Improving EigenVoices-based techniques and *SMLLR* for Speaker Adaptation by combining *EV* and *SMLLR* techniques or using Genetic Algorithms

Fabrice Lauri

Speech Group
LORIA - INRIA
B.P. 239 - 54506 Vandœuvre-Les-Nancy
France
{lauri,illina,fohr}@loria.fr

Abstract

This paper constitutes a study of several classical and original methods for a speaker adaptation of the acoustic hidden Markov models of an automatic speech recognition system (ASRS).

Most of today's real applications require that the speaker adaptation process continuously improves the performance of the underlying ASRS, as more utterances are pronounced by a new speaker. The first part of this article is dedicated to this problem. We begin by introducing the *Structural EigenVoices* approach (*SEV*). Compared to *EigenVoices* (*EV*), *SEV* improves the performance of an ASRS with more sentences, well beyond the point where the *EV* system has reached its limit. We then describe four methods that combine the advantages of *Structural Maximum Likelihood Linear Regression* (*SMLLR*) and *EigenVoices*-based techniques (*EV* or *SEV*). We show experimentally that one of them, *SEV*→*SMLLR*, can improve the performance of an ASRS at least as significantly as *SMLLR*, *EV* and *SEV*, irrespective of the amount of adaptation utterances used.

The second part of our work is focused on the use of genetic algorithms for rapidly adapting acoustic models. Whereas all of the standard adaptation methods (e.g. *SMLLR*, *SMAP*, *EV*, etc.) are based on the *E.-M.* procedure and thus provide a single local optimal solution, genetic algorithms are theoretically able to provide several global optimal solutions. We experimentally show that: (1) genetic algorithms and *EV* both equivalently improve the performance of an ASRS, and (2) combining genetic algorithms and *EV* further improves the performance of an ASRS.

Key words: Speaker adaptation, Structural *EigenVoices*, genetic algorithms, Hidden Markov Models.

Technical report – LORIA-INRIA Lorraine *February 2008*

1 Introduction

Reducing acoustic mismatches due to speaker variability between the training conditions and the testing conditions is a major problem in automatic speech recognition systems (ASRS). This problem is particularly difficult for rapid adaptation, i.e. when the available amount of adaptation data is low. Among the speaker adaptation techniques which tackle this problem, *SMLLR*[1] [12], Structural Maximum A Posteriori (*SMAP*) [19,20], *EigenVoices*-based techniques [7–9,14–16] and methods combining *SMLLR* and *EigenVoices* [1–3,10,21] have been shown to rapidly improve the performance of an ASRS.

Whereas *SMLLR*[2] becomes efficient only after a certain number of adaptation utterances have been pronounced, *EigenVoices* can improve the performance of an ASRS even if only a single adaptation utterance (less than ten seconds of speech) has been produced. This outstanding result of *EigenVoices* can be explained by the fact that unlike *SMLLR*, it makes use of *a priori* information about the inter-speaker variations, which enables the system to estimate far fewer required parameters than *SMLLR*. However, the performance of the *EigenVoices* technique quickly saturates as more adaptation data becomes available. These conclusions can be drawn in batch mode as well as in incremental mode.

In *SMAP*, the gaussians are organized in a tree whose structure is similar to the one used by *SMLLR*, except that each leaf node contains only one gaussian. Each node in that tree represents a cluster of gaussians which shares a common bias vector representing the mismatch between the training conditions and the conditions of use. All the bias vectors are estimated using the MAP criterion. Then, the bias vector of each leaf node is applied to the corresponding gaussian mean. The MAP criterion combines an estimation made on the basis of the adaptation data with an estimation from prior knowledge about the gaussian means. Thanks to its highly constrained structure, *SMAP* is able to deliver good results when only a limited amount of adaptation data is available, while still preserving good asymptotic properties (i.e. converges to the performance of the SD models) as the size of adaptation data increases. Nevertheless, for some tasks in supervised batch mode, *SMAP* is not as efficient as *EV* when few adaptation utterances are available, and it is only as

[1] In this article we will distinguish Maximum Likelihood Linear Regression (MLLR) [13] from Structural Maximum Likelihood Linear Regression (*SMLLR*) [12]. *MLLR* uses gaussian classes designed by hand to adapt the acoustic models whereas Structural *MLLR* is able to build a gaussian tree from which gaussian classes are determined automatically in function of the amount of adaptation data.

[2] As *SMLLR* and *EigenVoices* are key techniques in this article, they will be described in details in section 2.1 and 2.2.1, respectively.

efficient as *SMLLR* when enough adaptation data becomes available (see appendix A). Moreover, *SMAP* gives worse results than *SMLLR* in incremental mode, irrespective of the amount of adaptation data (see appendix A).

Based on these observations, several recent speaker adaptation techniques have either tried to improve the *EigenVoices* or to integrate the advantages of *SMLLR* and *EigenVoices* scheme to rapidly adapt the speaker-independent (SI) models to a new speaker.

Kernel *EigenVoices* [14–16] is an extension of *EigenVoices* whose eigenvectors are computed using non-linear kernel principal component analysis (PCA), whereas the regular version of *EigenVoices* uses linear PCA to obtain the eigenvectors.

Botterweck [2] extends the standard *EigenVoices* technique to large-vocabulary continuous speech recognition problems by training the acoustic models of each speaker with the help of *SMLLR* and *MAP*.

In the Eigen-MLLR approaches [3,21], the eigenspace representing the inter-speaker variations is built using *Principal Component Analysis (PCA)* from the parameters of the linear regression matrices, which are obtained for each training speaker by the use of *SMLLR*. The regression matrices which were computed for the adapted models of the new speaker are then constrained to be located in the space spanned by the first eigen-matrices. These methods can thus solve the problem of huge memory requirements of the *EigenVoices* technique, since the number of parameters of the regression matrices is much lower than the parameters of a speaker-independent system (SIS).

X. L. Aubert [1] applied the concept of Eigen-MLLRs to unsupervised speaker enrollment[3] in a large vocabulary recognition system. His work presents two ways of dealing with several *MLLR* transformations, either by joining the individual transforms with respect to the Eigen-MLLR vector space, or by keeping them separated.

All of the above mentioned approaches have shown that the integration of *SMLLR* and *EigenVoices* adaptation is fairly robust and reliable. Nevertheless, all of them are performed in batch mode with few adaptation data. Yet, most of the real applications require that adaptation techniques are able to improve the performance of the underlying ASRS continuously, as more utterances are pronounced by a new speaker.

