論文 / 著書情報
Article / Book Information

Title	A Stream-Weight and Threshold Estimation Method Using Adaboost for Multi-Stream Speaker Verification
Author	Taichi Asami, Koji Iwano, Sadaoki Furui
Journal/Book name	IEEE ICASSP 2006, Vol. 5, No. , pp. 1081-1084
発行日 / Issue date	2006, 5
権利情報 / Copyright	(c)2006 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

A STREAM-WEIGHT AND THRESHOLD ESTIMATION METHOD USING ADABOOST FOR MULTI-STREAM SPEAKER VERIFICATION

Taichi Asami, Koji Iwano, and Sadaoki Furui

Tokyo Institute of Technology, Department of Computer Science
2-12-1-W8-77 Ookayama, Meguro-ku, Tokyo, 152-8552 Japan
{taichi, iwano, furui}@furui.cs.titech.ac.jp

ABSTRACT

This paper proposes an automatic stream-weight and threshold estimation method for noise-robust speaker verification using multi-stream HMMs integrating segmental and prosodic information. The proposed method simultaneously optimizes stream-weights and a decision threshold by combining the linear discriminant analysis (LDA) and Adaboost techniques. Experiments were conducted using Japanese connected digit speech contaminated by white noise with various SNRs. In this experiment, a target ratio of false acceptance rate (FAR) and false rejection rate (FRR) was set by 1:1 so as to adjust them to approach an equal error rate (EER). Experimental results show that the proposed method effectively estimates stream-weights and thresholds so that FARs and FRRs are adjusted to EERs in most of the SNR conditions.

1. INTRODUCTION

Increasing noise-robustness is one of the key issues for constructing practical speaker verification systems. We have proposed a noise-robust speaker verification method which uses prosodic information in combination with segmental (spectral envelope) information [1]. This method uses a multi-stream HMM technique to integrate fundamental frequency (F_0) features and MFCC features; we call the method "multi-steam speaker verification". We have shown that verification performance can be increased in noisy environments by combining MFCC features with F_0 features extracted by the Hough transform based noise-robust method [2].

In order to construct practical systems, system parameters such as stream-weights of multi-stream HMMs and decision thresholds need to be estimated before verification [3, 4, 5]. As for stream-weights, we have already proposed an optimization method [3] using the linear discriminant analysis (LDA) and Adaboost technique [6]. In this method, the stream-weights are automatically optimized according to the noise conditions of a development set. Experimental results using Japanese connected digit speech contaminated with white noise showed that optimum stream-weights were obtained by the proposed method in various SNR conditions.

In this paper, we propose an automatic threshold estimation method for multi-stream speaker verification based on the same framework as[3]. Since the optimum threshold of the multi-stream speaker verification is variable according to the setting of stream-weights, it is necessary to simultaneously estimate the threshold and the stream-weights. The threshold needs to be determined according to a target balance of false acceptance (FA) and false rejection (FR), which is given by a system developer.

This paper is organized as follows. Section 2 explains a speaker verification method using multi-stream HMMs integrating segmental and prosodic information. In Section 3, our stream-weight and threshold optimization method based on the LDA and the Adaboost is explained. Experimental results are presented in Section 4, and Section 5 concludes this paper.

2. SPEAKER VERIFICATION USING MULTI-STREAM HMMS

Our speaker verification method [1] integrates segmental and prosodic information using multi-stream HMMs. The strategy for integration is explained below.

2.1. Integration of segmental and prosodic features

Each segmental feature vector has 25 elements consisting of 12 MFCC, their deltas and the delta log energy. The window length is 25ms and the frame interval is 10ms. Cepstral mean subtraction (CMS) is applied to each utterance.

The prosodic feature vector consists of $\log F_0$ and $\Delta \log F_0$. F_0 is extracted by a noise-robust method based on the Hough transform [2]. The segmental and prosodic feature vectors are combined at each frame to build a segmental-prosodic feature vector.

