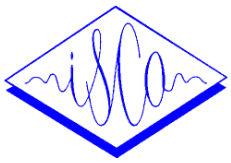

ESCA Workshop on Automatic Speaker Recognition, Identification, and Verification
Martigny, Switzerland
April 5-7, 1994

Similarity Normalization Method for Speaker Verification Based on A Posteriori Probability

Tomoko Matsui and Sadaoki Furui

Abstract— **This paper proposes two methods for creating a pooled model for all registered speakers to reduce the enormous amount of calculation needed by the similarity normalization method for speaker verification based on a posteriori probability. The proposed methods perform the same as or better than the original method and the amount of calculation is reduced significantly. Speaker verification is tested by using separate populations of customers and impostors in order to evaluate performance under practical conditions. The speaker (and text) verification error rates are roughly 1.6 times larger than if the same population is used for both customers and impostors. Using 15 customers and a separate group of 15 impostors, one proposed method achieves a speaker verification error rate of 1.6% for text-independent verification and a speaker and text verification error rate of 1.1%, which is about half that with the original method in text-prompted verification.**

Keywords— **Similarity normalization, speaker verification, a posteriori probability, pooled model.**

1. Introduction

In speaker verification, the similarity value between the input speech and the reference model or template of the speech for the person whose identity is being claimed is calculated and compared with a threshold. The identity claim of a speaker is accepted when the similarity value exceeds the threshold and is rejected when it is smaller. The similarity value has a wide range for different texts spoken at different times, even by the same speaker, so it is difficult to set stable thresholds.

Higgins et al. [4] proposed the similarity normalization method, which is based on the likelihood ratio and Rosenberg et al. [5] examined the effectiveness of Higgins' method. We [1][2] reported a normalization method based on a posteriori probability. Our method normalizes the similarity value between the input speech and the model for the claimed speaker by subtracting the average value of the (n highest) similarities for the models of all registered speakers. In Section 5.1, We compare the performance of Higgins' method with our method. Both have a common disadvantage: theoretically, the similarity values between the input speech and the models for all registered speakers must be calculated, and the amount of calculation is linearly proportional to the population of the registered speakers.

In [4] and [5], cohort speakers were selected from the registered speakers for each customer, and the similarity values for all the registered speakers were approximated by those for the cohort speakers. It is difficult, however, to select a proper set of cohort speakers. In this paper, the summation of the similarity values for the models for all registered speakers is approximated by the similarity value for a pooled model, which is formed by pooling the features of all the registered speakers. Two methods for creating the pooled model are proposed. We investigate how all of the registered speakers' utterances can be used to create an effective pooled model with reasonable computational effort. In both methods, HMMs (hidden Markov models) are used for the pooled model.

In our previous work [1]-[3], speaker verification was performed choosing one speaker to be the customer and the other registered speakers as impostors and rotating through all the speakers. Thus, the models of the impostors were included in the normalization, which causes problems, because the impostor population could be much larger than the registered speaker population. Therefore, as pointed out in [5], it is not realistic for the population of impostors to be the same as that for the registered speakers who are used for normalization, or for a model of an impostor to be used for every normalization. In this paper, speaker verification is performed by using separate populations of customers and impostors.

In our previous experiments [1][2], the logarithm of the summation of the similarity values for normalization was approximated by the summation of the logarithms of the similarity values. Section 5.2 investigates the effect of this approximation.

2. Speaker Verification

We used two speaker recognition methods to verify the performance of our normalization method.

One is the text-independent method [3]. A one-state text-independent HMM is made for each speaker; the accumulated likelihood of the input speech frames for the HMM is used for recognition decision.

The other is the text-prompted method [1][2]. A new key text is prompted for each recognition event, and the recognizer accepts the input utterance only when it decides that the true speaker correctly uttered the prompted sentence. Speaker-specific phoneme HMMs for each speaker are created during the training phase. During speaker and text verification, a phoneme-concatenated HMM corresponding to the key text is made, and the accumulated likelihood of the input speech frames for the HMM is compared with a threshold to decide whether to accept or reject the speaker.

T. Matsui and S. Furui are with the NTT Human Interface Laboratories. Address: 3-9-11, Midori-cho, Musashino-shi, Tokyo 180, Japan. E-mail: matsui@speech-sun.ntt.jp, furui@speech-sun.ntt.jp.

