Speaker Identification System Using Hidden Markov Model Based on Local Distribution Membership

Agus Buono[1], Benyamin Kusumoputro[2]
pudesha@yahoo.co.id, kusumo@cs.ui.ac.id
1. Computational Intelligence Research Lab, Dept. of Computer Science,
Bogor Agriculture University Dramaga Campus, Bogor – West Java, Indonesia
2. Computational Intelligence Research Lab, Faculty of Computer Science, University of Indonesia
Depok Campus, Depok 16424, PO.Box 3442, Jakarta, Indonesia

Abstract

The work described in this paper addresses an application of Hidden Markov Model (HMM) with modification in observation probability using two approaches for membership values, i.e. Euclidean distance base and kernel function base. Mel-Frequency Cepstrum Coefficients (MFCC) used as feature extraction. In this research we use "pudesha" as a keyword to identify each speaker and modeled by left-right HMM which its states are clustered using Fuzzy C-Mean clustering. Each new observation, we compute the membership value to every cluster and choose the biggest one as the membership value of the observation to appropriate hidden state. In the unguided-utterance and limited training data, experimental results show that our methods recognize better than classical HMM that uses Normal Distribution as observation probability. It is also showed that the use of Normal distribution for observation probability leads to a singularity problem in computing the inverse of covariance matrix, especially for limited training data. In our proposed approaches, the singularity problem will not occurs, since we do not need to compute the inverse of covariance matrix.

Keywords: *Hidden Markov Model, Mel-Frequency Cepstrum Coefficients, Fuzzy Clustering, Euclidean Distance, Kernal Function*

1. Introduction

Hidden Markov Model that has been widely used for decades in speech recognition and speaker identification has never been claimed perfect for speech modeling [1]. There are some disadvantages [2] with one Gaussian HMM, especially in its assumptions, i.e. normality and independently, and constraint due to limited training data.

The research described here is aimed on developing HMM model as speech classifiers for use in automatic speaker identification. Our improvements are based on the observation probability that uses a membership function. We propose two membership functions, i.e. one based on Euclidean distance and the other based on kernel function. Our approaches could handle non-Normality and singularity problem in standard HMM.

2. Problem Description

The motivation behind our research is to identify a person based on its speech or voice characteristics (a.k.a. Automatic Speaker identification – ASI). Figure 1 presents the generic speaker identification system [3].

Speech is a complicated signal produced as a result of several transformations occurring at several different levels: semantic, linguistic, articulator and acoustic [4]. Differences in these transformations appear as differences in the acoustic properties of the speech signal. Moreover, there are some sources of variability in speech signal, i.e. emotion (stress or duress), aging, and sickness.

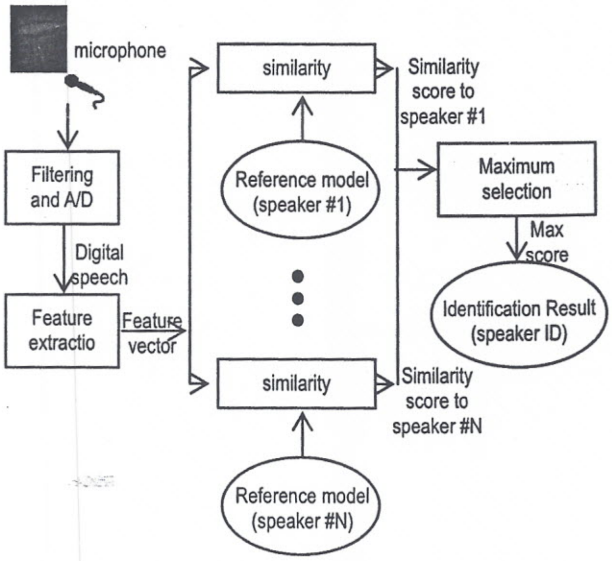

microphone
Filtering and A/D
Digital speech
Feature extractio
Feature vector
similarity
Reference model (speaker #1)
Similarity score to speaker #1
Maximum selection
Max score
Identification Result (speaker ID)
similarity
Similarity score to speaker #N
Reference model (speaker #N)

Figure 1. Generic Speaker-Identification System

In the context of HMM, an utterance is modeled by a directed graph where a node/state represents one articulator configuration that we could not observe directly (hidden state). A graph edge represents transition from one configuration to the successive configuration in the utterance. We model this transition by a matrix, A. In reality, we only know a speech signal produced by each configuration, which we call observation state or observable state. In Gaussian HMM, observable state is a random variable and assumed has Normal or Gaussian distribution with mean vector μi and covariance matrix Σi; (i=1, 2, 3,..., N; N is number of hidden states).Because of the source of variability, each state has a multimodal distribution instead of unimodal distribution. In Gaussian HMM, we handle this problem using mixture Gaussian distribution.

