Journal of System Simulation

Volume 27 | Issue 11 Article 13

8-5-2020

Improved Support Vector Pre-extracting Algorithm in Speech Recognition Application

Hao Rui
1. College of Information Management, Shanxi University of Finance & Economics, Taiyuan 030006, China; ;

Yanbo Niu
2. College of Information Engineering, Taiyuan University of Technology, Taiyuan 030024, China; ;

Xiu Lei
3. College of Statistics, Shanxi University of Finance & Economics, Taiyuan 030006, China;

Improved Support Vector Pre-extracting Algorithm in Speech Recognition Application

Abstract

Abstract: Support vector machine (SVM) training is difficult for large-scale data set of speech recognition. A new SVM pre-extracting algorithm was proposed. *On the one hand, kernel Fuzzy C-Means clustering was separately performed on each class of original data set. All the cluster centers were as a representative set of each class. On the other hand, according to the geometric distribution of support vectors and combined with the classification strategy of one-versus-one for SVM multi-class classification algorithm, boundary samples were extracted as support vectors for SVM to training and prediction. The algorithm was applied to embedded speech recognition system.* Experiments indicate that this method improves the efficiency of training but also maintains the high recognition rate.

Keywords

support vector, multi-class classification, kernel fuzzy C-Means clustering, sample pre- extracting, speech recognition system simulation

Recommended Citation

Hao Rui, Niu Yanbo, Xiu Lei. Improved Support Vector Pre-extracting Algorithm in Speech Recognition Application[J]. Journal of System Simulation, 2015, 27(11): 2714-2721.

第 27 卷第 11 期　　系统仿真学报©　　Vol. 27 No. 11
2015 年 11 月　　Journal of System Simulation　　Nov., 2015

Improved Support Vector Pre-extracting Algorithm in Speech Recognition Application

Hao Rui[1], *Niu Yanbo*[2], *Xiu Lei*[3]

(1. College of Information Management, Shanxi University of Finance & Economics, Taiyuan 030006, China; 2. College of Information Engineering, Taiyuan University of Technology, Taiyuan 030024, China; 3. College of Statistics, Shanxi University of Finance & Economics, Taiyuan 030006, China)

Abstract: Support vector machine (SVM) training is difficult for large-scale data set of speech recognition. A new SVM pre-extracting algorithm was proposed. *On the one hand, kernel Fuzzy C-Means clustering was separately performed on each class of original data set. All the cluster centers were as a representative set of each class. On the other hand, according to the geometric distribution of support vectors and combined with the classification strategy of one-versus-one for SVM multi-class classification algorithm, boundary samples were extracted as support vectors for SVM to training and prediction. The algorithm was applied to embedded speech recognition system.* Experiments indicate that this method improves the efficiency of training but also maintains the high recognition rate.

Keywords: support vector; multi-class classification; kernel fuzzy C-Means clustering; sample pre-extracting; speech recognition system simulation

改进的支持向量预选取方法在语音识别中的应用

郝瑞[1]，牛砚波[2]，修磊[3]

(1. 山西财经大学信息管理学院，太原 030006；2. 太原理工大学信息工程学院，太原 030024；3. 山西财经大学统计学院，太原 030006)

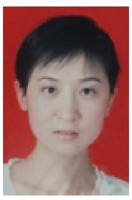
摘要：对于大规模数据量的语音识别问题，支持向量机的训练成为一个难题。预选取支持向量是解决这一难题的方法之一。提出一种新的支持向量预选取算法.*一方面对原数据集的每类数据分别进行核模糊 C 均值聚类，将所有的聚类中心作为每类数据的表征集；另一方面根据支持向量的几何分布含义并借鉴支持向量机的多类分类算法中一对一方法的思路提取原数据集的边界样本作为预选取支持向量进行训练和预测，并将该算法应用于嵌入式语音识别系统中，*实验结果表明：该方法提高了语音识别系统的训练效率，降低了计算代价，同时保持了较高的识别率。