2.2. Integration of segmental and prosodic models

The proposed method was evaluated using four-connected-digit speech in Japanese. Since timing of the change of F_0 transitions, such as "rising" and "falling", is highly related to that of CV syllable transitions in Japanese connected digit speech, segmental and prosodic features are integrated in our method using syllabic unit HMMs. The integrated syllable HMM is denoted by "SP-HMM (Segmental-Prosodic HMM)".

In order to make SP-HMMs, S-HMMs (Segmental HMMs) and P-HMMs (Prosodic HMMs) are first trained separately by segmental and prosodic features. Then, the S-HMMs and the P-HMMs are combined to construct SP-HMMs. Gaussian mixtures in the segmental stream of SP-HMMs are tied with corresponding S-HMM mixtures, while the mixtures in the prosodic stream are tied with corresponding P-HMM mixtures.

2.3. Multi-steam modeling

SP-HMMs are modeled as multi-stream HMMs. In recognition, the log-probability $b_j(O_{sp}^t)$ of generating t-th frame segmental-prosodic

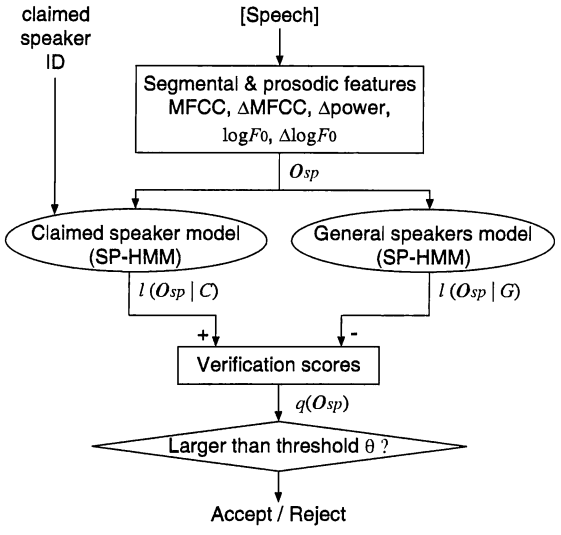

Fig. 1. Flow of speaker verification process.

observation O_{sp}^t at state j is calculated by:

$$b_j(O_{sp}^t) = \lambda_s b_j(O_s^t) + \lambda_p b_j(O_p^t) \quad (1)$$

where $b_j(O_s^t)$ is the log-probability of generating a segmental feature vector O_s^t, and $b_j(O_p^t)$ is the log-probability of generating a prosodic feature vector O_p^t. λ_s and λ_p are weighting factors for the segmental and prosodic streams, respectively. They are constrained by $\lambda_s + \lambda_p = 1$ $(0 \leq \lambda_s, \lambda_p \leq 1)$.

2.4. Verification score

The verification score after observing a feature set O is denoted by $q(O)$, which is calculated as

$$q(O) = l(O|C) - l(O|G) \quad (2)$$

where $l(O|C)$ is a frame-averaged log-likelihood value with claimed speaker's SP-HMM C and $l(O|G)$ is a frame-averaged log-likelihood value with general speaker's SP-HMM G.

The log-likelihood values for segmental-prosodic feature vector O_{sp} are defined using Eq. (1) as follows:

$$l(O_{sp}|C) = \lambda_s l(O_s|C) + \lambda_p l(O_p|C), \quad (3)$$
$$l(O_{sp}|G) = \lambda_s l(O_s|G) + \lambda_p l(O_p|G). \quad (4)$$

Then, the verification score $q(O_{sp})$ is calculated as

$$q(O_{sp}) = \lambda_s q(O_s) + \lambda_p q(O_p). \quad (5)$$

If the score is larger than a threshold value θ, the speaker is accepted as the claimed speaker. Therefore, the discriminant function is $z = q(O_{sp}) - \theta$. If z is positive, the speaker is accepted, and if it is less than or equal to 0, the speaker is rejected as being an imposter. The flow of speaker verification process is shown in Fig. 1.