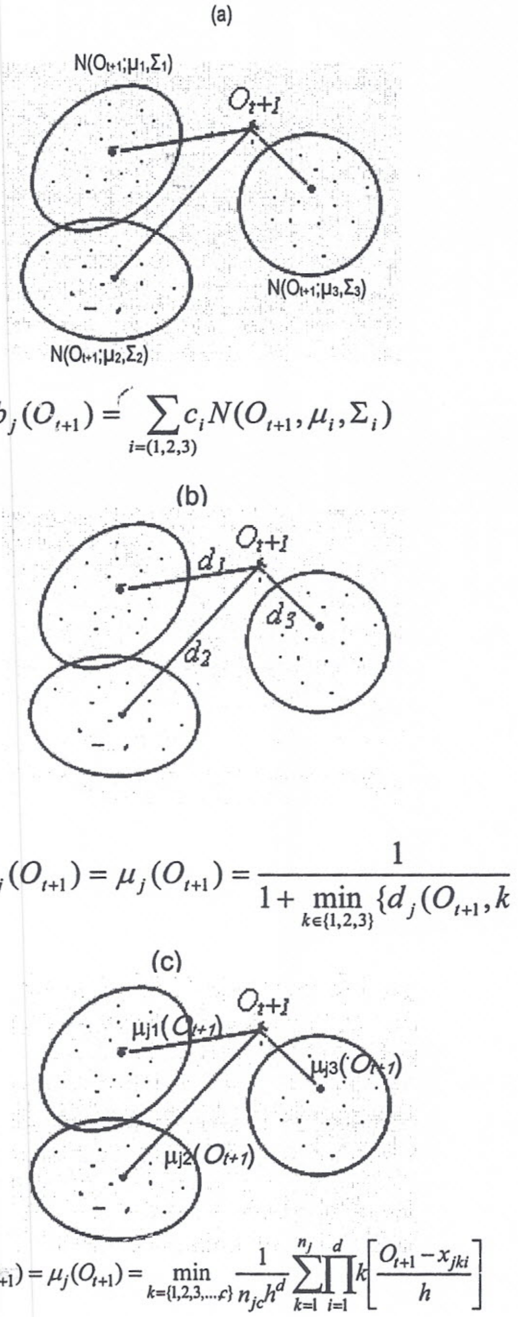
Actually, there is no guarantee with the assumption of Gaussian distribution and in the implementation aspect, sometime we have a singularity problem in the computation the inverse of covariance matrix, especially for the mixture Gaussian distribution and limited training data. Our approaches to handle this problem are as follow: firstly, we accommodate the phenomena of multimodal distribution by clustering each state into several clusters. Secondly, the observation probability replaced by membership value defined by Euclidean distance or kernel function. The membership value computed for each cluster and we choose the biggest one as observation probability. So our approaches relax from Gaussian distribution assumption and free from singularity problem. Figure 2 presents the comparison of three methods, i.e. Gaussian, Euclidean distance, and kernel function.

3. Speaker Identification Using HMM

Definition. A stochastic process [5] is a family of random variable, $\{X_t\}_{t=1}^{\infty}$, where t is a parameter running over a suitable index set T. If the process has the following property: given the value of X_t, the value of X_s for s>t are independent from X(u) for u<t, we call it a Markov Chain. In the condition that state space (range of possible values for the random variable X_t) is a finite or countable set and index set T={0, 1, 2, ..} the process called discrete time Markov Chain. If the value of random variable X_t only depends on X_{t-1} and it is independent with the index t, we call it first order stationer discrete time Markov Chain. If we could not observe the state sequence of the chain (hidden state), and we only can observe its observation sequence produced by the appropriate hidden state sequence, we call it · Hidden Markov Model (HMM).

