Class-Discriminative Weighted Distortion Measure for VQ-Based Speaker Identification

Tomi Kinnunen and Ismo Kärkkäinen

University of Joensuu, Department of Computer Science,
P.O. Box 111, 80101 JOENSUU, FINLAND
{tkinnu,iak}@cs.joensuu.fi

Abstract. We consider the distortion measure in vector quantization based speaker identification system. The model of a speaker is a codebook generated from the set of feature vectors from the speakers voice sample. The matching is performed by evaluating the distortions between the unknown speech sample and the models in the speaker database. In this paper, we introduce a weighted distortion measure that takes into account the correlations between the known models in the database. Larger weights are assigned to vectors that have high discriminating power between the speakers and vice versa.

1 Introduction

It is well known that different phonemes have unequal discrimination power between speakers [14, 15]. That is, the inter-speaker variation of certain phonemes are clearly different from other phonemes. This knowledge should be exploited in the design of speaker recognition [6] systems. Acoustic units that have higher discrimination power should contribute more to the similarity or distance scores in the matching.

The description of acoustic units in speech and speaker recognition is often done via *short-term spectral features*. Speech signal is analyzed in short segments (frames) and a representative feature vector for each frame is computed. In speaker recognition, *cepstral coefficients* [5] along with their 1st and 2nd time derivatives (Δ- and ΔΔ-coefficients) are commonly used. Physically these represent the shapes of the vocal tract and their dynamic changes [1, 2, 5], and therefore carry information about the *formant structure* (vocal tract resonant frequencies) and dynamic formant changes.

In *vector quantization* (*VQ*) based speaker recognition [3, 8, 9, 10, 16], each speaker (or *class*) is presented by a codebook which approximates his/her data density by a small number of representative *code vectors*. Different regions (*clusters*) in the feature space represent acoustically different units.

The question how to benefit from the different discrimination power of phonemes in VQ-based speaker recognition returns into question how to assign *discriminative weights* for different code vectors and how to adopt these weights into the distance or similarity calculations in the matching phase. As a motivating example, Fig. 1 shows two scatter plots of four different speakers cepstral code vectors derived from the TIMIT speech corpus. In both plots, two randomly chosen components of the 36-dimensional cepstral vectors are shown. Each speakers data density is presented as a codebook of 32 vectors. As can be seen, different classes have strong overlap. However, some speakers do have code vectors that are far away from all other classes. For instance, speakers marked by ”•” and ”Δ” in the rightmost plot have both such code vectors that are especially good for discriminating them from other speakers.

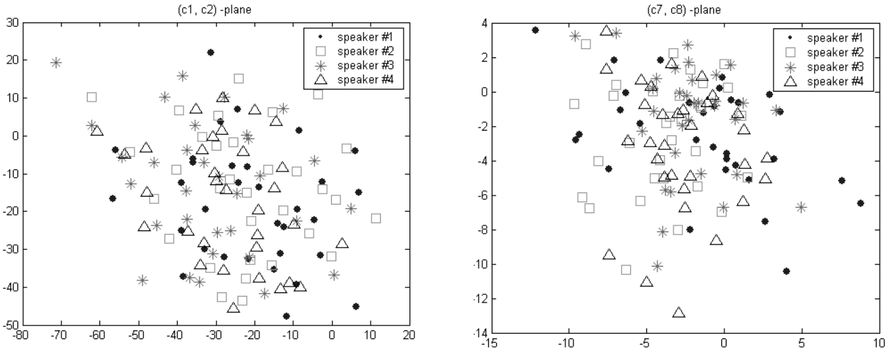

Fig. 1. Scatter plots of two randomly chosen dimensions of four speakers cepstral data from TIMIT database

There are two well-known ways for improving class separability in pattern recognition. The first one is to improve separability in the *training phase* by *discriminative training algorithms*. Examples in the VQ context are *LVQ* [12] and *GVQ* [8] algorithms. The second discrimination paradigm, *score normalization*, is used in the *decision phase*. For instance, matching scores of the client speaker in speaker verification can be normalized against matching scores obtained from a *cohort set* [3].

