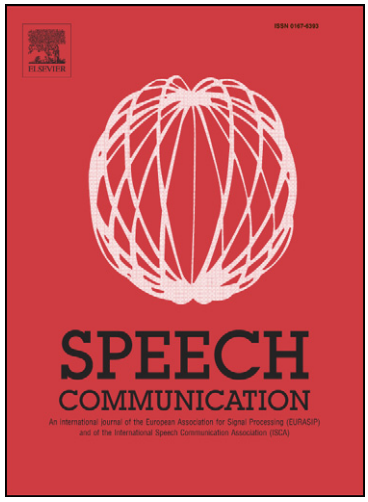
Optimizing feature complementarity by evolution strategy: Application to automatic speaker verification

C. Charbuillet, B. Gas, M. Chetouani, J.L. Zarader



Optimizing feature complementarity by evolution strategy: Application to automatic speaker verification

C. Charbuillet, B. Gas, M. Chetouani, J.L. Zarader

PII: S0167-6393(09)00007-7
DOI: 10.1016/j.specom.2009.01.005
Reference: SPECOM 1776

To appear in: *Speech Communication*

Received Date: 2 December 2007
Revised Date: 14 January 2009
Accepted Date: 21 January 2009

Please cite this article as: Charbuillet, C., Gas, B., Chetouani, M., Zarader, J.L., Optimizing feature complementarity by evolution strategy: Application to automatic speaker verification, *Speech Communication* (2009), doi: 10.1016/j.specom.2009.01.005

Optimizing Feature Complementarity by Evolution Strategy: Application to Automatic Speaker Verification

C. Charbuillet, B. Gas, M. Chetouani, J.L. Zarader,

Université Pierre et Marie Curie-Paris6, UMR 7222 CNRS, Institut des Systèmes Intelligents et Robotique (ISIR), Ivry sur Seine, F-94200 France

Abstract

Conventional automatic speaker verification systems are based on cepstral features like Mel-scale Frequency Cepstrum Coefficient (MFCC), or Linear Predictive Cepstrum Coefficient (LPCC). Recent published works showed that the use of complementary features can significantly improve the system performances. In this paper, we propose to use an evolution strategy to optimize the complementarity of two filter bank based feature extractors. Experiments we made with a state of the art speaker verification system show that significant improvement can be obtained. Compared to the standard MFCC, an Equal Error Rate (EER) improvement of 11.48% and 21,56% was obtained on the 2005 Nist SRE and Ntimit databases, respectively. Furthermore, the obtained filter banks picture out the importance of some specific spectral information for automatic speaker verification.

Key words:
Feature Extraction, Evolution Strategy, Speaker Verification

1. Introduction

Automatic speaker verification (ASV) is now extended across several domains. Applications include security access control, telephone banking transactions, surveillance, audio-indexing and forensic speaker recognition. The front-end of state of the art speaker verification systems is based on the estimation of the spectral envelope of the short term signal, e.g. Mel-scale Frequency Cepstrum Coefficient (MFCC), or Linear Predictive Cepstrum Coefficient (LPCC) (Reynolds, 2002). However, these methods were initially designed for speech recognition and, consequently, they are not the most suitable for speaker recogni-

Email addresses: christophe.charbuillet@lis.jussieu.fr (C. Charbuillet), bruno.gas@upmc.fr (B. Gas), mohamed.chetouani@upmc.fr (M. Chetouani), jean-luc.zarader@upmc.fr (J.L. Zarader).

tion tasks. To improve ASV performances, several approaches have been proposed to optimize the feature extractor for a specific task (Katagiri et al., 1998). These methods consist of simultaneously learning the parameters of both the feature extractor and the classifier (Chetouani et al., 2005). These procedures are based on the optimization of a criterion, which can be the Maximization of the Mutual Information (MMI) (Torkkola, 2003) or the Minimization of the Classification Error (Miyajima et al., 2001).

In this paper we propose to use an Evolution Strategy (ES) to design a feature extraction system adapted to the speaker verification task.

Recent progress in speaker verification has created interest in new and challenging tasks. To increase utility in forensic application, the Nist 2004, 2005 and 2006 speaker recognition evaluations have added cross-channel and cross-language tasks (Przybocki et al., 2006). Research has been supported by the creation of the Mixer and Reading Corpora by the Linguistic Data Consortium (Cieri et al., 2006). Most of the systems used for these evaluations are based on the state of the art cepstral Gaussian Mixture Model using a Universal Background Model (GMM-UBM) system. Recent improvements were obtained by the means of three different classification approaches: discriminative techniques based on support vector machines (SVM) (Campbell et al., 2004), channel compensation in model space (Yin et al., 2006), and integration of high level information (Reynolds et al., 2003). Feature transformation techniques were also well exploited to remove cross-channel effects. These include well-known and widely used blind transformation such as cepstral mean subtraction, RASTA filtering, spectral subtraction and feature warping (Pelecanos and Sridharan, 2001). More recently, model based feature transformations were proposed by Reynolds et al. (2003) and by Vair et al. (2006) with feature mapping and channel factor based feature transform approaches, respectively.
Feature extraction still remains widely based on the estimation of the cepstral envelope of the short term signal. Feature extraction methods described in the Nist literature are MFCC (Mel Frequency Cepstrum Coefficient), LPCC (Linear Predictive Cepstrum Coefficient) and PLP (Perceptual Linear Predictive). However, it should be noted that

