Towards a Speaker Voice Recognition Method Based on KPFVQ

Jie Yang
Department of Mechanical Engineering, The Shanghai Second Polytechnic University, Shanghai, China
Email: jeanyang1218@hotmail.com

***Abstract*—To address the problem that fuzzy kernel speaker voice recognition method sensitive to outlier and noise as well as slow training, a Kernel-function based Possibilistic Fuzzy Vector Quantization (KPFVQ) was proposed. The method combines typical possibilistic clustering of fuzzy C-mean, thus suppressing the sensitivity. It also uses kernel mapping for vector quantization and match decision on voice features in the high-dimensional feature space. The characteristic differences among different samples were emphasized so that it is easier to distinguish voice and voice, voice and noise. The experiment results show that the proposed algorithm in the paper can achieve better recognition effect both to relatively clean voice and that with noise. Its training speed improves greatly as the voice length increases, which can achieve the real-time effect.**

***Index Terms*—kernel method, fuzzy C-mean, vector quantization, speaker voice recognition**

I. Introduction

As a biometric authentication technology, speaker recognition has broad application prospect in many fields. The Vector Quantization (VQ) technology plays a very important role in speaker recognition as an efficient data compression and encoding method. [1]. However, the traditional VQ based on hard clustering strictly divides sample feature vectors into regulated class, while ignoring varying degrees of overlap between feature vectors. In recent years, the Fuzzy C-Means (FCM) method based on software division was applied to speaker recognition [2], [3]. It introduces uncertainty thought through membership function to implement effective extension of hard clustering algorithm. Using FCM method, the quantization error of code can be reduced, the recognition performance of which is significantly improved compared with conventional methods. However, the above two methods have not optimize sample features but directly perform clustering with features. Effectiveness of these methods largely depends on sample distribution, which plays no effect on non-hyper sphere data as well as variety noise pollution data. At the same time, the voice feature distribution is too complicated to determine its specific structure in advance. Therefore, the above clustering algorithms cannot accurately describe complex speaker voice features. Subsequently, Lin *et al.* [4] introduced kernel method and fuzzy C-mean clustering into speaker recognition. It mainly addressed to term recognition. The Fuzzy Kernel Vector Quantization method (FKVQ) used Mercer kernel function to map data in mode space into high-dimensional space to expand difference among features [5]. In this way, complicated speaker voice feature can be accurately classified and the recognition rate can be improved. After all, FKVQ is based on FCM, so it only gets local optimal classification results. In addition, it needs normalized constraints on fuzzy membership function. The computation complexity increases, which limits its applications. The typical idea in Possibilistic C-Means (PCM) relaxed constraints on sample membership, so it has advantages in the processing of outlier and noise data [6].

The paper uses improved hybrid C-means clustering, namely Possibilistic Fuzzy C-Means (PFCM) that combined FCM and PCM for vector quantization design. It both takes closely linked membership of sample data on feature center into account and considers typical of features. The combination of these two not only decreased sensitivity to initial value, but also suppressed effect on outlier and noise [7]. In addition, kernel learning method is likely to achieve better VQ result on complicated voice structure. The computation of parameters in the high-dimensional space may also be simplified. The paper aims at replacing traditional K-means with kernel-based possibilitic fuzzy C-means clustering. It is organized as follows: Section 2 gives VQ method based on possibilitic fuzzy C-mean; Section 3 performs simulation experiment on proposed method and analyzes results; Section 4 concludes our work.

II. Vector Quantization Based on Possibilitic Fuzzy C-Mean

A. Kernel Function Method

In the PFCM, Euclidean distance is used to compute $D_{ij} = \|x_j - v_i\|$ between gravity vector v_i and voice sample vector x_j. However, the Euclidean distance is not sufficient to describe distance measure of vector quantization in the complicated environment. The paper uses kernel function to compute D_{ij}.