An HMM has specified completely with three components, i.e. initial state distribution, Л; transition probability matrix, A; and observation probability matrix, B. HMM is notated by λ = (A, B, Л), where:

- A: NxN transition matrix with entries $a_{ij}=P(X_{t+1}=j|X_t=i)$, N is the number of possible hidden states
- B: NxM observation matrix with entries $b_{jk}=P(O_{t+1}=v_k|X_t=j)$, k=1, 2, 3, ..., M; M is the number of possible observable states
- Л: Nx1 initial state vector with entries $\pi_i=P(X_1=i)$



$$b_j(O_{t+1}) = \sum_{i=(1,2,3)} c_i N(O_{t+1}, \mu_i, \Sigma_i)$$

$$b_j(O_{t+1}) = \mu_j(O_{t+1}) = \frac{1}{1 + \min_{k\in\{1,2,3\}} \{d_j(O_{t+1}, k}$$

$$b_j(O_{t+1}) = \mu_j(O_{t+1}) = \min_{k=\{1,2,3,\ldots,c\}} \frac{1}{n_{jc}h^d} \sum_{k=1}^{n_j} \prod_{i=1}^{d} k\left[\frac{O_{t+1} - x_{jki}}{h}\right]$$

Figure 2. Three methods in computation of observation probability with three clusters for state j, (a) Gaussian HMM, (b) Euclidean base, (c) Kernel function base

For Gaussian HMM, B consists of a mean vector and a covariance matrix for each hidden state, µi and Σi, respectively, i=1, 2, 3, …, N. The value of bj(Ot+1) is N(Ot+1,µj,Σj), where :

$$N(\mu_j,\Sigma_j)=\frac{1}{(2\pi)^{d/2}|\Sigma_j|^{1/2}}\exp\left[-\frac{1}{2}(O_{t+1}-\mu_j)\Sigma_j^{-1}(O_{t+1}-\mu_j)'\right] \quad (1)$$

There are three problems with HMM [1], i.e. evaluation problem, P(O|λ); decoding problem, P(Q|O, λ); and training problem.

Gaussian HMM in Speaker Identification

Figure 3 describes the process of speaker identification using Gaussian HMM.

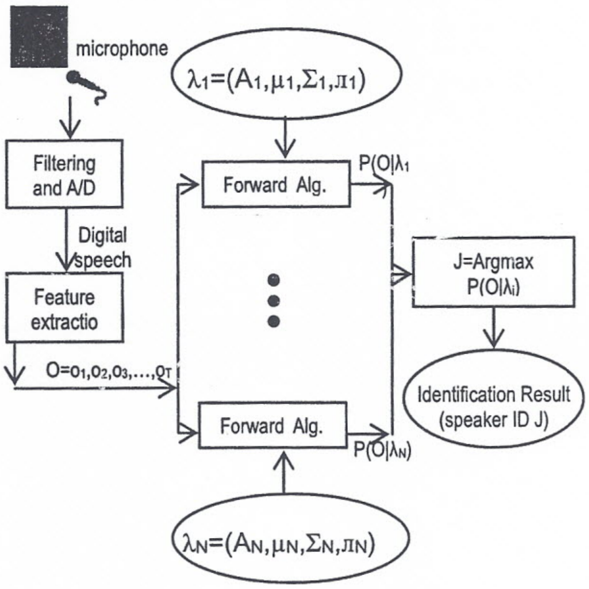


Figure 3. Block diagram of speaker identification using Gaussian HMM

In this research we use Mel-Frequency Cepstrum Coefficients (MFCC) by Do in 1994 [3] as feature extraction. The HMM of each speaker is trained using k-segmental algorithm developed by Dugad in 1996 [6]. Forward algorithm [6] used for evaluating a new observation sequence to compute P(O|λi), i=1, 2, 3, …, n (n is the number of speakers). In this scheme, the value of observation probability bj(Ot+1)=P(Ot+1|Xt=j) is computed by (1).