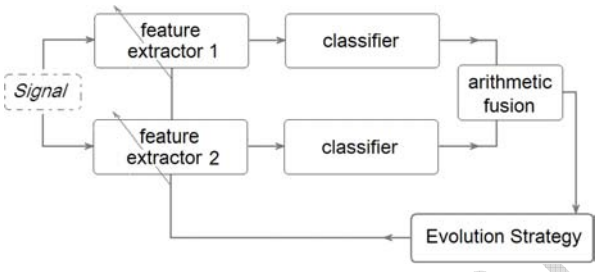


Figure 1. *Complementary feature extraction optimization*

MFCC is the most used method of feature extraction.

Currently we have an alternative and increasingly used approach which consists of fusing heterogeneous systems. These approaches can be classified into two categories: fusion of systems using different classifiers (Farrell et al., 1998) and fusion of systems based upon different features. Our study deals with the second principle. Complementarity of the LPCC and MFCC were pointed out by Zhiyou et al. (2003) and later on by Campbell et al. (2007). Poh Hoon Thian et al. (2004) showed that significant improvements can be obtained by combining Linear Frequency Cepstral Coefficient (LFCC) with spectral subband centroid features.

In this article, we propose to use an evolution strategy to optimize the complementarity of feature extractors. This approach is illustrated by Fig. 1. The main contributions of our work are the following:

– we propose an algorithm that optimizes the feature extraction complementarity of two speaker verification systems,

– we applied this algorithm to the optimization of the filter banks of cepstral feature extractors. Experiments we made using different optimization conditions show the existence of a unique solution. This allows us to depict the importance of specific spectral information for speaker verification,

– the obtained feature extraction system can be easily integrated on a state of the art speaker verification system by an appropriate tuning of the LFCC feature extractor.

This article is structured as follows: a description of the proposed algorithm is given in section 2. Section 3 presents the experiments we made and the obtained results. The conclusion and the perspectives of this study are given in section 4.

2. Proposed algorithm

Evolutionary Algorithms (EAs) are nature-inspired optimization methods. The basic idea is that of "natural selection", i.e. the principle of "the survival of the fittest". This class of algorithms has been successfully applied to the speech processing domain, in particular with the use of Genetic algorithm (GA). Chin-Teng et al. (2000) proposed to use GA to the feature transformation problem for speech recognition and Zamalloa et al. (2006) worked on a GA based feature selection for speaker recognition. In the later study, a GA is used to select the most important characteristics of the cepstral feature vector to reduce the system complexity.

In this paper, we propose an Evolution Strategy (ES) (Beyer and Schwefel, 2002) that optimizes the complementarity of two feature extractors. We present an application to the optimization of the filter bank of two cepstral feature extractors.

This section is organized as follow: first the optimization criterion we used is presented in section 2.1. Then, the ES used to minimize the presented criterion is described in section 2.2. Finally, the proposed algorithm is discussed and the related works are presented in section 2.3 .

2.1. *Optimization criterion*

The principle of our approach is to fuse two speaker verification systems based on complementary feature extractors. The fusion we used is a weighted sum between two GMM based speaker verification systems. This fusion is given by:

$$L_f = \alpha L_1 + (1 - \alpha) L_2 \tag{1}$$

where L_1 and L_2 are respectively the log likelihood (LLK) produced by the two GMM systems to fuse, L_f is the resulting LLK and $\alpha \in [0; 1]$ is the fusion weight.

The performance measure we used is the Equal Error Rate (see section 3.2.3). In the rest of this paper, the EER obtained by the evaluation of the system S on the database $\mathcal{B}$ will be given by:

$$EER^{\mathcal{B}}[S] \tag{2}$$

In a generic way, the *optimal fusion* of the two systems S_1 and S_2 on a database $\mathcal{B}$ can be represented by:

$$S_f = S_1 \oplus^{\mathcal{B}} S_2 \tag{3}$$

where S_f is the system resulting of the optimal fusion of the system S_1 and S_2 and $\mathcal{B}$ is the database used for the fusion tuning. In our case, the estimation of the weight α is made to minimize $EER^{\mathcal{B}}[S_f]$. This is done by testing all α values in $[0; 1]$ with a step of 0.001.

Let $S_{(C1)}$ and $S_{(C2)}$ be two speaker verification systems based on the feature extractors $C1$ and $C2$. The aim of our algorithm is to find $C1$ and $C2$ which minimize the Feature Complementary Criterion (FC-Criterion) defined by:

$$EER^{\mathcal{B}_V}[S_{(C1)} \oplus^{\mathcal{B}_C} S_{(C2)}] \tag{4}$$

where $\mathcal{B}_V$ is a validation database and $\mathcal{B}_C$ is a cross validation one. These two databases must be independent to represent real word application. The next subsection describes the algorithm we used to minimize this criterion.

