S5.8

Text-Independent Speaker Recognition by Trajectory Space Comparison

Yifan GONG, Jean-Paul HATON
CRIN/INRIA, BP 239, 54506 Vandoevre, France

Abstract

We present the principle of *trajectory space comparison* for text-independent speaker recognition and some solutions to the space comparison problem based on vector quantization. The comparison of recognition rate of different solutions is reported. Experimental system achieved 99.5% text-independent speaker recognition rate for 23 speakers, using 5 phrases for training and 5 for test. A speaker-independent continuous speech recognition system was built in which this principle is used for speaker adaptation.

1 Introduction

Speaker recognition consists in identifying one speaker among a set of speakers using some test utterances, with samples of speeches of each speaker having been given. The criterion is certain variation among speakers. Researchers in the speech processing field have been interested in automatic speaker recognition problem. In [1], pointers to the work references can be found.

There are three sources of speech variations between speakers: they are the differences in vocal cords and vocal tract shape, in speaking style, and in what speakers choose to say. We are interested in text independent speaker recognition so that only the first source of variations is exploited.

Motivated by the success of word-based vector quantization speech recognition systems, in [2] a system for speaker recognition based on vector quantization codebook has been proposed. During the training phase, the system constructs a codebook by unsupervised classification for each speaker. During the recognition phase, a test utterance is frame-by-frame vector quantization coded using respectively the codebook of each speaker and the recognized speaker is assigned as the one whose codebook has produced the minimum distorsion of the vector quantization coding averaged over the whole test utterance. In a vocabulary-dependent evaluation, i.e. using the same 10-digits vocabulary for training and for testing, a 98% recognition rate for 100 speakers was reported.

However, for the formulation of [2] if the speech parameter subspace relating to speaker i is included in that relating to speaker j a recognition error may take place. In fact, only the subset of vectors in the codebook of j which represent the subspace common to i and j will be used to compute the average coding distorsion. The information in the remaining vectors of the codebook of j will not contribute to the recognition process. In this case the average coding distorsion will have no significant difference for the two speakers and therefore the recognition result is unreliable.

In this paper we introduce a new principle for speaker recognition, which is based on the *comparison of trajectory space* of speakers. In section-2 we describe the principle of speaker recognition by space comparison and four solutions to the problem of space comparison. In section-3 experimental results of speaker recognition based on our proposal are presented and compared. Finally, in section-4 further work to improve the performance is indicated.

2 Principle

Represented in a p-dimensional parametric space R^p, a running speech signal is a moving point. Due to physical constraints of the sound producing system, for a specific speaker $i \in \mathcal{S}$ where $\mathcal{S}$ is a set of speakers, the resulting trajectory of the moving point reaches only a subspace $E_i \subset R^p$.

The basic hypotheses of the proposed method are:

H-1 For different speakers, the subspaces are different:

$$\forall i, j \in \mathcal{S} \quad E_i \neq E_j. \tag{1}$$

H-2 In the recognition phase, a speaker reproduces a similar subspace as that produced in the training phase.

$$\forall i \in \mathcal{S} \quad E_i^T \simeq E_i^R \tag{2}$$

where E_i^T is the parameter subspace reached by the trajectory of speech signal of the speaker i during the *training* phase, and E_i^R is that during the *recognition* phase.

H-1 requires that the parametric space R^p be suitably chosen so that different speakers produce different subspaces. H-2 requires that the training space of a speaker be reproductible in the recognition phase.

These hypotheses mean that the recognition is dependent on the subspaces formed by a trajectory but not on the way how the trajectory is traced. Therefore inherently this approach is text-independent, i.e. the text of the test phrases is not important to the recognition rate. Under these hypotheses, the speaker recognition problem can be solved by space comparison using the following formula, where I_r is the recognized speaker:

$$I_r \leftarrow argmax_{i \in \mathcal{S}} \delta(E_i^T, E_i^R) \tag{3}$$

CH2847-2/90/0000-0285 $1.00 © 1990 IEEE

where $\delta(x,y) : R^p \times R^p \rightarrow [0,1]$ is a *similarity* measure between the spaces x and y, which gives higher value if two spaces are more similar.

