for density estimation 11]. For variable *x*, standard histograms simply partition *x* into distinct bins of width Δ_i and then count the number n_i of observations of *x* falling in bin *i*. In order to turn this count into a normalized probability density, we simply divide by the total number *N* of observations and by the width Δ_i of the bins to obtain probability values for each bin given by

$$p_i = \frac{n_i}{N\Delta_i} \tag{2}$$

for which it is easily seen that $\int p(x)dx = 1$. This gives a model for the density *p*(*x*) that is constant over the width of each bin, and often the bins are chosen to have the same width $\Delta_i = \Delta$.

In practice, the histogram technique can be useful for obtaining a quick visualization of data in one or two dimensions but is unsuited to most density estimation applications. One obvious problem is that the estimated density has discontinuities that are due to the bin edges rather than any property of the underlying distribution that generated the data. Another major limitation of the histogram approach is its scaling with dimensionality. If we divide each variable in a *D*-dimensional space into *M* bins, then the total number of bins will be M^D. This exponential scaling with *D* is an example of the curse of dimensionality. In a space of high dimensionality, the quantity of data needed to provide meaningful estimates of local probability density would be prohibitive.

Kernel density estimation (KDE) is a non-parametric way to estimate the probability density function of a random variable which was proposed by Rosenblatt and Parzen. Therefore it is also termed the Parzen-window method.

As the probability of a vector *X* lies in field *R* is:

$$P = \int_R p(X)dX \tag{3}$$

Let $\chi = \{x_1, \cdots, x_N\}$ be an independent and identically distributed sample drawn from some distribution with an unknown density *p*(*x*). Because each data point has a probability *P* of falling within *R*, the total number *k* of points that lie inside *R* will be distributed according to the binomial distribution:

$$P_k = \begin{pmatrix} N \\ k \end{pmatrix} P^k (1-P)^{N-k} \tag{4}$$

the probability P can be estimated by following

$$\hat{P} = \frac{k}{N} \tag{5}$$

If, however, we also assume that the region *R* is sufficiently small that the probability density *p*(*x*) is roughly constant over the region, then we have

$$P = \int_R p(\mathbf{x})\, d\mathbf{x} = (p\mathbf{x}) \tag{6}$$

Where *V* is the volume of *R*.

Combining (5) and (6), we obtain the density estimate in the form

$$\hat{p}(\mathbf{x}) = \frac{k / N}{V} \tag{7}$$

Let the region *R* to be a small hypercube centred on the point *x* at which we wish to determine the probability density. In order to count the number K of points falling within this region, we define the following function

$$\varphi(\mathbf{u}) = \begin{cases} 1 & |u_j| \le \frac{1}{2}, j = 1,\cdots \\ 0 & otherwis \end{cases} \tag{8}$$

Let h_n is the width of the region *R*, then its volume can be computed by

$$V_n = h_n^d \tag{9}$$

and the total number of data points lying inside this cube will therefore be

$$k_n = \sum_{i=1}^{n} \varphi\left(\frac{\mathbf{x} - \mathbf{x}_i}{h_n}\right) \tag{10}$$

Substituting this expression into (7) then gives the following result for the estimated density at *x*

$$\hat{p}_n(\mathbf{x}) = \frac{k_n / n}{Vn} = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{V_n} \varphi\left(\frac{\mathbf{x} - \mathbf{x}_i}{h_n}\right) \tag{11}$$

Clearly, $\hat{p}_n(\mathbf{x})$ subject to the following two conditions

$$\begin{aligned} &\hat{p}_n(\mathbf{x}) \ge 0 \\ &\int \hat{p}_n(\mathbf{x}) d\mathbf{x} = \end{aligned} \tag{12}$$

which ensure that the resulting probability distribution is nonnegative everywhere and integrates to one.