关键词：支持向量；多类分类；核模糊 C 聚类；样本预选取算法；语音识别系统仿真

中图分类号：TP391.9　　文献标识码：A　　文章编号：1004-731X (2015) 11-2714-08

引言

Support vector machine (SVM) based on statistical learning theory of VC dimension theory and structure risk minimum principle has been successfully applied in speech recognition[1-3]. However, with the increase of scale of speech recognition system, SVM algorithm complexity exponentially rises followed by quadratic programming (QP) problems, leading to large amount of calculation and slow training speed. Thus SVM is incapable of application of large-scale data that has



Received: 2014-12-24　　Revised: 2015-03-30;
Foundation item: Shanxi Scholarship Council of China (2009-28); Natural Science Foundation of Shanxi Province (2009011022-2);
Biography: Hao Rui (1978-), W, PhD, Lecturer. Research interests: Artificial Intelligence.

become a major factor affecting the development of SVM.

When SVM is used to classification, not all the training samples are functional for classification, but only a few called support vectors are functional. Moreover the support vectors are geometry distributed in the boundary of the original data. Therefore, if this part of the support vectors are extracted from the original data set before training and SVM can be trained only by support vectors, the number of training samples can be obviously reduced while maintaining the recognition accuracy of the model. Various methods for the support vectors pre-extracting have been proposed. Currently SVM pre-extracting algorithms are grouped into the following categories: the chunking algorithm[4] uses iterative methods to find all the support vectors. However, the choice of initial chunking is arbitrary which leads to increase the number of iterations to affect SVM training speed. The decomposing algorithm[5] brakes down the original QP problems into the smaller QP sub-problems. It solves the problem of high space complexity. But for the large-scale problems which have more support vectors, it still has the defects of slow convergence speed. The methods exploiting the geometric distribution of the training data have been reported such as K-NN algorithm[6], center distance ratio method[7], sample density method[8] and vector projection method[9-11] etc. These methods above are simple and effective, but they have the large amount of calculation and are not suitable for large-scale data problems.

In this paper, we propose an improved SVM pre-extracting algorithm of support vectors. We use kernel fuzzy C-Means clustering to extract typical sample points as a representative set of the original data. Then we extract a certain ratio sample points which are adjacent to other class points, as boundary vectors. The algorithm can not only extract support vectors fast and efficiently but also reduce the number of training samples and the requirements of computation while maintaining the high recognition rate.

1 Non-linear SVM

For non-linear separable problem, given a data set

$$T=\{(x_1,y_1),(x_2,y_2)\cdots,(x_n,y_n)\}\in(R^d\times Y)^n \quad (1)$$

In this case $x_i\in R^d, y_i\in Y=\{1,-1\}$, $1\leqslant i\leqslant n$ are the input data. Kernel function $K(x_i,x_j)$ and penalty parameter $C>0$ are introduced to construct and solve convex QP problems then

$$\min_{\alpha} w(\alpha)=\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j K(x_i,x_j)-\sum_{i=1}^{n}\alpha_i$$
$$\text{S.T.}:\sum_{i=1}^{n}\alpha_i y_i=0 \quad (2)$$

In this case $0\leqslant\alpha_i\leqslant C, 1\leqslant i\leqslant n$ is the Lagrange coefficient. We solve Eq. (1) to obtain $\alpha^*=(\alpha_1^*,\alpha_2^*,\cdots\alpha_n^*)^T$ then the decision function is given by the following:

$$f(x)=\mathrm{sgn}\left(\sum_{i=1}^{n}\alpha_i^* y_i K(x_i^*,x)+b^*\right) \quad (3)$$

In this case $\alpha_i^*\in R$ is the expansion coefficient, associated with x_i^* called support vectors in this paper. From Eq.(3) we can see not all of the training samples are functional, but only the training samples corresponding to non-zero component α_i^* of solution α^* for quadratic programming problem are functional to the decision function. Namely, only the corresponding training samples of support vectors can contribute to the decision function.