Set $X = \{x_1, x_2, \cdots, x_N\}$ as a limited data set in the input space R_d; x_j is a d-dimensional vector. With non-

Manuscript received May 10, 2015; revised December 7, 2015.

doi: 10.18178/ijsps.4.5.422-426

linear mapping $\Phi(\cdot)$, the input mode space is mapped into a high-dimensional feature space *H*.

$$\begin{aligned} X = \{x_1, x_2, \cdots, x_N\} \rightarrow \\ \Phi(X) = \{\Phi(x_1), \Phi(x_2), \cdots, \Phi(x_N)\} \end{aligned} \tag{1}$$

Then:

$$D_{ij} = \left\| x_j - v_i \right\| \rightarrow \widehat{D}_{ij} = \left\| \Phi(x_j) - \Phi(v_i) \right\|, \forall i, j \tag{2}$$

With the kernel function meet Mercer conditions [5], there is:

$$K(x_k, x_j) = \Phi(x_k) \cdot \Phi(x_j), \forall k, j \tag{3}$$

The Gaussian radial basis function is used:

$$K(x_k, x_j) = \exp\left(-\frac{\left\| x_i - x_j \right\|}{2\delta^2} \right) \tag{4}$$

So the right side of (2) can be rewritten as:

$$\begin{aligned} (\widehat{D}_{ij})^2 = \left\| \Phi(x_j) - \Phi(v_i) \right\|^2 = \\ K(x_i, x_j) - 2K(x_j, v_i) + K(v_i, v_j) = 2 - 2K(x_j, v_i) \end{aligned} \tag{5}$$

B. Possibilitic Fuzzy C-Mean Vector Quantization

The vector quantization just replaces traditional K-mean with possibility fuzzy C-mean clustering based on kernel function. Expand PFCM algorithm mentioned above to high-dimensional space, thus the objective function of Kernel-function based Possibilistic Fuzzy C-Means (KPFCM) can be obtained as:

$$\begin{aligned} J_m^{\Phi}(X, U, T, V) = \underbrace{\sum_{i=1}^{C} \sum_{j=1}^{N} (au_{ij}^m) D_{ij}^2 +} \\ \underbrace{\sum_{i=1}^{C} \sum_{j=1}^{N} (bt_{ij}^p) D_{ij}^2 + \eta \sum_{i=1}^{C} \sum_{j=1}^{N} (1 - t_{ij})^p} \end{aligned} \tag{6}$$

where, *m* and *p* are fuzzy weight; *a*>0, *b*>0; *C* is number of classes and *N* is total number of samples; η is a constant. In the initialization, there is:

$$\eta = \frac{1}{m^2 NC} \sum_{i=1}^{N} \left\| \Phi(x) - \Phi(\overline{x}) \right\| \tag{7}$$

where:

$$\overline{x} = \frac{1}{N} \sum_{i=1}^{N} x_i$$

Let $\boldsymbol{T} = \left[t_{ij} \right]_{C \times N}$ as typical value matrix, $\boldsymbol{U} = \left[u_{ij} \right]_{C \times N}$ as fuzzy membership matrix, $\boldsymbol{V} = \left[v_{ij} \right]_{C \times d}$ as central vector matrix, where:

$$\begin{gathered} 0 \le u_{ij}, t_{ij} \le 1, i = 1, \cdots, C, j = 1, \cdots, N \\ \sum_{i=1}^{C} u_{ij} = 1 \end{gathered} \tag{8}$$

It can be seen from constraints in (8) that the membership of the *j*-th cluster in x_j is related to that of other *C*-1 cluster centers. The u_{ij} shows compatibility of x_j on the *j*-th cluster; t_{ij} is possibility that x_j belongs to the *j*-th cluster, which is independent with that x_j belongs to other *C*-1 clusters but related to expression of the *j*-th cluster. Thus, the contribution of each sample to every cluster can be highlighted. The dissimilarity between noise and outlier on each cluster is marked as small typical value, the effect of which on cluster can be greatly decreased. The first part of objective function in (6) is to seek for optimal cluster that impacted by initial state less but likely to be affected by noise or outlier. The second part of (6) seeks for inherent clusters impacted by initial constraints. The combination of two parts will optimize the clustering.

