Cluster-based Support Vector Machines in Text-Independent Speaker Identification

Sheng-Yu Sun, C.L. Tseng, Y.H. Chen, S.C. Chuang, H.C. Fu
Department of Computer Science and Information Engineering,
National Chiao-Tung University, Hsinchu, Taiwan,
E-mail: {sysun, tsengcl, yuehhong, scchuang, hcfu}@csie.nctu.edu.tw

***Abstract*— Based on Statistical learning theory, Support Vector Machines(SVM) is a powerful tool for various classification problems, such as pattern recognition and speaker identification etc. However, Training SVM consumes large memory and long computing time. This paper proposes a cluster-based learning methodology to reduce training time and the memory size for SVM. By using k-means based clustering technique, training data at boundary of each cluster were selected for SVM learning. We also applied this technique to text-independent speaker identification problems. Without deteriorating recognition performance, the training data and time can be reduced up to 75% and 87.5% respectively.**

I. Introduction

Using GMM to perform speaker identification usually can reach rather high accuracy [13]; however the rate drops when the number of speakers grows. SVM is the discriminative classifier while GMM is the generative probability classifier [1]. We can not clearly control the identification rate due to the uncertainty in probability, while SVM provides a theoretical value of error upper bound and a more complete math model to adjust the number of support vectors, the VC dimension and others to minimize the error upper bound [2], [3]. SVM claims that if data of different classes are properly transformed to the space with high enough dimension, we can find suitable classifiers to classify them. SVM possesses user-friendly property and a sound theory basis [2], but it requires larger memory size when dimension gets higher and database bigger. Furthermore, calculating complexity would also rise. How to conquer the hardship of speeding up SVM and reduce the data required is worth of further studying. There have been numerous methods proposed to speed up SVM, such as decomposition methods; "chunking method [3], [4]" proposed by Boser, Guyon and Vapnik; [5]by Osuna, Freund and Girosi; and SMO (Sequential Minimal Optimization) [6]. Furthermore, the pre-processing methods are also applicable, e.g. [7] lowers the dimension and speeds up training process by picking M feature vectors of higher importance in the feature space, and then transforms data from the input space to the space constituted by M feature vectors. On the other hand, [8] selects M feature vectors of higher importance in the input space. The selection is done by SVM method, which makes it twice to use SVM during the training process. Data selection is also employed to minimize database. The data are repeatedly clustered and trained by SVM to find its support vectors[9]. The disadvantage of these pre-processing methods is time consuming. To save the processing time, we propose cluster-based learning methodology. By K-means clustering we can select important data effectively as the training data. The rest of this paper is organized as the following. Section 2 shortly describes SVM. Section 3 introduces clustering methodology and the framework of this paper. Results and experimental procedure is recorded in Section 4. Finally, the conclusions and our future works are given in Section 5.

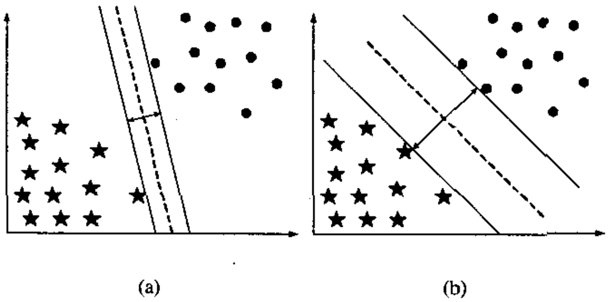


Fig. 1. (a)The separable hyperplane with smaller margin (b)The separable hyperplane with larger margin. It is the so-called optimal separable hyperplane (OSH).

II. Overview of Support Vector Machines

The paper only give a brief description of SVM. The interested reader is referred to [2], [3] for a more detailed description. SVM is a binary classifier, which searches for the optimal decision boundary in two classes of data. Basically, there are two cases of SVM, the separable and the non-separable which can be further classified to be linearly separable, linearly non-separable, non-linearly separable, and non-linearly non-separable.

