A New Score Normalization Method for Speaker Verification with Virtual Impostor Model

Jung Gon Kim, Woo-Yong Choi*, Sung Bum Pan*, Hyung Soon Kim

Department of Electronics Engineering
Pusan National University, Korea
* Information Security Research Division
Electronics and Telecommunications Research Institute, Korea
{gnome,kimhs}@pusan.ac.kr, {wychoi4,sbpan}@etri.re.kr

Abstract
User authentication system using a smart card or authentication token is drawing more and more attention. The limitation of the size of the RAM inside the processor for user authentication necessitates a verification algorithm with the efficient usage of the memory. In this paper, we present a score normalization method, which is suitable for embedded speaker verification system. Proposed score normalization method does not need physical or actual impostor model but utilizes the client model’s mean and covariance vector to imitate an impostor model. Therefore, no additional memory for impostor model is required. Furthermore, most parts of score evaluation process for impostor model are identical with that of client model. As a result, computational burden for virtual cohort model is trivial. We evaluate the performance of our proposed system in seven channel conditions and two channel compensation methods. In our experiment, the performance of our proposed speaker verification system is always superior to un-normalized likelihood based system in various channel condition. And, when comparing with un-normalized likelihood score based speaker verification system, average error rate reduction of the proposed system is higher than 6.7%.

1. Introduction

Many state of the art hidden Markov model (HMM) or Gaussian mixture model (GMM) based speaker verification systems use likelihood ratio score of a spoken password (or pass-phrase) to decide the claimed identity of speaker is correct or incorrect[1][2]. Because un-normalized likelihood score of client model is easily affected by environmental mismatch between model training condition and test condition. It is hard to determine the optimal threshold. However the likelihood ratio scoring method tends to be more robust to environment, because optimal threshold is less variable. If environmental mismatch occurs during the test session, the score of impostor model is also affected by environmental condition. So overall likelihood ratio score is less variable. As a result, determining the optimal threshold is relatively easy.

In likelihood ratio scoring or normalized likelihood scoring method, the score is calculated as the ratio of the client model and impostor model. If \( \lambda_c \) is the client model and \( Y \) is the sequence of acoustic feature vector extracted from spoken password, likelihood ratio is computed as follows:

\[
\text{score}(Y \mid \lambda_c) = \frac{P(Y \mid \lambda_c)}{P(Y \mid \lambda_i)}
\]

where And \( \lambda_i \) is model for impostor, \( P(Y \mid \lambda_c) \) is the likelihood of the feature vector sequence with respect to the model for client and \( P(Y \mid \lambda_i) \) is the likelihood of the feature vector sequence for impostor.

In likelihood ratio scoring based speaker verification system, the verification performance is depends on the quality of impostor model. So, how to make an impostor model is very important part of building a speaker verification system. There are some consideration points to guarantee the quality of an impostor model. The first is the environmental conditions of client model and impostor model. If environmental condition of client and impostor model is different, the verification system is rather sensitive to changes in the environment than a same environmental condition. The second is acoustic resolution of impostor model. Because impostor model must cover overall acoustic space except the neighboring area of client model, impostor model must have sufficient acoustic resolution. To take a proper acoustic resolution, impostor model’s mixture size is generally larger than client[3]. However, unfortunately, it means that a verification system needs more memory and more computing power. So, if we desire to embed the verification system in a smart card or authentication token, these problems must be a seriously considered.

In recent years, various methodologies for impostor modeling were proposed, for example cohort model[1], universal background model[4], concatenating the speaker independent phone model[3]. Because, those models were proposed for text independent or fixed-phrase text dependent speaker verification system, which assumes the phonetic transcription of client password is unimportant or available. It is hard to adapt to a password selectable text dependent verification system, which have no information about the transcription of spoken word.

Our goal in this work is to build a compact and flexible user selectable text dependent speaker verification system, which can be embedded in smart card. To this end, in this paper, we propose a new impostor modeling method, which is suitable to embedded speaker verification system. The proposed method does not need physical or actual impostor.
model but modifies the variance of client model to imitate an impostor model. Therefore, no additional memory for impostor model is required. Furthermore, computational burden for impostor model is ignorable.

In the next section, we explain our new score normalization scheme. In section 3, we will present the experimental results in various mismatched channel conditions. At last, conclusion is presented in section 4.

2. Virtual impostor model

As mentioned in the previous section, a basic requirement of impostor model is sufficient acoustic resolution, in other words, low likelihood score on the neighborhood of the mean of client model and high score on the other acoustic space. As shown we can see in figure 1, if a single Gaussian (or model) is used as a impostor Gaussian (or model), it can not cover all neighboring acoustic space of client. So, to cover the neighboring acoustic space of client, mixture of Gaussian or model should be used. In that case, required memory size for impostor model becomes much larger than that for client model. Furthermore, the computational burden of likelihood evaluation for impostor model is much heavier than that of client model. Considering the limitation of the size of the RAM and computing power of a smart card, we need more compact impostor model.