Therefore any function $\phi(u)$ subject to the following two conditions can be chosen as kernel functions.

$$\begin{aligned} &\varphi(u) \ge 0 \\ &\int \varphi(u) du = \end{aligned} \tag{13}$$

Several common kernel functions are given by the following

(a) Rectangle window

$$\phi(u)=\begin{cases}1, |u| \le \frac{1}{2}\\ 0.其他\end{cases} \tag{14}$$

(b) Gaussian window

$$\phi(u)=\frac{1}{\sqrt{2\pi}}\exp\{-\frac{1}{2}u^2\} \tag{15}$$

3. Algorithm Design

With respect to small sample text classification problem, there are no sufficient samples for training. Thus the traditional classifiers tent to be over-fitting and cannot obtain a good performance. For small sample text classification problem, in this paper, we proposed a small sample text classification algorithm based on kernel density estimation, denoted by TCAKDE.

The main idea of this algorithm is that obtaining the probability density function of the original text data by kernel density estimation method. Thus the original training set can be expanded and the traditional classifiers can obtain a better classification performance. Table 1 below gives TCAKDE algorithm.

Table 1. TCAKDE Algorithm

```
Input: original small sample training set
  $T=\{(x_i,0), i=1,\dots,n_0\}\bigcup\{(x_i,1), i=1,\dots,n_1\}$ , classifier C, kernel function $\varphi(u)$
output：text classification model TM
process
  $\varphi(u)=\frac{1}{\sqrt{2\pi}}\exp\left\{-\frac{1}{2}u^2\right\}$    //select Gaussian window as kernel function
f=KDE(T, $\varphi(u)$);    // estimate the density function f on training set T by kernel density estimation method
 AT= Auxiliary_Sample_Generation(f)；   // construct auxiliary training set AT by the density function f
 $D=T\bigcup AT$ ；    // obtain the new training set
TM=C(D)   // train classifier C on the new training set D, and obtain the text classification model TM
```

4. Experiments

4.1. Experimental Data

To test the performance of SSTCAKDE algorithm, web page data has been selected to conduct text classification experiment in this paper. The selected data set comes from the news text which have been collected by sohu news site. To facilitate the experiment, merely four types of news topic including sci-tec, talk, education and economic have been extracted from the complex news content to conduct classification test. Meanwhile, in order to make the experimental data meet the demands of the algorithm proposed in this paper, the small number of text training data has been selected from each news topic in this experiment. In this experiment, for each type of news topics, 200 samples have been selected to conduct training, and 600 samples have been selected to conduct training. Using the method in literature [12] to conduct preprocessing of the collected new text data to get the training data and test data.

4.2. Indexes of Classification Performance

In order to evaluate the performance of the algorithm more accurately, this experiment does not use traditional classification accuracy as the evaluation index, but chose precision (abbreviated as P) and recall (abbreviated as R)[13]. The calculation formula is as follows:

$$P = \frac{n_1}{n_2} \text{ , } R = \frac{n_1}{n_3} \tag{16}$$

where n_1 represents the number of pages to be classified correctly, n_2 represents the number of pages to be classified in this kind, n_3 represents the total number of pages belonging to this class. Clearly it can be seen only when the precision and recall rates are higher, the performance of the algorithm is superior.

4.3. Experimental Method

In order to test the performance of TCAKDE algorithm, we compare KNN algorithm and SVM algorithm on the selected dataset. Detailed process is as follows: Set KNN as the classifier for text classification in SSTCAKDE algorithm and compare its result with that of KNN algorithm. Set SVM as the classifier in SSTCAKDE algorithm and compare its result with that of SVM. We select the *C*-SVM algorithm as the SVM algorithm, where C is a penalty factor. Since radial basis function (RBF) has a good adaptability on non-linear, and high dimensional data set, this experiment selects Gaussian kernel as kernel function:

$$K(x,y) = \exp(\frac{-\|x-y\|^2}{2\sigma^2}) \tag{17}$$

where σ is a width parameter, "x" and "y" are *n*-dimensional vectors in the original feature space.

As this experiment is a multi-classification problem, so we select one-against-all (1-v-r) approach[14], which is to transform a *c*-class problem into *c* two-class problems, where one class is separated from the remaining ones. In this experiment, the best σ and *C* in *C*-SVM algorithm and K in KNN algorithm are obtained by 10-fold cross-validation[15].