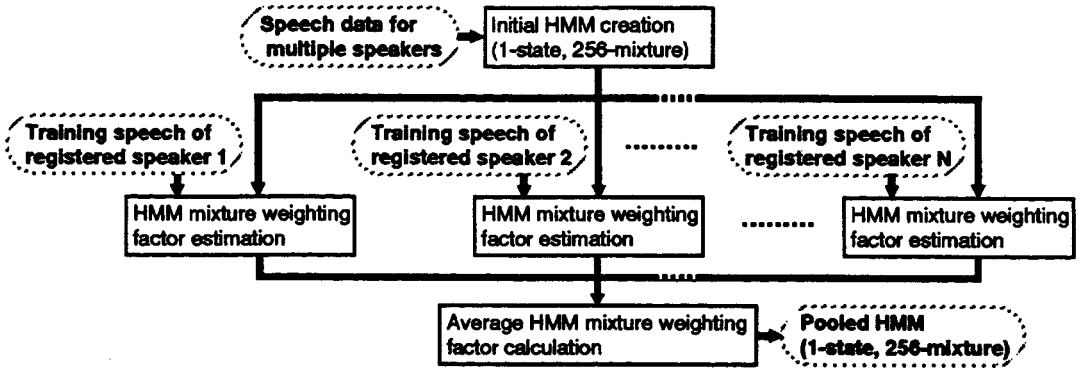

Fig. 1. Block diagram of Method A for making a pooled model.

3. Similarity Normalization Method

3.1 Basics

In text-independent verification, the a posteriori probability used in the normalization method is given by

$$p(s_c|x) = \frac{p(x|s_c) \times p(s_c)}{\sum_i \{p(x|s_i) \times p(s_i)\}} \approx \frac{p(x|s_c)}{\sum_i p(x|s_i)},$$

where s_i is a speaker and s_c is the claimed speaker. The $p(s_i)$ is the probability for speaker i, and is assumed to be a constant for all speakers. The $p(x|s_c)$ is the probability for the claimed speaker's HMM. In [2], $\sum_i p(x|s_i)$ was approximated by the summation of the n highest likelihoods for all registered speakers including the claimed speaker.

In text-prompted verification, the a posteriori probability used in the normalization method is given by

$$p(s_c, t_c|x) = \frac{p(x|s_c, t_c) \times p(s_c, t_c)}{\sum_i \sum_j \{p(x|s_i, t_j) \times p(s_i, t_j)\}} \approx \frac{p(x|s_c, t_c)}{\sum_i \sum_j p(x|s_i, t_j)},$$

where t_j is a text and t_c is the prompted text. The $p(s_i, t_j)$ is the simultaneous probability for speaker i and text j and is assumed to be a constant for all combinations of speakers and texts. The $p(x|s_c, t_c)$ is the probability of the claimed speaker's HMM corresponding to the prompted text. In [1], $\sum_i \sum_j p(x|s_i, t_j)$ was approximated by the summation of the n highest likelihoods by using parallel phoneme HMM networks for all registered speakers including the claimed speaker.

3.2 Pooled model

The amount of calculation for $\sum_i p(x|s_i)$ and $\sum_i \sum_j p(x|s_i, t_j)$ is enormous. To cope with this problem, we propose two new methods that approximate these summations by using the likelihood of a pooled model for all registered speakers and texts. The following sections show two methods of making the pooled model.

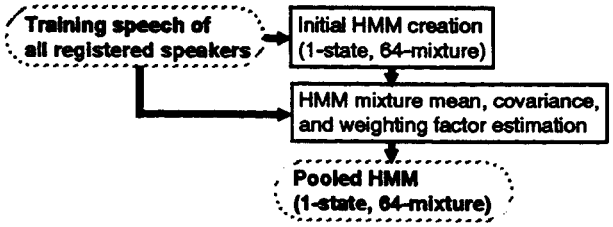

Fig. 2. Block diagram of Method B for making a pooled model.

3.2.1 Method A

Fig. 1 shows a block diagram of Method A. In this method, speech data for multiple speakers who are different from the registered speakers is used for creating an initial model (a 1-state, 256-mixture Gaussian HMM) that universally covers the feature space of speakers. We used 9,000 sentences uttered by 30 male and 30 female speakers as the speech data. For each registered speaker, training speech was applied to the initial model and only the mixture-weighting factors were estimated by using the Baum-Welch algorithm. Then, for each mixture, the weighting factors calculated for all of the registered speakers were averaged to create a pooled HMM (1-state, 256-mixture HMM).

When a new speaker is added as a registered speaker, the pooled HMM is updated by estimating the mixture-weighting factors of the initial model using the training speech of the new speaker and recalculating the average of the weighting factors for all speakers including the new speaker. This updating procedure does not need a lot of computation.

3.2.2 Method B

Fig. 2 shows a block diagram of Method B. The pooled model is a 1-state, 64-mixture Gaussian HMM made using the training speech of all registered speakers by using the Baum-Welch algorithm to estimate the mixture mean, covariance matrices, and weighting factors

Although this method is procedurally simple, the mixture mean, covariance matrices, and weighting factors must be reestimated using the training speech of all speakers every time a new speaker is registered.

