BOOSTED BINARY FEATURES FOR NOISE-ROBUST SPEAKER VERIFICATION

Anindya Roy[1,2], *Mathew Magimai.-Doss*[1], *Sébastien Marcel*[1]

[1]Idiap Research Institute, Martigny, Switzerland
[2]École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland

ABSTRACT

The standard approach to speaker verification is to extract cepstral features from the speech spectrum and model them by generative or discriminative techniques. We propose a novel approach where a set of client-specific binary features carrying maximal discriminative information specific to the individual client are estimated from an ensemble of pair-wise comparisons of frequency components in magnitude spectra, using Adaboost algorithm. The final classifier is a simple linear combination of these selected features. Experiments on the XM2VTS database strictly according to a standard evaluation protocol have shown that although the proposed framework yields comparatively lower performance on clean speech, it significantly outperforms the state-of-the-art MFCC-GMM system in mismatched conditions with training on clean speech and testing on speech corrupted by four types of additive noise from the standard Noisex-92 database.

Index Terms— Speaker verification, binary features, speaker-specific features, noise robustness, Adaboost

1. INTRODUCTION

The standard approach to speaker verification is to parameterize the short-term magnitude spectra extracted from speech frames typically by cepstral coefficients [1] and model these parameters using standard techniques like Gaussian Mixture Models (GMM) [1]. In this work, we propose a novel approach that aims to extract speaker specific information directly from the magnitude spectrum. In this approach, a small set of binary features, typically numbering 20 to 30, are iteratively selected from a very large set of features according to their discriminative ability on the training data. These features are data-driven and optimized for each individual client. The final classifier is a weighted linear combination of single stump classifiers using the selected features.

The motivation for the proposed binary features is the recent success of binary-valued features based on pixel comparison like Local Binary Patterns (LBP), Modified Census Transform and Haar features [2] in the vision research community particularly for fast object detection. These features are robust to illumination variations since their value depends only on the comparison of two pixel values, not on the pixel values themselves. In this work, we mapped this approach to extract features for speaker verification, using the 1-D spectral vectors as object instances to be classified as either belonging to the client or impostor classes, analogous to face *vs* non-face classification problem in vision. These binary features are discriminatively selected for each client individually using Adaboost [3], a standard ensemble learning technique. While testing, the model can be evaluated and a decision can be taken relatively fast since the classifier is a simple weighted linear combination of binary outputs, each depending on a comparison operation on two frequency components of the spectrum. Experiments show that the intrinsic illumination-robustness of such features in the vision domain possibly leads to their robustness against several additive noise types in the speech domain. We have compared the proposed framwork with the standard Mel Frequency Cepstral Coeffecient (MFCC)-GMM framework [1].

The rest of the paper is organized as follows. In Sec.2, we describe the proposed speaker verification framework. We describe our experiments in Sec.3. In Sec.4, we discuss the results and highlight certain aspects of our method. Finally, Sec.5 outlines the main conclusions of our work.

2. THE PROPOSED FRAMEWORK

2.1. Binary Features

In the first step, the input speech waveform is blocked into frames and a spectral transform T is applied to it to yield a sequence of spectral magnitude vectors. Let $\overrightarrow{\mathbf{X}} = [X(1), \cdots, X(N)]^T$ be an instance of such a vector. The spectral transform T can be either 1) a simple N_0-point Discrete Fourier Transform (DFT) (In this case, $\overrightarrow{\mathbf{X}}$ comprises of one half of the magnitude spectrum components since they are symmetric, and $N = \frac{N_0}{2} + 1$.) or 2) DFT followed by Mel filtering [1] (In this case, $\overrightarrow{\mathbf{X}}$ represents the Mel filter outputs and N = number of filters). The proposed binary features are calculated on the vector $\overrightarrow{\mathbf{X}}$ as follows. A binary feature $\phi_i : \Re^N \rightarrow \{0, 1\}$ is defined completely by the following 3 parameters: two indices $k_{i,1}$,$k_{i,2}$ which can vary from 1 to N but cannot be equal and one threshold parameter, θ_i, selected according to a certain criterion (ref. Sec. 2.2). For the DFT case, the $\{k_{i,j}\}$ represent frequency indices. For the Mel filter case, they represent indices of Mel filters. The feature ϕ_i is defined as,