A. Linear Support Vector Machines

In the simplest linearly separable case, given a data set $\{\mathbf{x}_i, y_i\}, i = 1, ..., l, y_i \in \{-1, +1\}, \mathbf{x}_i \in \mathbf{R}^d$, we hope to find a hyperplane to separate positive and negative data in this data set. Consider the family of the decision functions

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b \qquad (1)$$

0-7803-8359-1/04/$20.00 ©2004 IEEE

and find suitable $\mathbf{w}_o, b_o$ such that $sgn(\mathbf{w}_o^T\mathbf{x}_i + b_o) = sgn(y_i)$. The ***margin*** is defined as the shortest distance from the separating hyperplane to the closest positive or negative data. The Optimal Separable Hyperplane(OSH) is the hyperplane that can completely separate the two classes while the margin is maximized as fig.1. $\mathbf{w}$ is determined by the following constrained optimization problem

$$min_{\mathbf{w}}\frac{1}{2}\|\mathbf{w}\|^2$$

subject to

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, i = 1, ..., n. \qquad (2)$$

$\mathbf{s}_i$ is called 'support vector" and is the vector nearest to OSH and satisfies the equation,

$$y_{s_i}(\mathbf{w}_o^T\mathbf{s}_i + b_o) = 1$$

, if $\mathbf{w}_o$ is the optimal solution.

In linear non-separable cases, SVM searches for the maximum margin and the minimum error. As a result, we need a new variable, ***slack variables***, ξ_i, and solve the optimization problem below:

$$\min_{\mathbf{w}}\frac{1}{2}\|\mathbf{w}\|^2 + c\sum_{i=1}^{n}\xi_i$$

subject to

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, i = 1, ..., n. \qquad (3)$$

where $c > 0, \xi_i > 0$. c is the misclassification penalty parameter and control the degree of penalty to the misclassification samples.

The above optimal problem can be solved by Lagrange method [2], [3], and its decision function is

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b = \sum_{i=1}^{N_s}\alpha_i y_{s_i}\mathbf{s}_i^T\mathbf{x} + b \qquad (4)$$

where $\alpha_i, i = 1, ..., N_s$, are positive Lagrange multipliers, $\mathbf{s}_i, i = 1, ..., N_s$, are the corresponding support vectors with $\alpha_i > 0$, and N_s is the number of support vectors.

B. Nonlinear Support Vector Machines

In the real world, linear classifier can solve only a few problems. Most problems are non-linear cases. How does SVM solve these problems? SVM transforms the data nonlinearly in input space to a higher dimension feature space, and finds OSH with linear support vector machines. Let

$$\Phi : R^d \rightarrow F$$

be a nonlinear transformation from input space to feature space and the decision functions defined as

$$f(\mathbf{x}) = \sum_{i=1}^{N_s}\alpha_i y_{s_i}\Phi(\mathbf{s}_i)^T\Phi(\mathbf{x}) + b. \qquad (5)$$

TABLE I

KERNEL FUNCTIONS

Classifier Type	Kernel function
Polynomial	$K(\mathbf{x}, \mathbf{x}_i) = [(\mathbf{x} \cdot \mathbf{x}_i) + 1]^p$
Radial basis function(RBF)	$K(\mathbf{x}, \mathbf{x}_i) = exp\{-\frac{\lvert\mathbf{x}-\mathbf{x}_i\rvert^2}{\sigma^2}\}$
Sigmoid function	$K(\mathbf{x}, \mathbf{x}_i) = tanh[k(\mathbf{x} \cdot \mathbf{x}_i) + \delta]$

To find OSH in feature space involves inner product of two vectors. Besides, feature space is a high dimension space, so it needs more effort to calculate. SVM employs Mercer kernels $k(\mathbf{s}_i, \mathbf{x})$, where $k(\mathbf{s}_i, \mathbf{x}) = \Phi(\mathbf{s}_i)^T\Phi(\mathbf{x})$, to speed up training process, so the decision functions become

$$f(\mathbf{x}) = \sum_{i=1}^{N_s}\alpha_i y_{s_i}k(\mathbf{s}_i, \mathbf{x}) + b \qquad (6)$$

This paper employs polynomial kernel functions as kernel functions for which SVM is one of the frequently adopted kernel functions as Table I shows.