4.4. Experimental Result Analysis

The experimental platform is as follows: Intel Core2 Duo CPU T6500, 2.10GHz, 4.00GB RAM, Windows 8 OS. The average classification result on precision is reported in Figure 2 and Figure 3, and the average classification result on recall is shown in Figure 4 and Figure 5.

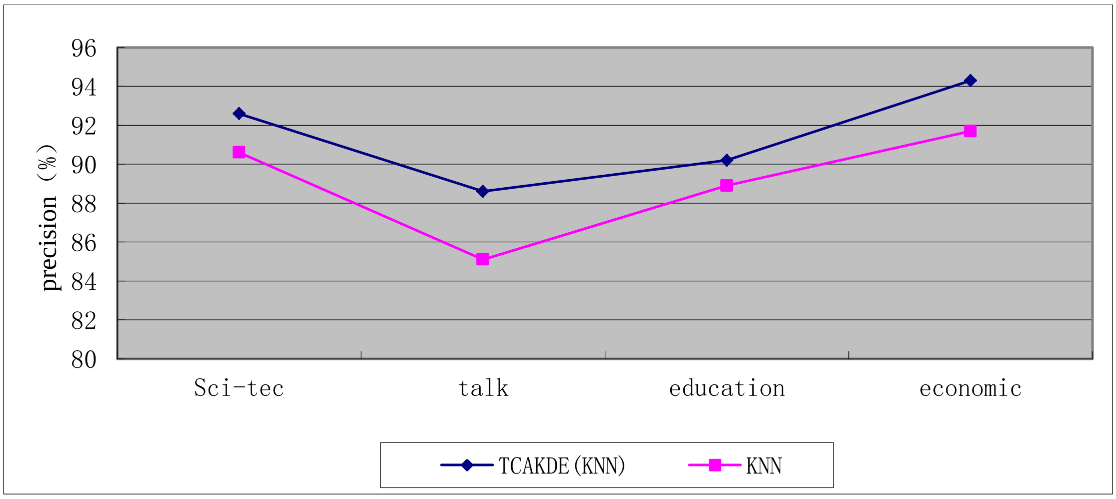

Figure 2. Precision Comparison on KNN Model

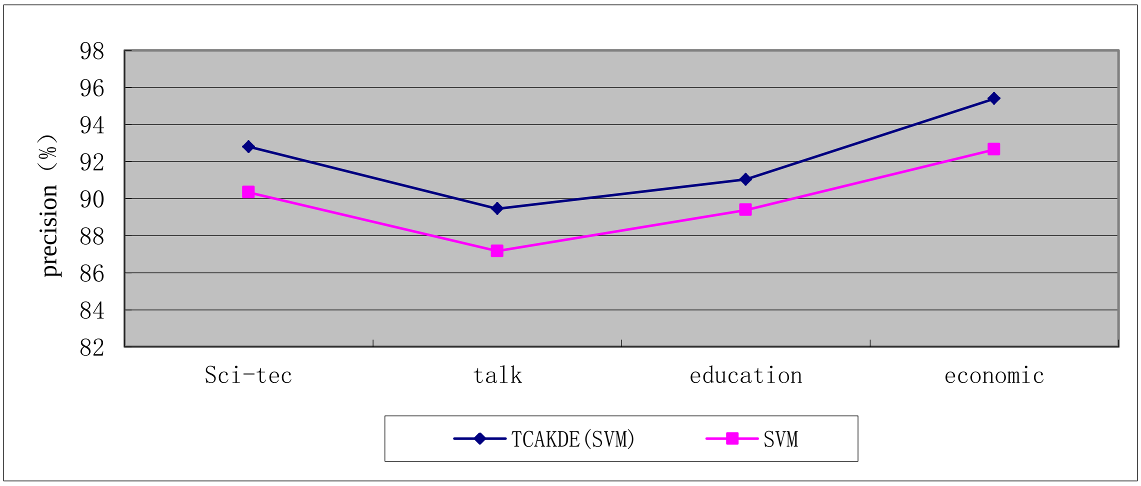

Figure 3. Precision Comparison on SVM Model

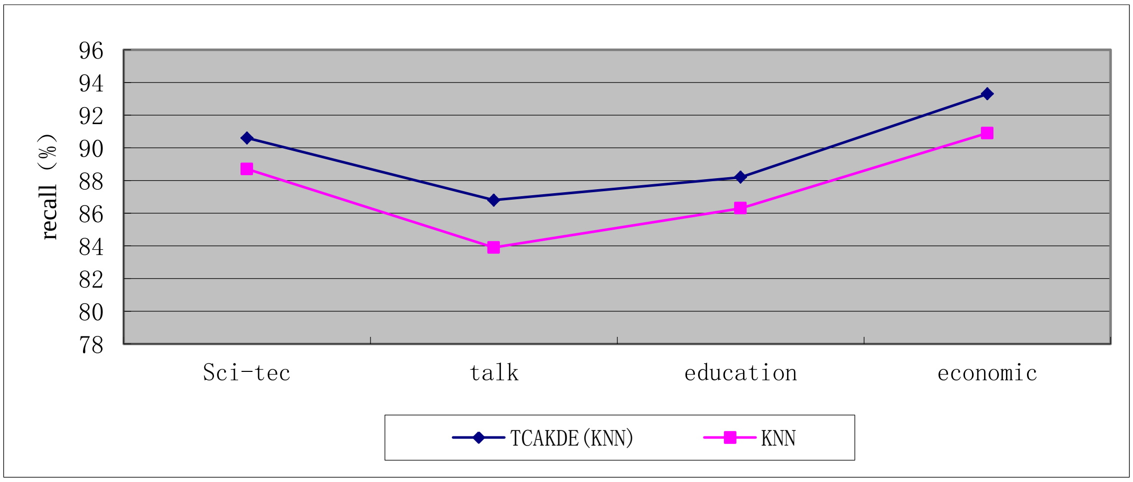

Figure 4. Recall Comparison on KNN Model