4. Proposed Methods

At least two problems occur in Gaussian HMM. Firstly, no one can guarantee with Gaussian assumption. Secondly, we often have problem in computing the inverse of covariance matrix because of the singularity, especially with larger dimension and limited training data. To handle this condition, in our proposed methods, the value of observation probability bj(Ot+1)=P(Ot+1|Xt=j) is approached by a membership value µj(Ot+1). There are two ways for computing observation probability, i.e. Euclidean base and kernel function base. In order to anticipate that observation distribution is multimodal, we cluster them using fuzzy clustering prior to computation observation probability step.

Euclidean Distance Base: In this approach, components of HMM for each speaker consists of transition probability A, initial state distribution Л, and matrix P_j, i.e. a center matrix c by d for each appropriate state j, j=1,2,3,…,N; N is the number of states in the model, d is the dimension of data and c is the number of clusters. We denote the matrix P_j as:

$$P_j=\begin{bmatrix}p_1\\p_2\\\cdots\\p_c\end{bmatrix}=\begin{bmatrix}x_{11}&x_{12}&\cdots&x_{1d}\\x_{21}&x_{22}&\cdots&x_{2d}\\\cdots&\cdots&\cdots&\cdots\\x_{c1}&x_{c2}&\cdots&x_{cd}\end{bmatrix}\quad \text{j=1,2,3,…,N} \quad (2)$$

For the new observation O_{t+1}, we compute:

$$D_j(O_{t+1},k)=\sum_{i=1}^{d}(x_{ki}-o_{t+1,i})^2 \quad (3)$$

for k=1, 2, 3, …, c; the membership value $\mu_j(O_{t+1})$ computed by the formula (j=1, 2, 3, …, N) :

$$b_j(O_{t+1})=\mu_j(O_{t+1})=\frac{1}{1+\min\limits_{k\in\{1,2,..c\}}\{d_j(O_{t+1},k)\}} \quad (4)$$

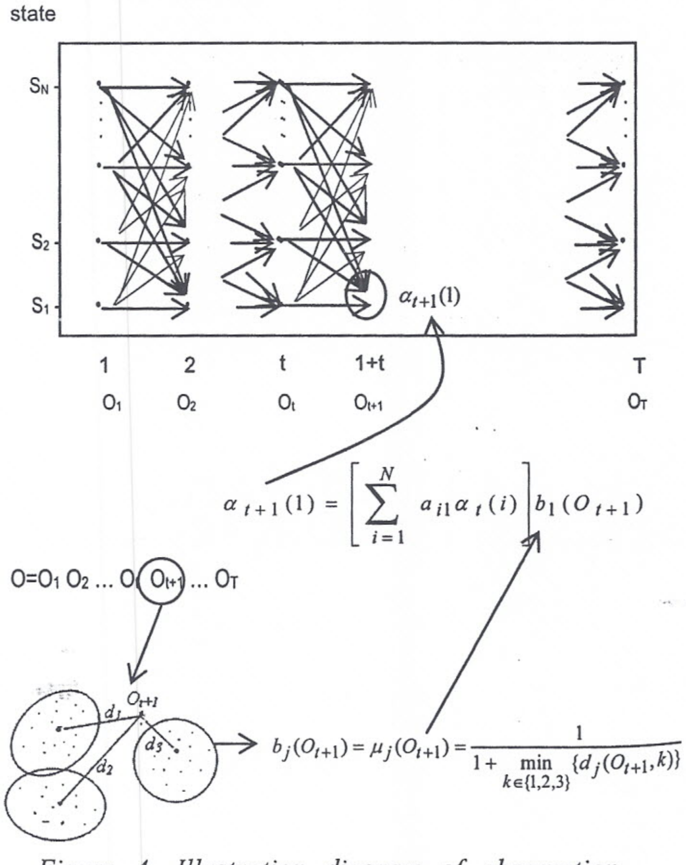

state
SN
S2
S1
$\alpha_{t+1}(1)$
1
2
t
1+t
T
O1
O2
Ot
Ot+1
OT
$\alpha_{t+1}(1)=\left[\sum_{i=1}^{N}a_{i1}\alpha_t(i)\right]b_1(O_{t+1})$
O=O1 O2 … Ot Ot+1 … OT
Ot+1
d1
d2
d3
$b_j(O_{t+1})=\mu_j(O_{t+1})=\frac{1}{1+\min\limits_{k\in\{1,2,3\}}\{d_j(O_{t+1},k)\}}$

