Speaker Discriminative Weighting Method for VQ-based Speaker identification

Tomi Kinnunen and Pasi Fränti

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

Abstract: We consider the matching function 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 similarity of the unknown speaker and the models in the database. In this paper, we propose to use weighted matching method 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. Experiments show that the new method provides significantly higher identification accuracy and it can detect the correct speaker from shorter speech samples more reliable than the unweighted matching method.

1. Introduction

Various phonetic studies have showed that different parts of speech signal have unequal discrimination properties between speakers. That is, the inter-speaker variation of certain phonemes are clearly different from other phonemes. Therefore, it would be useful to take this knowledge into account when designing speaker recognition systems.

There are several alternative approaches to utilize the above phenomen. One approach is to use a front-end pre-classifier that would automatically recognize the acoustic units and give a higher significance for units that have better discriminating power. Another approach is to use weighting method in the front-end processing. This is usually realized by a method called *cepstral liftering*, which has been applied both in the speaker [3,9] and speech recognition [1]. However, all front-end weighting strategies depend on the parametrization (vectorization) of the speech and, therefore, do not provide a general solution to the speaker identification problem.

In this paper, we propose a new weighted matching method to be used in vector quantization (VQ) based speaker recognition. The matching takes into account the correlations between the known models and assigns larger weights for code vectors that have high discriminating power. The method does not require any *a priori* knowledge about the nature of the feature vectors, or any phonetic knowledge about the discrimination powers of the different phonemes. Instead, the method adapts to the statistical properties of the feature vectors in the given database.

2. Vector Quantization in Speaker Recognition

In VQ-based recognition system [4, 5, 6, 8], a speaker is modeled as a set of feature vectors generated from his/her voice sample. The speaker models are constructed by clustering the feature vectors in K separate clusters. Each cluster is then represented by a *code vector*, which is the centroid (average vector) of the cluster. The resulting set of code vectors is called a *codebook*, and it is stored in the speaker database.

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 should be set reasonably high since the previous results indicate that the matching performance improves with the size of the codebook [5, 7, 8]. For the clustering we use the *randomized local search* (RLS) algorithm as described in [2].

The matching of an unknown speaker is then performed by measuring the similarity/dissimilarity between the feature vectors of the unknown speaker to the models (codebooks) of the known speakers in the database. Denote the sequence of feature vectors extracted from the unknown speaker as $X = \{x_1, ..., x_T\}$. The goal is to find the best matching codebook C_{best} from the database of N codebooks $C = \{C_1, ..., C_N\}$. The matching is usually evaluated by a *distortion measure*, or *dissimilarity measure* that calculates the average distance of the mapping $d: X \times C \to \mathbf{R}$ [5, 8]. The best matching codebook can then be defined by the codebook that *minimizes* the dissimilarity measure.

Instead of the previous approaches, we use a *similarity measure*. In this way, we can define the weighting matching method intuitively more clearly. Thus, the best matching codebook is now defined as the codebook that *maximizes* the similarity measure of the mapping $s: X \times C \to \mathbf{R}$, i.e.:

$$C_{\text{best}} = \arg\max_{1 \le i \le N} \{s(X, C_i)\}. \tag{2.1}$$

Here the similarity measure is defined as the average of the inverse distance values:

$$s(X, C_i) = \frac{1}{T} \sum_{t=1}^{T} \frac{1}{d(\boldsymbol{x}_t, \boldsymbol{c}_{\min}^{i,t})}, \tag{2.2}$$

where $\boldsymbol{c}_{\min}^{i,t}$ denotes the nearest code vector to $\boldsymbol{x}_t$ in the codebook C_i and $d: \mathbf{R}^{\mathrm{P}} \times \mathbf{R}^{\mathrm{P}} \to \mathbf{R}$ is a given distance function in the feature space, whose selection depends of the properties of the feature vectors. If the distance function d satisfies $0 < d < \infty$, then s is a well-defined and $0 < s < \infty$. In the rest of the paper, we use *Euclidean distance* for simplicity. Note that in practice, we limit the distance values to the range $1 < d < \infty$ and, thus, the effective values of the similarity measure are $0 < s < 1$.

3. Speaker Discriminative Matching

Consider the example shown in Fig. 1, in which the code vectors of three different speakers are marked by rectangles, circles and triangles. There is also a set of vectors from an unknown speaker marked by stars. The region at the top rightmost corner cannot distinct the speakers from each other since it contains code vectors from all speakers. The region at the top leftmost corner is somewhat better in this sense because samples there indicate that the unknown speaker is not "triangle". The rest of the code vectors, on the other hand, have much higher discrimination power because they are isolated from the other code vectors.