$$\phi_i(\overrightarrow{\mathbf{X}}) = \begin{cases} 1 & \text{if } X(k_{i,1}) - X(k_{i,2}) \geq \theta_i, \\ 0 & \text{if } X(k_{i,1}) - X(k_{i,2}) < \theta_i. \end{cases} \tag{1}$$

From the range of the k_i values, the total number of such binary features is $N(N-1)$. Let $\Phi = \{\phi_i\}_{i=1}^{N(N-1)}$ represent the complete set of such features.

2.2. Feature selection

Out of the complete set of binary features Φ, a certain number of features are iteratively selected *for each client* according to their discriminative ability with respect to that client. This selection is based on the Discrete Adaboost algorithm [3] with weighted sampling,

This work was supported by the Swiss National Science Foundation (projects 200020-122062 and 51NF40-111401) and the FP7 European MOBIO project (IST-214324). The authors would like also to thank Dr. Nelson Morgan and Dr.Francesco Orabona for their comments and advice.

which is widely used for such binary feature selection tasks [2] and is known for its robust performance [3]. The algorithm, which is to be run once for each client, is as follows:

Algorithm: Feature selection by Discrete Adaboost

```
Inputs: $N_{tr}$ training vectors $\{\overrightarrow{\mathbf{X}}_j\}_{j=1}^{N_{tr}}$, the corresponding class labels, $y_j \in \{0,1\}$ (0:impostor, 1:client), $N_f$, the number of features to be selected, $N_{tr}^*$, the number of training vectors to be randomly sampled at each iteration ($N_{tr}^* < N_{tr}$).

  • Initialize the weights $\{w_{1,j}\} \leftarrow \frac{1}{2N_{tr}^{(0)}}, \frac{1}{2N_{tr}^{(1)}}$ for $y_j = 0, 1$ respectively, where $N_{tr}^{(0)}$ and $N_{tr}^{(1)}$ are the number of impostor and client training vectors respectively.

  • Repeat for $n = 1, 2, \cdots N_f$:

      – Normalize weights, $w_{n,j} \leftarrow \frac{w_{n,j}}{\sum_{j'=1}^{N_{tr}} w_{n,j'}}$
      – Randomly sample $N_{tr}^*$ training vectors, according to the distribution $\{w_{n,j}\}$
      – For each $\phi_i$ in $\Phi$, choose $\theta_i$ to minimize misclassification error, $\epsilon_i = \frac{1}{N_{tr}^*}\sum_{j=1}^{N_{tr}^*} \mathbf{1}_{\{\phi_i(\overrightarrow{\mathbf{X}}_j) \neq y_j\}}$ over the sampled set.
      – Select the next best feature, $\phi_n^* = \phi_i^*$ where $i^* = \arg\min_i \epsilon_i$
      – Set $\beta_n \leftarrow \frac{\epsilon_{i^*}}{1-\epsilon_{i^*}}$
      – Update the weights, $w_{n+1,j} \leftarrow w_{n,j}\beta_n^{\mathbf{1}_{\{\phi_n^*(\overrightarrow{\mathbf{X}}_j)=y_j\}}}$

Output: The sequence of selected best features $\{\phi_n^*\}_{n=1}^{N_f}$.
```

For the database and framing parameters used (ref. Sec.3), N_{tr} was around 80,000, and $N_{tr}^{(1)}$, which varies for each client, was around 350. N_{tr}^* was set to 4000 and N_f to 30. Figure 1 shows the distribution of the selected binary features $\{\phi_n^*\}_{n=1}^{N_f}$ for the DFT case, in terms of their frequency indices $(k_{n,1}, k_{n,2})$ and the equivalent value in Hz (at $f_s = 8$kHz). It is observed that the client-specific features are spread relatively uniformly throughout the spectrum, with slightly higher concentration below 1kHz and above 2.5kHz.