Figure 4. Illustration diagram of observation probability using Euclidean distance

By using this approach, step 3 and 4 of the k-segmental algorithm for HMM training developed by Dugad [6] becomes:

Step 3: compute center matrix c by d, P_j, j=1,2,3,…,N

a. classify each observation with label state j into c clusters using fuzzy C-mean clustering, j=1, 2, 3, …, N

b. assign the center of cluster k (k=1, 2, 3, …, c) for state j as the j^{th} row of center matrix P_j (j=1, 2, 3, …, N)

step 4: compute observation probability, bj(Ot+1)= µj(Ot+1) using (4)

Figure 4 describes an illustration for this approach.

Kernel Function Base : In this scheme we assign a value of empirical distribution that is predicted by kernel methods to $b_j(O_{t+1})$. Hence, we do not need to assume any theoretical distributions. We do two types of computation, the first computation requires no (or without) state clustering, and the second one uses state clustering prior to predict $b_j(O_{t+1})$. Kernel function formula to predict the empirical distribution of O_{t+1} given state j is:

a. Without state clustering :

$$b_j(O_{t+1}) = \frac{1}{n_j h_1 h_2 ... h_d} \sum_{k=1}^{n_j} \prod_{i=1}^{d} k\left[\frac{O_{t+1}, i - x_{jki}}{h_i}\right] \quad (5)$$

n_j : the number of frames with state label j

k(z) : kernel Gaussian, i.e.

$$k(z) = \frac{1}{\sqrt{2\pi}} \exp\left(-0.5z^2\right)$$

b. With state clustering :

$$b_j(O_{t+1}) = \max_{k=\{1,2,3,...,c\}} \frac{1}{n_{jc} h_1 h_2 .. h_d} \sum_{k=1}^{n_j} \prod_{i=1}^{d} k\left[\frac{O_{t+1} - x_{jki}}{h_i}\right] \quad (6)$$

c : number of clusters

In those formulas, h_i is smoothing parameter for i^{th} dimension. In order to reduce the bias and mean integrated square error of the prediction, h_i formulated as follow [7] :

$$h_i = 1{,}04\ s\ n^{-1/5} \quad (7)$$

where s is standard deviation of its variable in the i^{th} dimension, n is number of data.

In the first computation, components of HMM for each speaker consist of transition probability A, initial state distribution Л, and feature vector of each state. In this method, k-segmental algorithm step 3 and 4 become:

step 3 : compute 1 by d smoothing parameter vector, Hj=[h1 h2 … hd], j=1, 2, 3, …, N, using formula (7).

step 4 : compute observation probability, bj(Ot+1)= µj(Ot+1) using (5)

Whereas, in the second one, components of HMM for each speaker consist of transition probability A, initial state distribution Л, and feature vector of each cluster state. K-segmental algorithm step 3 and 4 become:

step 3 : compute c by d smoothing parameter matrix, Hj, j=1,2,3,…,N, using formula (7) (c is number of clusters for state j).

$$H_j = \begin{pmatrix} h_{11} & h_{12} & \cdots & h_{1d} \\ h_{21} & h_{22} & \cdots & h_{2d} \\ \cdots & \cdots & \cdots & \cdots \\ h_{c1} & h_{c2} & \cdots & h_{cd} \end{pmatrix}$$

step 4 : compute observation probability, bj(Ot+1)= µj(Ot+1) using (6)

5. Experiments and Results

Data: The speech data consists of several short unguided utterances of an isolated word 1.28 second length sampled at 11 kHz. Ten speakers utter the word "*pudesha*" in a real condition for 40 times, which yield 400 utterances (in each speaker, 30 utterances for model training and the rest for model testing). We use hamming window for windowing each unguided utterances with 30 millisecond window length without overlapping. For each window, we compute 13 MFCC coefficients formulated by Dugad [6].

We conducted four experiments, i.e. Gaussian HMM, HMM Euclidean distance base with state clustering, HMM Kernel function base without state clustering, and HMM Kernel function base with state clustering.

HMM Structure: In this research we use left right HMM with 7 hidden states as illustrated in Figure 5.

