ALIZE/SpkDet: a state-of-the-art open source software for speaker recognition

Jean-François Bonastre[1], *Nicolas Scheffer*[1], *Driss Matrouf*[1], *Corinne Fredouille*[1],
Anthony Larcher[1], *Alexandre Preti*[1], *Gilles Pouchoulin*[1], *Nicholas Evans*[1,2],
Benoît Fauve[2] *and John Mason*[2]

[1]Laboratoire d'Informatique d'Avignon (LIA), UAPV, France
[2]Speech and Image Group, Swansea University, Wales, UK
[1]{firstname.name}@univ-avignon.fr, [2]{b.g.b.fauve.191992,J.S.D.Mason}@swansea.ac.uk

Abstract

This paper presents the ALIZE/SpkDet open source software packages for text independent speaker recognition. This software is based on the well-known UBM/GMM approach. It includes also the latest speaker recognition developments such as Latent Factor Analysis (LFA) and unsupervised adaptation. Discriminant classifiers such as SVM supervectors are also provided, linked with the Nuisance Attribute Projection (NAP). The software performance is demonstrated within the framework of the NIST'06 SRE evaluation campaign. Several other applications like speaker diarization, embedded speaker recognition, password dependent speaker recognition and pathological voice assessment are also presented.

1. Introduction

Indicated by the growing number of participants in the international NIST speaker recognition evaluations (SRE) [1], text-independent automatic speaker verification (ASV) has experienced an increasing interest in recent years. At the same time meaningful improvements in performance have been achieved with error rates roughly halving over the last three NIST SRE campaigns. These two phenomena are not directly linked since the best performances invariably come from a limited number of sites, mostly with previous experience in such campaigns. And any exceptions to this observation tend to relate to site combinations where perhaps one or more of the partners have previous SRE experience. The key point here is the high level of research and technology investment required to reach and remain close to the ever moving state-of-the-art. In this context, combined effort across sites can clearly help; the simplest and most common example of this is the sharing of system scores with score-level fusion. At the other extreme of such cooperation is open source software, the potential of which is far more profound.

This paper presents open source software: the ALIZE/SpkDet packages. The realization has taken place in the context of the open source ALIZE toolkit [2, 3] developed by the LIA in the framework of the French Research Ministry Technolangue[1] programme. ALIZE comes from the collaboration within the ELISA consortium [4], which grouped the efforts of several European laboratories in order to participate in NIST evaluation campaigns (mainly NIST-SRE and NIST Rich Transcription evaluation) from 1998 to 2004. ALIZE/SpkDet is a package within ALIZE tailored specifically to ASV; ALIZE/SpkDet was one of the chosen reference systems in the framework of the BioSecure Network of Excellence[2]; it was also used by several institutions for the NIST'06 SRE. Feature extraction comes from SPro [5]. All these systems are distributed through open source licences.

Recent key developments include support vector machines (SVMs) [6], the associated Nuisance Attribute Projection compensation (NAP) [7], and Factor Analysis (FA) [8, 9]. These developments stem mainly from a collaboration between the LIA and the Swansea University.

This paper presents an overview of the ALIZE/SpkDet software. After a short technical description in Section 2, Section 3 presents a demonstration of the toolkit performance, using the latest NIST SRE database. In Section 4, some work concerning unsupervised adaptation of the client models is described and evaluated using the NIST'05 and '06 SRE frameworks. Even if the software is mainly designed for text-independent speaker recognition, several other applications use ALIZE/SpkDet. Section 5 presents some examples of these developments. Finally, Section 6 concludes with ideas on the immediate future of the project.

2. Overview of ALIZE/SpkDet

The ALIZE project was initiated by the ELISA consortium under the responsibility of the LIA and initially funded by the French Research Ministry Technolangue program during 2003/2004. The project was continued using LIA and BioSecure funds. ALIZE was also retained for two French National Research Agency (ANR) projects, BIOBIMO and MISTRAL, allowing support until the end of 2008.

2.1. ALIZE main objectives

The main objectives of ALIZE are:

- to propose a toolkit facilitating the development of new ideas, with a (proven) state-of-the-art level of performance;
- to encourage laboratories to evaluate new proposals using the toolkit on both standard databases and protocols such as the NIST SRE evaluations;
- to help the understanding of speaker recognition algorithms (like EM training, MAP adaptation or Viterbi algorithm), parameters and limits;
- to test new commercial applications;
- to facilitate the exchanges and knowledge transfer between the academic laboratories and between academic laboratories and companies.

[1] http://www.technolangue.net/

[2] http://biosecure.info

Through the MISTRAL and BIOBIMO projects, some new functionalities are currently developed, like client/server or embedded architectures, text dependent speaker recognition systems and multimodal user authentication systems.

2.2. An illustration of ALIZE specificities

ALIZE is open source software developed in C++ following an object oriented UML method. The general architecture of the toolkit is based on a split of the functionalities between several software servers. The main servers are the feature server, which manages acoustic data, the mixture server which deals with models (storage, modification, tying of components, saving/reading...), and the statistics server which implements all the statistical computations (EM-based statistic estimations, likelihood computation, viterbi alignment, etc.) This architecture presents several advantages:

- Each server is accessible thanks to a very short list of high level functions and the low level functions such as memory management are usually hidden from the user;
- Each server is optimized and updated separately (thanks to this relative independence, developing new functionalities inside a new server is also very easy);
- Several instances of the same server could be launched at the same time (particularly useful for multi-stream or multimedia systems);
- The user-code presents the same structure, organized between the main servers, helping the source code development and understanding;
- Finally, the software architecture allows easily to distribute the servers on different threads or computers.