2.2. *Evolution strategy for complementary optimization*

Our method is based on the evolution of two populations ($\mathcal{P}_1$ and $\mathcal{P}_2$) of feature extractors under a mutation, evaluation and selection loop. This method is described by **Algorithm 1**.

2.2.1. *Population definition*

Each individual of the populations represents a linear filter bank, defined by its minimum and maximum frequencies:

$$\mathbf{a} = \begin{cases} \mathbf{y} = \{F_{min}, F_{max}\} & \in \mathbb{R}^2 \\ \mathbf{F} & \in \mathbb{R} \end{cases} \tag{5}$$

where $\mathbf{a}$ represents an individual, F_{min} and F_{max} are the minimum and maximum frequencies of the filter bank and $\mathbf{F}$ represents the fitness. The two populations of filter banks are defined by:

Algorithm 1 Evolution strategy for complementary optimization

1: $t := 0$
2: initialize($\mathcal{P}_1^{t=0}$)
3: initialize($\mathcal{P}_2^{t=0}$)
4: **while** stop_criterion **do**
5: $\tilde{\mathcal{P}_1}^t := mutation(\mathcal{P}_1^t)$
6: $\tilde{\mathcal{P}_2}^t := mutation(\mathcal{P}_2^t)$
7: $\{\hat{\mathcal{P}_1}^t, \hat{\mathcal{P}_2}^t\} := evaluation(\tilde{\mathcal{P}_1}^t, \tilde{\mathcal{P}_2}^t)$
8: $\mathcal{P}_1^{t+1} := selection(\hat{\mathcal{P}_1}^t)$
9: $\mathcal{P}_2^{t+1} := selection(\hat{\mathcal{P}_2}^t)$
10: $t := t + 1$
11: **end while**

$$\begin{aligned} \mathcal{P}_1 &= \{\mathbf{a}_1^1, ..., \mathbf{a}_\lambda^1\} \\ \mathcal{P}_2 &= \{\mathbf{a}_1^2, ..., \mathbf{a}_\lambda^2\} \end{aligned} \tag{6}$$

where λ is the number of individuals.

2.2.2. *Initialization*

The first step of the algorithm consists of a random initialization of the $\mathbf{y}$ vector of each individual. The initialization method we used is given by:

$$\begin{aligned} \mathbf{y} &:= [U(0, Fe/2), U(0, Fe/2)] \\ \mathbf{y} &:= sort(\mathbf{y}) \end{aligned} \tag{7}$$

where $U(a, b)$ represents a random variable uniformly distributed on $[a; b]$ and Fe is the sampling frequency of the signals. The *sort* function aims at ensuring $F_{min} < F_{max}$.

2.2.3. *Mutation*

The mutation operator aims at exploring the search space. It consists of a short random variation applied to each individual of the population. The mutation method we used is given by:

$$\tilde{\mathbf{y}} := \mathbf{y} + r \cdot [N(0, 1), N(0, 1)] \tag{8}$$

where $N(0, 1)$ is a random variable with standard normal distribution and r represents the mutation rate.

2.2.4. *Evaluation*

The evaluation operator represents the main contribution of our algorithm. At each generation, all combinations of feature extractors are evaluated and the resulting Equal Error Rates (EER) are memorized. At the end of this process, the fitness of an individual is defined as the lowest EER obtained (e.g. the EER corresponding to the best combination including this feature extractor). Consequently, complementary couples of filter banks tend to emerge. This operator is given by **Algorithm 2**:

Algorithm 2 Evaluation operator

1: $E \in \mathbb{R}^{2\lambda}$
2: **for** $i = 1$ to λ **do**
3: **for** $j = 1$ to λ **do**
4: $E(i, j) := EER^{\mathcal{B}_{EV}}[S(\mathbf{a}_i^1) \oplus^{\mathcal{B}_{EV}} S(\mathbf{a}_j^2)]$
5: **end for**
6: **end for**
7: **for** $i = 1$ to λ **do**
8: $\mathbf{F}_i^1 := \min_j[E(i, j)]$
9: **end for**
10: **for** $j = 1$ to λ **do**
11: $\mathbf{F}_j^2 := \min_i[(E(i, j)]$
12: **end for**

Line n°4 of **Algorithm 2** refers to a reduced version of the FC-Criterion defined by equation 4. Here, the same database $\mathcal{B}_{EV}$ (called evolution database) is used both for fusion tuning and EER estimation. This approximation strongly reduces the computational cost of our algorithm. As we will see in section 3.4, the solution obtained by the use of this reduced criterion satisfactorily generalizes according to the FC-Criterion. $S(\mathbf{a}_i^1)$ represents a speaker verification system using the filter bank defined by the i^{th} individual of population $\mathcal{P}_1$ and $\mathbf{F}_i^1$ represents the fitness of this individual (*ditto* for $\mathcal{P}_2$).

2.2.5. *Selection*

The *Selection* operator picks out the μ best feature extractors of the current population. These individuals are then cloned according to the evaluation results to produce the new generation $\mathcal{P}^{t+1}$ composed of λ individuals. The selection operator we used is given by **Algorithm 3**. In this pseudocode, $U(0, 1)$ (line 5) represents a random variable uniformly distributed on $[0; 1]$.