III. DATA SELECTION BY CLUSTERING

Clustering is the method to classify data. Since the data in the same cluster usually has similar characteristic, we can select some data in the same cluster rather than all of them to reduce the amount of training data. There are various ways of clustering, such as K-means, SOM, GMM, etc. However, clustering speed is critical to speed up SVM training. The clustering method employed in this paper is K-means clustering. The original K-means algorithm takes long time to calculate. Some of more efficient versions of the algorithm have been proposed [10], [11]. The algorithm [11] proposed by Alsabti et al. is to save data in Kd-tree [12], and is properly pruned to minimize the effort to calculate distance of all points in every iteration. We adopt the algorithm to speed up the K-means clustering.

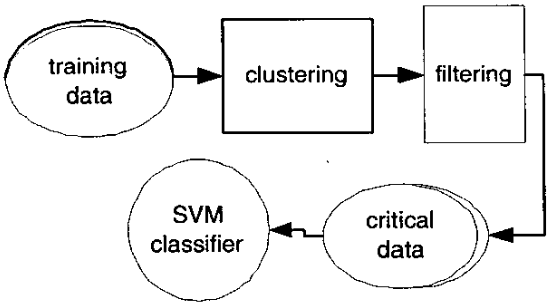


Fig. 2. The framework of the proposed method

A. Our Framework

Figure 2 shows the framework of our system. First, we cluster the data and filter out ***critical data***, which significantly affect SVM learning. The filtering algorithm is described as follows.

```
Let CS be the set of critical data
Initial CS ← ∅
for each cluster $C_j$
    for each data $\mathbf{x}_{ij}$, where $\mathbf{x}_{ij} \in C_j$
        if $distance(\mathbf{x}_{ij}, \mathbf{c}_j) > T_j$, where $\mathbf{c}_j$ is the centroid
        of $C_j$ and $T_j$ is a threshold
            $CS \leftarrow \mathbf{x}_{ij}$
    end for
    $CS \leftarrow \mathbf{c}_j$
end for
```

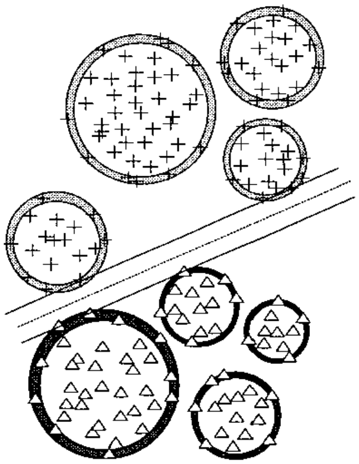


Fig. 3. The original training data with two classes and four clusters each.