If we assume that the covariance matrix of client model is diagonal, we can calculate the score of a input vector using the following equation:

$$\log p(y_i | \lambda_i) = \sum_{d=1}^{D} \frac{1}{2} \left[ \left( y_{i,d} - \mu_{\lambda_{i,d}} \right)^2 / \sigma_{\lambda_{i,d}}^2 + \log(2\pi\sigma_{\lambda_{i,d}}^2) \right]$$

(2)

$$= \frac{1}{2} \left[ \sum_{d=1}^{D} \left( y_{i,d} - \mu_{\lambda_{i,d}} \right)^2 / \sigma_{\lambda_{i,d}}^2 + \sum_{d=1}^{D} \log(2\pi\sigma_{\lambda_{i,d}}^2) \right]$$

(3)

where $D$ is a total dimension of input acoustic feature vector. $t$ is time index and $i$ is state index.

In that case, score of impostor model can be calculated as follows:

$$\log p(y_i | \lambda_{\pi}) = \sum_{d=1}^{D} \frac{1}{2} \left[ \left( y_{i,d} - \mu_{\lambda_{\pi,d}} \right)^2 / \alpha \sigma_{\lambda_{\pi,d}}^2 + \log(2\pi\alpha\sigma_{\lambda_{\pi,d}}^2) \right]$$

$$= \frac{1}{2} \left[ \sum_{d=1}^{D} \left( y_{i,d} - \mu_{\lambda_{\pi,d}} \right)^2 / \alpha \sigma_{\lambda_{\pi,d}}^2 + \sum_{d=1}^{D} \log(2\pi\alpha\sigma_{\lambda_{\pi,d}}^2) + D\log \alpha \right]$$

(4)

where $\alpha$ is a constant larger than one. which is to be determined experimentally. Using $\alpha$, we can control the impostor model’s variance.

Because virtual cohort model utilizes the client model’s mean and covariance, as we can see from (3) and (4), most parts of score evaluation process are identical with that of client model. As a result, computational burden virtual cohort model is trivial.

2.2. Further reduction of computational burden using grand variance

In the previous section, we addressed the computational efficiency of virtual cohort model. If we use grand variance[5] and constant transition probability for all states, further computational reduction can be achieved. Because score of virtual cohort model is proportional to client model’s score, if all states of client model share the same variance vector, i.e. grand variance, instead of using different variance vector for each state, resulting optimal path of Viterbi decoding with virtual cohort model is identical to that with client model. Furthermore, because the second term of (3) has no concern with input vector, if we use grand variance, the second term becomes a constant and does not effect on optimal path. So, we can safely neglect the second term of (3) during Viterbi decoding process. In addition, transition probabilities are related with prosody information of client speaker, however, due to the limited size of training data, transition probability is not reliable and we do not use transition probabilities. Using those simplifications, Viterbi decoding score with client model can be evaluated with only the first term of (3).

$$\text{score}(y_i, i) = \sum_{d=1}^{D} (y_{i,d} - \mu_{\lambda_{i,d}})^2 / \sigma_{\lambda_{i,d}}^2$$

(5)

Finally, log likelihood for client model is calculated as follows:

$$\log P(Y | \lambda) = \frac{-1}{2} \left[ \text{decoding\_score} + T \sum_{d=1}^{D} \log(2\pi\sigma_{\lambda_{d}}^2) \right]$$

(6)
where decoding_score is the score through optimal path.

And, because client model and virtual impostor model share the same mean vector and transition probability, optimal path of client model and virtual impostor model is same. Therefore we can reuse the value of decoding_score and log likelihood for virtual cohort model can be easily calculated as follows:

\[
\log P(Y|\lambda_c) = \frac{1}{2} \left( \text{decoding_score}/\alpha + a \sum_{n=1}^{a} (2\pi\sigma^2_n) + D \log \alpha \right)
\]

As a result, by using two simplifications that all states share a grand variance and constant transition probability, client model and virtual cohort model can share the same Viterbi decoding result. Furthermore, most of constant parameters can be pre-calculated in model training procedure. Thus, comparing with un-normalized score based decision, the additional computational burden of score normalization with virtual impostor model is trivial.

Final score normalization in verification stage is as follows:

\[
\text{score} = \frac{\log P(Y|\lambda_c)}{\log P(Y|\lambda_v)}
\]

where \(Y\) is input feature vector, \(\lambda_c\) is client model, and \(\lambda_v\) is virtual impostor model.

3. Experimental results

3.1. Channel simulator

To simulate channel distortion, we had approximated frequency responses of 7 commercial dynamic microphones with polynomial curve fitting. Using those polynomial functions, we made 7 FIR filters, those frequency response are similar to those of 7 dynamic microphones’. Figure 3 shows frequency responses of FIR filters.