2.3. Feature Modelling and Classifier structure

For each client, the selected features are combined linearly to give a strong classifier F [3]:

$$F(\overrightarrow{\mathbf{X}}) = \sum_{n=1}^{N_f} \alpha_n \phi_j(\overrightarrow{\mathbf{X}}). \tag{2}$$

The weights $\{\alpha_n\}$ are calculated to minimize the exponential loss [3] and normalized to sum to unity for each client, $\alpha_n = \frac{log(\beta_n)}{\sum_{n'=1}^{N_f} log(\beta_{n'})}$. Since a decision is only required at the utterance level and not at the frame level, the responses $F(\overrightarrow{\mathbf{X}})$ of each frame $\overrightarrow{\mathbf{X}}$ in an utterance are added and normalized by the number of frames, to obtain the final score S for the utterance. This is compared with a preset threshold to decide if the utterance was made by a client or an impostor. This preset threshold Θ is calculated by minimizing the Equal Error Rate [1] on a separate Development set. (ref. Sec.3.)

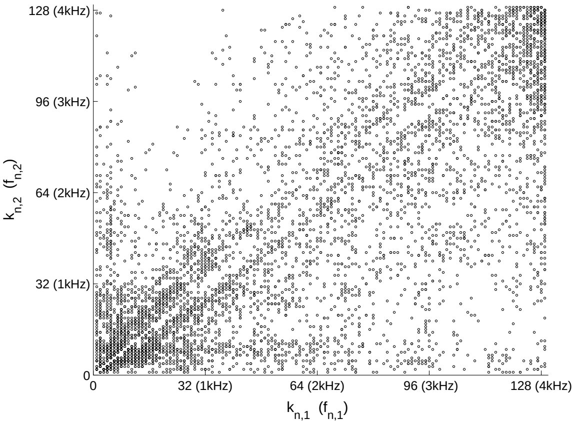

Fig. 1. Distribution of the selected binary features $\{\phi_n^*\}_{n=1}^{N_f}$ for all clients in the database, in terms of their frequency indices $(k_{n,1}, k_{n,2})$ and the equivalent value in Hz (at $f_s = 8$kHz).

3. SPEAKER VERIFICATION EXPERIMENTS

3.1. Description of the database used

All experiments are performed on the standard XM2VTS audio database [4], [5] having 200 clients and 95 impostors. Utterances of around 5 *sec* duration are recorded across 4 sessions, 2 utterances per session. Sampling frequency f_s = 8 kHz. Speech is relatively clean (SNR≥30dB), there is a certain amount of session variability between the 4 sessions. For all experiments under the *mismatched* condition (Sec.3.3), the noisy speech utterances were obtained by adding randomly selected segments from the standard Noisex-92 database [6] to the original speech from the XM2VTS database, at 7 different SNR levels. Four noise types were used, white, pink, babble and factory noise [6].

3.2. Description of the systems tested

In the proposed framework, the following 5 systems were tested. The primary system *BBF* uses a frame length of 20ms and 50% overlap, a silence removal step based on frame energies, retaining 20% of the higher energy frames during training and 10% while testing, a 256-point DFT and a spectral subtraction step which subtracts the mean of the 15% lowest energy frames from all the retained frames. The binary features are calculated directly from Fourier spectra. Since the spectrum is symmetric, half of it is discarded, giving N =129 frequency points and a total of 16512 binary features. Out of this, the number of selected features N_f is 30. A variant of this system *BBFa* is exactly the same but without the spectral subtraction step. Variant *BBFq* uses only a quarter of the full Fourier spectrum, i.e, till 1 kHz, instead of the full 4 kHz, motivated by the concentration of the selected features (using the full spectrum) below 1kHz (ref. fig. 1). The other variants *BBFmxx* use Mel spectra instead of Fourier spectra, i.e. the spectral vectors $\overrightarrow{\mathbf{X}}$ represent Mel filter outputs. We report using 24 and 40 filters (*BBFm24*, *BBFm40* respectively).