We propose in this article to analyze the most widely used adaptation techniques, namely *SMLLR* and *EV*, as well as several original approaches capable of adapting the acoustic models of an ASRS. Each technique will be evaluated on the same speech corpus, and with the same experimental conditions.

The first part of this article deals with some of the investigations we carried out to study how the performance obtained by techniques combining *Eigen-*

[3] in the sense that the first decoded words pronounced by a new speaker are employed to adapt the speaker-independent HMM models.

Voices and *SMLLR* evolves as more adaptation data becomes available. In [10], we presented four different possible methods which integrate the concepts of both *SMLLR* and *EigenVoices* for speaker adaptation in supervised batch mode and in unsupervised incremental mode. In this article, we show new results obtained for each one of these techniques.
The second part of this article is dedicated to the problem of rapid speaker adaptation of acoustic models, which means the case where only one or up to five utterances are available. To tackle this problem, we proposed to use genetic algorithms in [11]. New results related to this technique are also presented in this paper.

The remainder of this paper is organized as follows. The Structural *MLLR* algorithm is introduced in Section 2.1. Section 2.2 reviews the regular version of the *EigenVoices* algorithm and presents our structural version: Structural *EigenVoices*. Section 2.3 describes four different methods which combine either *EigenVoices* and *SMLLR* techniques, or Structural *EigenVoices* and *SMLLR* techniques to improve the performance of a speaker-independent system. Section 3 proposes two adaptation techniques based on genetic algorithms. Section 4 evaluates all of the methods mentioned above, using speech data from the *Resource Management (RM)* corpus. A detailed analysis of their complexity and memory requirements is given in section 5. Finally, concluding remarks and future research issues are given in Section 6.

2 Combining EigenVoices and Structural MLLR

Several methods for improving the performance of *EigenVoices* and *SMLLR*, irrespective of the available amount of adaptation data, are presented in this section.
For many tasks, *EV* performs better than *SMLLR* when only a few seconds (e.g. 5 or 10 sec.) of adaptation data is available, while *SMLLR* is more successful in case of large data sets [10,18]. In this case, it is well justified to combine them to obtain a technique which could be efficient for any number of adaptation utterances.
SMLLR and *EV* will be introduced in sections 2.1 and 2.2.1, respectively. The Structural *EigenVoices* technique will be presented in section 2.2.2, followed by the four methods integrating the advantages of both *SMLLR* and *EigenVoices*.

2.1 Structural MLLR

The Structural version of *MLLR* [12] is able to adjust the number of linear regression matrices $\{W_1, W_2, \ldots, W_{N_R}\}$ that will be applied to the gaussian mean vectors according to the available amount of adaptation data. This flexibility is made possible by the use of a binary tree structure which clusters the gaussians of the SI-models. Each tree node G_i, together with a transformation matrix W_i, is called a regression class. Let $\gamma_t(g_{(i,m)})$ be the occupation probability of the gaussian m of the regression class i at time t, and $S_i = \sum_{m=1}^{M_i} \sum_{t=1}^{T} \gamma_t(g_{(i,m)})$ be the number of observations associated with the set of M_i gaussians belonging to the regression class i. For each leaf node which possesses more than θ_{SMLLR} observations, i.e. $S_i \geq \theta_{SMLLR}$, the associated matrix is estimated by using the set of gaussians of the node. For the leaf nodes that don't have enough observations, the associated matrix is estimated by using the gaussians of the closest parent node which has enough observations. This regression classes generation process can update the parameters of the gaussians by using robust estimated transformation matrices.
The adapted gaussian mean vector $\hat{\mu}_{(i,m)}$ of the gaussian m and the class i is then obtained by the following equation:

$$\hat{\mu}_{(i,m)} = W_i \; \xi_{(i,m)} \tag{1}$$

where $\xi_{(i,m)}$ is the extended vector of the gaussian mean $\mu_{(i,m)}$ such that $\xi_{(i,m)} = [1 \; \mu^{'}_{(i,m)}]^{'}$.

2.2 EigenVoices

The *EigenVoices* technique makes use of *a priori* information about the inter-speaker variations in order to constrain the adapted models such that they are located in a dimensionality reduced speaker-space. The speaker-space is obtained by applying a dimensionality reduction technique [4] to a set of N_S supervectors of dimension D, extracted from N_S speaker-dependent (SD) models. Each supervector is made up of the parameters that have to be adapted. In the case of gaussians, the supervector consists of the concatenation of the gaussian mean vectors of each model currently present in the speaker-dependent system. If N_G is the number of gaussians of a speaker-dependent system, and N_D is the dimension of each such a gaussian mean vector, then the supervector will be of dimension $D = N_G \times N_D$.
This off-line step yields N_S supervectors of dimension D, called the eigenvectors. The reduced speaker-space is made up of the first K eigenvectors

[4] Principal Component Analysis (PCA) for example

$\{e_1, e_2, \ldots, e_K\}$ with $K < N_S << D$. Together with an origin e_0 [5], these K *EigenVoices*, which capture most of the variation of the training data, span the reduced speaker-space of dimension K.

2.2.1 Regular version

In the regular version of the *EigenVoices* technique [8,9], a new speaker is first located by a vector of $K+1$ weights $\{w_0, w_1, \ldots, w_K\}$ in the reduced speaker-space. All of the gaussian mean vectors $\hat{\mu}_i$ of the adapted models are then updated using the equation

$$\hat{\mu}_i = \sum_{k=0}^{K} w_k \, e_k^{(i)} \qquad \forall i = 1, 2, \ldots, N_G \tag{2}$$

where N_G is the total number of gaussians of the speaker-adapted system, and $e_k^{(i)}$ is the N_D-dimensional sub-vector of the k-th eigenvector related to the i-th gaussian.
The $K+1$ weights are generally estimated using *Maximum Likelihood Eigen-Decomposition (MLED)* [18] in order to maximize the likelihood of the adaptation data. The other HMM parameters remain identical to the corresponding SI-model parameters.