Speaker recognition by trajectory space comparison implies therefore to solve two problems: *subspace representation* and *comparison.*

Since modeling parametrically p-dimensional arbitrary spaces with associated relevant similarity is a difficult task. We use non-parametric space representation. In order to represent the subspaces, we vector-quantify subspaces of all speakers, i.e.: for each speaker i a codebook C_i is constructed using training phrases. Subspaces are thus represented using a limited number of centroids stored in the codebooks.

In the following we describe four proposals for computing similarity between spaces, which are based on vector quantization. The input speech signal is converted into some parameter space, resulting in the sequence of L vectors:

$$s_1, s_2, ...s_t, ...s_L.$$

The c^{th} centroid of the codebook of the speaker i is denoted by C_i^c, which is a vector with the same dimension as $\{s_t\}$. The number of codes of the codebook C_i is M_i. The similarity of the vector x to the vector y is $l(x,y)$.

Using vector quantization, in the recognition phase a test utterance is compared to each of all codebooks C_i, $\forall i \in \mathcal{S}$, each giving a space similarity measure S_i. The final speaker is assigned as the one who gives best space similarity:

$$I_r \leftarrow argmax_{\forall i \in \mathcal{S}} S_i. \quad (4)$$

In the following we give algorithms of the four techniques with no intention of doing implementation optimization. All variables and all elements in a array are to be treated as having been initialized to zero.

Algorithm-1

Algorithm-1 (Fig-1) performs vector quantization for each test speech signal frame. After each coding, it cumulates the coding similarity of the best mathed code in the codebook. The resulting space similarity is the average of mean coding similarity of all codes. The mean coding similarity of a code is the cumulated coding similarity divided by the number of times that code has been used for coding the signal. This algorithm is characterized by coding, individual code averaging on similarity and code similarity averaging and thus is called CODE-ICAVE-CAVE.

for $t = 1$ to L do
  $c \leftarrow argmax_{c=1}^{M_i} l(s_t, C_i^c)$
  $s \leftarrow \max_{p=1}^{M_i} l(s_t, C_i^p)$
  $CodeSimilarity_c \leftarrow CodeSimilarity_c + s$
  $CodeCounter_c \leftarrow CodeCounter_c + 1$
done
$S_i \leftarrow \frac{1}{M_i} \sum_{1 \le c \le M_i} \frac{CodeSimilarity_c}{CodeCounter_c}$

Figure 1: Algorithm CODE-ICAVE-CAVE

Algorithm-2

In performing vector quantization for each test speech signal frame, Algorithm-2 (Fig-2) updates the best similarity resulted from coding, for each code in the codebook. The resulting space similarity is the average of best similarities for all codes. Characterized by coding, individual code maximizing on similarity and code similarity averaging, this algorithm is called CODE-ICMAX-CAVE.

for $t = 1$ to L do
  $c \leftarrow argmax_{c=1}^{M_i} l(s_t, C_i^c)$
  $s \leftarrow \max_{p=1}^{M_i} l(s_t, C_i^p)$
  $MaxCodeSimilarity_c \leftarrow \text{Max}(MaxCodeSimilarity_c, s)$
done
$S_i \leftarrow \frac{1}{M_i} \sum_{1 \le c \le M_i} MaxCodeSimilarity_c$

Figure 2: Algorithm CODE-ICMAX-CAVE

Algorithm-3

Algorithm-3 (Fig-3) computes first the best coding similarity for each code, for all test frames and all codes. It then gives the average best coding similarity as result. The average is performed over all codes. This algorithm performs individual code maximizing on similarity and averaging on code similarity, and is therefore called ICMAX-CAVE.

Algorithm-4

As fourth algorithm we implemented the method described in [2] which uses also vector quantization. Algorithm-4 (Fig-4) computes the cumulated best vector quantization similarity during test frame coding, and averages this cumulated value over the number of test frames. Due to the optimization procedure some centroids in the codebook which yield low similarity may not be taken into account in the space similarity S_i. Consequently information in that codebook may not be fully exploited. The principle of this algorithm is coding and averaging similarities over time and is called CODE-TAVE.

for $t = 1$ to L do
  for $c = 1$ to M_i do
    $s \leftarrow l(s_t, C_i^c)$
    $MaxCodeSimi_c \leftarrow \text{Max}(MaxCodeSimi_c, s)$
  done
done
$S_i \leftarrow \frac{1}{M_i} \sum_{1 \le c \le M_i} MaxCodeSimi_c$

Figure 3: Algorithm ICMAX-CAVE

$$\boxed{S_i \leftarrow \frac{1}{L}\sum_{t=1}^{L} \max_{p=1}^{M_i} l(s_t, C_i^p)}$$

Figure 4: Algorithm CODE-TAVE

For speaker recognition, all these four algorithms require $L \times M_i$ times of vector similarity evaluations and thus are of the same algorithmic complexity.