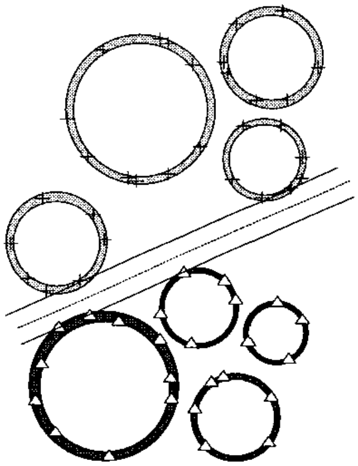
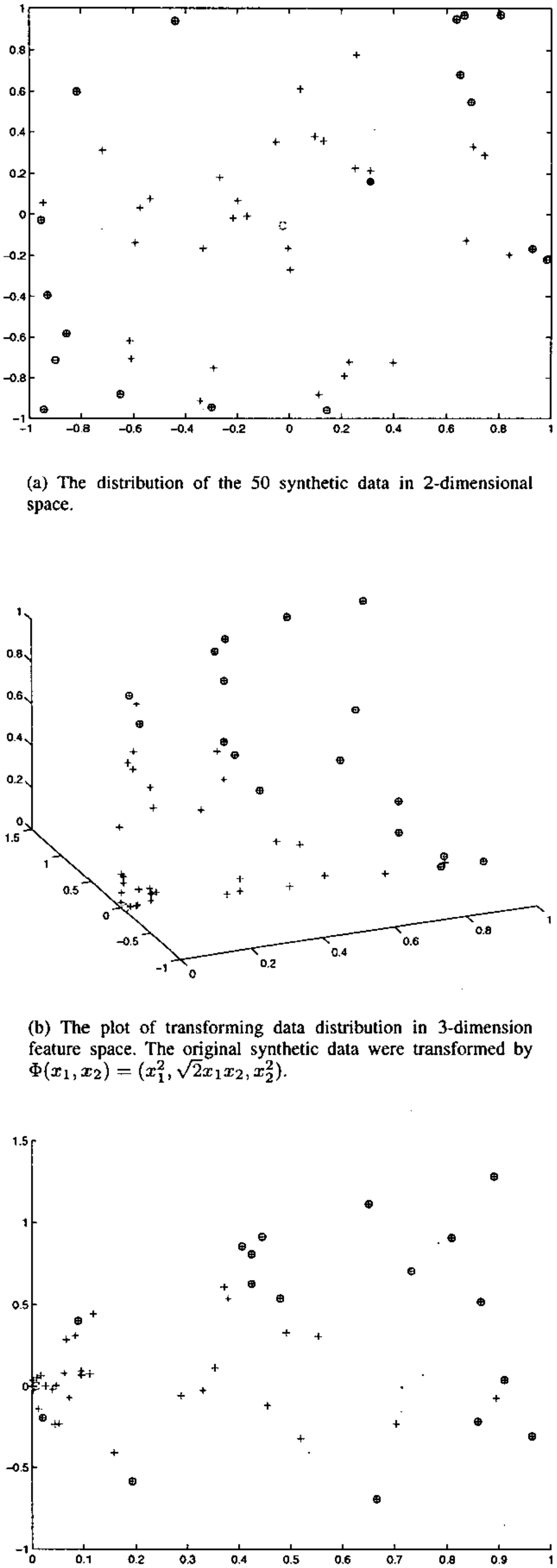


Fig. 4. The training data after selecting

Every cluster C_j is corresponding to a threshold T_j. T_j varies with the size of a cluster. T_j and radius of a cluster is positively related. The longer the radius is, the bigger the Tj



(a) The distribution of the 50 synthetic data in 2-dimensional space.

(b) The plot of transforming data distribution in 3-dimension feature space. The original synthetic data were transformed by $\Phi(x_1, x_2) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$.

(c) The vertical view of (b) as we can see that the selected data are almost distributed around the boundary of the distribution.

Fig. 5. The figures exhibit a 3-dimension feature space corresponding to a non-linear transformation in 2-dimension input space. The circle and "C" mark the selected data.

is. Because the data that decisively affects SVM classifiers is those at boundary of each class, the data over the cluster is selected to be our *critical data*, shown as Fig. 3, 4. Fig.3 is the distribution of original data which the light and dark gray areas represent the data at the boundary among all clusters of data. Fig.4 shows the SVM classifier found by training data, which are at boundary of each class and also the *critical data*. Although the OSH found by SVM is the OSH in feature space for the training data, we believe that data originally distributed over the input space would be as well located mostly over the feature space when nonlinear transformation of polynomial kernel function transforms the data in input space to feature space. To describe the statement we make up 50 synthetic data points and distribute them in 2-dimensional input space (shown as Fig. 5a). Then, we transform data into 3-dimensional feature space with non-linear transformation $\Phi(\mathbf{x}) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$, where $\mathbf{x} = (x_1, x_2)$ (shown as Fig. 5b). Fig. 5 exhibits that the data in the boundary of the input space mostly distribute over the outside in the feature space through the non-linear transformation. In addition to that, the center of each cluster represents the main characteristic of the cluster. When the data of clusters is transformed to feature space, the boundary of the other side is settled as well. We take the data over the boundary and centroid of the cluster in input space as our *critical data.* Fig 5c is the vertical view of Fig. 5b, and it shows that the selected data mostly distributed over the boundary through the figure.