In this paper, we introduce a third alternative for improving class separability and apply it to speaker identification problem. For a given set of codebooks, we assign discriminative weights for each of the code vectors. In the matching phase, these weights are retrieved from a look-up table and used in the distance calculations directly. Thus, the time complexity of the matching remains the same as in the unweighted case.

The outline of this paper is as follows. In Section 2, we shortly review the baseline VQ-based speaker identification. In Section 3, we give details of the weighted distortion measure. Experimental results are reported in Section 4. Finally, conclusions are drawn in Section 5.

2 VQ-Based Speaker Identification

Speaker identification is a process of finding the best matching speaker from a speaker database, when given an unknown speakers voive sample [6]. In VQ-based speaker identification [8, 9, 11, 16], vector quantization [7] plays two roles. It is used both in the training and matching phases. In the training phase, the speaker models are constructed by clustering the feature vectors in K separate clusters. Each cluster is represented by a *code vector* $\boldsymbol{c}_i$, which is the centroid (average vector) of the cluster. The resulting set of code vectors is called a *codebook*, and notated here by $C^{(j)} = \{\boldsymbol{c}_1^{(j)}, \boldsymbol{c}_2^{(j)}, ..., \boldsymbol{c}_K^{(j)}\}$. The superscript ($j$) denotes speaker number.

In the codebook, each vector represents a single acoustic unit typical for the particular speaker. Thus, the distribution of the feature vectors is represented by a smaller set of sample vectors with similar distribution than the full set of feature vectors of the speaker model. The codebook size should be set reasonably high since the previous results indicate that the matching performance improves with the size of the codebook [8, 11, 16]. For the clustering we use the *randomized local search* (RLS) algorithm [4] due its superiority in codebook quality over the widely used LBG method [13].

In the matching phase, VQ is used in computing a *distortion* $D(X, C^{(i)})$ between an unknown speakers feature vectors $X = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_T\}$ and all codebooks $\{C^{(1)}, C^{(2)}, ... , C^{(N)}\}$ in the speaker database [16]. A simple decision rule is to select the speaker i^* that minimizes the distortion, i.e.

$$i^* = \arg\min_{1 \le i \le N} D(X, C^{(i)}) \, . \tag{1}$$

A natural choice for the distortion measure is the average distortion [8, 16] defined as

$$D(X, C) = \frac{1}{T} \sum_{\boldsymbol{x} \in X}^{T} d(\boldsymbol{x}, \boldsymbol{c}_{NN[\boldsymbol{x}]}) \, , \tag{2}$$

where $NN[\boldsymbol{x}]$ is the index of the nearest code vector to $\boldsymbol{x}$ in the codebook and $d(.,.)$ is a *distance measure* defined for the feature vectors. In words, each vector from the unknown feature set is quantized to its nearest neighbor in the codebook and the sum of the distances is normalized by the length of the test sequence. A popular choice for the distance measure d is the Euclidean distance or its square. In [15] it is justified that Euclidean distance of two cepstral vectors is a good measure for the dissimilarity of the corresponding short-term speech spectra. In this work, we use squared Euclidean distance as the distance measure.

In the previous work [10] we suggested an alternative approach to the matching. Instead minimizing distortion, maximization of a *similarity measure* was proposed. However, later experiments have pointed out that it is difficult to define a natural and intuitive similarity measure in the same way as distortion (2) is defined. For that reason, we limit our discussion to distortion measures.

3 Speaker Discriminative Matching

As an introduction, consider the two speakers codebooks illustrated in Fig. 2. Vectors marked by ”•” represent an unknown speakers’ data. Which one is this speaker? We can see that the uppermost code vector $\boldsymbol{c}_2^{(1)}$ is actually the only vector which clearly turns the decision to the speaker #1. Suppose that there wasn’t that code vector. Then the average distortion would be approximately same for both speakers. There are clearly three regions in the feature space which cannot distinguish these two speakers. Only the code vectors $\boldsymbol{c}_2^{(1)}$ and $\boldsymbol{c}_3^{(2)}$ can make the difference, and they should be given a large discrimination weight.