3 Experimental Results

The idea of space comparison for speaker recognition was evaluated on a speech corpus of 23 speakers, 7 females and 16 males, arbitrarily selected from the laboratory staffs. Each speaker was asked to utter 15 different sentences in 3 repetitions. These sentences were extracted from a speech data base designed for speaker recognition [3]. No attempt was made in phrase selection so that parameter difference between speakers might be increased. Sentences in each repetition, numbered $1, 2, ...15$, are organized into 3 training-testing groups, as described in Table-1. This arrangement assures that in each test group the text of sentences used in training and testing are different. In all tests, vector quantization codebook was trained from training sentences only. Totally 207 ($23 speakers \times 3 groups \times 3 repetitions$) tests were carried out.

Test Group	Training	Recognition
G_1	1-5	6-10
G_2	6-10	11-15
G_3	11-15	1-5

Table 1: Each repetition was divided into 3 group of sentences for the evaluation

The speech signal was sampled at 16kHz and pre-empha -sized using the filter

$$H(z) = 1. - 0.94z^{-1}.$$

A $25.6ms$ Hamming window is used to perform short time analysis [4], resulting in a parameter vector sequence. The time interval between two successive analysis windows was $10ms$. The vector quantization codebook training algorithm used is the LBG algorithm proposed in [5].

We performed a preliminary comparative tests on parameter space, similarity measures, dimension of space, and number of codes. We observed that:

- Linear prediction coding coefficients (lpcc) with Itakura likelihood ratio and lpcc-derived lifted cepstral coefficients [6] with Euclidean distance gave similar result.
- 16 coefficients gave good result.
- Codebook size of 128 performed slightly better than codebook size of 64.

Based on these observations, we used 16^{th} autocorrelation linear prediction coding [4] as parameter space. The vector-to-vector similarity measure is based on the Itakura likelihood ratio [7], converted into the range of (0,1) using $y = \frac{1}{x}$.

Tab-2 reports the evaluation results of 207 tests using the four algorithms presented in section-2. Within this table, in the form $n(a_1/a_2/a_3)$, a_i is the number of errors for the i^{th} group and n is the number of errors for the 3 groups of test for a given repetition. M is the codebook size. For $M = 128$ only one repetition was evaluated while for $M = 64$ three repetitions were evaluated.

We obtained a text-independent speaker recognition rate of 99.5% using 16^{th} lpcc space, 64 codes, 207 tests, and 5 training sentences (about 10 seconds) for each speaker and 5 for test.

An example of values of S_i in Eq-4 computed by the Algorithm-3 (ICMAX-CAVE) is given in Tab-3 and Tab-4, in which there is no recognition error.

4 Conclusion

We proposed a new principle for text-independent speaker recognition based on trajectory space comparison. We implemented a set of algorithms computing similarity between parameter spaces using vector quantization. The comparison of experimental speaker recognition results is given.

In our test condition, Algorithm-3 (ICMAX-CAVE) obtained a text-independent speaker recognition rate of 99.5% using 64 codes. The proposed method has given significantly better recognition rate than that of [2].

It is essential that the centroid of codes not used in coding a test utterance be also taken into account in speaker recognition decision making in order to improve recognition rate.

Present single speaker continuous speech recognition systems give very good precision. If speakers could be recognized, or their acoustically most similar speakers could be identified, and if the references of basic recognition units of a certain number of speakers are available, then such systems could be turned into speaker-independent systems with reliable results. Using the result of this work, we have built a speaker-independent continuous speech understanding system [8] in which phoneme reference are dynamically changed to the acoustically most similar speaker's reference.

Further test is needed in order to evaluate the effect of training utterance length, test utterance length, number of codes and number of speakers on the recognition rate.

References

[1] D. O'Shaughnessy. *Speech Communication Human and Machine.* Addison-Wesley Publishing Company, 1987.