2.3. *Related works*

The evolution strategy we use is directly derived from the multimembered $(\mu/\rho, \lambda)$-ES describes by Beyer and Schwefel (2002). λ is the number of offspring, μ is the number of parents and ρ refers to the number of parents involved in the procreation

Algorithm 3 Selection operator

1: Select the μ bests individuals:
$\breve{\mathcal{P}}^t = \{\mathbf{a}_1, ..., \mathbf{a}_\mu\}$, $\mu < \lambda$
2: Normalize the fitness of $\breve{\mathcal{P}}^t$ to $[0; 0.5]$
3: **while** number of individual of $\mathcal{P}^{t+1} < \lambda$ **do**
4: **for** $i = 1$ to μ **do**
5: **if** $\mathbf{F}_i < U(0, 1)$ **then**
6: $\mathcal{P}^{t+1} \overset{Add}{\leftarrow} \mathbf{a}_i$
7: **end if**
8: **end for**
9: **end while**

of one offspring during the recombining process. We use the simplest case without recombination $\rho = 1$ (cloning), usually denoted by (μ, λ)-ES. As we will see, this simple ES version satisfactorily solves our optimization problem.

2.4. *Minimizing the over fitting effects*

The over fitting effect is a common problem in machine learning (Mitchell, 1997). To avoid this effect, several approaches were proposed for evolutionary computation such as cross validation (CV), early stopping (ES), complexity reduction (CR), noise addition (NA) or random sampling technique (RST): Paris et al. (2004), Yi and Khoshgoftaar (2004), Ross (2000). In our application we combined the cross validation and random sampling technique described in the following:

RST consists in using a randomly selected subset of training data to evaluate the individual's performance. This subset is extracted from the *global train database* (describes in section 3.2). Each generation of individuals is evaluated on a new subset.

CV technique consists in evaluating the generalization capacity of an individual by testing it on data which does not belong to the training nor to the test database.

Each generation of individuals is evaluated on a new subset extracted using the RST technique. For each generation we evaluate and memorize the performances of the best individual of the population on a cross validation base. The algorithm is stopped when a stagnation of the performances is observed. Then, the best individual of the best generation on the cross validation base is selected and evaluated on the test database.

3. Experiments

3.1. *Speaker verification system*

All experiments we made are based on a state of the art Gaussian Mixture Model based on Universal Background Model (GMM-UBM) speaker verification system. This system, is the LIA SpkDet provided by the University of Avignon [1], France.

3.1.1. *Front-end*

First, the speech signal is segmented into frames by a 20 ms window processing at 10 ms frame rate. Next, cepstral feature vectors are extracted from the speech frames. The first derivatives are then added to the feature vectors. Last, a speech activity detector (SAD) is used to discard silence/noise frames.

3.1.2. *GMM system*

The system used for ES filter bank evaluations is a GMM with diagonal covariance matrix composed of 16 mixture components. The use of this reduced system was imposed by the computational cost of ES. The evaluations of the filter banks obtained by ES are done using a GMM system using different number of mixture components (16, 32, 64, 128, 256, 512, and 1024).

3.1.3. *Baseline systems*

We used two different baseline systems to compare the results obtained by ES. These systems are based on the standard LFCC and MFCC feature extractors using 24 filters and 16 cepstrum coefficients. The linear and the Mel scaled filter bank are scaled to the $[300Hz; 3400Hz]$ frequency interval.

3.1.4. *Computational cost*

The computational cost of a filter bank evaluation during the evolution process with a 16 mixture components system is of about 10 minutes on a 3GHz Pentium computer. Consequently, the computational cost of an evolution run of 40 individual during 50 generations is of about 17 days. For our experiment we used a cluster system of

[1] LIA web site: http://www.lia.univ-avignon.fr

$8 \times 3GHz$ computers, able to reduce the computational cost to approximately 2 days.

3.2. *Databases and evaluation protocol*

Two different corpus were used for our experiments. The main one, the 2005 Nist SRE database, was used for the filter bank evolution. We evaluated the performances of the obtained filter banks on both the 2005 Nist SRE database and the Ntimit one. These two databases are detailed in the following.

3.2.1. *The 2005 Nist databases*

The 2005 Nist corpus is extracted from the Mixer and Transcript Reading Corpora (Cieri et al., 2006). This corpus is dedicated to cross-channel and cross-language speaker recognition research. It is composed of conversational telephone speech passed through different channels (land-line, cordless or cellular) and eight different types of handsets. The number of utterances produced by each speaker vary from 1 to 30 with an average of 8. Signals are sampled to 8 kHz. We used for our experiments utterances of 2min 30s corresponding to the 1conv4w-1conv4w 2005 Nist SRE evaluation plan.
The dataset we used for the filter bank evaluation during the evolution process (called evolution database) is made up using the random sampling technique described in section 2.4. At each generation signals from 10 males and 10 females are extracted from a *global train* database of 30 males and 30 females. These extracted signals compose the *evolution* databases. The *cross validation* database is composed of signals from 30 males and 30 females. The *validation* database is made up of 100 males and 100 females. It is important to point out that the speakers involved in these three databases are different.
Speaker models were trained using one utterance of 2min 30s per speaker. The rest of the utterances were used as tests. Experiments were performed by testing all the models with all the test utterances. Table 1 shows the number of true tests and imposter tests for each database.