2.2.2 Structural version

The structural version of the *EigenVoices* approach (SEV) [10] borrows the flexibility of *SMLLR* by integrating a gaussian binary tree structure to adjust the adaptation parameters (the weights) with the available amount of adaptation data. Its flexibility enables *SEV* to avoid the early saturation encountered by its regular counterpart when more adaptation data is available.
As illustrated in figure 1, a class i in *SEV* represents a tree node G_i with a set w_i of $K+1$ weights such that $w_i = \{w_{(i,0)}, w_{(i,1)}, \ldots, w_{(i,K)}\}$. These weights will only be applied to the corresponding gaussian mean vectors $\mu_{(i,m)}$ that belong to the node G_i, using the following equation

$$\hat{\mu}_{(i,m)} = \sum_{k=0}^{K} w_{(i,k)} \, e_k^{(i,m)} \qquad \forall i = 1, 2, \ldots, N \text{ and } \forall m = 1, 2, \ldots, N_i \tag{3}$$

where N is the total number of classes, and N_i the number of gaussians grouped together within the class i; $e_k^{(i,m)}$ is the N_D-dimensional sub-vector

[5] e_0 can be the average supervector of all of the SD models or the supervector s_{SIS} extracted from the SI models.

related to the m-th gaussian of the class i in the k-th eigenvector.
The process for generating the classes in *SEV* is the same process as the one in *SMLLR*: the weights of a class are only estimated if more than θ_{SEV} observations have been gathered for that particular class.
The number of weights associated with a class in Structural *EigenVoices* is lower than the number of parameters of a linear regression matrix associated with a class in *SMLLR*. As the larger the amount of adaptation data, the more adaptation parameters can be reliably estimated, the value of θ_{SEV} will always be lower than the value of θ_{SMLLR}. For this reason, *SEV* is able to adapt more gaussian means than *SMLLR* for the same amount of adaptation data.
We conducted preliminary experiments using *SEV* with different values of θ_{SEV}. We observed that it yielded only poor results when few adaptation utterances were used, mainly because the weights could not be estimated precisely enough. We came to the conclusion that a critical amount of adaptation data should be gathered before carrying out the adaptation process based on the tree of classes. On the one hand, this implies that the classes generation process in *SEV* is triggered only if the total number of observations is greater than some predetermined threshold α_{SEV}. On the other hand, an adaptation with *SEV* will be equivalent to an adaptation with *EV*if the total number of observations is lower than α_{SEV}.

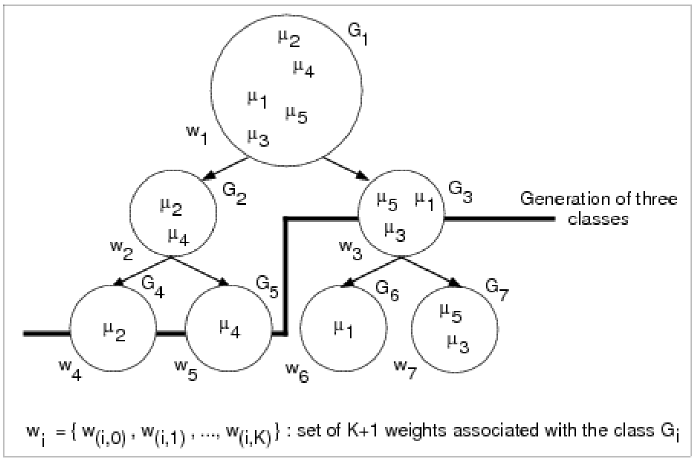
μ_2 μ_4 μ_1 μ_5 μ_3
G_1
w_1
μ_2 μ_4
G_2
w_2
μ_5 μ_1 μ_3
G_3
Generation of three classes
w_3
G_4
G_5
G_6
G_7
μ_2
μ_4
μ_1
μ_5 μ_3
w_4
w_5
w_6
w_7
$w_i = \{ w_{(i,0)}, w_{(i,1)}, \ldots, w_{(i,K)} \}$: set of K+1 weights associated with the class G_i

Fig. 1. *SEV*approach. In this example, three classes have been generated, according to the available amount of adaptation data. Thus, three sets of $K+1$ weights will be estimated. The weights of the set w_i are then applied to the gaussians grouped together within the class G_i.

2.3 Combining SMLLR with EigenVoices

We developed four possible approaches that integrate the concepts of *SMLLR* and *EigenVoices*-based techniques in a previous work [10]. Since the ordering,

in which *SMLLR* and *EigenVoices*-based techniques (*EV* or *SEV*) are combined, may influence the performance of the resulting technique, we tested each combination seperatly.
These four techniques, *EV*→*SMLLR*, *SEV*→*SMLLR*, *SMLLR*→*EV* and *SMLLR* →*SEV*, can easily be applied for the speaker adaptation task both in supervised batch mode and in unsupervised incremental mode. The two former approaches are based on the assumption that *SMLLR* adaptation is more efficient after an adaptation with one of the *EigenVoices*-based (*EV* or *SEV*) techniques. The two latter approaches assume that an *EigenVoices*-based adaptation is more robust after an adaptation with *SMLLR*.

2.3.1 Approaches EV→SMLLR and SEV→SMLLR

These two approaches (figure 2) consist of first obtaining an adapted model with the help of either the classical *EigenVoices* technique (*EV*→*SMLLR*) or Structural *EigenVoices* (*SEV*→*SMLLR*). These models are then used as initial models by *SMLLR* to produce the final adapted models.

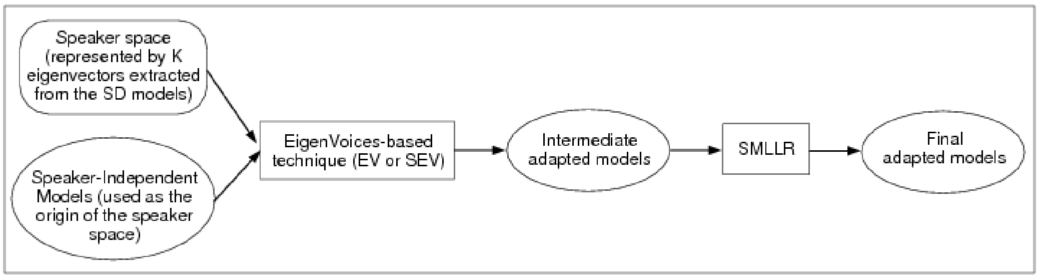
Speaker space (represented by K eigenvectors extracted from the SD models)
Speaker-Independent Models (used as the origin of the speaker space)
EigenVoices-based technique (EV or SEV)
Intermediate adapted models
SMLLR
Final adapted models

Fig. 2. Approaches *EV*→*SMLLR* and *SEV*→*SMLLR*. The origin of the speaker space used by the *EigenVoices*-based technique (*EV* or *SEV*) is the supervector extracted from the speaker-independent models, as usual.