3. AUTOMATIC STREAM-WEIGHT AND THRESHOLD OPTIMIZATION METHODS

3.1. Estimation by the LDA

As described in section 2.4, speaker verification by SP-HMM uses the following discriminant function z:

$$z = q(O_{sp}) - \theta \quad (6)$$
$$= \lambda_s q(O_s) + \lambda_p q(O_p) - \theta. \quad (7)$$

Since z is a linear function, the stream-weights and the threshold are estimated as coefficients of a linear function obtained by the LDA. The Estimation process is as follows. First, segmental and prosodic scores, $q(O_s)$ and $q(O_p)$, calculated from both a claimed speaker's and an impostor's data included in the development set are plotted in two-dimensional space composed by $q(O_s)$ and $q(O_p)$. Then, the LDA is applied to the space so as to obtain the discriminant function z which distinguishes score distribution of claimed speakers from that of impostors. Since the obtained function $z = a_s q(O_s) + a_p q(O_p) - b$ does not satisfy $a_s + a_p = 1$, it is transformed so that the sum of the coefficients becomes 1. The estimated values of the stream-weights and the threshold are

$$\hat{\lambda}_s = \frac{a_s}{a_s + a_p}, \quad \hat{\lambda}_p = \frac{a_p}{a_s + a_p}, \quad \hat{\theta} = \frac{b}{a_s + a_p}. \quad (8)$$

Thus, all the parameters are estimated according to the LDA criterion which maximizes discriminant performance between claimed speakers and impostors.

3.2. Optimization by the Adaboost

The Adaboost, a class of boosting algorithms, constructs a high performance classifier by combining sequentially trained simple classifiers [6]. In our optimization method, the linear discriminant functions obtained by the LDA are used as simple classifiers for the Adaboost. By doing so, we can estimate more accurate weights and thresholds than those obtained by only using the LDA. The stream-weights and the threshold are optimized according to a target ratio of false acceptance rate (FAR) and false rejection rate (FRR), $FAR : FRR = \alpha : (1 - \alpha)$, given by the system developer.

Details of the optimization algorithm is as follows, where n represents the number of data in the development set and T represents the number of iterations. Let $\{x_i\}$ $(i = 1, \cdots, n)$ be the development data plotted into the two-dimensional space composed by $q_s(O_s)$ and $q_p(O_p)$, and $\{w_i\}$ $(i = 1, \cdots, n)$ be the weights of each data.

1. Initialize the weights of data $w_i = 1/n$.
2. Iterate the following processes for $t = 1, \cdots, T$.
   (a) Choose n samples from $\{x_i\}$ allowing duplications, using $\{w_i\}$ as a probability distribution.
   (b) Obtain a linear discriminant function
   $$z_t = \lambda_s^{(t)} q_s(O_s) + \lambda_p^{(t)} q_p(O_p) - \theta^{(t)} + \delta_{t-1} \quad (9)$$
   by applying the LDA to the resampled data $\{x_i'\}$, where δ_{t-1} is an offset value for adjusting the balance between FAR and FRR to the target ratio $\alpha : (1 - \alpha)$. The offset value is obtained from the $\{t-1\}$-th boosting iteration, where the initial value δ_0 is set at 0.

(c) Classify all the data in the development set $\{x_i\}$ using z_t, and calculate FAR, FRR, and the weighted discriminant errors ϵ_t, ϵ_{FA}, and ϵ_{FR}:

$$FAR_t = \frac{\sum_{i:\ x_i \text{ is FA}} 1}{\sum_{i:\ x_i \text{ is an imposter}} 1}, \quad (10)$$

$$FRR_t = \frac{\sum_{i:\ x_i \text{ is FR}} 1}{\sum_{i:\ x_i \text{ is a claimed speaker}} 1}, \quad (11)$$

$$\epsilon_t = \sum_{i:\ \text{misclassify } x_i} w_i, \quad (12)$$

$$\epsilon_{FA} = \frac{\sum_{i:\ x_i \text{ is FA}} w_i}{\sum_{i:\ x_i \text{ is an imposter}} w_i}, \quad (13)$$

$$\epsilon_{FR} = \frac{\sum_{i:\ x_i \text{ is FR}} w_i}{\sum_{i:\ x_i \text{ is a claimed speaker}} w_i}. \quad (14)$$