2 A novel support vector pre-extracting method

From derivation formula of nonlinear SVM and

第 27 卷第 11 期 系统仿真学报 Vol. 27 No. 11
2015 年 11 月 Journal of System Simulation Nov., 2015

support vector geometric distribution structure, we come to the conclusion that not all of samples play a decisive role in classification for given input samples. Only a small part of the samples (the support vectors) support the classification hyper-plane and the rest of the samples for the establishment of the classification hyper-plane are redundant information and even noise. Hence, if we directly input all the samples into SVM to classification and modeling without any sample pretreatment, on the one hand, a lot of computing and storage space are wasted, especially when dealing with large-scale data; on the other hand, the redundant information and noise input may lead to a drop down the classification accuracy and generalization ability. If according to certain rules and principles, useful support vectors are extracted out from the original training samples before the training, the above problems can be improved. Thus, we put forward a new support vector pre-extracting method. Firstly, from the perspective of data mining, clustering method is used to extract typical representative sample point in each class, namely all samples in each class are characterized with a small number of key sample points. Then, extract boundary vectors that are adjacent to the other class as support vectors to classification and modeling. The method can retain the maximum classification information of the original sample. Thus the higher recognition rate is obtained.

2.1 The representative samples extraction based on the kernel fuzzy C-means clustering

For large-scale data, the sample distribution of the input space tends to be dense. Each area distributes a large number of samples, which leads to sample overlapping and redundancy. Each area can use the key samples representing all of the original sample data to achieve the purpose of streamlining large data samples while almost no loss of sample information. The method of clustering divides the data into several subsets according to the similarity, and each subset is represented by the center of the clustering. The kernel fuzzy C-means clustering (KFCM) is a clustering algorithm based on the objective function. KFCM is the typical method to the non-linear data-set partition which advantage lies in the feature differences between the samples are expanded; the structure relation of the data sets are simplified; the clustering is easier to be implemented in the feature space because the method maps the data into a high dimensional feature space employing nonlinear mapping. Thus in this paper we choice KFCM clustering method to extract the representative samples in the original training sample. The objective function is shown by the following:

$$J=(U,V)=\sum_{i=1}^{n}\sum_{j=1}^{l}u_{ij}^{\Upsilon}\left\|\Phi(x_j)-\Phi(v_i)\right\|^2 \tag{4}$$

where $x_j\,(j=1,2,...,l)$ is the feature vectors of samples; n is the number of clustering; $U=[u_{ij}]_{n\times l}$ is the membership degree matrix; $V=[v_i]$ is the clustering center matrix; Υ is defined by the weight coefficient of membership degree. Let the kernel function be $K(x,y)=\Phi(x)^T\,\Phi(y)$ then

$$J=\sum_{i=1}^{n}\sum_{j=1}^{l}u_{ij}^{\Upsilon}[K(x_j,x_j)-2K(x_j,v_i)+K(v_i,v_i)] \tag{5}$$

Gaussian kernel is an excellent performance of nonlinear kernel which can better calculate the nonlinear classification hyper-plane. Thus we employ Gaussian kernel as follows:

$$K(x,y)=\exp\left(-\frac{\|x-y\|^2}{2\delta^2}\right) \tag{6}$$

If Gaussian kernel is employed, $K(x,x)=1$.The objective function is stated in expression (7).

$$J = 2\sum_{i=1}^{n}\sum_{j=1}^{l} u_{ij}^{\Upsilon}\left(1 - K\left(x_j, v_i\right)\right) \tag{7}$$

According to the Lagrange multiplier method, we obtain the membership matrix U and cluster center matrix v^i as follows:

$$u_{ij} = \frac{\left[1 - K\left(x_j, v_i\right)\right]^{1/1-\Upsilon}}{\sum_{i=1}^{n}\left[1 - K\left(x_j, v_i\right)\right]^{1/1-\Upsilon}} \tag{8}$$

$$v_i = \frac{\sum_{j=1}^{l} u_{ij}{}^{\Upsilon} K(\mathrm{x}_j, v_i) x_j}{\sum_{j=1}^{l} u_{ij}{}^{\Upsilon} K(x_j, v_i)} \tag{9}$$

KFCM algorithm extracts the cluster centers of each class. The extracted cluster centers are as a representative set for each class. Let n_i denote the number of clusters of the i_{th} class, where $i(1 \leqslant i \leqslant m)$. Let m denote the number of classes. The initial cluster centers are randomly generated. Obtain the cluster centers of the i_{th} class $V^i = [v_1^i, v_2^i, \cdots, v_{n_i}^i]$ and the class-labels of cluster centers $Y^i = [i, i, \cdots, i]_{1\times n_i}$, $(1 \leqslant i \leqslant m)$. Sum up all of the cluster centers and obtain the typical sample set of training sample X, that is:

$$V = [V^1, V^2, \cdots, V^m], Y = [Y^1, Y^2, \cdots, Y^m]$$

2.2 Boundary vectors extraction

Support vectors from the geometric position, are distributed the border area adjacent to the two classes, which is the nearest sample points from the different class. We give the boundary vectors extraction process.