3.1 Weighted Distortion Measure

We define our distortion measure by modifying (2) as follows:

$$D(X,C)=\frac{1}{T}\sum_{\boldsymbol{x}\in X}^{T} f(w_{NN[\boldsymbol{x}]})d(\boldsymbol{x},\boldsymbol{c}_{NN[\boldsymbol{x}]})\,. \qquad \textbf{(3)}$$

Here $w_{NN[x]}$ is the weight associated with the nearest code vector, and f is a non-increasing function of its argument. In other words, code vectors that have good discrimination (large weight) tend to decrease the distances *d*; vice versa, non-discriminative code vectors (small weight) tend to increase the distances. Product $f(w)d(\boldsymbol{x},\boldsymbol{c})$ can be viewed as an operator which ”attracts” (decreases overall distortion) vectors $\boldsymbol{x}$ that are close to $\boldsymbol{c}$ or the corresponding weight w is large. Likewise, it ”repels” (increases overall distortion) such vectors $\boldsymbol{x}$ that are far away or are quantized with small w.

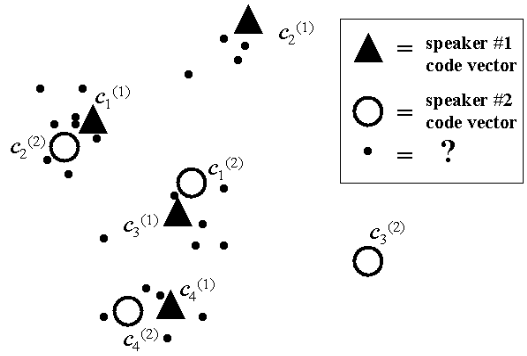

Fig. 2. Illustration of code vectors with unequal discrimination powers

An example of a quantization of a single vector is illustrated in Fig. 3. Three speakers' code vectors and corresponding weights are shown. For instance, the code vector at location (8, 4) has a large weight, because there are no other classes' presentatives in its neighborhood. The three code vectors in the down left corner, in turn, have all small weights because they all have another classes' representative near. When quantizing the vector marked by ×, the unweighted measure (2) would give the same distortion value $D \cong 7.5$ for all classes (squared Euclidean distance). However, when using the weighted distortion (3.1), we get distortion values $D_1 \cong 6.8$, $D_2 \cong 6.8$ and $D_3 \cong 1.9$ for the three classes, respectively. Thus, × is favored by the class #3 due to the large weight of the code vector. We have not yet specified two important issues in the design of the weighted distortion, namely:

- How to assign the code vector weights,
- Selection of the function *f*.

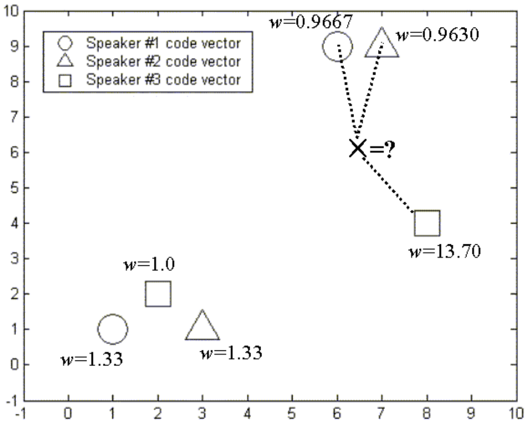

Fig. 3. Weighted quantization of a single vector

In this work, we fix the function *f* as a decaying exponential of the form

$$f(w) = e^{-\alpha w}, \tag{4}$$

where α >0 is a parameter that controls the rate of decay. In the above example, α=0.1.

3.2 Assigning the Weights

The weight of a code vector should depend on the minimum distances to other classes code vectors. Let $\boldsymbol{c} \in C^{(j)}$ be a code vector of the *j*th speaker. Let us denote the index of its nearest neighbor in the *k*th codebook simply by $NN^{(k)}$. The weight of $\boldsymbol{c}$ is then assigned as follows:

$$w(\boldsymbol{c}) = \frac{1}{\sum_{k \neq j} 1/d(\boldsymbol{c}, \boldsymbol{c}_{NN^{(k)}})}. \tag{5}$$

In other words, nearest code vector from all other classes are found, and the inverse of the sum of inverse distances is taken. If some of the distances equals 0, we set $w(\boldsymbol{c}) = 0$ for mathematical convenience. The algorithm is looped over all code vectors and all codebooks.