For comparison, following 3 reference systems were tested. 1) *MC33*: A state-of-the-art system using 33 features [1] (16 MFCC (from 24 filters), 16 Δ-MFCC and Δ-energy), silence removal by bi-Gaussian modelling [1] and Cepstral Mean Substraction(CMS) [1]. Frame length and overlap are same as in *BBF*. Modelling is by 32

Systems tested		Dev. set (EER%)	Test set *a priori* thr.	Test set *a post.* thr.
Reference systems	MC33	1.8	1.4	1.5
	MC16	1.7	3.4	2.8
	MS24	6.5	5.9	5.8
Proposed systems	**BBF**	**4.3**	**9.1**	8.2
	BBFa	4.7	10.8	9.2
	BBFq	8.5	11.4	11.5
	BBFm24	5.5	9.8	9.3
	BBFm40	**5.0**	**8.6**	8.3

Table 1. Verification performance (HTER %) under matched condition.

Gaussian UBM-GMM system [1]. 2) *MC16*: It uses 16 MFCC features modelled by 32 Gaussian UBM-GMM, silence removal based on frame energies as in *BBF*, and no CMS. This second system using only static features was motivated by the fact that the proposed binary features exploit information from a single frame only. 3) *MS24*: It uses log spectra from a Mel filterbank with 24 filters to model a 32 Gaussian UBM-GMM system. It uses the same spectral substraction setup as *BBF*. This system was included in order to find whether the noise-robustness of the proposed framework is due to use of spectra instead of cepstra or is it an intrinsic property of the binary features themselves, because spectral features have been generally observed to be more robust than cepstral features in noisy conditions, for speech applications.

3.3. Experimental conditions

Two different conditions were tested. 1) Matched-clean condition: The standard Lausanne Protocol variant 1 [5] associated with the XM2VTS database was followed. According to this protocol, first utterance from sessions 1, 2 and 3 (Training set) are used for training. For training a client model, the remaining speakers in the client set are treated as impostors. Second utterance from same 3 sessions (Development set) are used to set the threshold Θ at Equal Error Rate (EER) [1]. It is a global threshold. For testing, the 2 utterances from the remaining session 4 and a dedicated impostor set different from all clients are used (Test set). Performance is reported in terms of the Half Total Error Rate (HTER) = $\frac{1}{2}$(False Acceptance Rate(FAR) + False Rejection Rate(FAR)) [1] on the Test set, using the *a priori* threshold Θ. 2) Mismatched-noisy condition: The same protocol was followed. Training and development (setting the threshold) was done on original clean speech but the testing was carried out on noisy speech [6] (ref. Sec.3.1).

3.4. Results

The verification performance (HTER%) under matched condition is shown in Table 1. We also report EER% on the Development set, and HTER % on the Test set with the threshold set *a posteriori* on the Test set. The mismatched condition is reported in Figs. 2, 3, 4 and 5 for white, pink, babble and factory noise types respectively, showing HTER % against SNR of the test speech. Results are discussed in Sec.4.

4. DISCUSSIONS

In matched-clean condition, the proposed framework is outperformed by the reference systems. A major reason can be due to channel variability between sessions 1,2,3 (used for training) and session 4 (used for testing). A slightly different protocol which takes into account this variability (selective training using all sessions) lowered the test HTER for *BBF* from 9.1% to 5.4%.

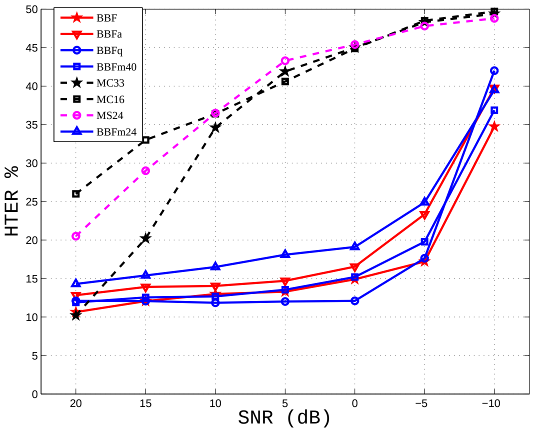

Fig. 2. Verification performance (HTER%) *vs.* SNR, mismatched condition: test speech corrupted additively by white noise.