While clustering, we apply K-means algorithm to data of each class respectively. Then, apply ’filtering” in selecting *critical data* from K clusters, and take the *critical data* as the training data for SVM.

B. Multi-classification Problems

To solve binary classification problem, we can directly apply SVM on the problem because SVM is basically designed to solve the binary classification. To tackle N-class classification problem, we have binary-SVM as the basis and construct N*(N-1)/2 classifiers where each one is trained on data from two classes. The method is the “one-against-another method“. The other one is “one-against-rest method“, which is less effective than the former one [14]. This paper adopts the “one-against-another method“.

IV. Experiments and Results

The paper adopts TCC-300 speech dataset [15], [16] as the experimental database. The silence part in the experimental utterances is removed before extracting feature. The first 13 dimensional MFCC and their first derivatives are used to form a 26 dimensional feature vector calculated every 10 ms with subsequent mel-cepstral mean subtraction to compensate for channel effects and normalization. We randomly pick 20 speakers out of total 40 speakers with equal number of male and female. A 30-second utterance is collected from each speaker as the training data, and several 5-second utterances are used as testing data. The LIBSVM [18] package was used to train an SVM for all the experiments in this research. The result shows that by selecting the number of clusters (K) to be 20, and choosing the polynomial kernel function with ordre of 2, coefficient 4 and penalty parameter 250, the proposed method can achieve better performance.

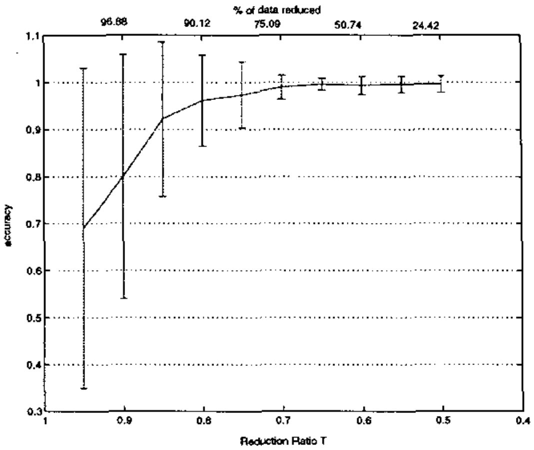


Fig. 6. The speaker identification accuracy curve of the proposed method, with k (=20) clusters and data reduction ratio T from 0.95 $\sim$ 0.5, among 20 speakers. The % of data reduced means the amount data points can be saved or deleted. The length of training utterance is 30 seconds and 5-second for testing utterance for a speaker. The mean and standard deviation of the accuracy values are higher than 99% and lesser than 0.03 respectively as T is selected to be ≤ 0.7.

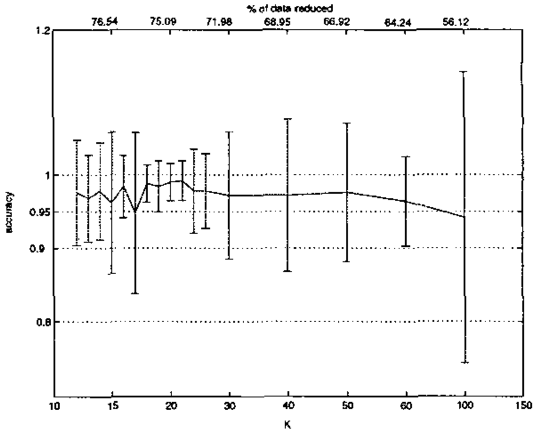


Fig. 7. The accuracy curve of our method with the reduction ratio T to be 0.7, and k=12 $\sim$ 100 for 20 speakers. The Standard deviation are very small when $18 \leq k \leq 21$.