As an example, consider the code vector located at (1,1) in Fig. 3. The distances (squared Euclidean) to the nearest code vectors in other classes are 2.0 and 4.0. Thus, the weight for this code vector is $w = 1/(1/2.0 + 1/4.0) = 1.33$. In the practical implementation, we further normalize the weights within each codebook such that their sum equals 1. Then all weights satisfy $0 \leq w \leq 1$, which makes them easier to handle and interpret.

4 Experimental Results

For testing purposes, we used a 100 speaker subset from the American English TIMIT corpus. We resampled the wave files down to 8.0 kHz with 16-bit resolution. The average duration of the training speech per speaker was approximately 15 seconds. For testing purposes we derived three test sequences from other files with durations 0.16, 0.8 and 3.2 seconds. The feature extraction was performed using the following steps:

- Pre-emphasis filtering with $H(z)=1-0.97z^{-1}$.
- 12th order mel-cepstral analysis with 30 ms Hamming window, shifted by 15 ms.

The feature vectors were composed of the 12 lowest mel-cepstral coefficients (excluded the 0th coefficient). The Δ- and $\Delta\Delta$-cepstral were added to the feature vectors, thereby implying 3×12=36-dimensional feature space.

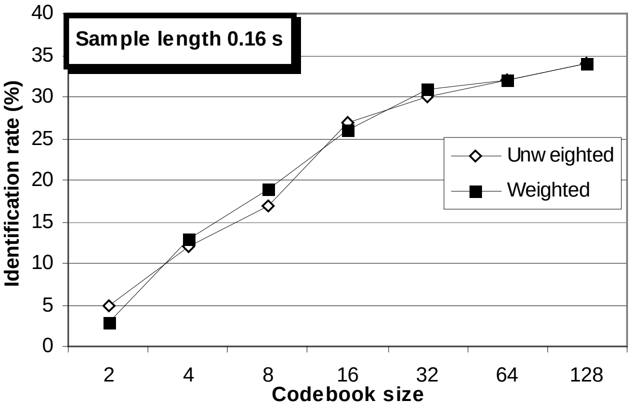

Fig. 4. Performance evaluation using ~0.16 s. speech sample (~10 vectors)

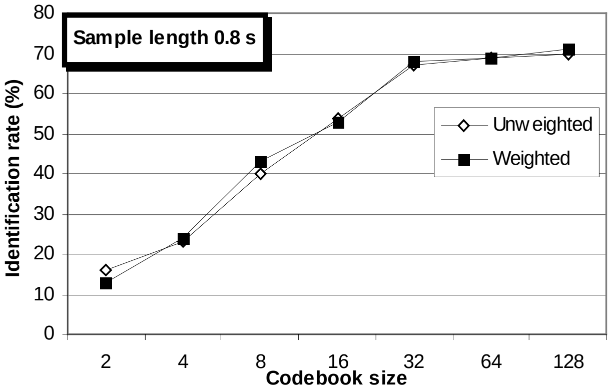

Fig. 5. Performance evaluation using ~0.8 s. speech sample (~50 vectors)

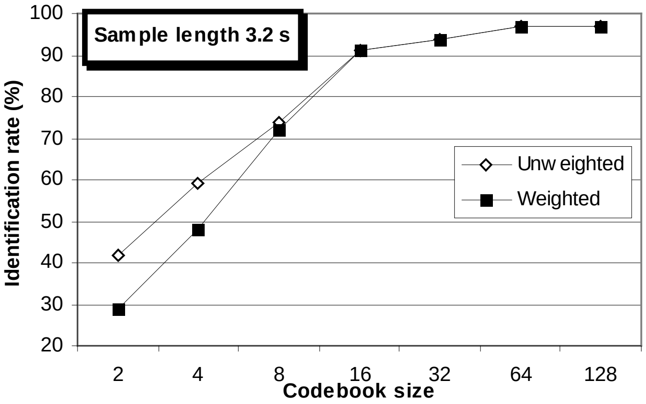

Fig. 6. Performance evaluation using ~3.2 s. speech sample (~200 vectors)

The identification rates by using the reference method (2) and the proposed method (3) are summarized through Figs. 4 - 6 for the three different subsequences by varying the codebook sizes from K = 2 to 128. The parameter α of (4) is fixed in all three experiments to α = 1.