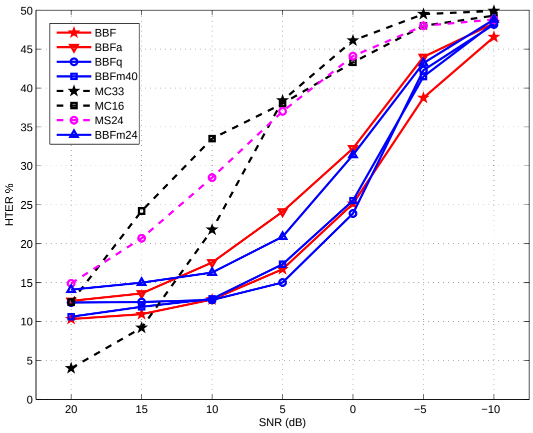

Fig. 3. Verification performance (HTER%) *vs.* SNR, mismatched condition: test speech corrupted additively by pink noise.

In mismatched-noisy condition, the proposed framework outperforms the reference systems significantly for medium to high levels of noise. In white noise case, improvement is visible from SNR=15dB. For other types, it is visible from SNR around 10dB. Please note that system *BBFa* is to be compared with *MC16* and not with *MC33* because it uses a similar restricted framework. It is noteworthy that *BBFq* compares reasonably well with other proposed systems even by using only a quarter of the spectrum. Further, the proposed framework performs significantly better than reference system *MS24* indicating that the noise-robustness of the proposed framework is more due to the intrinsic robustness of the binary features. A brief feature level analysis of the robustness of the proposed features against the four noise types is shown in Fig.6 where the variation in probability of the first selected feature value, $P(\phi_1^*(\overrightarrow{\mathbf{X}}) = 1)$ for a client from the database is plotted against noise level, for both the client and all impostors. The separation

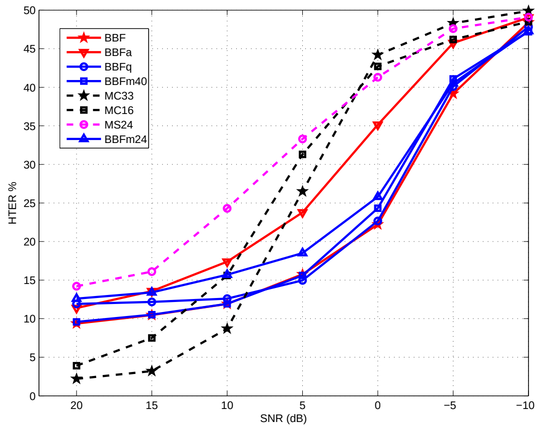

Fig. 4. Verification performance (HTER%) *vs.* SNR, mismatched condition: test speech corrupted additively by babble noise.

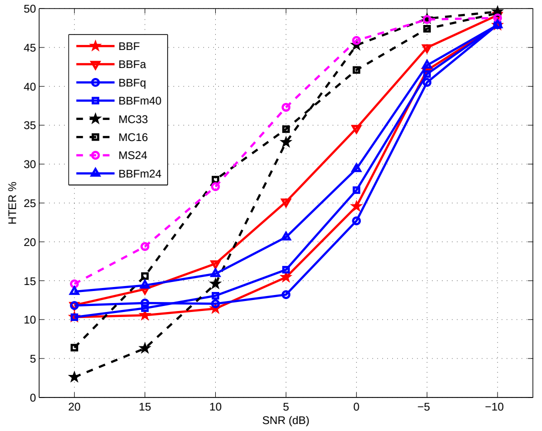

Fig. 5. Verification performance (HTER%) *vs.* SNR, mismatched condition: test speech corrupted additively by factory noise.

between client and impostor probabilities remain relatively stable over a wide SNR range, which can possibly lead to stable scores over the same range (ref. Eqn.2).