Compute the minimum value of (6) about u_{ij}, t_{ij} and v_{ij} in turn, there are:

$$u_{ij} = \left[\sum_{k=1}^{c} \left(\frac{1 - K(x_j, v_i)}{1 - K(x_j, v_k)} \right)^{\frac{1}{m-1}} \right]^{-1}, \forall i.j \tag{9}$$

$$t_{ij} = \left[1 + \left(\frac{2b}{\eta} \left(1 - K(x_j, v_i) \right) \right)^{\frac{1}{p-1}} \right]^{-1}, \forall i.j \tag{10}$$

and

$$\Phi(v_i) = \frac{\sum_{k=1}^{N} (au_{ik}^m + bt_{ik}^p) \Phi(x_k)}{\sum_{k=1}^{N} (au_{ik}^m + bt_{ik}^p)}, \forall i \tag{11}$$

As the (11) cannot compute directly, multiple $\Phi(v_i)^T$ to both sides so that it can be rewritten as:

$$K(x_{j,} v_i) = \frac{\sum_{k=1}^{N} (au_{ik}^m + bt_{ik}^p) K(x_k, x_j)}{\sum_{k=1}^{N} (au_{ik}^m + bt_{ik}^p)}, \forall i, j \tag{12}$$

In order to obtain codebook with speaker voice feature, estimate central vector of each cluster with (13) after clustering.

$$v_i = \frac{\sum_{k=1}^{N} (au_{ik}^m + bt_{ik}^p) x_k}{\sum_{k=1}^{N} (au_{ik}^m + bt_{ik}^p)}, \forall i \tag{13}$$

The specific steps for VQ codebook estimation are as following:

Step 1: Initialization. Fix value of *a*, *b*, *c*, *m* and *p*, where $a > 0$, $b > 0$, $n > c > 1$, $+\infty > m$, $p > 1$. Set initial loop value *r*=1 and maximum loop number as $r_{\max}$. Compute value of η with (7) and set ***V***(0) as initial cluster center.

Step 2: The loop steps include:

(a) Update membership matrix $\boldsymbol{U}(r)$ with (9).

(b) Update typical value matrix $\boldsymbol{T}(r)$ with (10).

(c) Update $K(x_j, v_i)$ with (12).

(d) Add *r*.

The loop ends until condition $\left\| \boldsymbol{T}(r) - \boldsymbol{T}(r-1) \right\| < \varepsilon$ or $r > r_{\max}$ is met.

Step 3: Compute cluster center using (13) and set it as the codebook after estimation.

C. Speaker Recognition Method

After obtained training codebook with above algorithm, conduct vector quantization on test feature vector sequence in the high-dimensional space to compute average quantization error as:

$$D(r)=[\sum_{i=1}^{C}\sum_{j=1}^{N}(au_{ij}^{m}+bt_{ij}^{p})\widehat{D}^{2}(\Phi(x_j),\Phi(v_i(r))) \\ +\eta\sum_{i=1}^{C}\sum_{j=1}^{N}(1-u_{ij})^{p}]\times\frac{1}{N_r} \qquad (14)$$

where, $\widehat{D}^{2}(\Phi(x_j),\Phi(v_i(r))$ is the distance of the i-th codebook between feature vector x_j and the r-th speaker.