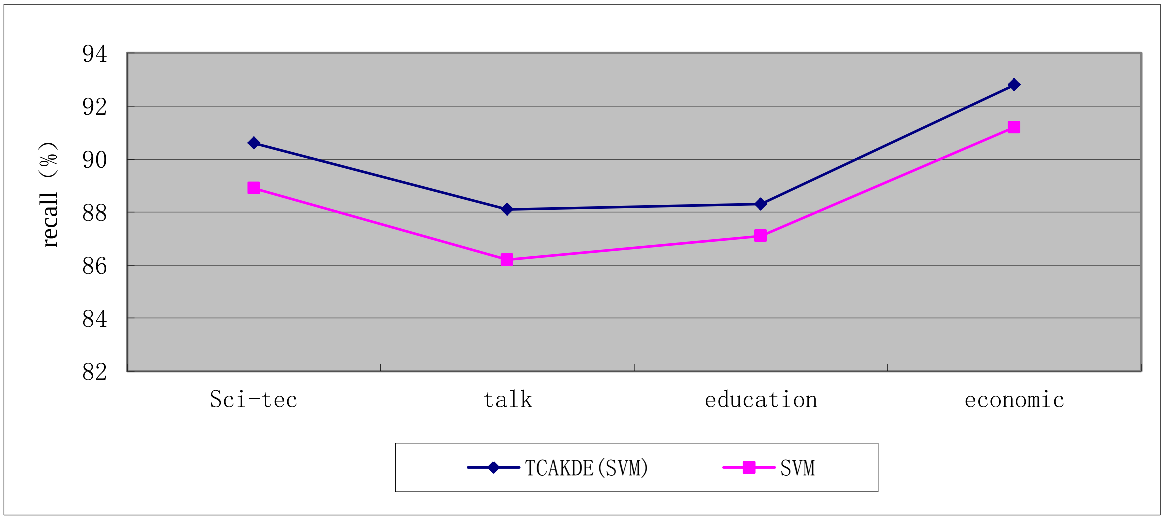

Figure 5. Recall Comparison on SVM Model

As shown in the above four figures, the proposed TCAKDE algorithm performs better than KNN and SVM models on classification results. This is mainly due to a reasonable expansion on the original training set for TCAKDE algorithm. Thus the traditional classifiers can be trained with more training samples, avoiding over-fitting. Moreover the experiment also indicates that this method is an effective text classification algorithms which is independent of classifiers.