2.3.2 Approaches SMLLR→EV and SMLLR→SEV

In the second set of approaches (figure 3), the two steps involved in the previous techniques are simply swapped. They consist of adapting the speaker-independent models using *SMLLR*. A supervector is extracted from the gaussian means of the adapted models, and it is then used as the origin of the speaker space in *EV* (in the case of *SMLLR*→*EV*) or *SEV* (in the case of *SMLLR*→*SEV*) to provide the final adapted models. Consequently, the first weight (w_0 in *EV* or $w_{(i,0)}$ for each class i in *SEV*) is no longer associated with the gaussian means of the speaker-independent system, but it is related to the gaussian means of the models yielded by *SMLLR*.

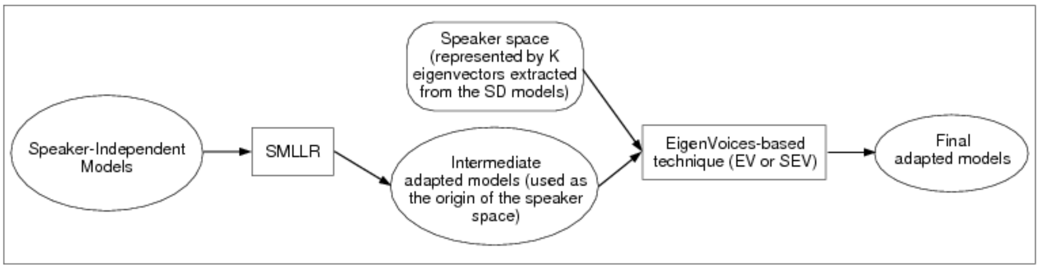
Speaker space (represented by K eigenvectors extracted from the SD models)
Speaker-Independent Models
SMLLR
Intermediate adapted models (used as the origin of the speaker space)
EigenVoices-based technique (EV or SEV)
Final adapted models

Fig. 3. Approaches *SMLLR→EV* and *SMLLR→SEV*. The origin of the speaker space used by the *EigenVoices*-based technique (*EV* or *SEV*) is the supervector extracted from the adapted models provided by *SMLLR*.

3 Using Genetic Algorithms for Rapid Speaker Adaptation

Every adaptation technique for acoustic models solves a numerical optimization problem. These techniques try to estimate the best parameters of the acoustic models by maximizing a gain function, the *log likelihood*.

All of the standard methods (*SMLLR*, *SMAP*, *EV*...) are suboptimal. As they are based on the Expectation-Maximization procedure to estimate the parameters of the acoustic models, they can only find a local optimal solution.

In this section, we propose to use genetic algorithms [17] in the framework of rapid speaker adaptation of acoustic models. The application of this family of algorithms was motivated by two major reasons. First, genetic algorithms can provide (at least) one global optimal solution, by exploring and exploiting a population of solutions. Second, they are also able to find good solutions when only few adaptation utterances are available. Indeed, they do not estimate the adaptation parameters, like the weights for *EV* or the linear regression matrices in the case of *SMLLR*. They rather try to update existing solutions whose quality is measured by a *fitness function*, depending on the log-likelihood of the adaptation data. This means that a good solution can even be found if only few adaptation utterances are available.

3.1 Classical Genetic Algorithms

Classical genetic algorithms are basically methods for solving numerical optimization problems. Similar to most optimization techniques, genetic algorithms look for the best solution in a given search space by maximizing a gain function. The search is guided by a simulation of Darwin's natural selection scheme, which associates the diversity of randomization and the survival of the fittest individuals.

Typically, genetic algorithms start from an initial population of candidate solutions (*individuals*), which they try to improve over a sequence of N_{IT} iterations in order to reach a population that contains better solutions. In the terminology of genetic algorithms, a solution is represented by one or more *chromosomes*. Each chromosome is represented as a vector of different *genes*. A gene represents one of the problem parameters of the associated optimization problem. A solution s is characterized by a *fitness function* $f(s)$, which represents a quality of adequacy to the considered problem.

To create a new population of solutions, standard genetic algorithms use three genetic operators within each iteration: the *reproduction* operator, the *mutation* operator, and the *selection* operator. The reproduction operator can be seen as a way of providing an exchange of information between candidate solutions. Once the children have been generated by the reproduction operator, they can be subjected to mutations. The idea behind the mutation procedure is the introduction of some variations within the population. Finally, the selection operator enables the fittest individuals of the current population to survive. These individuals then constitute the next population. The general principle of a genetic algorithm is summed up in figure 4.

The special characteristics of the genetic algorithms we used for the speaker adaptation problem are defined in more detail in the next section.

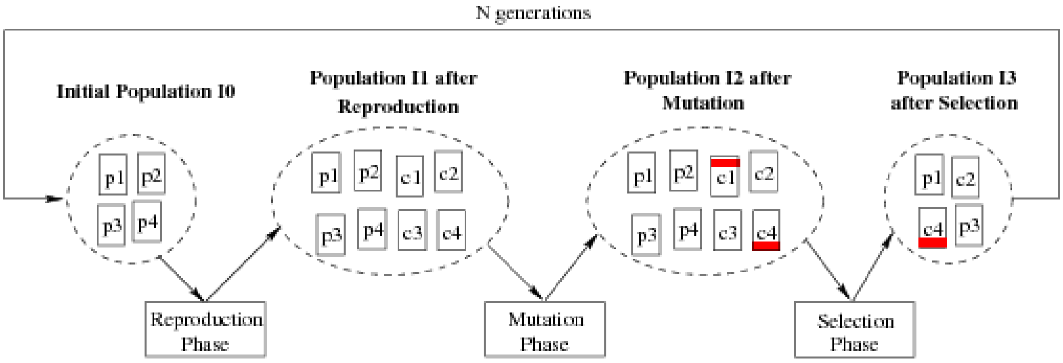

Fig. 4. Principle of a genetic algorithm

3.2 Genetic Algorithms applied for Speaker Adaptation

In our case, a candidate solution of the genetic algorithm is a (super)vector consisting of all the gaussian mean vectors of all the models currently included in the speech recognition system. A gene is a particular gaussian mean vector. The *fitness function* $f(s)$ for a candidate solution s is defined by:

$$f(s) = \frac{exp\left(\frac{log\ p(O/M_s)}{T}\right)}{\sum_i\ exp\left(\frac{log\ p(O/M_i)}{T}\right)} \tag{4}$$

where O, M_s and T represent the adaptation data, the acoustic models of the solution s, and the number of frames of the adaptation data, respectively.