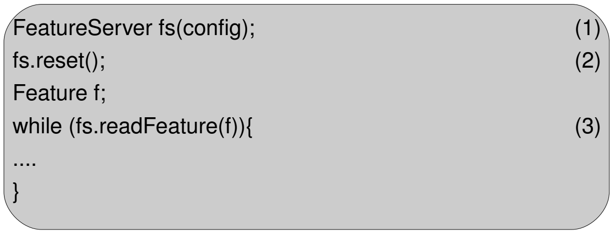
2.2.1. *Data reading and synchronization*

All the acoustic data management is delegated to the feature server. It relies on a two step procedure: feature server initialization and a reading call for accessing each feature vector. The reading of an acoustic stream is performed frame by frame following a feature loop as illustrated in figure 1. The user synchronizes the reading process thanks to the reading calls. Therefore, the same source code is employed for off-line (file based) processing and for on-line (open microphone) processing (the only difference occurs in the server configuration). Accessing at different time instants in the audio stream (go backward or forward, go to frame x) is also allowed. Memory management is achieved according to two simple rules:

- the user defines the size of the data buffer, in time, from unlimited to one frame. If the buffer shows a limited size, while the user requests a frame out of the buffer, the server will return a '*frame non available message*' (the same message is sent if no frame is available in an open microphone mode).
- the user defines the memory footprint for the data server. An integrated buffering service is provided for accessing very large datasets (several hundred hours of speech).

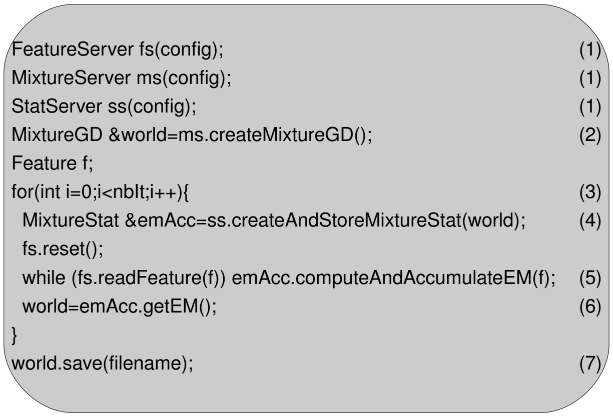
2.2.2. *EM/ML world model training*

For training a GMM world model, the user has to initialize three servers, the feature server for the data, the mixture server for managing the Gaussian components (and mixture models) and the statistics server for estimating the statistics. Figure 2 shows the skeleton of an example EM training procedure, beginning from scratch (the model is randomly initialized by default but the initialization could be easily modified).



```
FeatureServer fs(config);                                  (1)
fs.reset();                                                (2)
Feature f;
while (fs.readFeature(f)){                                 (3)
....
}
```

Figure 1: Acoustic data management in ALIZE. (1) Server init., (2) Server reset, (3) Reading call in the reading loop.



```
FeatureServer fs(config);                                                  (1)
MixtureServer ms(config);                                                  (1)
StatServer ss(config);                                                     (1)
MixtureGD &world=ms.createMixtureGD();                                     (2)
Feature f;
for(int i=0;i<nbIt;i++){                                                   (3)
  MixtureStat &emAcc=ss.createAndStoreMixtureStat(world);                  (4)
  fs.reset();
  while (fs.readFeature(f)) emAcc.computeAndAccumulateEM(f);               (5)
  world=emAcc.getEM();                                                     (6)
}
world.save(filename);                                                      (7)
```

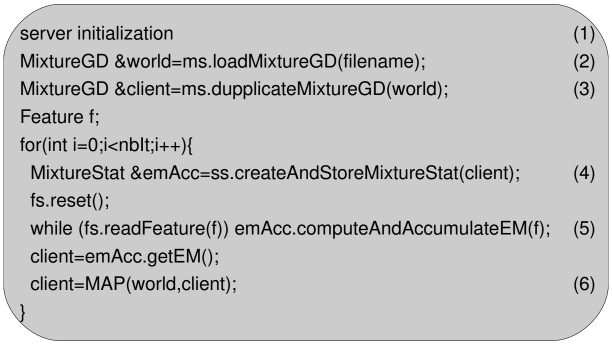
Figure 2: EM based world model training. (1) Servers init., (2) Init. the world model, (3) EM it. loop, (4) Reset the stat. accumulator, (5) Feature reading loop and statistcics accuulation, (6) Get the stat and copy it into the world model, (7) Save the model.

2.2.3. *EM/MAP speaker model estimation*

For deriving a speaker model from the world model using a MAP adaptation algorithm, the user builds exactly the same program as the previous one. Only two differences have to be highlighted: the client model is initialized as a copy of the world model and the final model (at each iteration) is the result of MAP(), a function involving both the world model (the a priori knowledge) and the statistics estimated on the client training data. Implementing some variants of the MAP algorithm will only take place in this MAP() function. This process is illustrated Figure 3.