3.2.2. *The Ntimit databases*

The NTIMIT database is composed of clean speech signals from the TIMIT database recorded over local and long-distance telephone loops. Each sentence was played through an "artificial mouth" coupled to a carbon-button telephone handset. The speech was transmitted through a local or long-distance central office and looped back for recording. Even if signals are sampled to 16kHz, useful bandwidth is reduced to 300-3400kHz. 10 utterances of 3s were recorded for each speaker.
We used 168 speakers of the test portion of the database for the Ntimit *evaluation* database. Speaker models were trained using 8 utterances totalizing 24s. The remaining two utterances of 3s each were individually used as tests. We used 50 males and 50 females of the train portion of the Ntimit database to create the *cross validation* database. Experiments were performed by testing all the models with all the test utterances. Table 2 shows the number of true tests and imposter tests for each database.

Table 1
Number of claimant and imposter trials for the 2005 Nist database

Database	true tests	imposter tests	total
Global train	622	17541	18163
Evolution	≈200	≈1800	≈2000
Cross validation	631	18316	18947
Validation	1332	115610	116942

Table 2
Number of claimant and imposter trials for the Ntimit database

Database	true tests	imposter tests	total
Cross validation	200	9800	10000
Validation	334	30804	31138

3.2.3. *Performance measures*

Speaker verification performances are reported using two different measures: The Equal Error Rate (EER) and the Detection Cost Function (DCF) used for the Nist SRE evaluation. These measures are derived from the false acceptation probability $P_{FA}(\theta)$ and the false-reject probability $P_{FR}(\theta)$ of the verification system. These probabilities are functions of the decision threshold θ.
The well known EER is defined by the false acceptation probability $P_{FA}(\theta_0)$ corresponding to a decision threshold θ_0 verifying $P_{FA}(\theta_0) = P_{FR}(\theta_0)$.
The DCF is defined by the following weighted sum:

$$DCF = \frac{C_{Miss} \cdot P_{FR} \cdot P_{target} + C_{FA} \cdot P_{FA} \cdot (1 - P_{target})}{NormFact} \quad (9)$$

where $C_{Miss} = 10$ and $C_{FA} = 1$ are the relative costs of detection errors and $P_{target} = 0.01$ is the *a priori* probability of the specified target speaker. This cost function is normalized by $NormFact = 0.1$ so that a system with no discriminative capability is assigned a cost of 1.0. The values of these parameters are given by the Nist SRE evaluation plan. The optimal decision threshold is calculated to minimize the DCF.

3.3. *Evolution*

We present in this section a set of 3 different evolution runs. These experiments were done using the ES settings given by Table 3. This parameters are defined on section 2.2. It is important to notice that the initial conditions (i.e. initial populations) of these 3 evolution runs were different.

Table 3
ES parameters

population size (λ)	20
number of selected individuals (μ)	5
mutation rate (r)	300Hz

Fig. 2 shows the evolution of the F_{min} and F_{max} parameters from the initialization to the 60^{th} generation. We report in this figure parameters from the μ selected parents of each generation for the 3 evolution runs. We can notice that all these experiments converge to a unique solution. Population $\mathcal{P}_1$ specializes on large filter banks ($[300Hz; 3400Hz]$) whereas population $\mathcal{P}_2$ focuses on a short spectrum zone ($[400Hz; 1300Hz]$). In the new section, the best filter banks of these 3 evolution runs are evaluated.

3.4. *Filter banks evaluation*

During the evolution, we evaluate the best individual of each generation on the cross validation database and memorize it. The evolution strategy is stopped when a stabilization of the population performance is observed. Then the best individual of the evolution is selected and tested on the test databases. We present here the best pair of filter banks obtained during the 3 evolution runs presented bellow, named {Fb1.a;Fb1.b},{Fb2.a;Fb2.b} and {Fb3.a;Fb3.b}. Table 4 presents their characteristics and Fig. 4 presents their performances on the Nist and Ntimit validation bases. Filter banks Fb3.a and Fb3.b are illustrated by Fig. 3. To interpret the following results, it is important to recall the condition used for the evolution:

- the database used for the filter bank evolution is exclusively extracted from the 2005 Nist corpus;
- the GMM system used has 16 mixture components;
- the evaluation criterion used is the EER.

Several experiments were made to evaluate possible overfitting of the evolution condition. These experiments were made using the following conditions:

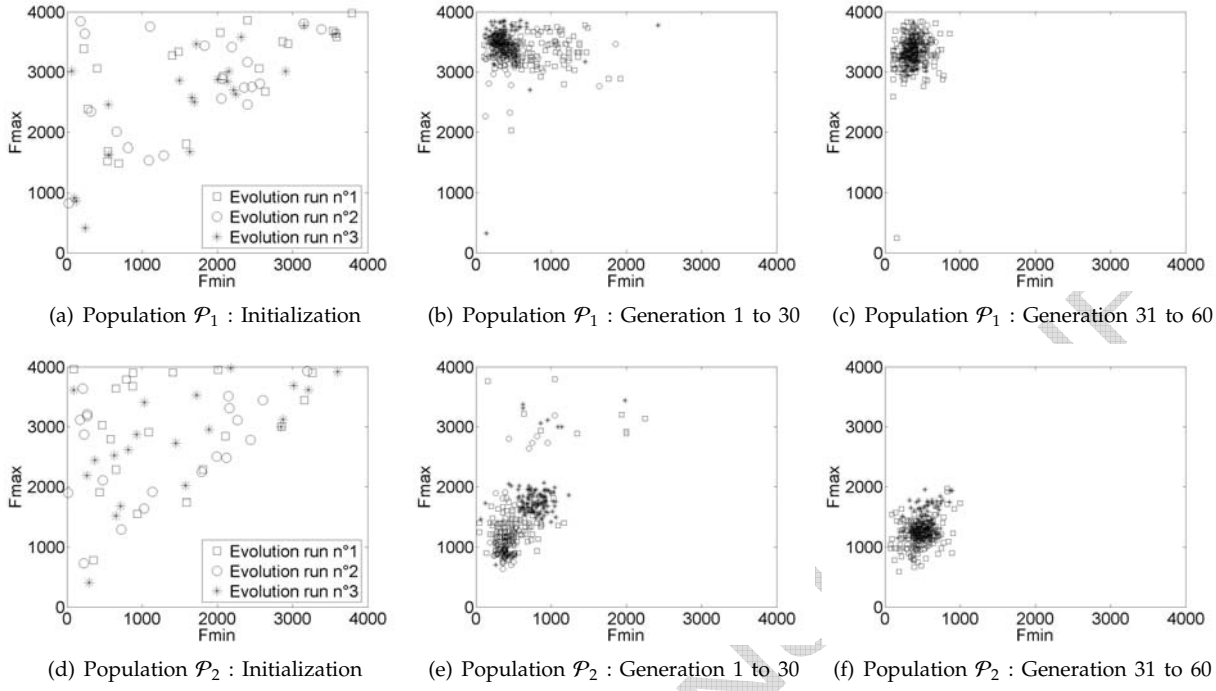
- the databases used for the tests are extracted from the 2005 Nist and the Ntimit corpus;
- filter banks are evaluated on GMM system using a number of mixture components of 16 and more (16 to 1024);
- performance measures used are both the EER and the DCF.

The presented results were made with real world conditions: for each test, the fusion weight α was estimate by the use of a cross validation database according to the FC-Criterion defined in section 2.1. The cross validation databases used for the fusion tuning are corpus dependent (Nist/Ntimit) and are detailed in section 3.2.

Table 4
Filter bank characteristics

Filter bank	F_{min} (Hz)	F_{max} (Hz)
Fb1.a	251	3278
Fb1.b	549	1349
Fb2.a	298	3294
Fb2.b	454	1270
Fb3.a	282	3168
Fb3.b	376	1270

The results we obtained show significant improvements compared to baseline systems. On the **2005 Nist database**, filter banks Fb3 obtained a relative **EER** improvement of 10.8% compared to the LFCC and of 11.48% compared to the MFCC



(a) Population $\mathcal{P}_1$: Initialization (b) Population $\mathcal{P}_1$: Generation 1 to 30 (c) Population $\mathcal{P}_1$: Generation 31 to 60

(d) Population $\mathcal{P}_2$: Initialization (e) Population $\mathcal{P}_2$: Generation 1 to 30 (f) Population $\mathcal{P}_2$: Generation 31 to 60

Figure 2. Evolution of F_{min} and F_{max} for each population

systems. The **DCF** relative improvements are respectively of 6.19% and 6.37%. On the **Ntimit database**, filter banks Fb3 obtained a relative **EER** improvement of 22.0% compared to the LFCC and of 21.56% compared to the MFCC systems. The **DCF** relative improvements are of 19.96% and 14.09% respectively. The relative improvement measure we used is given by:

$$\frac{BestEER[S_b] - BestEER[S_i]}{BestEER[S_b]} \times 100 \qquad (10)$$

where S_b is a baseline system, S_i is the improved system and $BestEER[]$ represents the best EER obtained according to the GMM complexity (same thing for DCF).

It is important to recall that the amount of data available for the speaker model training is of 24s for the Ntimit database and of 2min 30s for the Nist database. The amount of data available for the model test is of 3s and of 2min 30s, respectively. The complementary information provided by the short filter bank seems to be more useful when a small amount of data is available.