TABLE I

VERIFICATION ERROR RATES (%) FOR TEXT-INDEPENDENT AND TEXT-PROMPTED METHODS USING VARIOUS NORMALIZATION TECHNIQUES.

session	text-independent speaker verification error rates (%)				text-prompted speaker&text verification error rates (%)			
	without	Σ	Method A	Method B	without	Σ	Method A	Method B
T1	3.0	0.6	0.8	0.8	1.8	1.8	0.6	0.8
T2	4.1	1.3	2.2	1.7	4.0	1.5	1.2	2.1
T3	4.9	1.9	2.6	2.0	3.7	2.2	1.1	1.2
T4	5.0	2.7	1.7	1.9	2.1	3.0	1.4	1.3
Average	4.3	1.6	1.8	1.6	2.9	2.1	1.1	1.4

4. EXPERIMENTS

4.1 Conditions

The database consists of sentence data uttered by 20 male and 10 female speakers; 10 male and 5 female speakers were used as customers and the remainder were used as impostors. The speech was recorded in five sessions (T0-4) over ten months. The cepstral coefficients were calculated by LPC analysis with an order of 16, a frame period of 8 ms, and a frame length of 32 ms. Ten sentences from session T0 were used for training, and five sentences from sessions T1, T2, T3, or T4 were individually used for testing. In the ten training sentences, the texts of half of them were the same for all customers and all sessions, while the other half differed from customer to customer and from session to session. The sentences for testing were different from those for training and were the same for all customers and impostors and all recording sessions. Six hundred utterances (30 people × 5 sentences × 4 sessions) were used for evaluation. The average duration of each sentence was 4.2 s.

The speaker verification error rate was used for text-independent verification. The threshold was set a posteriori to equalize the probability of false acceptance and false rejection. The continuous HMM (1-state, 64-mixture) was used as the model for each registered speaker.

The speaker and text verification error rate was used for text-prompted verification. The threshold was set a posteriori in the same way as described above. In these experiments, we also used the speech data of texts that differed from the prompted texts but were uttered by the true speaker as data that should be rejected. The tied-mixture HMM (3-state, 256-mixture) was used as the phoneme model for each registered speaker. The number of phonemes was 41.

4.2 Results

TABLE I lists the speaker verification error rates for text-independent recognition, and the speaker and text verification error rates for text-prompted recognition. Here, "without" means a method without normalization and "$\sum$" means the original method that calculates the summation of the likelihoods over all registered speakers. For text-independent verification, the performances of $\sum$ and Methods A and B were almost the same. For text-prompted verification, the error rates for Methods A and B were roughly 35-50% of those for "without," and 50-70% of those for $\sum$. The amount of calculation for normalization for Methods A and B was much smaller than that for $\sum$. These results confirm the effectiveness of Methods A and B.

TABLE II

VERIFICATION ERROR RATES (%) USING COMMON OR SEPARATE POPULATIONS AS CUSTOMERS AND IMPOSTORS.

population	text-independent		
	without	Method A	Method B
separate	4.3	1.8	1.6
common	3.2	1.2	0.6
population	text-prompted		
	without	Method A	Method B
separate	2.9	1.1	1.4
common	2.2	0.9	0.6

TABLE II compares the performances for the two experimental conditions: "separate" means using separate populations for customers and impostors and "common" means using a common population for customers and impostors. The speaker (and text) verification error rates for the separate case were roughly 1.6 times larger than those for the common case.

5. DISCUSSION

5.1 Likelihood ratio vs. a posteriori probability

As mentioned in Section 1, Higgins' method [4] is based on the likelihood ratio, whereas our method [1][2] is based

on a posteriori probability. These methods are slightly different: our method for calculating the summation of the likelihoods includes the likelihood of the claimed speaker, which is excluded in Higgins' method. Here we compare the performances of these two methods under the experimental conditions using separate populations of customers and impostors.

Fig. 3 shows the speaker (and text) verification error rates using Higgins' method and our method. When n for "n highest" is small, Higgins' method performed better than our method. In our method, when n is small and the claimed speaker is true, the likelihood of the claimed speaker tends to be chosen and used for normalization. In this case, the normalized value becomes smaller than with Higgins' method and the input speech may be mistakenly rejected. When n is big, the performances of Higgins' and our methods are almost the same. Therefore, the effectiveness of Methods A and B will not be changed by either including or excluding the claimed speaker since the summation of likelihoods is performed over many speakers.

5.2 Calculation for logarithm of summation

With HMM-based methods, log-likelihoods are usually used in practice. In [1] and [2], the logarithm of the summation of the likelihoods, $\log(\sum p)$, for normalization was approximated by the summation of the logarithms of the likelihoods, $\sum \log(p)$. By using this approximation, low likelihood values are emphasized. Here, we examine the difference in performance caused by this approximation.