2.2.4. *Score computation*

The score computation follows the same structure as the two previous programs. Figure 4 shows an example of the (log) likelihood computation for several client models, using a n-top Gaussian computing. For top Gaussian computing, the user needs only to set an optional flag (DETERMINE_TOP_DISTRIBS) during the corresponding call for memorizing the winning components and to set the flag to a different value (USES_TOP_DISTRIBS) for using it for the other calls. An implicit component tying is also implemented and helps to save both computational time and memory.



```
server initialization                                               (1)
MixtureGD &world=ms.loadMixtureGD(filename);                        (2)
MixtureGD &client=ms.dupplicateMixtureGD(world);                    (3)
Feature f;
for(int i=0;i<nbIt;i++){
  MixtureStat &emAcc=ss.createAndStoreMixtureStat(client);          (4)
  fs.reset();
  while (fs.readFeature(f)) emAcc.computeAndAccumulateEM(f);        (5)
  client=emAcc.getEM();
  client=MAP(world,client);                                         (6)
}
```

Figure 3: Client model estimation by MAP algorithm. (1) Servers initialization, (2) Load the world model, (3) Create the client model by world duplication, (4) Create the statistics accumulator, (5) Compute the stat on the client training data, (6) Estimate the resulting model as a function between the $apriori$ knowledge (world) and the current stat, copy it into client model.

2.2.5. *Discriminant classifiers*

Discriminant classifiers like the SVM were proposed during the past years in several works as in [10, 11, 12, 13]. These classifiers are usually applied to GMM supervectors. A GMM supervector is composed of the means of a classical GMM system, as initially proposed by [13]. Libsvm library[3] is used for the basic SVM functionalities.

2.2.6. *Session variability modeling*

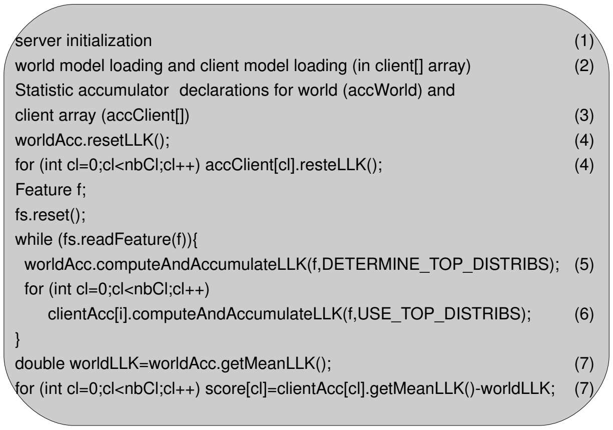
Some of the most important developments in ASV over recent years relate to strategies that address the ”mismatch factor”. ALIZE/SpkDet includes a set of functionalities related to the Factor Analysis proposed by [8] and the Nuisance Attribute Projection proposed by [7]. In these approaches the goal is to directly model the mismatch rather than to compensate for their effects as it was done with H-norm and T-norm. This involves estimating the variabilities from a large database in which each speaker is recorded in multiple sessions. The underlying hypothesis is that a low dimensional ”session variability” subspace exists with only limited overlap on speaker specific information. Both Factor Analysis and NAP were developed inside ALIZE/SpkDet, using the SVDLIBC[4] toolkit for the singular value decomposition.

2.2.7. *Factor Analysis implementation inside ALIZE/SpkDet*

This section describes more precisely the Factor Analysis approach and its basic implementation in ALIZE/SpkDet. A speaker model can be decomposed into three different components: a speaker-session-independent component, a speaker dependent component and a session dependent component. A GMM mean supervector is defined as the concatenation of the GMM component means. Let D be the dimension of the feature space, the dimension of a supervector mean is MD where M is the number of Gaussian in the GMM. A speaker and session independent model is usually estimated in speaker verification to represent the inverse hypothesis: the UBM model. Let this model being parameterized by $\theta = \{\mathbf{m}, \mathbf{\Sigma}, \alpha\}$. In the following, (h, s) will indicate the session h of the speaker s. The factor analysis model, in our case the eigenchannel MAP estimator, can be written as:

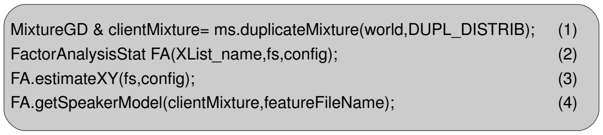
$$\mathbf{m}_{(h,s)} = \mathbf{m} + \mathbf{D}\mathbf{y}_s + \mathbf{U}\mathbf{x}_{(h,s)}, \quad (1)$$

where $\mathbf{m}_{(h,s)}$ is the session-speaker dependent supervector mean, $\mathbf{D}$ is $MD \times MD$ diagonal matrix, $\mathbf{y}_s$ the speaker vector (a MD vector), $\mathbf{U}$ is the session variability matrix of low rank R (a $MD \times R$ matrix) and $\mathbf{x}_{(h,s)}$ are the channel factors, a R vector (theoretically $\mathbf{x}_{(h,s)}$ does not dependent on s). Both $\mathbf{y}_s$ and $\mathbf{x}_{(h,s)}$ are normally distributed among $\mathcal{N}(0, I)$. $\mathbf{D}$ satisfies the following equation $\mathbf{I} = \tau\mathbf{D}^t\mathbf{\Sigma}^{-1}\mathbf{D}$ where τ is the *relevance factor* required in the standard MAP adaptation ($\mathbf{D}\mathbf{D}^t$ represents the *a priori* covariance matrix of $\mathbf{y}_s$).



```
server initialization                                                              (1)
world model loading and client model loading (in client[] array)                   (2)
Statistic accumulator  declarations for world (accWorld) and
client array (accClient[])                                                         (3)
worldAcc.resetLLK();                                                               (4)
for (int cl=0;cl<nbCl;cl++) accClient[cl].resteLLK();                              (4)
Feature f;
fs.reset();
while (fs.readFeature(f)){
  worldAcc.computeAndAccumulateLLK(f,DETERMINE_TOP_DISTRIBS);                      (5)
  for (int cl=0;cl<nbCl;cl++)
       clientAcc[i].computeAndAccumulateLLK(f,USE_TOP_DISTRIBS);                   (6)
}
double worldLLK=worldAcc.getMeanLLK();                                             (7)
for (int cl=0;cl<nbCl;cl++) score[cl]=clientAcc[cl].getMeanLLK()-worldLLK;         (7)
```