3.2. Database

To evaluate the performance of our speaker verification system, we used ETRI speaker recognition database[1]. Total number of speakers in the database is 250, and they can be classified three groups according to the interval of session recording. Among the 250 people, 100 people’s recording interval was a week, another 100 people’s recording interval was a month, and the other 50 people’s recording interval was three months. Each person speaks two-digit strings, four-digit strings and short sentences. Each utterance is repeated five times per a session, and total number of session is four. In our experiment, we used 100 person’s four-digit strings. Recoding interval of those people is a week. The number of four-digit string per a person is 10.

3.3. Baseline system

Our baseline system is HMM based password selectable text dependent speaker verification system. 12th order MFCC feature vector was extracted from the speech signal every 10 msec using a 20 msec window. Whole-word HMM model was used as a client model. Considering the lack of training data and computational efficiency, we used diagonal grand covariance matrix. The number of state is automatically determined using cepstral distance of input feature data. Baseline system employed un-normalized log likelihood score of client model and threshold is determined using average log likelihood score of training data. In the performance evaluation of virtual cohort model, a single global threshold was used. 5 utterances were used to make a client model. Training data was uncorrupted, channel-free data, and to simulate the environmental mismatched condition, test database was filtered with 7 FIR filters.

3.4. Channel compensation methods

As channel compensation methods, we used global cepstral mean subtraction (CMS), and fast pole filtering[6]. CMS is very effective method to remove channel effect. However it also remove speaker information. The goal of pole filtering is preservation of speaker information by broadening the peaks of formants. Equation (9) shows fast pole filtering process.

\[
y^{PF}_d(t) = y_d(t) - m^{PF}_d(t)
\]

where

\[
m^{PF}_d(t) = \frac{1}{T} \sum_{i=1}^{T} y^i_d(t)
\]

and \(y_d(t)\) is d-th order cepstral coefficient of t frame. \(m^{CMS}_d\) is average value of d-th order cepstral coefficient. and \(\gamma \leq 1\).

In case of \(\gamma = 1\), the result is identical with global CMS. Fast pole filtering was originally proposed as a channel compensation for LPCC parameter, however final result in cepstrum domain, i.e. equation (10), can be interpreted as a
low pass filtering in cepstral domain. So, we can also use (10) for MFCC as a channel compensation method.

3.5. Score normalization results with virtual impostor model

As mentioned in the previous section, the variance of virtual impostor model is expanded by $\alpha$. Figure 4 shows the equal error rates (EER) of uncorrupted, channel-free data with CMS and fast pole filtering ($\gamma = 0.95$) according to the value of $\alpha$.

![Figure 4: EER(%) according to the value of alpha](image)

From the figure 4, we can see that virtual impostor model works in wide range of $\alpha$ and the performance of virtual impostor model is robust to the value of $\alpha$. From this result, we use $\alpha = 100$ in the subsequent experiments.

Tables 1 and 2 show EERs of un-normalized score based system and normalized score (with virtual impostor model) based system according to different channel conditions. As a channel compensation method, we applied CMS in table 1, and fast pole filtering in table 2.

| Table 1: EER(%) for different channel conditions (CMS) |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Clean          | Ch1             | Ch2             | Ch3             | Ch4             | Ch5             | Ch6             | Ch7             |
| Unnormalized   | 3.66            | 3.66            | 3.70            | 3.73            | 3.90            | 3.63            | 3.72            | 3.70            |
| Virtual        | 3.37            | 3.42            | 3.40            | 3.46            | 3.69            | 3.46            | 3.46            | 3.43            |

| Table 2: EER(%) for different channel conditions (Fast pole filtering, $\gamma = 0.95$) |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Clean          | Ch1             | Ch2             | Ch3             | Ch4             | Ch5             | Ch6             | Ch7             |
| Unnormalized   | 3.67            | 3.71            | 3.66            | 3.71            | 3.84            | 3.60            | 3.78            | 3.71            |
| Virtual        | 3.37            | 3.37            | 3.40            | 3.59            | 3.38            | 3.47            | 3.33            | 3.34            |

As we can see from tables 1 and 2, the performance of score normalization with virtual impostor model is always superior to un-normalized score based system. In case of CMS, average error reduction rate (ERR) of the proposed method was 6.77%, and in case of fast pole filtering, ERR was 8.14%. Comparing with other score normalization method, performance improvement of our proposed system is relatively small. However, considering that the proposed method does not use any auxiliary database for impostor model and memory usage and computational burden of the proposed method is trivial. These results show the usefulness of our proposed score normalization method.

4. Conclusions

In this paper, we proposed virtual impostor model, which is suitable for embedded user selectable text dependent speaker verification system. Because the proposed impostor model shares the mean and transition probability of client model and variance of impostor model is a proportional to a client model, most part of likelihood evaluation for impostor model is identical with client model. Therefore memory usage and computational burden for score normalization becomes trivial.

5. Acknowledgements

This research was supported by Information Security Research Division, Electronics and Telecommunications Research Institute, Korea.

6. References