Let us consider the unknown speaker "star", whose sample vectors are concentrated mainly around three clusters. One cluster is at the top rightmost corner and it cannot distinct, which speaker the sample vectors originate from. The second cluster at the top leftmost corner can rule out the speaker "triangle" but only the third cluster makes the difference. The cluster at the right middle indicates only to the speaker "rectangular" and, therefore, we can conclude that the sample vectors of the unknown speaker originate from the speaker "rectangular".

The situation is not so evident if we use the unweighted similarity score of the formula (2.2). It gives equal weight to all sample vectors despite the fact that they do not have the same significance in the matching. Instead, the similarity value should depend on two separate factors: the distance to the nearest code vector, and the discrimination power of the code vector. Outliers and noise vectors that do not match well to any code vector should have small impact, but also vectors that match to code vectors of many speakers should have smaller impact on the matching score.

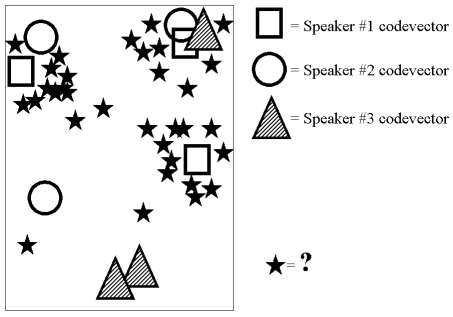

Fig. 1: Illustration of code vectors having different discriminating power.

3.1 Weighted similarity measure

Our approach is to assign weights to the code vectors according to their discrimination power. In general, the weighting scheme can be formulated by modifying the formula (2.2) as follows:

$$s_w(X, C_i) = \frac{1}{T}\sum_{t=1}^{T} \frac{1}{d(\boldsymbol{x}_t, \boldsymbol{c}_{\min}^{i,t})} \cdot w(\boldsymbol{c}_{\min}^{i,t}), \qquad (3.1)$$

where *w* is the *weighting function*. When multiplying the local similarity score, $1/d(\boldsymbol{x}_t, \boldsymbol{c}_{\min}^{i,t})$, with the weight associated with the nearest code vector, $\boldsymbol{c}_{\min}^{i,t}$, the product can be thought as a local operator that moves the decision surface towards more significant code vectors.

3.2 Computing the weights

Consider a database of speaker codebooks $C_1,...,C_N$. The codebooks are post-processed to assign weights for the code vectors, and the result of the process is a set of weighted codebooks $(C_i, W_i), i = 1,..., N$, where $W_i = \{w(\boldsymbol{c}_{i1}),..., w(\boldsymbol{c}_{iK})\}$ are the weights assigned for the *i*th codebook. In this way, the weighting approach does not increase the computational load of the matching process as it can be done in the training phase when creating the speaker database. The weights are computed using the following algorithm:

```
PROCEDURE ComputeWeights(S: SET OF CODEBOOKS) RETURNS WEIGHTS
FOR EACH Ci IN S DO                 % Loop over all codebooks
    FOR EACH cj IN Ci DO            % Loop over code vectors
       sum := 0;
       FOR EACH Ck , k ≠ i, IN S DO          % Find nearest code vector_
           dmin := DistanceToNearest(cj, Ck); % _ from all other codebooks
           sum := sum + 1/dmin;
       ENDFOR
       w(cij) := 1/sum;
   ENDFOR;
ENDFOR;
```

4. Experimental Results

For testing purposes, we collected a database of 25 speakers (14 males + 11 females) using sampling rate of 8.0 kHz with 16 bits/sample. The average duration of the training samples was 66.5 seconds per speaker. For matching purposes we recorded another sentence of the length 8.85 seconds, which was further divided into three different subsequences of the lengths 8.85 s (100%), 1.77 s (20%) and 0.177 s (2%).

The feature extraction was performed using the following steps:

- High-emphasis filtering with filter $H(z) = 1 - 0.97z^{-1}$.
- 12^{th} order mel-cepstral analysis with 30 ms Hamming window, shifted by 10 ms.

The feature vectors were composed of the 12 lowest mel-cepstral coefficients (except the 0^{th} coefficient, which corresponds to the total energy of the frame). We concatenated the feature vectors also with the Δ- and $\Delta\Delta$-coefficients (1^{st} and 2^{nd} time derivatives of the cepstral coefficients) to capture the dynamic behavior of the vocal tract. The dimension of the final feature vector is therefore 3×12 = 36.