This *fitness function* represents in fact an average *log likelihood* on the acoustic frames. It has the same property as the standard log likelihood: the higher the $f(s)$, the higher the likelihood $p(O/M_s)$. The term $exp\left(\frac{log\ p(O/M_s)}{T}\right)$ is an approximation of the probability $p(O/M_s)$, whose value cannot in general be computed, because it falls out of the precision range for variables of current computers. This term is then normalized, in order to ensure that the fitness of every individual s is within the range [0; 1].
The initial population is composed of N_I individuals, which are the $N_S + 1$ supervectors extracted from the N_S speaker-dependent systems, and from the speaker-independent system.

3.2.1 Reproduction Operator

The reproduction operator consists of (1) selecting pairs of parents among the individuals of the current population, and (2) merging the chromosomes of both parents of a pair to generate two children (or offsprings).
We chose the elitist strategy to select the parents who constitute a pair. An elitist reproduction suggests that a dominant individual (the best individual of the current population) will be merged several times with different other individuals. The best individual of the current population is thus guaranteed to be a member of a pair of parents. The other parent is an individual selected with a probability proportional to its fitness function. The higher the value of its fitness function, the more likely the corresponding individual will be selected as a parent. This means that each individual can be selected several times.
If N_I is the number of individuals in the current population, then $N_I/2$ pairs of parents are defined in this step.
Once all of the $N_I/2$ pairs of parents have been defined, the parents of each pair are merged to generate two offsprings. The merging of two parents consists of swapping (*crossing-over step*) and combining (*interpolation step*) groups of genes to generate the genes of two offsprings.
If, for example, two parents p_1 and p_2 are represented by vectors containing 3 genes:

$$p_1 = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \text{ and } p_2 = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

then crossing the chromosomes after the second gene and defining the interpolation factor as $i_f \in [0; 1]$ would produce two children c_1 and c_2, such that:

$$c_1 = \begin{bmatrix} a_1 * i_f + b_1 * (1 - i_f) \\ a_2 * i_f + b_2 * (1 - i_f) \\ b_3 * i_f + a_3 * (1 - i_f) \end{bmatrix} \text{ and } c_2 = \begin{bmatrix} b_1 * i_f + a_1 * (1 - i_f) \\ b_2 * i_f + a_2 * (1 - i_f) \\ a_3 * i_f + b_3 * (1 - i_f) \end{bmatrix}$$

The number N_{CP} of crossing points (in our example $N_{CP} = 1$) and the interpolation factor i_f are parameters of the algorithm and remain unchanged for all iterations. The position of each crossing point is generated at random for each pair of parents.

3.2.2 Mutation Operator

Let μ_i be the mean of the i-th gaussian (or *gene*) of a given supervector, σ_i the variance related to the gaussian i, and p_m the probability of mutation of a gene. As illustrated in figure 5, the mutation operator consists of carrying out the following two operations for each gene i of the children created during the reproduction step. Firstly, a random number $r \in [0; 1[$ is generated. Secondly, if $r < p_m$, the new mean $\hat{\mu}_g$ of this gene i is computed such that:

$$\hat{\mu}_i = \mu_i + s * \sigma_i \tag{5}$$

where s is a random number generated within the range $[-\gamma_m; \gamma_m]$. γ_m is the coefficient of mutation. It represents the degree of conservation of a gene: the higher γ_m, the more radically a gene may be altered by mutations. p_m and γ_m are both parameters of the algorithm and remain constant during all iterations.

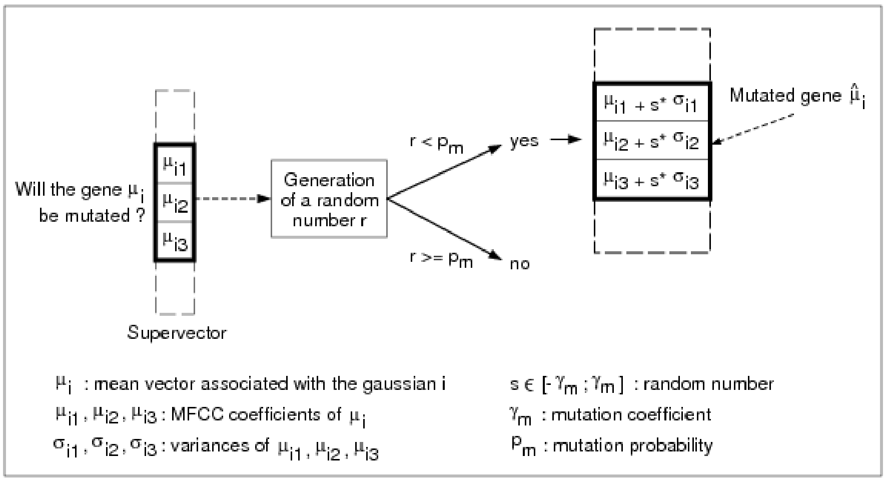
Will the gene μ_i be mutated ?
μ_{i1}
μ_{i2}
μ_{i3}
Supervector
Generation of a random number r
r < p_m
yes
r >= p_m
no
$\mu_{i1} + s^* \sigma_{i1}$
$\mu_{i2} + s^* \sigma_{i2}$
$\mu_{i3} + s^* \sigma_{i3}$
Mutated gene $\hat{\mu}_i$
μ_i : mean vector associated with the gaussian i
$\mu_{i1}, \mu_{i2}, \mu_{i3}$: MFCC coefficients of μ_i
$\sigma_{i1}, \sigma_{i2}, \sigma_{i3}$: variances of $\mu_{i1}, \mu_{i2}, \mu_{i3}$
$s \in [-\gamma_m ; \gamma_m]$: random number
γ_m : mutation coefficient
p_m : mutation probability

Fig. 5. Mutation operator

3.2.3 Selection Operator

The N_I parents of the next population are chosen using the $(\mu + \lambda)$ strategy [17, page 162]. This strategy consists of elaborating the population of the next iteration from the N_I best individuals (according to their fitness) of the current population (parents and generated children). This strategy of selection is commonly used in the framework of genetic algorithms when the application domain is dynamic [17], i.e. when the input data regularly varies. This is usually the case in speech recognition tasks.

3.3 Combining GA with EV

This approach, illustrated in figure 6, consists of three steps. First, a population of N_I potential systems adapted to the new speaker is obtained by the genetic algorithm *GA*. Second, the N_B best systems among these N_I systems are selected to be included into the set of the N_S SD systems used by the regular version of *EigenVoices*. Third, the *EigenVoices* technique is finally applied to the speaker-independent system using an initial speaker space of $N_B + N_S$ systems.