[2] F. Soong, A. Rosenberg, L. Rabiner, and B. Juang. A vector quantization approach to speaker recognition. In *Proc. Int. Conf. on Acoustics, Speech and Signal Processing*, pages 387–390, 1985.

[3] O. Mella and M. C. Haton. Méthologie d'étude de la pertinence de paramètres phonétiques et acoustiques pour la reconnaissance du locuteur. In *Actes du séminaire sur la variabilité du locuteur*, Luminy, 1989.

[4] L. R. Rabiner and R. W. Schafer. *Digital Processing of Speech.* Prentice-Hall, Englewood Cliffs, N.J., 1978.

[5] Y. Linde, A. Buzo, and R. M. Gary. An algorithm for the vector quantizer design. *IEEE. Trans. on Communication*, com. 28(1), Jan. 1980.

[6] B. H. Juang, L. R. Rabiner, and J. G. Wilpon. On the use of bandpass lifting in speech recognition. In *Proc. Int. Conf. on Acoustics, Speech and Signal Processing*, pages 765–768, 1986.

[7] N. Nocerino, F. K. Soong, L. R. Rabiner, and D. H. Klatt. Comparative study of several distorsion measures for speech recognition. In *Proc. IEEE Int. Conf. Acoust., Speech and Signal Processing*, pages 25–28, 1985.

[8] Y. Gong, F. Mouria, and J. P. Haton. Un système de reconnaissance de la parole continue sans segmentation. In *Actes du 7ème congrès Reconnaissance des Formes et Intelligence Artificielle*, page , AFCET, INRIA, Paris, France, Nov. 1989.

Methode:	CODE-ICAVE-CAVE	CODE-ICMAX-CAVE	ICMAX-CAVE	CODE-TAVE
1 (M=128)	-	0 (0/0/0)	0 (0/0/0)	0 (0/0/0)
1 (M=64)	3 (1/1/1)	3 (1/0/2)	1 (0/1/0)	1 (1/0/0)
2 (M=64)	3 (1/2/0)	0 (0/0/0)	0 (0/0/0)	3 (1/1/1)
3 (M=64)	2 (0/0/2)	2 (0/0/2)	0 (0/0/0)	5 (3/1/1)
total (M=64)	8	5	1	9
error (%)	3.86	2.42	0.48	4.35

Table 2: Speaker recognition results for four methods. Number of tests: 207 (23 *speakers* × 3 *groups* × 3 *repetitions*)

File	am	aq	as	bz	cf	cs	df	fsc	gaf	gh	gm
am	0.835	0.751	0.737	0.761	0.774	0.778	0.750	0.750	0.760	0.746	0.766
aq	0.730	0.826	0.738	0.768	0.744	0.753	0.777	0.810	0.744	0.774	0.760
as	0.767	0.803	0.852	0.800	0.808	0.777	0.795	0.813	0.806	0.782	0.783
bz	0.785	0.786	0.763	0.858	0.763	0.795	0.785	0.820	0.768	0.756	0.810
cf	0.765	0.735	0.768	0.736	0.876	0.758	0.748	0.755	0.796	0.705	0.771
cs	0.789	0.776	0.744	0.790	0.771	0.838	0.801	0.787	0.747	0.760	0.782
df	0.773	0.792	0.792	0.787	0.799	0.793	0.851	0.796	0.800	0.790	0.806
fsc	0.776	0.803	0.788	0.812	0.787	0.796	0.808	0.858	0.781	0.766	0.798
gaf	0.771	0.754	0.762	0.747	0.823	0.742	0.771	0.754	0.865	0.729	0.769
gh	0.778	0.807	0.802	0.801	0.796	0.784	0.814	0.815	0.794	0.868	0.791
gm	0.767	0.786	0.772	0.785	0.767	0.779	0.785	0.792	0.776	0.765	0.835
gyf	0.771	0.793	0.790	0.789	0.776	0.764	0.780	0.800	0.789	0.798	0.796
jfm	0.768	0.719	0.692	0.741	0.743	0.772	0.746	0.719	0.720	0.703	0.757
jg	0.737	0.791	0.772	0.774	0.757	0.736	0.777	0.821	0.770	0.750	0.769
jlc	0.781	0.730	0.697	0.733	0.740	0.752	0.728	0.729	0.708	0.731	0.733
jmp	0.771	0.789	0.783	0.797	0.767	0.768	0.775	0.795	0.777	0.791	0.778
jph	0.788	0.743	0.706	0.762	0.762	0.763	0.760	0.752	0.747	0.717	0.760
mcp	0.744	0.737	0.750	0.726	0.797	0.722	0.737	0.745	0.797	0.732	0.736
ms	0.759	0.768	0.717	0.784	0.736	0.762	0.760	0.779	0.727	0.765	0.783
nc	0.759	0.766	0.766	0.759	0.818	0.759	0.775	0.780	0.798	0.745	0.783
of	0.722	0.758	0.755	0.733	0.810	0.726	0.768	0.763	0.814	0.737	0.745
ot	0.722	0.740	0.754	0.738	0.793	0.715	0.737	0.728	0.773	0.672	0.731
sm	0.736	0.756	0.771	0.747	0.833	0.747	0.773	0.789	0.807	0.744	0.758