The following observations can be made from the figures. The proposed method does not perform consistently better than the reference method. In some cases the reference method (unweighted) outperforms the proposed (weighted) method, especially for low codebook sizes. For large codebooks the ordering tends to be opposite. This phenomenon is probably due to the fact that small codebook sizes give a poorer representation of the training data, and thus the weight estimates cannot be good either.

Both methods give generally better results with increasing codebook size and test sequence length. Both methods saturate to the maximum accuracy (97 %) with the longest test sequence (3.2 seconds of speech) and codebook size K=64. In this case, using codebook K=128 does not improve accuracy any more.

5 Conclusions

We have proposed a framework for improving class separability in pattern recognition and evaluated the approach in the speaker identification problem. In general, results show that with proper design VQ-based speaker identification system can achieve high recognition rates with very short test samples while model having low complexity (codebook size K = 64). Proposed method adapts to a given set of classes represented by codebooks by computing discrimination weights for all code vectors and uses these weights in the matching phase. The results obtained in this work show no clear improvement over the reference method. However, together with the results obtained in [10] we conclude that weighting indeed can be used to improve class separability. The critical question is: how to take full advantage of the weights in the distortion or similarity measure? In future work, we will focus on the optimization of the weight decay function f.

References

1. Deller, J.R. Jr., Hansen, J.H.L., Proakis, J.G.: *Discrete-time Processing of Speech Signals*. Macmillan Publishing Company, New York, 2000.
2. Fant, G.: *Acoustic Theory of Speech Production*. The Hague, Mouton, 1960.
3. Finan R.A., Sapeluk A.T., Damper R.I.: ”Impostor cohort selection for score normalization in speaker verification,” *Pattern Recognition Letters*, **18**: 881-888, 1997.
4. Fränti, P., Kivijärvi, J.: “Randomized local search algorithm for the clustering problem,” *Pattern Analysis and Applications*, **3**(4): 358-369, 2000.
5. Furui, S.: ”Cepstral analysis technique for automatic speaker verification,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, **29**(2): 254-272, 1981.
6. Furui, S.: ”Recent advances in speaker recognition,” *Pattern Recognition Letters*, **18**: 859-872, 1997.
7. Gersho, A., Gray, R.M., Gallager, R.: *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1991.
8. He, J., Liu, L., Palm, G.: ”A discriminative training algorithm for VQ-based speaker identification,” *IEEE Transactions on Speech and Audio Processing*, **7**(3): 353-356, 1999.
9. Jin, Q., Waibel, A.: “A naive de-lambing method for speaker identification,” *Proc. ICSLP 2002*, Beijing, China, 2000.
10. Kinnunen, T., Fränti, P.: ”Speaker discriminative weighting method for VQ-based speaker identification,” *Proc. 3rd International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA))*: 150-156, Halmstad, Sweden, 2001
11. Kinnunen, T., Kilpeläinen, T., Fränti P.: ”Comparison of clustering algorithms in speaker identification,” *Proc. IASTED Int. Conf. Signal Processing and Communications (SPC)*: 222-227, Marbella, Spain, 2000.
12. Kohonen T.: *Self-Organizing Maps*. Springer-Verlag, Heidelberg, 1995.
13. Linde, Y., Buzo, A., Gray, R.M.: ”An algorithm for vector quantizer design,” *IEEE Transactions on Communications*, **28**(1): 84-95, 1980
14. Nolan, F.: *The Phonetic Bases of Speaker Recognition*. Cambridge CUP, Cambridge, 1983.
15. Rabiner, L., Juang B.: *Fundamentals of Speech Recognition*. Prentice Hall, 1993.
16. Soong, F.K., Rosenberg, A.E., Juang, B-H., Rabiner, L.R.: ”A vector quantization approach to speaker recognition,” *AT&T Technical Journal*, **66**: 14-26, 1987.