The identification rates are summarized through Fig. 2-4 for the three different subsequences by varying the codebook sizes from *K*=1 to 256.

The proposed method (weighted similarity) outperforms the reference method (unweighted similarity) in all cases. It reaches 100% identification rate with $K \geq 32$ using only 1.7 seconds of speech (corresponding to 172 test vectors). Even with a very short test sequence of 0.177 seconds (17 test vectors) the proposed method can reach identification rate of 84% whereas the reference method is practically useless.

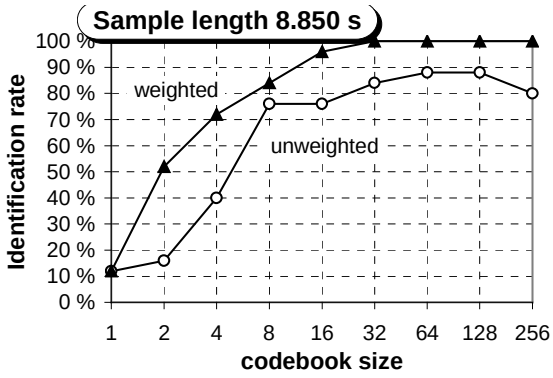

Fig. 2. Performance evaluation using the full test sequence.

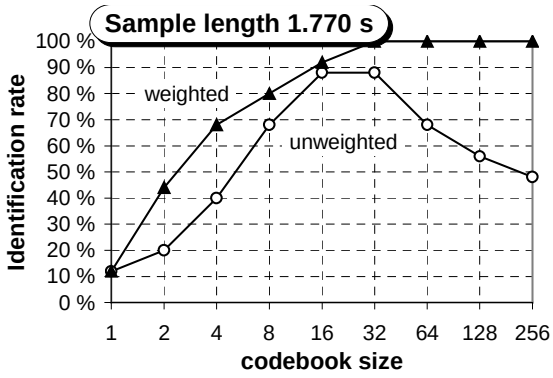

Fig. 3. Performance evaluation using 20 % of the test sequence.

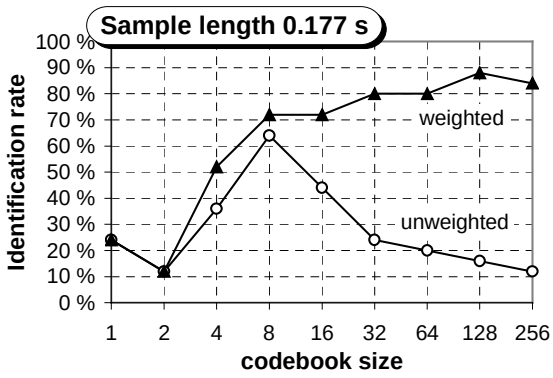

Fig. 4. Performance evaluation using 2 % of the test sequence.

5 Conclusions

We have proposed and evaluated a weighted matching method for text-independent speaker recognition. Experiments show that the method gives tremendous improvement over the reference method, and it can detect the correct speaker from much shorter speech samples. It is therefore well applicable in real-time systems. Furthermore, the method can be generalized to any other pattern recognition tasks because it is not designed for any particular features or distance metric.

References

[1] Deller Jr. J.R., Hansen J.H.L., Proakis J.G.: *Discrete-time Processing of Speech Signals*. Macmillan Publishing Company, New York, 2000.

[2] Fränti P., Kivijärvi J.: "Randomized local search algorithm for the clustering problem", *Pattern Analysis and Applications*, **3**(4): 358-369, 2000.

[3] Furui S.: "Cepstral analysis technique for automatic speaker verification". *IEEE Transactions on Acoustics, Speech and Signal Processing*, **29**(2): 254-272, 1981.

[4] 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.

[5] 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.

[6] Kyung Y.J., Lee H.S.: "Bootstrap and aggregating VQ classifier for speaker recognition". *Electronics Letters*, **35**(12): 973-974, 1999.

[7] Pham T., Wagner M., "Information based speaker identification", *Proc. Int. Conf. Pattern Recognition (ICPR*), **3**: 282-285, Barcelona, Spain, 2000.

[8] 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.

[9] Zhen B., Wu X., Liu Z., Chi H.: "On the use of bandpass liftering in speaker recognition", *Proc. 6th Int. Conf. of Spoken Lang. Processing* (*ICSLP*), Beijing, China, 2000.