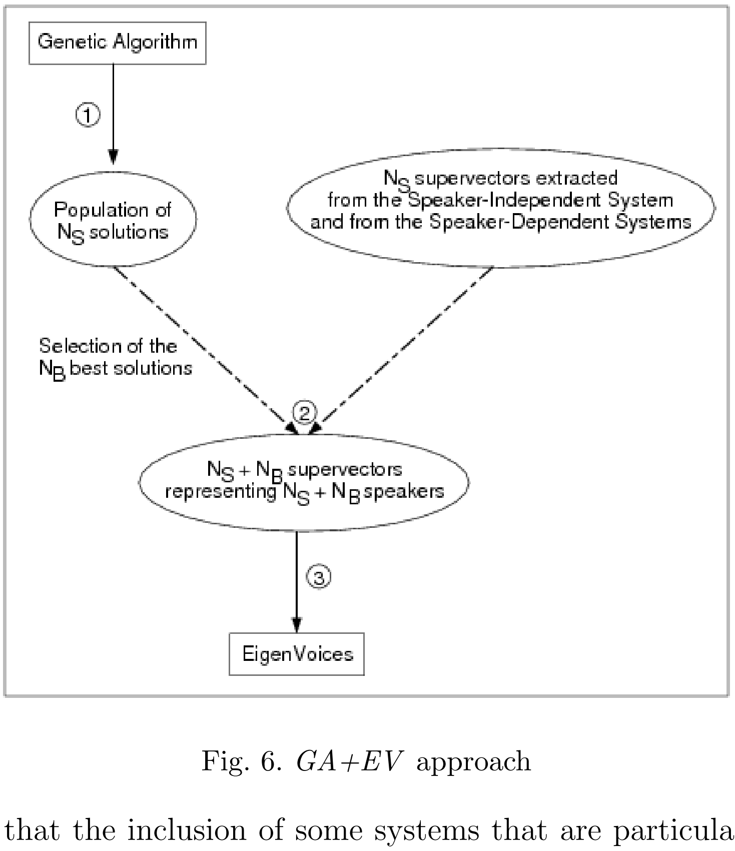
Genetic Algorithm
①
Population of
N_S solutions
N_S supervectors extracted
from the Speaker-Independent System
and from the Speaker-Dependent Systems
Selection of the
N_B best solutions
②
$N_S + N_B$ supervectors
representing $N_S + N_B$ speakers
③
EigenVoices

Fig. 6. *GA+EV* approach

We assume that the inclusion of some systems that are particularly adapted to the new speaker into the initial speaker space of N_S systems will make it closer to the new speaker. The estimation of the weights by the *EigenVoices* approach will thus be more robust.

4 Experimental Evaluation

4.1 Experimental Conditions

4.1.1 Training, Adaptation and Evaluation

All of the previous techniques have been implemented and included into the automatic speech recognition system ESPERE [6] [5] and evaluated on the *Resource Management (RM)* corpus.
The speech signals in *RM* are sampled at 16 kHz and were parameterized into the 11 MFCCs $C1$ to $C11$ and the 12 first and 12 second order time derivatives of $C0$ to $C11$, yielding a 35-dimensional feature vector.
The acoustic units in the speaker-independent system, and in each speaker-dependent system are represented by 45 HMMs with 3 states, and 1 HMM with one state to handle silence and short pauses. The probability density function of each state is modelled by a mixture of 32 gaussians.

The speaker-independent training set of RM1 was used to train the acoustic models of the speaker-independent system and the speaker-dependent systems. This set groups together 72 American native speakers (23 female and 49 male speakers). Each speaker pronounced 40 training utterances, which means a total of 2880 utterances. The acoustic models of the speaker-independent system were trained by performing 20 iterations of the Baum-Welch algorithm. Each speaker-dependent system was obtained by adapting the speaker-independent system using 10 iterations of *Structural Maximum A Posteriori (SMAP)* [20].

For the adaptation phase and the recognition phase, we used the speech data from 16 speakers (7 female and 9 male speakers) of the speaker-dependent set RM2. Each speaker uttered 600 training sentences, to be used only during the adaptation phase. For the recognition phase, a total of 1280 utterances were tested: 120 utterances per speaker for four of them and 100 utterances per speaker for eight of them. Speech recognition experiments were conducted by using the regular *word-pair* grammar of *RM.*

[6] ESPERE is a first order HMM-based speech recognition toolbox developed at LORIA.

4.1.2 Parameterizing of the adaptation techniques

The *LBG* method combined with the *K-means* procedure[7] was used to build the gaussian tree handled by *SMLLR* and the Structural version of *EigenVoices.*
Sufficient statistics were computed differently in batch mode and in incremental mode. In batch mode, all adaptation sentences were used to compute the statistics of each gaussian. In incremental mode, the accumulation of the sufficient statistics was carried out using the procedure proposed by Digalakis in [4]. This procedure consists of setting to zero all of the statistics associated with the gaussians before the first utterance has been pronounced. Then, for each new available adaptation utterance, the sufficient statistics of each gaussian g are updated by adding the statistics computed for the current n-th utterance using the previously adapted models. These sufficient statistics are used to estimate the adaptation parameters.

EigenVoices was parameterized to estimate 51 weights [8]. 50 weights are related to the 50 first eigenvectors and one weight is associated with the supervector s_{SIS} extracted from the SIS. The supervector s_{SIS} was used as the origin of the reduced speaker space.
SEV was parameterized using $\theta_{SEV} = 60$ and $\alpha_{SEV} = 800$.
The initial population of the genetic algorithm is made up of the 72 speaker-dependent systems and the speaker-independent system. Thus, $N_I = 73$. The genetic algorithm was also parameterized with:

- number of iterations $N_{IT} = 20$,
- number of crossing points $N_{CP} = 1$,
- interpolation factor $i_f = 0.4$,
- mutation probability $p_m = 0.001$ and
- mutation coefficient $\gamma_m = 1.0$

For each technique, the given parameterizing was determined empirically.

4.2 Experimental Results

Subsequent results represent the average word error rate (*WER*) for sixteen test speakers. The confidence interval equals to ±1% and was computed using a

[7] The *Mahalanobis* distance was employed as the distance measure between a gravity center of a node and a gaussian.