In the following experiments, we exercised different data reduction by varying ratio $T (= T_j/r_j$, where r_j is the radius of C_j)(see the filtering algorithm in Section III-A) and various cluster number k. As shown in Fig. 6, when the reduction ratio T is selected as 0.7, which means three quarter of data points were not used, the identification rates reaches 0.99 with

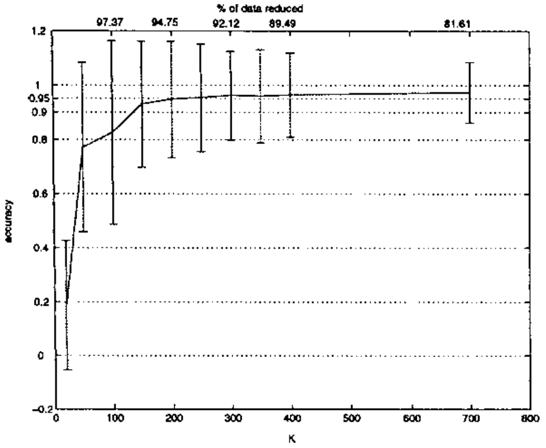


Fig. 8. The accuracy curve of the method proposed by [17], with the number of cluster k from 50 to 700, for 20 speakers. The length of training utterance is 30 seconds and 5-second for test utterance of a speaker. The method achieves good accuracy but with some what large standard deviation.

standard deviation 0.025. When the T value becomes larger, the reduced data points becomes lesser. Since the accuracy value stays almost constant, when the T value decreases from 0.65 to 0.5. Thus, we suggest the T value to be selected between 0.65 and 0.7 for better identification performance.

The second set of experiments are tried to vary the number of cluster k, to see whether it affects the performance. As shown in Figure 7, the accuracy rate varies between 0.992 and 0.94, while k changes from 12 to 100. However, the variance of the identification accuracy changes greatly. The small STD values corresponds to the cluster number k between 18 to 21. Beyond this range, the STD value becomes larger as shown in Figure 7. We suspect that the nature number of cluster could be in that range.

In addition, we have performed experiments by using the method proposed by [17]. As shown in Figure 8, when the cluster number k is smaller than 150, the performance is relatively poor. Then, the accuracy rises over 0.9, as k gets larger than 150. The overall variance seems somewhat large, it is in the range of (0.112~0.241). Furthermore, the clustering time of [17] for k = 700 is 350 times longer than our method for k = 20.

Moreover, we compare identification performance between SVM and GMM(32). As shown in Table II, the proposed data reduction SVM and the traditional SVM, we can see that the performance of these two methods are somewhat comparable to each other. However, the training time has been reduced more than 87.5% and the data storage also has been saved from 25% to 75%. When there are 10 speakers or less to be tested, the identification rates of GMM(32) and SVM method are comparable. When there are more than 20 speakers, the identification rate of GMM(32) is lowest among three methods. When there are 40 speakers, the identification rate of the proposed method obviously rises much more than that of GMM(32). The results points out that (1) our method possesses better robustness, (2) the data removed by our method almost has no effect on identification rate of SVM, (3) the method is able to reduce training time when the identification rate is not varied, and (4) the SVM classifier obtained by the proposed method maintains good robustness. Although our method lowers a little accuracy compared with the traditional SVM, it speeds up training process and reduces the number of support vectors. The identification time is proportional to the number of support vectors. If the accuracy is more important than other factors, we can quickly determine the parameters using our method and train an SVM with these parameters using the traditional method. Otherwise, we can get good performance using our method for speaker-identification problems.