The offset value δ_t is determined by the following equation:

$$\delta_t = \beta \cdot \frac{1 - FAR_t/FRR_t}{1 + FAR_t/FRR_t}. \quad (15)$$

(d) Calculate c_t as the weight of z_t by the following equations:

$$c_t = c_{\epsilon_t} \cdot c_{d_t}, \quad (16)$$

$$c_{\epsilon_t} = \frac{1}{2}\log\frac{1-\epsilon_t}{\epsilon_t}, \quad (17)$$

$$c_{d_t} = \frac{1}{2}\log\frac{1-d_t}{d_t}, \quad (18)$$

$$d_t = |(1-\alpha)\cdot FAR_t - \alpha \cdot FRR_t|. \quad (19)$$

(e) Update w_i by the following formula:

$$w_i = \begin{cases} w_i \times e^{-c_{cost_t}} & (i\text{: classify } x_i \text{ accurately}) \\ w_i \times e^{c_{cost_t}} & (i\text{: misclassify } x_i) \end{cases}$$

where c_{cost_t} is defined by using a cost function $cost_t$ as follows:

$$c_{cost_t} = \frac{1}{2}\log\frac{1-cost_t}{cost_t}, \quad (20)$$

$$cost_t = (1-\alpha)\cdot\epsilon_{FA} + \alpha\cdot\epsilon_{FR}. \quad (21)$$

(f) Normalize $\{w_i\}$ to meet $\sum_{i=1}^{n} w_i = 1$.

3. Let the conclusive classifier z be the weighted majority vote of z_t:

$$z = \sum_{t=1}^{T}(c_t z_t). \quad (22)$$

4. Normalize the coefficients of z so that the sum of them becomes 1.

5. Set the normalized coefficients as estimated stream-weights and the threshold.

In the original Adaboost algorithm, the conclusive classifier z is defined by $z = \sum_{t=1}^{T}\{c_t \times sign(z_t)\}$. However, it cannot be directly used for stream-weight estimation, since its form is not a linear discriminant function. Thus, we approximate z by $z = \sum_{t=1}^{T}(c_t z_t)$ as shown in Eq.(22).

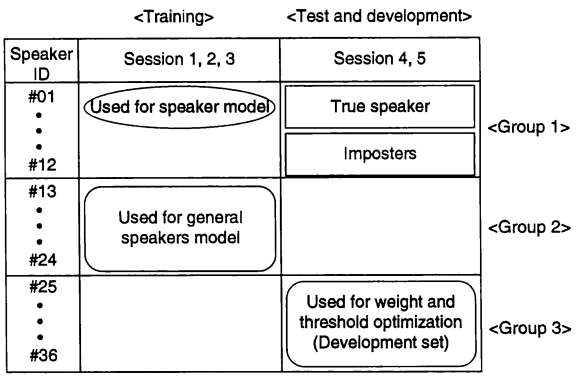

Fig. 2. Training, testing and development sets for the verification experiment when the speaker M01 is used as the claimed speaker.

4. EXPERIMENTS

4.1. Database

Speech data were recorded at five sessions separated by intervals of approximately one month. The data were collected from 36 male speakers and sampled at 16kHz with a 16bit resolution. Each speaker uttered 50 strings of four connected digits in Japanese at each session.

The set of data recorded at sessions $1 \sim 3$ were used for training and data recorded at sessions 4 and 5 were used for parameter optimization and testing. The database was separated into three groups in terms of speakers as shown in Fig. 2. Figure 2 shows the case where speaker M01 was used as the claimed speaker. The general speaker's model was trained using utterances by all the speakers in the speaker group 2, which did not include the claimed speaker nor the set of data used for optimizing the stream-weights and the threshold. In the case where speaker #01 was used as the claimed speaker, an additional experiment was conducted, in which the general speaker's model was trained by the data of speaker group 3 and the weights and the threshold were optimized by the data of speaker group 2. There are six combinations of the training set, the development set and the testing set. The result averaged over the six experiments was used for evaluation.