5. Conclusion

In this paper, we proposed a kernel density text classification algorithms. The algorithm can obtain a reasonable expansion on training set by using kernel density estimation method which has a good density function estimation performance. Therefor the traditional classifier can avoid over-fitting effectively and perform better on small sample text classification problems.

References

[1] M. Bagdouri, W. Webber and D. D. Lewis, “Towards minimizing the annotation cost of certified text classification[C]”, Proceedings of the 22nd ACM international conference on Conference on information & knowledge management. ACM, (**2013**), pp. 989-998.

[2] W. Meng, L. Lanfen and W. Jing, “Improving short text classification using public search engines [M]”, Integrated Uncertainty in Knowledge Modelling and Decision Making. Springer Berlin Heidelberg, (**2013**), pp. 157-166.

[3] A. Gelman, J. B. Carlin and H. S. Stern, “Bayesian data analysis [M]”, CRC press, (**2013**).

[4] M. Cassotti, D. Ballabio and V. Consonni, “Prediction of acute aquatic toxicity toward Daphnia magna by using the GA-kNN method [J]”, Alternatives to laboratory animals: ATLA, vol. 42, no. 1, (**2014**), pp. 31-41.

[5] R. V. Sharan and T. J. Moir, “Comparison of multiclass SVM classification techniques in an audio surveillance application under mismatched conditions [C]”, Digital Signal Processing (DSP), 2014 19th International Conference on. IEEE, (**2014**), pp. 83-88.

[6] Y. Jiang, Z. Nan and S. Yang, “Risk assessment of water quality using Monte Carlo simulation and artificial neural network method [J]”, Journal of environmental management, (**2013**), 122: pp. 130-136.

[7] W. L. Buntine, “Decision tree induction systems: a Bayesian analysis [J]”, arXiv preprint arXiv: (**2013**), 1304.2732.

[8] A. M. Fouad, M. Saleh, A. F. Atiya, “A Novel Quota Sampling Algorithm for Generating Representative Random Samples given Small Sample Size [J]”, International Journal of System Dynamics Applications (IJSDA), vol. 2, no. 1, (**2013**), pp. 97-113.

[9] C. D. Manning, P. Raghavan and H. Schütze, “Introduction to information retrieval [M]”, Cambridge: Cambridge university press, (**2008**).

[10] W. Li, L. Sun and D. Zhang, ”Text Classification Based on Labeled-LDA Model [J]”, journal of computers, vol. 4, (**2008**), pp. 620-627.

[11] C. M. Bishop, “Pattern recognition and machine learning [M]”, New York: springer, (**2006**).

[12] J. Lan, H. Shi and X. Li, “Associative web document classification based on word mixed weight [J]”, Computer Science, vol. 38, no. 3, (**2011**), pp. 187-19.

[13] X. Yu, J. Yang and Z. Xie, “Training SVMs on a bound vectors set based on Fisher projection [J]”, Frontiers of Computer Science, vol. 8, no. 5, (**2014**), pp. 793-806.

[14] J. Yang, X. Yu and Z. Q. Xie, “A novel virtual sample generation method based on Gaussian distribution [J]”, Knowledge-Based Systems, vol. 24, no. 6, (**2011**), pp. 740-748.

[15] R. Kohavi, “A study of cross-validation and bootstrap for accuracy estimation and model selection”, In: Wermter S, Riloff E, Scheler G, eds. Proc. 14th Joint Int. Conf. Artificial Intelligence. San Mateo, CA: Morgan Kaufmann, (**1995),** pp. 1137-1145.

Authors

Yang Liu, he was born in Changcun of Jilin Province, China, in 1976. He received Master degree from Northeast Normal University, China, in 2007. Now he is a PhD candidate in College of GeoExploration Science and Technology of Jilin University, China. At the same time, he is working for department of Computer Science and Technology of Changchun Normal University as a lecturer. His research interests include Data Mining and Software engineering.

Liguo Han, he was born in Yushu of Jilin Province, China, in 1961. He is a professor and a doctoral supervisor in College of GeoExploration Science and Technology of Jilin University, China. His research interests include Data mining and application of geophysical.