TABLE II

SPEAKER IDENTIFICATION PERFORMANCE

	our method	SVM	GMM(32)
5 speakers	99.6%	100%	100%
10 speakers	98.6%	100%	98.9%
20 speakers	99.0%	100%	93.8%
40 speakers	97.6%	100%	80.3%

V. CONCULSION

We propose a cluster-based data selection method, which can properly filter the data with SVM to significantly reduce the size of database and speed up SVM training process without having any effect on performance. Besides, we found it quite interesting when each cluster center is polynomial transformed to the feature space, the data are automatically located at the boundary of the cluster. We hope to have further study on such spaces transforming cases, e.g. Gaussian, Sigmoidal, etc., and find the generality in them.

ACKNOWLEDGMENT

This work was supported in part by National Science Council, ROC, under the grant NSC-91-2213-E009-215.

REFERENCES

[1] Xin Dong and Wu Zhaohui, 'Speaker Recognition Using Continuous Density Support Vector Machines," ELECTRONICS LETTERS 16th August 2001

[2] Vladimir N. Vapnik, *Statistical Learning Theory*, John Wiley and Sons, Inc., New York, 1998.

[3] Christopher J.C. Burges, 'A Tutorial on Support Vector Machines for Pattern Recognition," Data Mining and Knowledge Discovery, 1998.

[4] B.E. Boser, I.M. Guyon, V.N. Vapnik, 'A Training Algorithm for Optimal Margin Classifi ers," In Proc. 5th ACM Workshop on Computational Learning Theory, pages 144-152, 1992.

[5] Edgar Osuna, Robert Freund, Federico Girosi, 'An Improved Training Algorithm for Support Vector Machines," in Proc. of the 1997 IEEE Workshop on Neural Network for Signal Processing, pages 276-285, 1997.

[6] John C. Platt, 'Fast Training of Support Vector Machines Using Sequential Minimal Optimization," In Advances in Kernel Methods: Support Vector Learning, MIT Press 1998.

[7] Baudat G., Anouar F., 'Kernel-based Methods and Function Approximation," In Proc. IJCNN, pages 1244-1249, July 2001.

[8] K.Z. Mao, 'Feature Subset Selection for Support Vector Machines Through Discriminative Function Pruning Analysis," IEEE Transactions on System, Man., and Cybernetics - part B: Cybernetics, 2003.

[9] Michael Schmidt, Herbert Gish, 'Speaker Identification via Support Vector Classifiers," IEEE ICASSP, 1996.

[10] A.K. Jain, M.N. Murty, 'Data Clustering: A Review," ACM Computing Surveys, vol. 31, no. 3, pp. 264-323, 1999.

[11] K. Alsabti, S. Ranka, and V. Singh, 'An Efficient k-means Clustering Algorithm," Proc. First Workshop High Performance Data Mining, Mar. 1998.

[12] J.L. Bentley, 'Multidimensional Binary Search Trees Used for Associative Searching," Comm. ACM, vol. 18, pp. 509-517, 1975.

[13] D. A. Reynolds and R. C. Rose, 'Robust text-independent speaker identification using Gaussian mixture Speaker Models," IEEE Transactions on Speech and Audio Processing,Vol. 3, No. 1, January 1995.

[14] Chih-Wei Hsu and Chih-Jen Lin, 'A Comparison of Methods for Multiclass Support Vector Machines," IEEE Transactions on Neural Networks, vol 13, pp. 415-425, 2002.

[15] TCC-300 speech database.. Association for Computational Linguistics and Chinese Language Processing, Institute of Information Science, Academia Sinica, Nangkang, Taipei, ROC. [Online]. Available: http://rocling.iis.sinica.edu.tw/ROCLING/MAT/TCC-300brief.htm

[16] Hsiao-Chuan Wang, 'Speech Corpora and ASR Assessment in Taiwan," In Proc. of Oriental COCOSDA Workshop 2000, Oct. 16, 2000, Beijing, China.

[17] Marcelo Barros de Almeida, Antonio de Padua Bragà and Joao Pedro Braga, 'SVM-KM: speeding SVMs learning with a priori cluster selection and k-means," IEEE 6 th Brazilian Symposium on Neural Networks, SBRN 2000.

[18] [Online] Available: http://www.csie.ntu.edu.tw/ cjlin/libsvm