White noise was added to the training set at a 30dB SNR level to increase robustness against noisy speech, and the development and testing sets were contaminated with white noise at 5, 10, 15, 20 and 30dB SNR conditions.

The experiments were conducted at the condition of $\alpha = 0.5$, that is, adjusting FAR and FRR to an equal error rate (EER). The number of iterations and the parameter β to determine the offset value δ_t were experimentally set to 200 and 0.005, respectively.

4.2. Experimental results

Table 1 shows the FARs and FRRs obtained by using the proposed optimization methods in each SNR condition. The results in "LDA only" indicate the results using the stream-weights and the threshold estimated by only the LDA; and, the results in "Adaboost" are obtained by the proposed optimization method using the boosting technique. The bottom line shows the EERs in which the stream-weights

Table 1. Comparison of verification results with different stream-weights and threshold optimization methods in various SNR conditions.

Optimization method		SNR (dB)				
		30	20	15	10	5
LDA only	FAR (%)	1.22	4.82	8.15	9.55	10.9
	FRR (%)	0.51	2.68	11.4	27.6	50.9
Adaboost	FAR (%)	1.47	3.72	9.35	17.2	26.4
	FRR (%)	0.68	3.43	8.66	15.6	23.2
Manual optimization	ERR (%)	0.73	3.39	9.21	16.3	24.3

and the threshold were manually optimized using the development set.

It has been confirmed that the Adaboost-based optimization method more effectively adjusts the FARs and FRRs to EERs than the method using only the LDA at all conditions except the 30dB SNR condition. Our preliminary closed-condition experiments using the test set for optimizing stream-weights and threshold show that the Adaboost-based method works well even at the 30dB SNR condition. This means that degradation of the optimization performance at the 30dB SNR condition is due to the mismatch of data distributions between the development set and the test set.

5. CONCLUSIONS

This paper proposed an automatic stream-weight and threshold estimation method using LDA and Adaboost for multi-stream speaker verification. Experimental results using Japanese connected digit speech contaminated with white noise show that the proposed method effectively estimates stream-weights and thresholds so that the FARs and FRRs are adjusted to reach EERs in most of the SNR conditions. The proposed method is, in principle, applicable to any FAR-FRR ratio.

Our future works include: 1) evaluating optimization performance of the proposed method at various α (FAR-FRR ratio) conditions, 2) evaluating performance when applying the proposed optimization method to other multi-stream speaker verification systems using a larger number of streams, and 3) investigating a parameter optimization method using a test set without having labelled speaker IDs, instead of using a development set.

6. ACKNOWLEDGEMENTS

This research is partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Young Scientists (B), 17700141, 2005.

7. REFERENCES

[1] K. Iwano, T. Asami, and S. Furui, "Noise-robust speaker verification using F_0 features," *Proc. ICSLP 2004*, Jeju Island, Korea, vol.2, pp.1417–1420 (2004-10).

[2] K. Iwano, T. Seki, and S. Furui, "Noise robust speech recognition using F_0 contour information," IEICE Transactions on Information and Systems, vol.E87-D, no.5, pp.1102–1109 (2004-5).

[3] T. Asami, K. Iwano, and S. Furui, "Stream-weight optimization by LDA and Adaboost for multi-stream speaker verification," *Proc. Interspeech 2005*, Lisbon, Portugal, pp.2185–2188 (2005-9).

[4] T. Matsui, T. Nishitani, and S. Furui, "Robust methods of updating model and a priori threshold in speaker verification," *Proc. ICASSP 1996*, Atlanta, GA, USA, vol.1, pp.97–100 (1996-5).

[5] J. Lindberg, J. Koolwaaij, H.-P. Hutter, D. Genoud, J.-B. Pierrot, M. Blomberg, and F. Bimbot, "Techniques for a priori decision threshold estimation in speaker verification," *Proc. RLA2C*, Avignon, France, pp.89–92 (1998-4).

[6] Y. Freund and R.E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Science*, vol.55, no.1, pp.119-139 (1997-8).