Figure 4: LLR score computation with n top Gaussian computing. (1) Servers intialization, (2) Load the models, (3) Create the statistic acc. (4) Reset the LLK accumulators for the world model and for each client model, (5) For a given frame and using the world model, determine the top Gaussian, memorize the individual component likelihood and compute/accumulate the world likelihood, (6) For the same frame, using the top components and the memorized values, compute and accumulate the likelihood for each client, (7) Get the mean log likelihood and compute the LLR per client.



```
MixtureGD & clientMixture= ms.duplicateMixture(world,DUPL_DISTRIB);      (1)
FactorAnalysisStat FA(XList_name,fs,config);                             (2)
FA.estimateXY(fs,config);                                                (3)
FA.getSpeakerModel(clientMixture,featureFileName);                       (4)
```

Figure 5: FA model training process. (1) Duplicate world model in clientMixture, (2) FA is the factor analysis object containing all decomposition parameters, (3) Estimate speakers and channels components given U, (4) Put the model $\mathbf{m} + \mathbf{D}\mathbf{y}_s$ in clientMixture, featureFileName is the name of one session.

The process of FA decomposition is illustrated Figure 5 for the model training step. A same process is used for the other steps like the llk computation.

3. Performance of the software

3.1. Experimental protocol

The male part of the NIST'05 primary task (1conv4w-1conv4w) is used for development (DevSet). For this condition, one side

[3] http://www.csie.ntu.edu.tw/∼cjlin/libsvm

[4] http://tedlab.mit.edu/∼dr/SVDLIBC/

of a 5-minute long conversation is available for testing and the same amount for training. All background training data, for the universal background model (UBM), T-norm [14], NAP, and FA come from the NIST'04 database. This procedure leaves the NIST'06 free for validation. The final comparisons are made on the NIST'06 core (required) condition which includes multiple languages (rather than the English only common condition).

Performance is assessed using DET plots and measured in terms of Equal Error Rate (EER) and the minimum of the decision cost (minDCF). The cost function is calculated according to the NIST criteria [15].

3.2. Description of the systems

This paragraph presents several systems in order to illustrate the large scale of techniques embedded in the software as well as the level of performance achieved using ALIZE/SpkDet (more details on the different systems are in [16]):

- GMM baseline (GMM). A classical GMM system is developed using ALIZE/SpeakerDet. The features are based on 19 linear filter-bank derived cepstra (computed using the open source SPro toolkit[5]). The feature vector is composed of 50 coefficients, 19 static coefficients, 19 delta and 11 delta-delta and the delta energy. A classical energy-based frame pruning system is applied before normalizing the recordings, file-by-file (cepstral mean subtraction and variance normalization). Feature mapping is also applied. The UBM model size is composed of 512 Gaussian components (with diagonal covariance matrices) and is involved in the speaker model estimation via a MAP adaptation procedure. This GMM baseline system is also used for the other systems;
- GMM Supervector Linear kernel (GSL). The GSL system uses a SVM classifier applied on GMM supervectors. The supervectors are taken directly from the GMM baseline system, giving in our case a vector size of 512*50;
- GMM Supervector Linear kernel + Nuisance Attribute Projection (GSL-NAP). GSL-NAP corresponds to a GSL system using the Nuisance Attribute Projection technique [7] in order to deal with intersession variabilities;
- Symmetrical latent Factor Analysis (SFA). Recently new approaches have been proposed by Kenny [8] with Factor Analysis (FA) in a generative framework. This approach was implemented into our software using an original symmetrical approach presented in [17].

3.3. Performance on NIST SRE 2006 database

	male 05		male 06		all 06	
	DCF	EER	DCF	EER	DCF	EER
GMM	3.37	8.67	3.94	8.47	4.04	9.14
GSL	2.79	8.02	3.37	6.88	3.35	7.20
GSL-NAP	**1.62**	5.28	2.07	4.33	2.26	5.02
SFA	1.94	**4.38**	2.17	4.78	-	-

Table 1: Performance in EER(%) and minDCF(x100) for the GMM, GSL, GSL-NAP and latent factor analysis GMM (SFA) on 05 development set, male'06 and male and female combined '06 validation sets

Performance is presented in Table 1 for all the independent systems, for the development set and for the different validation sets (issued from NIST-SRE 2006). The results confirm that NAP and SFA techniques bring a significant improvement compared to the classical GMM system. Compared to the official results of NIST SRE 2006, available in [18], it is clear that ALIZE/SpkDet systems are among the few systems able to achieve an EER in the region of 5%.

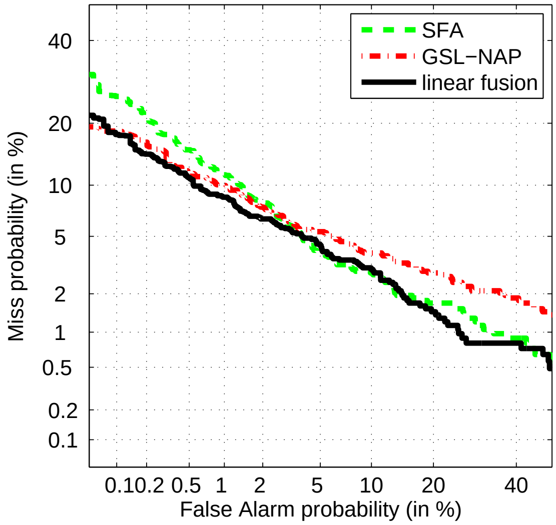


Figure 6: Det plots for GSL-NAP, SFA and their unweighted linear fusion, rank=40, T-norm (devSet)

Figure 6 shows the results of an unweighted linear fusion between the 2 best systems namely SFA and GSL-NAP[5]. The improvement shows their complementarity, even if the two systems share the same acoustical representation, the same GMM parameters (UBM) and the same session variability parameters.