Definition 1: Given the sample set $V^i = [v_1^i, v_2^i, \cdots, v_{n_i}^i]$ mapped in the high-dimensional feature space by the nonlinear mapping function Φ. The class center is denoted as follows:

$$v^i{}_0 = \frac{1}{n_i}\sum_{k=1}^{n_i}\Phi(v_k^i) \tag{10}$$

Definition 2: The distance between the two classes of samples mapped in the high-dimensional feature space by the nonlinear mapping function Φ can be calculated as

$$d(v_i, v_j) = \|\Phi(v_i) - \Phi(v_j)\| = \sqrt{K(v_i, v_i) - 2K(v_i, v_j) + K(v_j, v_j)} \tag{11}$$

Given the sample set of two classes $V^i = [v_1^i, v_2^i, \cdots, v_{n_i}^i]$ and $V^j = [v_1^j, v_2^j, \cdots, v_{n_j}^j]$. The class centers are $v^i{}_0$ and $v^j{}_0$ respectively. Thus the distance between the sample v^i and the two class centers are represented by

$$d(v^i, v_0^j) = \sqrt{K(v^i, v^i) - \frac{2}{n_j}\sum_{k=1}^{n_j} K(v^i, v_k^j) + \frac{1}{n_j^2}\sum_{k=1}^{n_j}\sum_{l=1}^{n_j} K(v_k^j, v_l^j)} \tag{12}$$

$$d(v^i, v_0^i) = \sqrt{K(v^i, v^i) - \frac{2}{n_i}\sum_{k=1}^{n_i} K(v^i, v_k^i) + \frac{1}{n_i^2}\sum_{k=1}^{n_i}\sum_{l=1}^{n_i} K(v_k^i, v_l^i)} \tag{13}$$

Theorem 1: Given the sample set of two classes $V^i = [v_1^i, v_2^i, \cdots, v_{n_i}^i]$ and $V^j = [v_1^j, v_2^j, \cdots, v_{n_j}^j]$. For any sample point v^i in the i_{th} class, if $DR(v^i) = d(v^i, v_0^i) / d(v^i, v_0^j) > \varepsilon$, where ε is threshold, v^i is distributed within border areas of two class which is belong to the support vectors set S^i and mark the sample, or v^i is out of the area and not marked.

For the SVM multi-classification problem, combined with the classification strategy of one-versus-one algorithm[12], extract the sample belonging to the border area adjacent to the two classes and mark them for combination of all the class centers. Then according to the number of occurrences, sort all the sample in each representative set by descending order. Finally, according to the proportion $P\left(0 < P < 1\right)$ which will extract the number of samples in the total samples, decide the number of retained samples in each class.

2.3 Detailed support vector pre-extracting process

Step 1. Initialization: set the accuracy of the

第 27 卷第 11 期 系统仿真学报 Vol. 27 No. 11
2015 年 11 月 Journal of System Simulation Nov., 2015

objective function e, fuzzy weighting factor Υ, the number of clusters n, the maximum number of iterations M_t;

Step 2. Input the training sample set X, employ KFCM clustering to extract the representative set $V^i, (1 \leqslant i \leqslant m)$, sequentially perform the following steps;

Step 3. Calculate and update the membership U^i of each division with Eq. (8);

Step 4. Calculate and update each cluster center V^i with Eq. (9);

Step 5. Repeat steps (3) and (4) until meeting the convergence criteria, namely $|J_t - J_{t-1}| < e$ or $t > M_t$. Otherwise, go to Step2;

Step 6. Sum up all the cluster centers of each class and get the representative set V;