[8] We carried out several adaptation/evaluation experiences using different number of weights (1, 2, 5, 10, 20, 30, 40, 50, 60 and 70) to determine which one enabled *EV* to deliver the best improvement of the SIS.

risk of 5%. The average *WER* of the speaker-dependent systems is $WER_{SDS} = 4.5\%$; the *WER* of the speaker-independent system is equal to $WER_{SIS} = 12.7\%$.

In order to measure the reduction in WER of the speaker-independent system when using a given adaptation technique, we used the following formula for computing its relative error rate (relWER) from its absolute WER (absWER):

$$relWER = \frac{WER_{SIS} - absWER}{WER_{SIS}} \times 100$$

Table 1 and figure 7 show the results of the regular version of *EigenVoices*, Structural *EigenVoices* and *SMLLR* for a supervised batch adaptation. One can observe that *EV* is better than *SMLLR* only when a single utterance has been used. Indeed, *EigenVoices* needs to estimate fewer parameters than *SMLLR* for the same amount of adaptation data, which can be done robustly in the case of an *EigenVoices* adaptation. From the fifth utterance, *SMLLR* gives a greater WER reduction than *EigenVoices*, which starts to saturate at this point. This is due to the limited number of adaptation parameters, which are unable to capture all of the information gathered by the adaptation data. As expected, Structural *EigenVoices* technique gives the same results than *EigenVoices* for a single utterance, and it gives significantly better results than *EV* when more adaptation utterances are used. It gives results equivalent to *SMLLR* when less than 100 utterances are used, but it is outdistanced by *SMLLR* from the 100-th utterance. *SEV* starts to saturate from the 300-th utterance, certainly for the same reason that standard *EigenVoices* adaptation saturates after the fifth utterance.

Table 1
Absolute WER of the proposed Structural EigenVoices approach compared with EigenVoices and *SMLLR* for a supervised batch adaptation according to the number of adaptation utterances. Relative WER is specified inside parentheses.

	1	5	10	30	50	100	300	600
SEV	**12.14**	**11.16**	**10.88**	10.29	**9.86**	9.49	8.81	9.11
	(4.4)	(12.1)	(14.3)	(19)	(22.4)	(25.3)	(30.6)	(28.3)
EV	**12.14**	11.85	11.83	11.82	11.85	11.79	11.77	11.80
	(4.4)	(6.7)	(6.8)	(6.9)	(6.7)	(7.2)	(7.3)	(7)
SMLLR	12.7	**10.95**	**10.63**	**9.89**	**9.65**	**8.72**	**6.92**	**6.31**
	(0)	(13.8)	(16.3)	(22.1)	(24)	(31.3)	(45.5)	(50.3)

Table 2 and figure 8 show the results of the regular version of *EigenVoices*, Structural *EigenVoices* and *SMLLR* for an unsupervised incremental adaptation. In this mode, the same conclusions that were formulated in the case of a supervised batch adaptation can be drawn, except that *SEV* gives even better

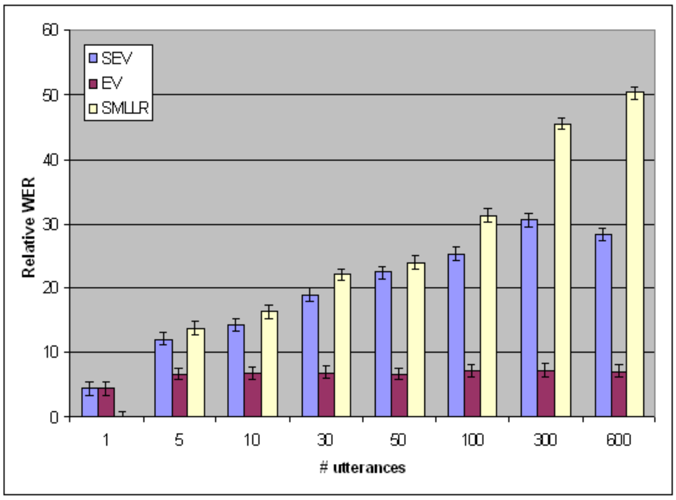

Fig. 7. Relative error reduction of the proposed Structural EigenVoices approach compared with EigenVoices and *SMLLR* for a supervised batch adaptation

results than *SMLLR* and *EV* for the three utterances case.

Table 2
Absolute WER of the proposed Structural EigenVoices approach compared with EigenVoices and *SMLLR* for a unsupervised incremental adaptation according to the number of adaptation utterances. Relative WER is specified inside parentheses.

	1	3	5	10	20	30	50	80
SEV	**12.26**	**11.60**	**11.26**	11.30	11.06	10.49	10.46	9.94
	(3.5)	(8.7)	(11.3)	(11)	(12.9)	(17.4)	(17.6)	(21.7)
EV	**12.26**	12.04	11.93	11.95	11.94	11.89	11.85	11.79
	(3.5)	(5.2)	(6.1)	(5.9)	(6)	(6.4)	(6.7)	(7.2)
SMLLR	12.7	11.80	**11.09**	**10.64**	**10.29**	**10.03**	**9.83**	**9.39**
	(0)	(7.1)	(12.7)	(16.2)	(19)	(21)	(22.6)	(26.1)

Table 3 and figure 9 present the results of the four proposed methods compared to *SMLLR* and *SEV* for a supervised batch adaptation. First, one can observe that all of the proposed methods improve the performance of the speaker-independent system for any available amount of adaptation data. Second, techniques that combine *SMLLR* and *SEV* yield always better results than techniques which combine *SMLLR* and *EV*. Compared to *EV→SMLLR* and *SMLLR→EV*, *SEV→SMLLR* and *SMLLR→SEV* not only benefit from the estimation method borrowed by the *EigenVoices*-based approaches (i.e. a linear combination of the SD models), but they also take advantage of the flexibility of *SEV*, such that better estimates can be obtained when more adaptation utterances are available. Both methods are particularly efficient for 30, 50 and 100 utterances, where they yield better results than the other methods, including *SMLLR*. Finally, one can notice that *SEV→SMLLR* globally