4. Unsupervised adaptation

A classical solution in order to improve the performance of a speaker recognition system is to increase the amount of information used to train the client model. However, this solution depends on the availability of such training data.
Unsupervised adaptation of the client models is a good solution to this data availability problem. In unsupervised adaptation, the client model is updated online, using data gathered from the test trials. The different approaches to unsupervised adaptation proposed in the literature rely mainly on a decision step to decide if a test trial belongs to the claimed speaker identity [19, 20, 21, 22, 23]. If the test trial is considered as client, it is used to either retrain the corresponding client model or to adapt it. The main drawback of such techniques remains the difficulty to set a decision threshold for selecting the trials: the performance gain relies on the number of client test trials detected, a false acceptance (an impostor trial is accepted as a client one) degrades the speaker model.
The work presented in this Section was done in collaboration with Thales Communications and is presented in details in [24]. The proposed approach addresses the threshold based decision step problem as no hard decision is used. The speaker model adaptation is applied to each test trial, even if it does not belong to the client but to an impostor. This "continuous adaptation"

[5] Tnorm is applied on both system scores before the fusion

method relies on a confidence measure on the tying between the test trial and the target speaker, used to weight each test information during the speaker model adaptation process.

4.1. Confidence measure estimation

The confidence measure is the estimation of the *a posteriori* probability of a test trial belonging to a given target speaker. This *a posteriori* probability is computed using two score models, one for the client scores and one for the impostor scores. Each score distribution is modelled by a 12 component GMM learned on a development set. The confidence measure is then computed using the WMAP approach [22, 25]. To avoid the problem of the re-estimation of the WMAP function after each adaptation step, we use only the initial target model, learned on a single session recording, to compute the score of the test trials.

4.2. Proposed adaptation function

The proposed adaptation function relies on the classical MAP algorithm [26], where only the mean parameters are updated. The empirical statistics are gathered from all the available data using the EM algorithm (initialized with the background model and maximizing the ML criterion). The statistics are then combined using the following rules:

- The statistics gathered from the initial voice excerpt used to train the target speaker model is associated with a confidence measure equal to 1;
- The statistics gathered from the different test trials are associated with the corresponding confidence measure;
- The empirical means and the corresponding occupancies are computed for each Gaussian component of the GMM, using all the EM statistics weighted by the corresponding confidence measures.

Finally, the adapted means (μ^i_{map}) for each Gaussian component (i) are computed using the background means (μ^i_{ubm}), the empirical means (μ^i_{emp}) and the occupancy values (n_i) using the classical MAP formula.

4.3. Experiments and results

All the experiments presented here are performed based upon the NIST'05 and '06 databases, all trials (det 1), 1conv-4w 1conv-4w, restricted to male speakers only. The baseline GMM system is the one described in Section 3. In addition, a version of the Latent Factor Analysis (LFA) is used at the feature level: in this case, the channel compensation is applied to each feature sets instead of the feature mapping. Figure 7 presents the results for the adapted system and the baseline on the NIST'05 database, for feature mapping and the LFA channel compensation techniques.

The results demonstrate the potential of the proposed method as it reaches a significant 27% DCF relative gain (and 37% in terms of EER) with the feature mapping (FM). When the LFA-normalized features are used, the DCF gain is about 20% (and 12.5% for the EER). The gain is smaller in this case, as expected because LFA is known to perform better channel compensation than FM, reducing the influence of the unsupervised adaptation on the channel effects.

In Figure 8, we attempt to analyze more precisely the behavior of our method. This figure shows the performance in terms of min DCF of the adapted system for each newly added test trial (n denotes the number of test trials added to the system)[6].

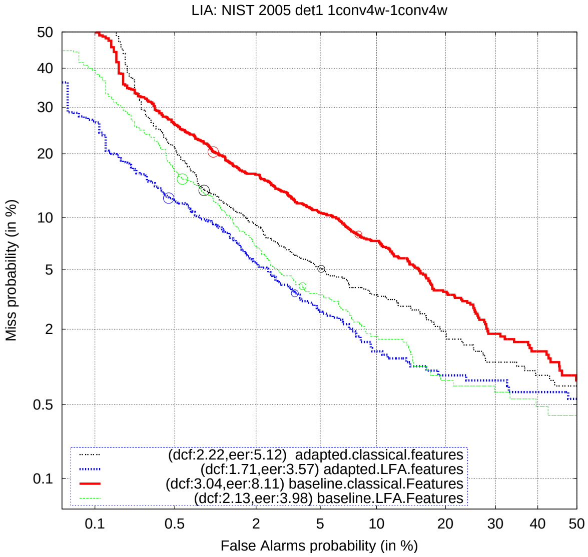


Figure 7: *Results for the adapted/baseline systems, NIST'05*

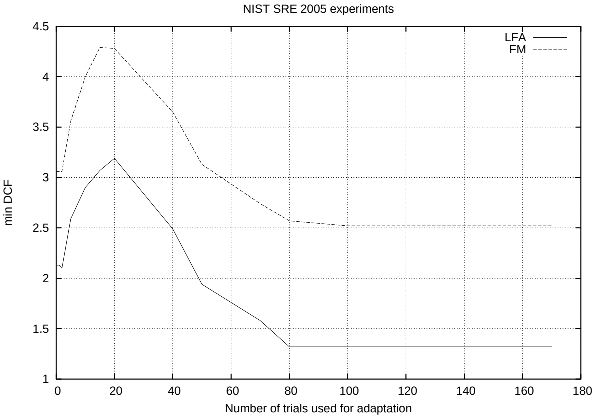


Figure 8: Step by step DCF for the two different sets of features on NIST'05

Results for the adapted system and the baseline are provided in Figure 9, for the NIST'06 database and using both FM and LFA. The results are disappointing compared to those of NIST'05. On this database, the unsupervised adaptation method introduces a significant loss. Different factors could explain this unexpected result with the most relevant thought to be:

- The percentage of impostor trials versus target trials differs between the databases. Whilst 9.0% of target tests are proposed in NIST'05, only 7.1% are present in NIST'06. Even if the difference seems quite small, it corresponds to 21% less target data for the 2006 database. Moreover, the number of test trials by target speaker is also smaller for NIST'06.
- When looking at the target and impostor score distribu-

[6] When the target models are updated using a new test trial, the entire test is recomputed, which differs from the NIST protocol where only the current trial and the next trials scores are computed using the new models.

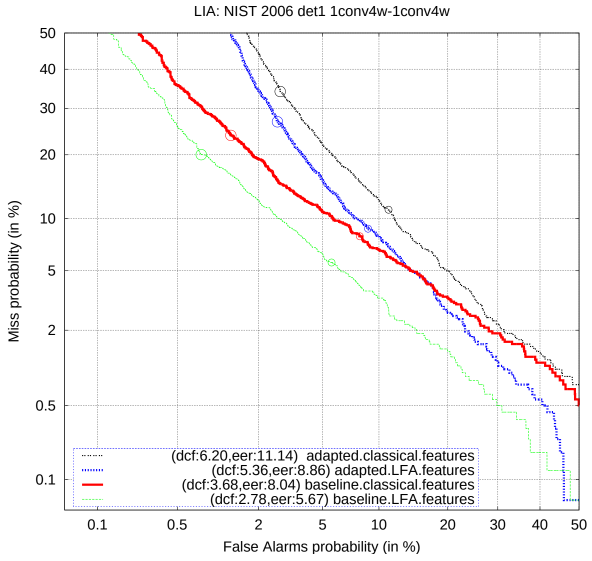


Figure 9: *Results for the adapted/baseline systems, NIST'06*

tions of the baseline system we observe that a large part of the errors comes from a small percentage of the impostor trials which obtained a very high score. This phenomena is significantly higher for the NIST'06 than for the NIST'05 database. It shows the obvious dependence between the baseline system and the adaptation process behavior.

5. Other applications

5.1. Embedded password dependent speaker verification

For embedded speaker recognition, the main issues are usually the memory and computational constraints. But realistic embedded applications generally require short utterances, few training data and very noisy environments, giving degraded speech quality. In order to deal with these application constraints, we moved from a text-independent speaker recognition scenario to a text-dependent authentication using passwords that were chosen by the clients. This section presents the LIA_EBD package based on ALIZE and dedicated to such embedded applications.

5.1.1. Description

Our approach merges a state of the art GMM/UBM system and a semi-continuous HMM architecture, following the ideas developed in [27] with a DTW algorithm. Hierarchical three-step models are created where a lower level model takes some of the parameters from the immediate upper level model. The top model is a classical UBM while the second one is a quite classical text-independent speaker model. The latter is adapted from the UBM, updating only the mean parameters using a MAP procedure. In LIA_EBD, only few UBM components are adapted and saved, in order to save both memory and computation time (non adapted mean parameters as well as weight and variance parameters are taken from the UBM model). In the lower level, a semi-continuous HMM (SCHMM) is built in order to take into account both phonetic and duration information. Each state of the SCHMM is linked to a GMM distribution. The SCHMM state-dependent distributions are adapted from the corresponding speaker model, moving only the weights. As for the second level model, only a few weights are adapted, the other parameters are taken from the upper level models (the corresponding speaker model and the UBM).

5.1.2. Database and Experiments

The system, currently in development, is evaluated using the Valid database [28] (as the final objective is to build a bimodal system: audio+video). This audio-video Database consists of five recording sessions composed of two sentences (the phonetic content is identical to that of the XM2VTS database[7]). Four of these sessions were recorded in variable and uncontrolled environments, the fifth session in a clean environment. 76 male speakers are used for the experiments.

	GMM	LIA_EBD
Passwords	1,81	3,09
Passwords & Wrong Sentences	2,50	3,00
Wrong Sentences	1,17	0,19

Table 2: Performance in EER (%) for the GMM and LIA_EBD on the Valid database with different test sets, male only. The experiments are performed by computing 100 client tests, 3275 impostor tests with the same password pronounced by the impostor cohort (the tests are referred to as *Passwords*) and 3275 tests with the second sentence pronounced by the impostor cohort (referred to as *Wrong Sentences*)

5.1.3. Comments

The results are promising as the system seems able to exploit both speaker dependent information and password dependent information (the EER decreases from 3.09% to 0.19% when the impostors pronounce a wrong password for the embedded system. The GMM system stays at 1.17% with the wrong sentences). The level of resources (memory and computation) remains very small for the LIA_EBD system. Of course, the presented results are preliminary as the database is very small (76 speakers, two sentences) but only a few system parameters have been optimized and much room remains for improvements.