Table 3: speaker recognition results for 23 speakers, G_2, second repetition, (ICMAX-CAVE algorithm, lpcc space, part-I)

File	gyf	jfm	jg	jlc	jmp	jph	mcp	ms	nc	of	ot	sm
am	0.741	0.773	0.765	0.811	0.741	0.783	0.740	0.766	0.727	0.721	0.727	0.744
aq	0.749	0.734	0.800	0.725	0.760	0.751	0.739	0.791	0.756	0.764	0.746	0.738
as	0.790	0.761	0.831	0.747	0.767	0.766	0.795	0.790	0.786	0.800	0.798	0.798
bz	0.770	0.758	0.806	0.770	0.774	0.788	0.741	0.804	0.729	0.764	0.743	0.754
cf	0.725	0.748	0.771	0.722	0.713	0.733	0.793	0.736	0.777	0.786	0.797	0.824
cs	0.747	0.778	0.770	0.773	0.769	0.770	0.753	0.791	0.743	0.773	0.732	0.747
df	0.782	0.799	0.810	0.768	0.777	0.788	0.785	0.808	0.787	0.806	0.791	0.775
fsc	0.798	0.767	0.823	0.769	0.769	0.802	0.764	0.804	0.766	0.780	0.772	0.777
gaf	0.741	0.769	0.765	0.735	0.740	0.756	0.806	0.755	0.767	0.793	0.808	0.804
gh	0.808	0.778	0.821	0.789	0.807	0.787	0.777	0.819	0.778	0.782	0.771	0.781
gm	0.775	0.767	0.787	0.734	0.782	0.762	0.760	0.783	0.745	0.778	0.763	0.758
gyf	0.859	0.752	0.812	0.758	0.789	0.806	0.760	0.808	0.752	0.774	0.770	0.756
jfm	0.697	0.821	0.720	0.771	0.717	0.745	0.728	0.754	0.713	0.721	0.702	0.729
jg	0.769	0.730	0.853	0.726	0.755	0.762	0.748	0.773	0.750	0.754	0.768	0.753
jlc	0.730	0.749	0.746	0.845	0.719	0.773	0.692	0.760	0.708	0.677	0.687	0.700
jmp	0.784	0.763	0.799	0.766	0.831	0.767	0.764	0.798	0.758	0.767	0.753	0.764
jph	0.734	0.753	0.762	0.789	0.717	0.841	0.749	0.774	0.731	0.705	0.700	0.713
mcp	0.729	0.732	0.750	0.717	0.728	0.729	0.838	0.735	0.781	0.789	0.783	0.789
ms	0.744	0.733	0.771	0.757	0.761	0.769	0.722	0.836	0.725	0.726	0.719	0.710
nc	0.747	0.761	0.777	0.736	0.739	0.743	0.803	0.780	0.850	0.796	0.788	0.797
of	0.744	0.737	0.764	0.707	0.733	0.737	0.813	0.754	0.792	0.855	0.813	0.798
ot	0.723	0.729	0.757	0.688	0.702	0.703	0.764	0.709	0.755	0.774	0.831	0.776
sm	0.737	0.743	0.777	0.730	0.744	0.740	0.804	0.771	0.795	0.821	0.811	0.852

Table 4: speaker recognition results for 23 speakers, G_2, second repetition, (ICMAX-CAVE algorithm, lpcc space, part-II)