Fig. 4 compares performances of $\log(\sum p)$ and $\sum \log(p)$. For $\sum \log(p)$, the error rate was the smallest when the three highest likelihoods were used in text-prompted verification, while the error rate was the smallest when the five highest likelihoods were used in text-independent verification. For $\log(\sum p)$, the error rate was the smallest when the likelihoods for all speakers in both text-prompted and -independent verification were used. The smallest error rates for $\log(\sum p)$ and $\sum \log(p)$ were almost the same, but it may be difficult to set n appropriately for $\sum \log(p)$.

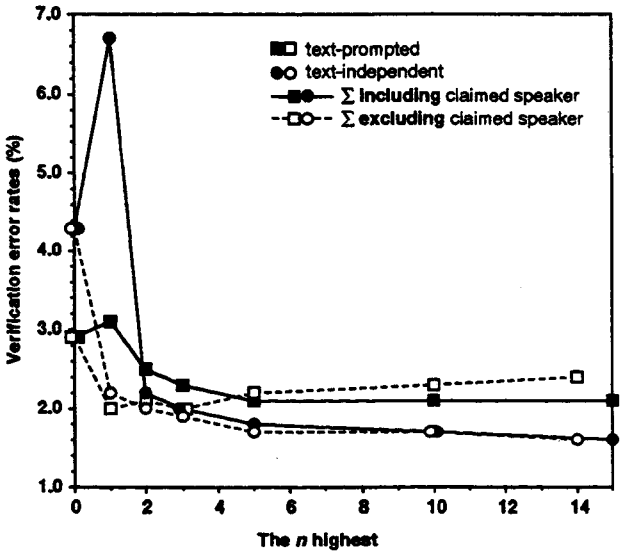

Fig. 3. Likelihood ratio vs. a posteriori probability.

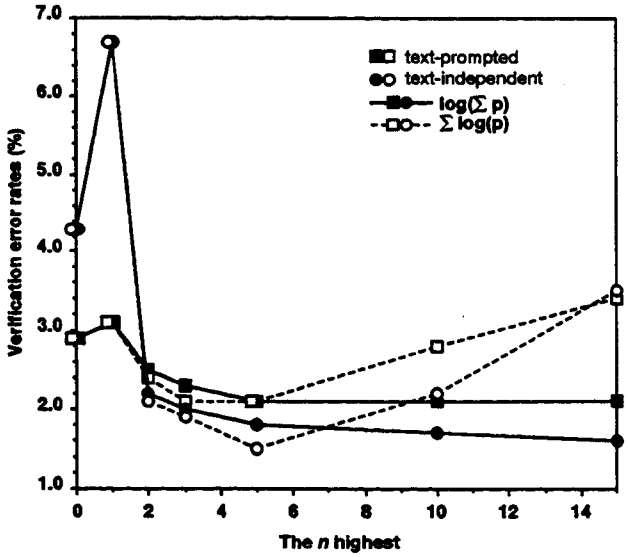

Fig. 4. $\log(\sum p)$ vs. $\sum \log(p)$ in normalization.

6. Conclusions

We investigated two methods for creating a pooled model of all registered speakers; these methods reduce the amount of calculation needed for normalization and perform the same as or better than the original method. To evaluate these methods more practically, speaker verification was performed by using separate populations of customers and impostors. The speaker (and text) verification error rates were roughly 1.6 times bigger than those using the same population for customers and impostors. One proposed method achieves a speaker verification error rate of 1.6% in text-independent verification and a speaker and text verification error rate of 1.1% in text-prompted verification.

Using a normalization method should allow speaker verification thresholds to be set easily. We are investigating methods for setting thresholds beforehand.

References

[1] T. Matsui and S. Furui, *Speaker Adaptation of Tied-Mixture-Based Phoneme Models for Text-Prompted Speaker Recognition*, Proc. ICASSP, Adelaide, 13.1, 1994.

[2] T. Matsui and S. Furui, *Concatenated Phoneme Models for Text-Variable Speaker Recognition*, Proc. ICASSP, Minneapolis, pp. II-391-394, 1993.

[3] T. Matsui and S. Furui, *Comparison of Text-Independent Speaker Recognition Methods Using VQ-Distortion and Discrete/Continuous HMMs*, Proc. ICASSP, San Francisco, pp. II-157-160, 1992.

[4] A. Higgins, L. Bahler, and J. Porter, *Speaker Verification Using Randomized Phrase Prompting*, Digital Signal Processing 1, pp. 89-106, 1991.

[5] A.E. Rosenberg, J. Delong, C.H. Lee, B.H. Juang, and F.K. Soong, *The Use of Cohort Normalized Scores for Speaker Verification*, Proc. ICSLP, Banff, pp. I-599-602, 1992.