5.2. Speaker diarization

The design of efficient indexing algorithms to facilitate the retrieval of relevant information is vital to provide easy access to multimedia documents. Acoustic-based information such as speaker turns, the number of speakers, speaker gender, speaker identity, other sounds (music, laughs) as well as speech bandwidth or characteristics (studio quality or telephone speech, clean speech or speech over music) has become mandatory for indexing algorithms. This section is dedicated to speaker-related tasks, also denoted speaker diarization in the NIST-RT evaluation terminology. The speaker diarization task consists in segmenting a conversation involving multiple speakers into homogeneous parts which contain the voice of only one speaker, and grouping together all the segments that correspond to the same speaker. The first part of the process is also called speaker

[7] www.ee.surrey.ac.uk/Research/VSSP/xm2vtsdb/

Corpus	Teleph.	BN	Meeting	Meeting
Year		News	head mic.	table mic.
2002	5.7	30.3	34.7	36.9
spring 2003	X	12.9	X	X
spring 2004	X	X	X	22.4

Table 3: 2002-2004 NIST evaluations results in diarization error rates for various corpora

change detection while the second one is known as the clustering process. Generally, no prior information is available regarding the number of speakers involved or their identities. Estimating the number of speakers is one of the main difficulties for the speaker diarization task. An original approach, denoted the *integrated* approach, in which all the steps involved in speaker diarization are performed simultaneously, was developed by the LIA, using ALIZE/SpkDet software. This work was done in the framework of the ELISA consortium and both classical *step-by-step* and *integrated* diarization approaches are proposed in the software, with a lot of functionalities in order to deal with different natures of audio files (conversational recordings, broadcast news or meeting information). This work is detailed in [29]. Table 3 presents a summary of the results obtained between 2002 and 2004 on different types of data, in the framework of the NIST evaluations (mainly on Rich Transcription). More recent results on meeting data were reported in [30][31].

5.3. Pathological voice assessment

In the medical domain, assessment of the pathological voice quality is a sensitive topic, involving multi-disciplinary domains. This task is usually realized perceputally by a jury composed of experts. The work presented in this paragraph proposes an automatic system derived from the ALIZE/SpkDet automatic speaker recognition technology to assess pathological voices, and more precisely on phonation disorders due to dysphonia.

Basically, the task consists in classifying a speech recording according to the G parameter of the Hirano's GRBAS[8] scale[32]), for which a normal voice is rated as grade 0, a slight dysphonia as 1, a moderate dysphonia as 2 and, finally, a severe dysphonia as 3. Table 4 presents the general classification results (given that a strict classification protocol was used, with a clear separation between training and testing set). The obtained results demonstrate the possibilities of this approach, even if the task is very difficult, due to its medical nature, the variability of the data (and the small number of data available) and the difficulty in obtaining a ground truth (done here by an expert jury).

	G=0	G=1	G=2	G=3
Speakers with a G=0	**19**	1	0	0
Speakers with a G=1	2	**12**	4	2
Speakers with a G=2	2	5	**11**	2
Speakers with a G=3	0	1	4	**15**

Table 4: Pathological voice assessment results. The rows separate the voice depending on the G values given by the expert jury; the columns represent the G values determined by the system

6. Conclusion and future

This paper presents the ALIZE/SpkDet open-source software. The ALIZE project began in 2003 within the ELISA consortium and under the direction of the LIA speaker recognition group. The basics of the project (open source, easy to understand, assessment during the NIST international evaluation campaigns) remain unchanged over the years.

The latest discriminant approaches and channel compensation functionalities have been added to the toolkit for the past year leading to state-of-the-art speaker recognition performance. The software is now regularly used by about 20 laboratories in the world, and support is provided to all the developers.

The software is dedicated to text-independent speaker recognition but it can also be used for a large range of applications. In addition to the applications presented in this paper, ALIZE/SpkDet software is also used for several other areas of research within the LIA laboratory (face and fingerprint recognition, emotion detection, topic detection for text processing, etc.) or by other users (for example signature/multimodal recognition [33]).

The main part of the development was done by the LIA team (thanks to several French Research Ministry funds) until 2005/2006. Recently, several external (not from the LIA) contributions have been proposed. In order to respond to the requests of the ALIZE/SpkDet developer community, the project has been migrated from a quite closed software engineering platform to a classical open-source open architecture. An important effort will also be dedicated to the documentation.

The main challenge to the ALIZE project is not at the technical level but is to propose a new concept inside the evaluation paradigm: the participation to the international evaluation campaign is viewed as a competition and quite in opposition to the open source idea. Regarding this challenge, it seems that the ALIZE project is a success. In order to emphasize this aspect, we wish to propose inside the ALIZE project some physical or virtual evaluation campaign specific tutorials, brainstorming and practical sessions.

7. References

[1] A. Martin and M. Przybocki, "The NIST speaker recognition evaluation series, National Institute of Standards and Technology's website, http://www.nist.gov/speech/tests/spk," .

[2] *ALIZE: open tool for speaker recognition*, Software available at http://www.lia.univ-avignon.fr/heberges/ALIZE/.

[3] J.-F. Bonastre, F. Wils, and S. Meignier, "Alize, a free toolkit for speaker recognition," in *ICASSP*, 2005.

[4] The ELISA consortium, "The ELISA consortium. the ELISA systems for the NIST'99 evaluation in speaker detection and tracking," *Digital Signal Processing*, vol. 10, pp. 143–153, 2000.

[5] G. Gravier, *SPro: speech signal processing toolkit*, Software available at http://gforge.inria.fr/projects/spro.