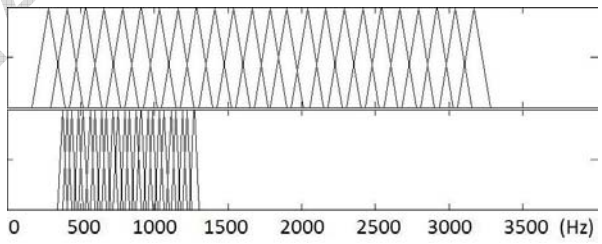
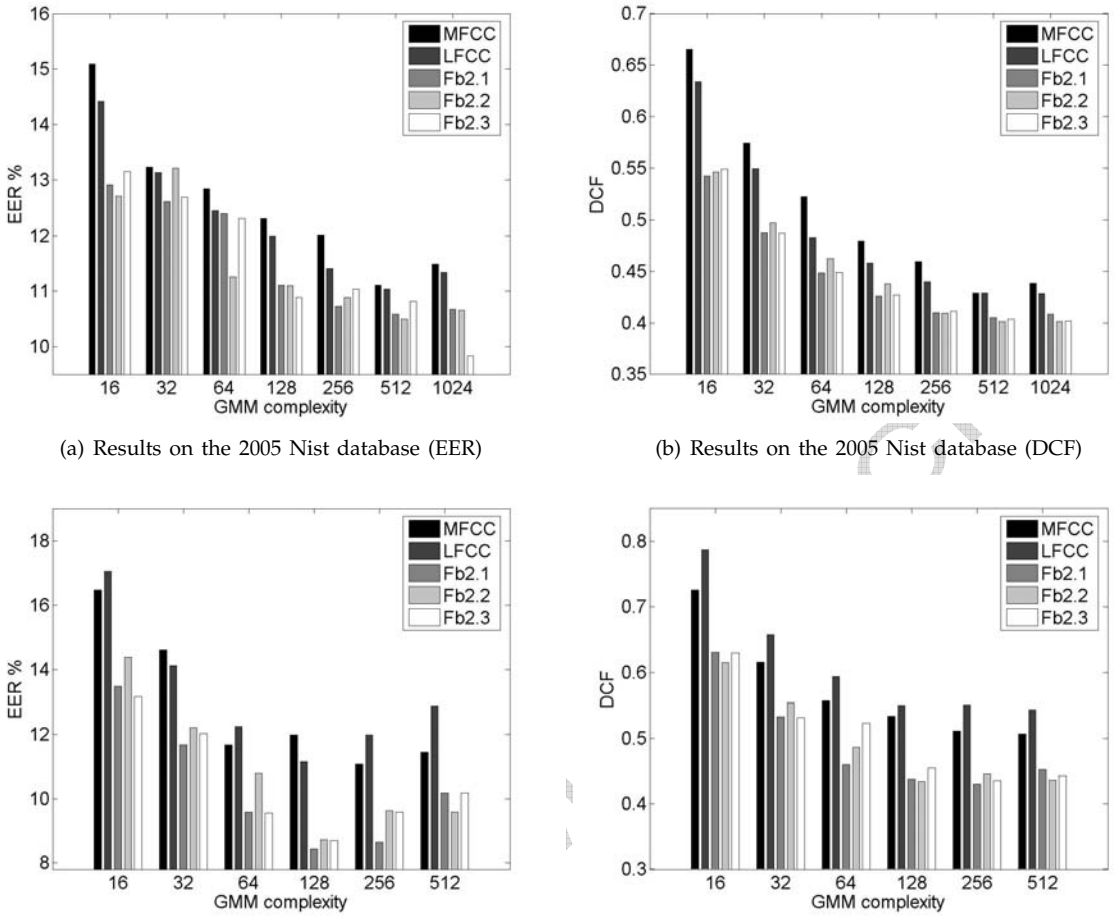


Figure 3. Filter bank Fb3.a (top) and Fb3.b (bottom)

4. Conclusion

In this paper, we proposed to use an Evolution Strategy (ES) to optimize the feature extraction system based on two complementary filter banks (CFB). The obtained CFB showed significant improvements on both the 2005 Nist and the Ntimit databases. Moreover, repetition of the optimization showed the robustness of the obtained solution according to ES initial conditions. The singularity and the characteristics of the obtained solutions allowed us to conclude that the frequency domain defined by $[376Hz; 1270Hz]$ contains important complementary speaker information.

The obtained improvements show that the traditional LFCC or MFCC feature extraction meth-



(a) Results on the 2005 Nist database (EER)

(b) Results on the 2005 Nist database (DCF)

(c) Results on the Ntimit database (EER)

(d) Results on the Ntimit database (DCF)

Figure 4. Results obtained on the 2005 Nist and Ntimit databases

ods are not able to provide an optimal cepstral representation for speaker verification. This was already pointed out by the researches of Campbell et al. (2007) which prove that significant improvements can be obtained by fusing two complementary cepstral systems based on LPCC and MFCC features, when the channel effects are removed. Thus, the following questions arise: can we obtain similar performances with a single feature extractor? Or, if this is not the case, should we reconsider the structure of conventional speaker verification systems?

Our future works will consist in exploring the second hypothesis by investigating the following problems:

– How many feature extractors should be used for an optimal cepstral representation?
– Which is the optimal way of combining these different features?

5. Acknowledgments

The authors would like to thank Gerard Chollet (CNRS-ENST, France) for his assistance during the 2006 Nist SRE campaign, Guillaume Gravier (CNRS-INRIA, France) and Jean François Bonastre (LIA, France) for providing the speaker verification systems we used, and for their guidance. We also want to thank Douglas Reynolds (MIT, USA) for his advices on the multi-feature approach.

References

H.-G. Beyer and H.-P. Schwefel. Evolution strategies, a comprehensive introduction. *Natural Computing*, 1:2–52, 2002.