After computation completed, the speaker corresponding to codebook with minimum average quantization error will be regarded as the recognition result.

$$result=\arg\min_{1\le r\le R}(D(r)) \qquad (15)$$

III. Simulation Experiment and Result Analysis

The experiment data uses ELSDSR voice database [8]. There are 23 persons for text-independent speaker recognition. Among them, 10 are girls and 13 are boys. Each speaker records 8 times. A voice segment of once record was selected as training voice and remaining for recognition. Firstly, the voice data was conducted endpoint detection. The obtained signal is then for pre-emphasis and Hamming window processing. The pre-emphasis factor is 0.95; window bandwidth is 256 sampling points; window shift 80 sample points. The 12-dimensional Mel Cepstral and its first-order dynamic cepstral are extracted, totally 24 dimensions. The first one is removed and the remaining 23 dimensions used as feature parameters of speaker.

A. Effect of Outlier and Noise on System Performance

There are outlier and noise in the voice samples inevitably. In order to verify processing of KPFCM on outlier and noise in the vector quantizing, the experiment was conducted. Firstly, the X_{10} in 2-dimensional X_{12} was clustered [9], [10], which is a two-dimensional dataset with 12 data points. Then, X_{12} was clustered after added noise $X_{11}(0, 0)$ and $X_{12}(0, 10)$. Compare change of cluster center before and after clustering, the result is shown in Table I. It can be seen that the algorithms all arrive at good cluster centers without affect from outlier and noise. Added noise data, the cluster centers shifts. The shift of KPFCM is the less than that of Fuzzy Kernel C-Means (FKCM) and PFCM.

The membership and typical value of noise is shown in Table II. Because of constraints of membership normalization in FKCM, two noise points are assigned same large value. These two algorithms assign corresponding typical value in accordance with contribution of sample on cluster result. Two isolated noise points have little effect on clustering result, which were given small typical values. The sample point $X_{12}(0,10)$ farther away from vector center is given less typical value. The KPFCM algorithm just assigns small typical value to noise points. It is inevitably there is noise and isolated points in the voice data, so it is impossible to make difference only depends on membership, which cannot achieve good recognition result. The vector quantization algorithm based on KPFCM, namely KPFVQ has better performance in respect to effect from noise.

TABLE I. Cluster Center of X_{12}

Algorithm	Central Vector	
	X_{10}	X_{12}
FKCM	(-3.34, 0)(3.34, 0)	(-2.98, 0.54)(2.98, 0.54)
PFCM	(-3.34, 0)(3.34, 0)	(-3.00, 0.49)(3.00, 0.49)
KPFCM	(-3.34, 0)(3.34, 0)	(-3.13, 0.4)(3.23, 0.4)

TABLE II. Membership and Typical Value of X_{11} and X_{12}

Noise	FKCM	PFCM		KPFCM	
	U	***U***	***T***	***U***	***T***
X_{11}	(0.5, 0.5)	(0.5, 0.5)	(0.012, 0.012)	(0.5, 0.5)	(0.010, 0.010)
X_{12}	(0.5, 0.5)	(0.5, 0.5)	(0.004, 0.004)	(0.5, 0.5)	(0.002, 0.002)

B. Training Time Comparison

A major advantage of KPFVQ is that its training time is significantly shorter than that of vector quantization based on FKCM, abbreviated as KFVQ. The 2-15s voice segments were selected for comparison. The computer CPU is Pentium 4 2.4GHz; memory 2GB. The MATLAB R2007b was used for simulation. Experiment result is shown in Fig. 1. It is seen that the training time of KFVQ is significantly longer than that of KPFVQ. It is inseparable of algorithm features. When KFVQ falls into local optimum, its search efficiency will extremely decline and training time significantly increase, while the training time of KPFVQ just only slightly increases.

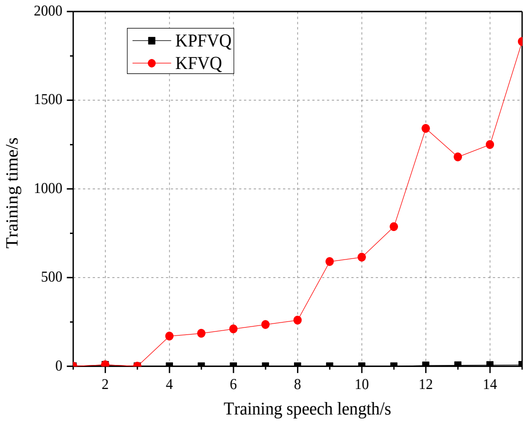

Figure 1. Training time of KFVQ and KPFVQ with different voice length.