Step 7. Calculate all samples in v^i. If $DR(v^i) > \varepsilon$ mark the sample point according to boundary vectors extraction process;

Step 8. Sort all the samples by descending order under the tag values. According to the proportion P, decide whether to belong to the boundary vector set S^i;

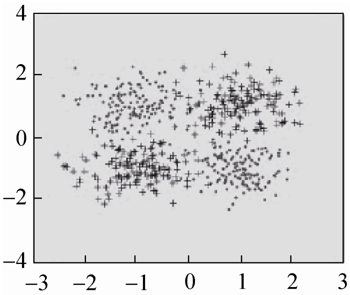
Step 9. Finally $S = S^1 \cup S^2 \cup \cdots \cup S^m$ is the pre-extraction support vector set.

3 Simulation experiments and analysis

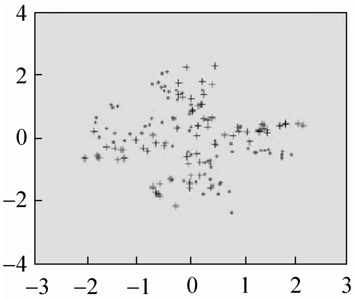
3.1 Synthetic data

We do simulation experiment for synthetic data to verify the effectiveness of the proposed method. The results are shown in Fig.1.

The training samples are the two randomly generated classes obeying Gaussian distribution. We firstly employ the SVM for training 600 original samples. As a result, 40% of the training samples are the support vectors and the training time takes 20s. On the contrary, after the application of pre-extracting algorithm, 95% of the training samples are the support vectors and the training time reduces to 1.5s. The experiment results show that the pre-extracting algorithm can improve the utilization rate of data and decrease redundant data in the process of the SVM training and The training speed is greatly improved due to reduction in the number of sample data.



(a) Original sample data



(b) Pre-extraction sample data

Fig. 1 Pre-extraction results

3.2 Embedded Speech Recognition Platform

In this section, we employ the hardware platform based on TI's DM6446 dual-core processor and use Linux embedded operating system. DM6446 chip employs ARM and DSP dual-core architecture. ARM subsystem employs 297 MHz ARM926 core and DSP part employs 594MHz C64X + DSP core. The kernel of development broad uses Linux 2.6.18. Cross-compiler tools use Arm-linux-gcc-3.4.1 and software packages uses Class Path -0.94. tar and Java VM 1.4.6.tar.

By cross-compiler, Java virtual machine is implanted to DM6446 development board. After testing, each part can work stable and the software can meet the project requirements from function and performance. Thus we can carry out the speech recognition simulation experiments by embedded speech recognition system. We use programming environment Matlab 7.0.

3.3 Experimental results and analysis

The experimental data from Korean non-specific small vocabulary isolated word speech database. We adopt the 16 personal pronunciations for 50 isolated words which obtain under the condition of different SNR (0dB, 15dB and CLEAN). Noise is Gaussian white noise. The characteristic sample data are obtained by the MFCC from original speech. Specific parameter configurations in the experiments: Gaussian kernel function is unified employed; With respect to the experimental parameters of the pre-extracting algorithm, we determine the reasonable values $\varepsilon = 0.3$, $P = 0.75$ by repeated experiments; For SVM, according to grid parameter optimization algorithm, we set $C = 1.0718$, $\delta = 0.2333$. The experiment process is that firstly the speech training set is preprocessed by the pre-extracting algorithm. Then the preprocessed training set emploies directed acyclic graph (DAG) multi-class SVM classification algorithm for speech recognition. The multiple classification algorithm of DAG is implanted to the DM6446 development board.

In order to compare execution efficiency and recognition accuracy of the proposed algorithm and standard SVM, we count up the training time and recognition rate respectively in 10 words, 20words, 30 words, 40 words, 50 words of sample sets, under the condition of different SNR. The statistical data shown in table 1, table 2 and table 3 are the mean values of repeat 5 times experiments. Comparisons of training time under the condition of different SNR are shown in Fig. 2, Fig. 3 and Fig. 4.