[6] W. Campbell, D. E. Sturim, D. Reynolds, and A. Solomonoff, "SVM based speaker verification using a GMM supervector kernel and NAP variability compensation," in *ICASSP*, 2006.

[8] The GRBAS scale is composed of 5 parameters: G-Global grade, R-Roughness, B-Breathiness, A-Astheny, S-Strain. The G parameter is often used for perceptual evaluation since it is one of the most reliable (followed by R and B) in terms of assessment variability.

[7] A. Solomonoff, W. M. Campbell, and I. Boardman, “Advances in channel compensation for SVM speaker recognition,” in *ICASSP*, 2005.

[8] P. Kenny and P. Demouchel, “Eigenvoices modeling with sparse training data,” *IEEE trans*, vol. 13, pp. 345–354, 2005.

[9] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, “Factor analysis simplified,” in *ICASSP*, 2005.

[10] J. Mariéthoz and S. Bengio, “A kernel trick for sequences applied to text-independent speaker verification systems,” IDIAP-RR 77, IDIAP, 2005.

[11] V. Wan, *Speaker Verification Using Support Vector Machines*, Ph.D. thesis, University of Sheffield, 2003.

[12] W.M. Campbell, J.P. Campbell, D.A. Reynolds, E. Singer, and P.A. Torres-Carrasquillo, “Support vector machines for speaker and language recognition,” *Computer Speech and Language*, vol. 20, pp. 210–229, 2006.

[13] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, “Support vector machines using GMM supervectors for speaker verification,” *IEEE Signal Processing Letters*, vol. 13, May 2006.

[14] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas, “Score normalization for text-independent speaker verification systems,” *Digital Signal Processing*, vol. 10, pp. 42–54, 2000.

[15] A. Martin and M. Przybocki, “NIST speaker recognition evaluation chronicles,” in *Odyssey*, 2004.

[16] B. Fauve, D. Matrouf, N. Scheffer, J.F. bonastre, and J. Mason, “State-of-the-art performance in text independent speaker verification through open source software,” *IEEE Transactions on Audio, Speech and Language Processing, to be published in September*, 2007.

[17] D. Matrouf, N. Scheffer, B. Fauve, and J.-F. Bonastre, “A straightforward and efficient implementation of the factor analysis model for speaker verification,” Interspeech, Antwerp, Belgium.

[18] M. Przybocki, A. Martin, and A. Le, “NIST speaker recognition evaluation chronicles - part 2,” Odyssey, 2006.

[19] C. Barras, S. Meignier, and J. L. Gauvain, “Unsupervised online adaptation for speaker verification over the telephone,” in *Odyssey , Toledo, Spain*, 2004.

[20] D. van Leeuwen, “Speaker adaptation in the nist speaker recognition evaluation 2004,” in *Interspeech, Lisbon, Portugal*, 2004.

[21] E.G. Hansen, R.E. Slyh, and T.R. Anderson, “Supervised and unsupervised speaker adaptation in the nist 2005 speaker recognition evaluation,” in *Odyssey, Puerto Rico, USA*, 2004.

[22] A. Preti and F. Capman J.-F. Bonastre, “A continuous unsupervised adaptation method for speaker verification,” in *CIS2E*, 2006.

[23] L.P. Heck and N. Mirghafori, “Unsupervised on-line adaptation in speaker verification: Confidence-based updates and improved parameter estimation,” in *Proc. Adaptation in Speech Recognition, Sophia Antipolis, France*, 2001.

[24] A. Preti, J.-F. Bonastre, D. Matrouf, F. Capman, and B. Ravera, “Confidence measure based unsupervised target model adaptation for speaker verification,” in *Interspeech, Antwerp, Belgium*, 2007.

[25] C. Fredouille, J-F. Bonastre, and T. Merlin, “Bayesian approach based-decision in speaker verification,” in *Odyssey, Crete, Grece*, 2001.

[26] F. Bimbot, J.-F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-Garcia, D. Petrovska-Delacretaz, and D. Reynolds, “A tutorial on text-independent speaker verification,” *EURASIP Journal on Applied Signal Processing*, vol. 4, pp. 430–451, 2004.

[27] J.F. Bonastre, P. Morin, and J.-C. Junca, “Gaussian dynamic warping (gdw) method applied to text-dependent speaker detection and verification,” in *Eurospeech, Geneve, Switzerland*, 2003.

[28] N. A. Fox, B. A. O’Mullane, and R. B. Reilly, “The realistic multi-modal valid database and visual speaker identification comparison experiments,” in *5th International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA-2005), New York, July 20-22*, 2005.

[29] S. Meignier, D. Moraru, C. Fredouille, J.-F. Bonastre, and L. Besacier, “Step-by-step and integrated approaches in broadcast news speaker diarization,” *Computer speech and Language*, vol. 20/2-3, April/July, pp. 303–330, 2006.

[30] C. Fredouille and G. Senay, “Technical improvements of the e-hmm based speaker diarization system for meeting records,” in *Machine Learning for Multimodal Interaction: 3rd International Workshop, MLMI 2006. Springer Lecture Notes in Computer Science Series*, 2006.

[31] C. Fredouille and Nicholas Evans, “The influence of speech activity detection and overlap on speaker diarization for meeting room recordings,” in *Interspeech, Antwerp, Belgium*, 2007.

[32] M. Hirano, “Psycho-acoustic evaluation of voice: Grbas scale for evaluating the hoarse voice,” in *Clinical Examination of voice, Springer Verlag*, 1981.

[33] K. Kryszczuk, J. Richiardi, P. Prodanov, and A. Drygajlo, “Reliability-based decision fusion in multimodal biometric verification systems,” *IEEE trans*, 2007.