W. M. Campbell, D. A. Reynolds, and J. P. Campbell. Fusing discriminative and generative methods for speaker recogntion: Experiments on switchboard and nfi/tno field data. In *Speaker and Language Recognition Workshop. IEEE Odyssey 2004*, pages 41–44, 2004.

W. M. Campbell, D. E. Sturim, W. Shen, D. A. Reynolds, and J. Navratil. The mit-ll/ibm 2006 speaker recognition system: High-performance reduced-complexity recognition. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, 2007.

M. Chetouani, M. Faundez-Zanuy, B. Gas, and J.-L. Zarader. *Lecture Notes in Computer Science*, chapter Non-linear Speech Feature Extraction for Phoneme Classification and Speaker Recognition, pages 344–350. Lecture Notes in Computer Science. Springer, 2005.

L. Chin-Teng, N. Hsi-Wen, and H. Jiing-Yuan. Ga-based noisy speech recognition using two-dimensional cepstrum. In *IEEE Transactions on Speech and Audio Processing*, volume 8, pages 664–675, 2000.

C. Cieri, W. Andrews, J. P. Campbell, G. Doddington, J. Godfrey, S. Huang, M. Liberman, A. Martin, H. Nakasone, M. Przybocki, and K. Walker. The mixer and transcript reading corpora: Resources for multilingual, crosschannel speaker recognition research. In *LREC 2006: Fifth International Conference on Language Resources and Evaluation*, 2006.

K. Farrell, R. Ramachandran, and R. Mammone. An analysis of data fusion methods for speaker verification. In *Acoustics, Speech, and Signal Processing, 1998. ICASSP '98. Proceedings of the 1998 IEEE International Conference on*, volume 2, pages 1129–1132 vol.2, 1998.

S. Katagiri, J. Biing-Hwang, and L. Chin-Hui. Pattern recognition using a family of design algorithms based uponthe generalized probabilistic descent method. In *Proceedings of the IEEE*, volume 86, pages 2345–2373, 1998.

T. Mitchell. *Machine learning*. McGraw-Hill Higher Education, 1997.

C. Miyajima, H. Watanabe, K. Tokuda, T. Kitamura, and S. Katagiri. A new approach to designing a feature extractor in speaker identification based on discriminative feature extraction. *Speech Communication*, 35(3-4):203–218, Oct. 2001.

G. Paris, D. Robilliard, and C. Fonlupt. *Lecture Notes in Computer Science*, chapter Exploring Overfitting in Genetic Programming, pages 267–277. Springer, 2004.

J. Pelecanos and S. Sridharan. Feature warping for robust speaker verification. In *Speaker and Language Recognition Workshop. IEEE Odyssey 2001*, 2001.

N. Poh Hoon Thian, C. Sanderson, S. Bengio, D. Zhang, and K. Jain Anil. Spectral subband centroids as complementary features for speaker authentication. *Lect. notes comput. sci.*, 3072:631–639, 2004.

M. Przybocki, A. Martin, and A. Le. Nist speaker recognition evaluation chronicles-part 2. *Speaker and Language Recognition Workshop, 2006. IEEE Odyssey 2006: The*, pages 1–6, 2006.

D. Reynolds. An overview of automatic speaker recognition technology. In *Acoustics, Speech, and Signal Processing, 2002. Proceedings. (ICASSP '02). IEEE International Conference on*, volume 4, pages 4072–4075, 2002.

D. Reynolds, W. Andrews, J. Campbell, J. Navratil, B. Peskin, A. Adami, Q. Jin, D. Klusacek, J. Abramson, R. Mihaescu, J. Godfrey, D. Jones, and B. Xiang. The supersid project: Exploiting high-level information for high-accuracy speaker recognition. In *Acoustics, Speech, and Signal Processing (ICASSP '03). IEEE International Conference on*, pages 784–787, 2003.

B. Ross. The effects of randomly sampledtraining data on program evolution. In *GECCO*, pages 443–450, 2000.

K. Torkkola. Feature extraction by non parametric mutual information maximization. *The Journal of Machine Learning Research*, 3:1415–1438, 2003. ISSN 1533-7928.

C. Vair, D. Colibro, F. Castaldo, E. Dalmasso, and P. Laface. Channel factors compensation in model and feature domain for speaker recognition. In *Speaker and Language Recognition Workshop. IEEE Odyssey 2006*, 2006.

L. Yi and T. Khoshgoftaar. Reducing overfitting in genetic programming models for software quality classification. In *High Assurance Systems Engineering, 2004. Proceedings. Eighth IEEE International Symposium on*, 2004.

S.-C. Yin, P. Kenny, and R. Rose. Experiments in speaker adaptation for factor analysis based

speaker verification. In *Speaker and Language Recognition Workshop. IEEE Odyssey 2006*, 2006.

M. Zamalloa, G. Bordel, J. L. Rodriguez, and M. Penagarikano. Feature selection based on genetic algorithms for speaker recognition. In *Speaker and Language Recognition Workshop. IEEE Odyssey 2006*, volume 1, pages 1–8, 2006.

M. Zhiyou, Y. Yingchun, and W. Zhaohui. Further feature extraction for speaker recognition. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 5, pages 4153–4158, 2003.