The experimental results indicate that under the different SNR, the training time is obviously reduced while maintaining the high recognition rate. Maximum reduction is from the original 1687s to 684s under the same condition. Relative to the standard SVM, it decreases by 59.4%. Furthermore, with increasing sample size the effect of the method is more prominent. The recognition rate of the training sample set bring down with vocabulary increase and SNR decrease. But as show in Fig. 3, the recognition rate still remains at a high level, even if SNR is 0dB, which is affected to some extent because the noise is relatively strong.

Table. 1 Comparison of the classifying performance (SNR=15dB)

Training algorithm	Vocabulary	nSV	Training time/s	SNR (15dB) /%
Standard SVM	50	1 204	1 485	92.71
Pre-extracting+SVM	50	674	699	91.03
Standard SVM	40	1 006	1 085	93.37
Pre-extracting+SVM	40	521	466	91.02
Standard SVM	30	694	590	92.14
Pre-extracting+SVM	30	397	318	91.56
Standard SVM	20	540	512	92.43
Pre-extracting+SVM	20	400	256	91.95
Standard SVM	10	270	267	94.67
Pre-extracting+SVM	10	140	80	92.01

第 27 卷第 11 期　　系统仿真学报　　Vol. 27 No. 11
2015 年 11 月　　Journal of System Simulation　　Nov., 2015

Table. 2　Comparison of the classifying performance (SNR=CLEAN)

Training algorithm	Vocabulary	nSV	Training time/s	SNR (CLEAN) /%
Standard SVM	50	1 195	998	96.52
Pre-extracting+SVM	50	610	578	94.43
Standard SVM	40	999	879	97.23
Pre-extracting+SVM	40	502	401	94.85
Standard SVM	30	678	542	97.08
Pre-extracting+SVM	30	364	261	95.01
Standard SVM	20	498	304	97.43
Pre-extracting+SVM	20	366	147	95.95
Standard SVM	10	251	180	98.67
Pre-extracting+SVM	10	132	78	96.03

Table. 3　Comparison of the classifying performance (SNR=0dB)

Training algorithm	Vocabulary	nSV	Training time/s	SNR (15dB) /%
Standard SVM	50	1 295	1 699	83.19
Pre-extracting+SVM	50	694	714	80.87
Standard SVM	40	1 016	1 107	83.03
Pre-extracting+SVM	40	599	483	81.79
Standard SVM	30	754	637	84.47
Pre-extracting+SVM	30	429	374	81.05
Standard SVM	20	565	539	85.76
Pre-extracting+SVM	20	416	274	83.25
Standard SVM	10	293	295	87.65
Pre-extracting+SVM	10	168	89	85.59