C. Comparison of Different Recognition Methods

In order to examine the impact of kernel function on system performance, compare the proposed method KPFVQ and Possibilistic Fuzzy Vector Quantization

(PFVQ) without kernel function. The codebook capacity takes 8, 16, 32 and 64. The error recognition rate curve of two methods under different codebook is shown in Fig. 2. It can be seen that the KPFVQ can get lower error rate under different codebook compared with PFVQ. It indicates that the introduction of kernel method enlarge distinguishing of speaker characteristics, thus improving system performance.

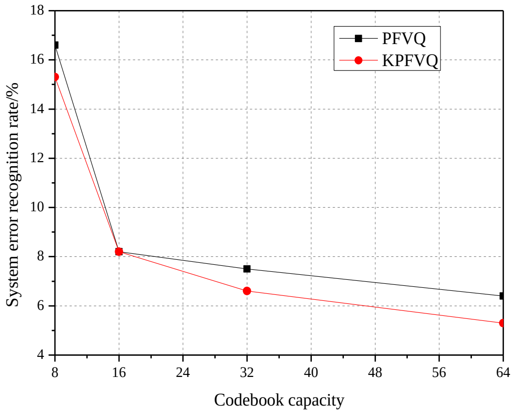

Figure 2. Error recognition rate comparison between KPFVQ and PFVQ.

Then the error recognition rate between KFVQ and KPFVQ were compared. Training data with 23 speakers was used. The codebook capacity is 16 and training speech length is 3-12 seconds. The split method was used in the beginning. The parameters were set as follows: m=0.15, p=1.15, a=6, b=1, e_2=5.5. The experiment result is shown in Fig. 3. It can be seen that the proposed method KPFVQ can obtain lower rate than KFVQ both in short voice and long voice. It is because KPFVQ combines advantages of fuzzy clustering and possibilistic clustering. The connection between central codebook and samples was considered. Meanwhile, the typical idea is introduced to avoid impact of bad samples on training.

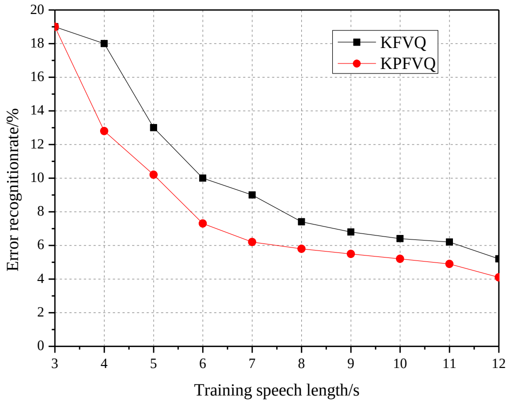

Figure 3. Error recognition rate comparison between KPFVQ and KFVQ.

D. *Comparison of Different Methods under Noise Environment*

The error recognition of speaker recognition system under noise environment was compared using PFVQ, KFVQ and KPFVQ. The training conditions are same as above experiments beside voice segment to be recognized added 45dB Gaussian noise. The parameters were set as follows: m=1.15, p=1.05, a=1, b=6, e_2=7.5. The recognition result is shown in Fig. 4. Compared with the former experiment, the impact of noise on algorithm can be decreased by adjusting typical of parameter enhancement algorithm. However, PFVQ cannot achieve this target, the error recognition rate of which is extremely high when the noise increases.