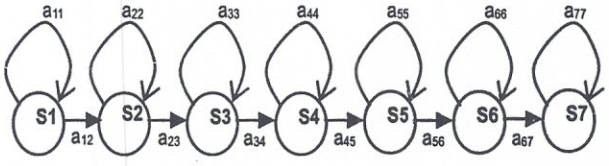


Figure 5. Left right HMM structure for word "pudesha"

Transition probability matrix, A, and initial state distribution, л, for the HMM are:

$$A = \{a_{ij}\} = \begin{pmatrix} a_{11} & a_{12} & 0 & 0 & 0 & 0 & 0 \\ 0 & a_{22} & a_{23} & 0 & 0 & 0 & 0 \\ 0 & 0 & a_{33} & a_{34} & 0 & 0 & 0 \\ 0 & 0 & 0 & a_{44} & a_{45} & 0 & 0 \\ 0 & 0 & 0 & 0 & a_{55} & a_{56} & 0 \\ 0 & 0 & 0 & 0 & 0 & a_{66} & a_{67} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \text{ and } \pi = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

With this structure, we train the model to set parameter values using training data, i.e. 30 utterances for each model. we use k-segmental algorithm developed by Rakesh [6] modified according to our approach, except for standard HMM, i.e. Gaussian HMM.

Results: Experimental results presented in the following Table 1.

Table 1. Comparison of recognition rates (%) for 10 speakers

Methods	Training Data		Testing Data	
	d=13	d=26	d=13	d=26
HMM's Gaussian	50	Fail	42	Fail
HMM Euclidean without state Clustering	70	70	54	54
HMM Euclidean with state Clustering	94.7	96.7	88	85
HMM Kernel Function without state Clustering	100	N/A	82	NA
HMM Kernel Function with state Clustering	100	NA	87	NA

We can see that there is no significant different in recognition rate between 13 MFCC coefficients and 26 MFCC coefficients. In other words, we can say that 13 MFCC coefficients are enough to represent the data. By these findings, we do not conduct experiment with kernel methods for 26 MFCC coefficients, so we put N/A (Not Available) in the aforementioned table.

By 13 MFCC coefficients, recognition rates for standard HMM only 50 % and 42 % for training data and testing data, respectively. By using 26 MFCC coefficients, standard HMM fails to train the model. Whereas, by using Euclidean distance without state clustering, recognition rate has increased more than 10 % compared to standard HMM. In order to improve recognition rate, we cluster each state into three clusters, and observation probability on index t+1 given state j is calculated according to the minimum distance from that observation to its clusters. By this approach, recognition rate increases from 42 % (standard HMM) to 88 % for testing data and from 50 % to 94.7 % for training data. The other advantages of this method are free from lack of Gaussian distribution and running well for limited training data and number of dimensions.

From the third and fourth row of the table, we can say that kernel function improves recognition rate as well as Euclidean distance with state clustering. But, there is one disadvantage with kernel method, i.e. we have to calculate the kernel function over all data training, so the training and testing process are very time consuming. For training data, recognition rate of HHM with kernel function is 100 %.

From Table 1 we can also infer that HMM with state clustering give a good result compared to without state clustering.

Table 2 presents recognition rate for testing data and each speaker with three comparable methods.

Table 2. Recognition rate for testing data and each speaker

Speaker ID	HMM Euclidean Base with state clustering	HMM kernel function base without state clustering	HMM kernel function base with state clustering	average
1 (male)	90	100	100	96
2 (male)	100	100	100	
3 (male)	90	100	100	
4 (male)	90	80	80	
5 (male)	100	100	100	
6 (male)	100	100	100	
7 (female)	100	60	80	70
8 (female)	90	40	50	
9 (female)	30	40	60	
10 female)	90	100	100	
	88	82	87	

From Table 2, we can say that female speaker more difficult to recognize compared to male speaker. In the average, recognition rate both for male and female speakers are 96.11 % and 70 %, respectively.

Table 3 presents classification result for the three methods in detail. From the table, we can say that error occurs for speaker number 8 and 9, because speaker number 8 and number 9 are sisters (sibling). The elder sister is 12 years old and the younger is 9 years. Their voices are very similar; hence it is difficult to differentiate between them using our bare ears. We can say that these methods are not running well for situation where the speakers are sibling and their ages are close, i.e. 3 years old in this case.