The proposed framework leads to significant reduction in computation time compared to the reference MFCC-GMM systems. While testing the client model, it involves only $N_f = 30$ comparison and addition operations per frame, which can even be hard-coded because the summation is over preset weights $\{\alpha_n\}$. In contrast, *MC33* requires 33 × 32 subtractions, 33 × 32 multiplications and 32 exponentiations. This makes the proposed system more practical for real-time operations. Another interesting aspect of the proposed framework is that the client models do not directly store spectral shape information. They only store discriminative frequency points $(k_{n,1}, k_{n,2})$ and thresholds. Thus, the proposed models may be more robust against efforts to reconstruct a synthetic voice model from stolen model parameters than an equivalent MFCC-GMM model, although such a claim remains to be validated.

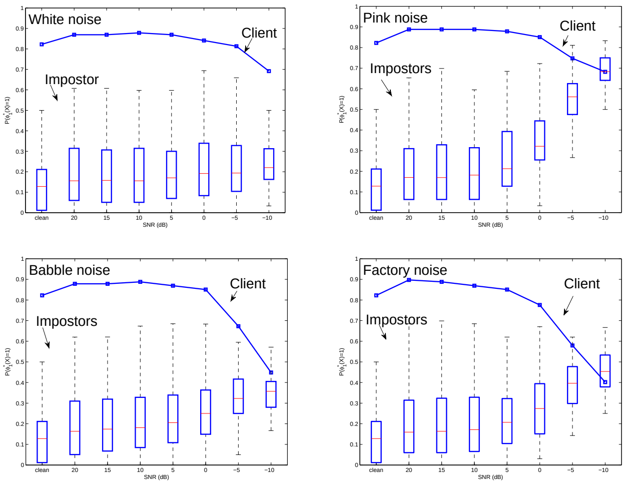

Fig. 6. Effect of 4 noise types on the proposed features, in terms of $P(\phi_1^*(X) = 1)$. The blue lines represent data from a particular client, the boxplots represent data over all impostors.

5. CONCLUSIONS

We propose a new set of binary features for speaker verification based on comparison of points in magnitude spectra. The features are selected individually for each client using Adaboost, are simple and relatively fast to calculate and show robustness against several additive noise types in mismatched conditions. As part of future work, the feature set could be augmented by joint modelling in the spectro-temporal plane. The features could be generalized to more than 2 frequency points to capture more speaker-specific information. Fusions between different proposed systems and between the proposed systems and the MFCC-GMM system could result in improved performance in both clean and noisy conditions.

6. REFERENCES

[1] F. Bimbot et al., “A Tutorial on Text-Independent Speaker Verification,” *EURASIP Journal on Applied Signal Processing*, , no. 4, pp. 431–451, 2004.

[2] Y. Rodriguez, “Face Detection and Verification using Local Binary Patterns,” PhD Thesis 3681, Ecole Polytechnique Federale de Lausanne, 2006.

[3] J. Friedman, T. Hastie, and R. Tibshirani, “Additive Logistic Regression: a Statistical View of Boosting,” *Annals of Statistics*, vol. 28, pp. 2000, 1998.

[4] K. Brady, M. Brandstein, T. Quatieri, and Dunn R., “An Evaluation of Audio-Visual Person Recognition on the XM2VTS corpus using the Lausanne Protocols,” in *ICASSP*, 2007.

[5] J. Luettin and G. Maitre, “Evaluation protocol for the extended M2VTS database (XM2VTSDB),” Idiap Comm. 98-05, Idiap, 2000.

[6] A.P. Varga, H.J.M Steeneken, M. Tomlinson, and D. Jones, “The NOISEX-92 study on the effect of additive noise on automatic speech recognition,” Technical report, DRA Speech Research Unit, 1992.