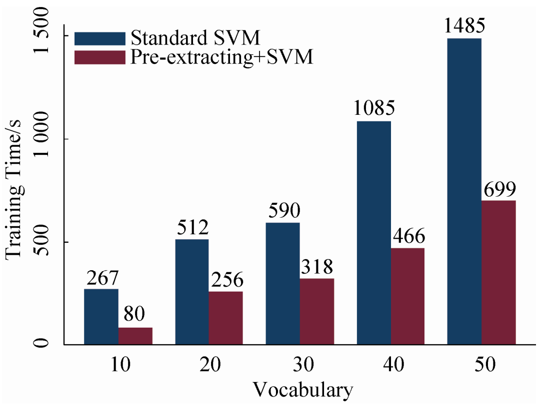


Fig. 2　Comparisons of training time when SNR=15dB

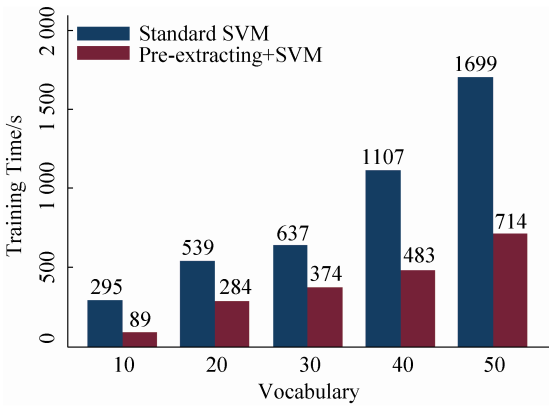


Fig. 4　Comparisons of training time when SNR=0dB

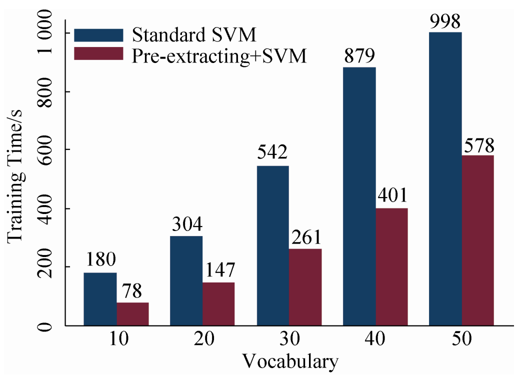


Fig. 3　Comparisons of training time when SNR=CLEAN

4　Conclusion

In this paper, we propose a novel support vector pre-extracting method which combines the advantages of kernel fuzzy C-Means clustering and boundary vector extraction algorithms. By reducing the redundancy of training samples, the sample pre-extracting algorithm is more focused on the optimization of support vectors. As a result, it is successfully applied in the embedded speech recognition system. The results of simulation experiment demonstrate that the method can improve the recognition speed while effectively guarantee the recognition accuracy of the model. Next work, we consider validating the proposed algorithm on large-scale data.

References:

[1] Hanilçi C, Ertaş F. Investigation of the effect of data duration and speaker gender on text-independent speaker recognition [J]. Computers & Electrical Engineering (S0045-7906), 2013, 39(2): 441-452.

[2] C H You, K A Lee, H Li. An SVM kernel with GMM-super vector based on the Bhattacharyya distance for speaker recognition [J]. IEEE Signal Processing Letters (S1070-9908), 2009, 16(1): 49-52.

[3] Sangeetha J, Jothilakshmi S. A novel spoken keyword spotting system using support vector machine [J]. Engineering Applications of Artificial in Intelligence (S0952-1976), 2014, 36(11): 287-293.

[4] Zhifeng Hao, Shu Yu, Xiaowei Yang, *et al*. Online LS-SVM Learning for Classification Problems Based on Incremental Chunk [C]// Lecture Notes in Computer Science (S0045-7906), 2004, 3173: 558-564.

[5] E Osuna, R Freund, F Girosi. An improved training algorithm machines [C]// Proceedings of the 1997 IEEE Workshop on Neural Net Processing. New York, USA: IEEE Press, 1997: 276-285.

[6] Palaniappan R, Sundaraj K, Sundaraj S. A comparative study of the svm and k-nn machine learning algorithms for the diagnosis of respiratory pathologies using pulmonary acoustic signals [J]. BMC Bioinformatics (S1471-2105), 2014, DOI: 10.1186/1471-2105-15-223.

[7] Jiao Licheng, Zhang Li, Zhou Weida. Pre-extracting Support Vectors for Support Vector Machine [J]. Electronica Sinica (S0372-2112), 2001, 29(3): 383-386.

[8] Zhang Bin, Tang Zhaohui, Zhu Hongqiu, *et al*. Novel Simplifying Method for Support Vector Machines and Its Application [J]. Journal of System Simulation (S1004-731X), 2012, 24(2): 344-247.

[9] Yang Jing, Yu Xu, Xie Zhiqiang. Support Vectors Pre-Extracting Method based on Improved Vector Projection [J]. Chinese Journal of Computers (S0254-4164), 2012, 35(5): 1002-1010.

[10] Ye Qiaolin, Ye Ning, Cui Jing, *et al*. Multisurface Support Vector Machines via Weight Vector Projection [J]. Pattern Recognition and Artificial Intelligence (S1003-6059), 2010, 23(5): 708-713.

[11] Zhang Jianpei, Zhao Ying, Yang Jing. Incremental Learning Algorithm of Support Vector Machine Based on Vector Projection [J]. Computer Science (S1002-137X), 2008, 35(3): 164-166.

[12] Arruti A, Mendialdua I, Sierra B, *et al*. New One Versus (One)(All) method: NOV@ [J]. Expert Systems with Applications (S0957-4174), 2014, 41(10): 6251-6260.