Table 3a. Classification result for testing data using HMM Euclidean distance base

Speaker ID	Recognized as speaker number									
	1	2	3	4	5	6	7	8	9	10
1	9	0	0	0	1	0	0	0	0	0
2	0	10	0	0	0	0	0	0	0	0
3	0	0	9	0	1	0	0	0	0	0
4	0	0	0	8	2	0	0	0	0	0
5	0	0	0	0	10	0	0	0	0	0
6	0	0	0	0	0	10	0	0	0	0
7	0	0	0	0	0	0	10	0	0	0
8	0	0	0	0	0	0	0	9	1	0
9	0	0	0	0	0	0	0	6	3	1
10	0	0	0	0	1	0	0	0	0	9

Table 3b. Classification result for testing data using HMM kernel function base without state clustering

Speaker ID	Recognized as speaker number									
	1	2	3	4	5	6	7	8	9	10
1	10	0	0	0	0	0	0	0	0	0
2	0	10	0	0	0	0	0	0	0	0
3	0	0	10	0	0	0	0	0	0	0
4	0	0	0	8	0	1	0	0	0	1
5	0	0	0	0	10	0	0	0	0	0
6	0	0	0	0	0	10	0	0	0	0
7	4	0	0	0	0	0	6	0	0	0
8	0	0	0	0	0	0	0	4	3	3
9	3	0	0	0	0	0	0	1	4	2
10	0	0	0	0	0	0	0	0	0	10

Table 3c. Classification result for testing data using HMM kernel function base with state clustering

Speaker ID	Recognized as speaker number									
	1	2	3	4	5	6	7	8	9	10
1	10	0	0	0	0	0	0	0	0	0
2	0	10	0	0	0	0	0	0	0	0
3	0	0	10	0	0	0	0	0	0	0
4	0	0	0	8	1	0	0	0	1	0
5	0	0	0	0	10	0	0	0	0	0
6	0	0	0	0	0	10	0	0	0	0
7	2	0	0	0	0	0	8	0	0	0
8	0	0	0	0	0	0	0	5	4	1
9	3	0	0	0	0	0	0	0	6	1
10	0	0	0	0	0	0	0	0	0	10

6. Conclusion

We have presented in this paper, the development of a speech recognition system based on modified HMM using euclidean distance and kernel function separately and a cascade with MFCC as feature extraction.

Experimental results show that the system could recognize and discriminate the speakers with recognition rate around 85 %. Whereas, the standard system only 42 %. This research also shows that recognition rate for the methods with state clustering better than without state clustering. If speakers are sibling, then recognition rate will drop significantly, especially for ones with similar ages. It is also confirmed from these experiments that 13 MFCC coefficients are good enough to represent the data.

References

[1] L.R. Rabiner. (1989). “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition”, *Proceeding IEEE*, Vol 77 No. 2, pp 257-289.

[2] Farbod H. and M. Teshnehlab. (2005). “Phoneme Classification and Phonetic Transcription Using a New Fuzzy Hidden Markov Model”. *WSEAS Transactions on Computers*. Issue 6, Vol. 4.

[3] Do, MN. (1994). “Digital Signal Processing Mini-Project: An Automatic Speaker Recognition System”. *Audio Visual Communications Laboratory, Swiss Federal Institute of Technology, Lausanne, Switzerland.* http://lcavwww.epfl.ch/ minhdo/asr_project/asr_project.pdf. [December 12 2006]

[4] J. Campbell. (1997). "Speaker Recognition: A Tutorial". *Proc. of the IEEE,* Vol 85, No. 9, pp 1437-1462.

[5] Taylor, H.M. and Samuel Karlin. (1984). *An Introduction to Stochastic Modeling*. Academic Press, Inc. Florida.

[6] Dugad, Rakesh. (1996). “A Tutorial on Hidden Markov Model. Technical Report, Departement of Electrical Engineering, Indian Institute of Technology, Bombay”.

[7] B.W. Silverman. (1990). Density Estimation for Statistics and Data Analysis. Chapman and Hall.