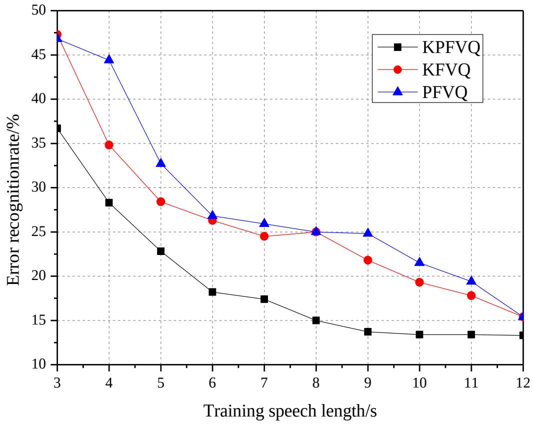

Figure 4. Error recognition rate comparison among PFVQ, KFVQ and KPFVQ under noise environment.

IV. Conclusion

The paper proposed a hybrid C-mean vector quantization speaker voice recognition method based on kernel and possibilistic fuzzy clustering. With simulation experiments, the capability to deal with noise and outlier was studied. It is found that KPFVQ has lower sensitivity to noise and outlier. The system error recognition rate can be decreased and training time greatly reduced. It can be seen that KPFVQ has better performance in the application of speaker recognition, which is more suitable for actual system than fuzzy kernel vector quantization method. Although KPFVQ mapped feature vector into high-dimensional space to enhance distinguish, the selection of kernel function depends on experiments or experiences. Therefore, how to determine kernel function in accordance with data distribution of speakers is our research focus in the future.

Acknowledgment

The research was supported by the constructing program of the key breeding discipline in Shanghai Second Polytechnic University Computer Application Technology (No.XXKPY1301).

References

[1] F. Soong, A. E. Rosenberg, L. R. Rabiner, and B. H. Juang, "A vector quantization approach to speaker recognition," in *Proc. IEEE Conf. on Acoustics, Speech and Signal Processing*, 1985, pp. 387-390.

[2] N. B. Karayiannis and P. I. Pai, "Fuzzy vector quantization algorithms," in *Proc. IEEE World Congress on Computational Intelligence*, 1994, pp. 1996-2001.

[3] D. Tran, M. Wagner, and V. L. Tu, "A proposed decision rule for speaker recognition based on fuzzy C-Means clustering," in *Proc. 5th International Conf. on Spoken Language Processing*, 1987, pp. 755-758.

[4] L. Lin and S. X. Wang, "A kernel method for speaker recognition with little data," in *Proc. 8th International Conf. on Signal Processing*, 2006, pp. 716-719.

[5] M. Girolami, "Mercer kernel-based clustering in feature space," *IEEE Trans. on Neural Networks*, vol. 13, pp. 780-784, May 2002.

[6] R. Krishnapuram and M. K. James, "A possibilistic approach to clustering," *IEEE Trans. on Fuzzy Systems*, vol. 1, pp. 98-110, May 1993.

[7] X. H. Wu and J. J. Zhou, "A novel possibilistic fuzzy C-means clustering," *Acta Electronica Sinica*, vol. 36, pp. 1996-2000, Oct. 2008.

[8] (2005). ELSDSR: English language speech database for speaker recognition. [Online]. Available: http://www2.imm.dtu.dk/~If/elsdsr/

[9] N. R. Pal, K. Pal, J. M. Leller, and J. C. Bezdek, "A possibilistic fuzzy C-means clustering algorithm," *IEEE Trans. on Fuzzy Systems*, vol. 13, pp. 517-530, Aug. 2005.

[10] N. R. Pal, K. Pal, J. M. Leller, and J. C. Bezdek, "A new hybrid C-means clustering model," in *Proc. IEEE International Conf. on Fuzzy Systems*, 2004, pp. 179-184.

Jie Yang was born in Lanzhou, China in Nov. 1976. She received master's degree from Fudan University in 2007. She is now with Computer and Information Academy, the Shanghai Second Polytechnic University, China. Her research interests now are speech signal processing and application of linear canonical transform in speech signal re-construction.