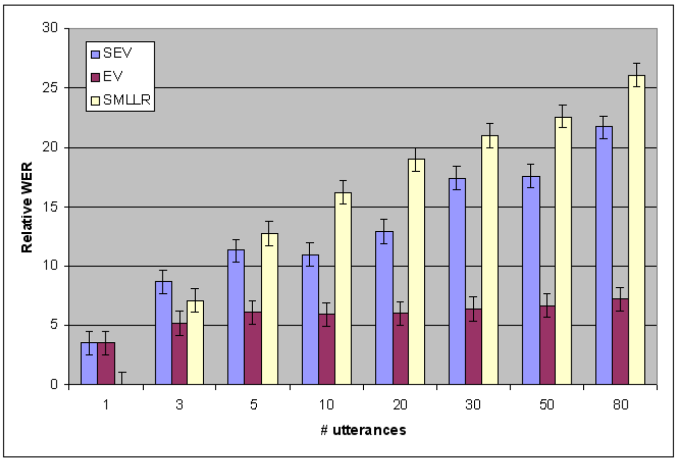

Fig. 8. Relative error reduction of the proposed Structural EigenVoices approach compared with EigenVoices and *SMLLR* for a unsupervised incremental adaptation

produces the best results when compared to the other methods. In particular, *SEV*→*SMLLR* is significantly better than *SMLLR* when between 50 and 80 utterances are used, according to the McNemar's test [9] exposed in appendix B.

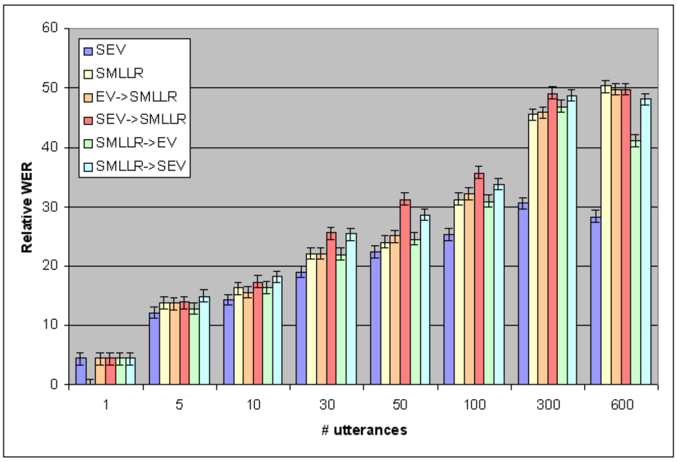

Fig. 9. Relative error reduction of the proposed approaches combining *SMLLR* and an EigenVoices-based scheme for a supervised batch adaptation.

Table 4 and figure 10 show the results of the four proposed methods compared with *SMLLR* and *SEV* for an unsupervised incremental adaptation. In this mode, there exists no method that is better than all the others, irrespective of the number of adaptation utterances used. Nevertheless, we can emphasize that *SEV* is the best technique when one or three utterances are used, and *SMLLR* should be used when more than five utterances are available. We ex-

[9] McNemar's test is used for deciding whether the difference in error rates between two speech recognition algorithms tested on the same data set is statistically significant.

Table 3
Absolute WER of the proposed approaches combining *SMLLR* and an EigenVoices-based scheme for a supervised batch adaptation according to the number of adaptation utterances. Relative WER is specified inside parentheses.

	1	5	10	30	50	100	300	600
SEV	**12.14**	11.16	10.88	10.29	9.86	9.49	8.81	9.11
	(4.4)	(12.1)	(14.3)	(19)	(22.4)	(25.3)	(30.6)	(28.3)
SMLLR	12.7	**10.95**	**10.63**	9.89	9.65	8.72	6.92	**6.31**
	(0)	(13.8)	(16.3)	(22.1)	(24)	(31.3)	(45.5)	(50.3)
EV→SMLLR	**12.14**	**10.96**	**10.72**	9.89	9.53	8.61	6.87	**6.38**
	(4.4)	(13.7)	(15.6)	(22.1)	(25)	(32.2)	(45.9)	(49.8)
SEV→SMLLR	**12.14**	**10.93**	**10.50**	**9.45**	**8.72**	**8.17**	**6.46**	**6.38**
	(4.4)	(13.9)	(17.3)	(25.6)	(31.3)	(35.7)	(49.1)	(49.8)
SMLLR→EV	**12.14**	**11.07**	**10.62**	9.91	9.58	8.78	6.74	7.48
	(4.4)	(12.8)	(16.4)	(22)	(24.6)	(30.9)	(46.9)	(41.1)
SMLLR→SEV	**12.14**	**10.81**	**10.39**	**9.47**	9.07	8.41	**6.52**	6.59
	(4.4)	(14.9)	(18.2)	(25.4)	(28.6)	(33.8)	(48.7)	(48.1)

plain the second-rate results of the four proposed methods by the following fact. For each of them, the statistics related to the current adaptation utterance are accumulated twice: one time before the *SMLLR* adaptation and the other time before the *EigenVoices*-based adaptation (*SEV* or *EV*). This double accumulation may thus influence on the estimation of the adaptation parameters. We are currently carrying out some investigations to study more deeply the impact of this step on the performance of the yielded adapted models.

Table 5 presents the results of the two proposed schemes *GA* and *GA+EV* compared with *EigenVoices*, for a rapid supervised batch adaptation with one adaptation utterance. *EV*, *GA* and *GA+EV* give the same improvement of performance of the speaker-independent system. They are statistically equivalent according to the McNemar's test shown in appendix B. Although the genetic algorithm has to estimate a huge number of parameters (about 150000 coefficients) with only few adaptation data (about 500 frames), it was nevertheless able to find good solutions.
We strongly believe that the quality of the adapted models could further be improved by increasing the number of SD models (and thus the number of training speakers) in the initial population.

Further experiments with genetic algorithm are in progress in supervised batch

Table 4
Absolute WER of the proposed approaches combining *SMLLR* and an EigenVoices-based scheme for an unsupervised incremental adaptation according to the number of adaptation utterances. Relative WER is specified inside parentheses.

	1	3	5	10	20	30	50	80
SEV	**12.26**	**11.60**	**11.26**	11.30	11.06	10.49	10.46	9.94
	(3.5)	(8.7)	(11.3)	(11)	(12.9)	(17.4)	(17.6)	(21.7)
SMLLR	12.7	**11.80**	**11.09**	**10.64**	**10.29**	**10.03**	**9.83**	**9.39**
	(0)	(7.1)	(12.7)	(16.2)	(19)	(21)	(22.6)	(26.1)
EV→*SMLLR*	**12.26**	**11.82**	**11.05**	**10.73**	10.53	10.48	10.43	10.46
	(3.5)	(6.9)	(13)	(15.5)	(17.1)	(17.5)	(17.9)	(17.6)
SEV→*SMLLR*	**12.26**	**11.84**	11.40	10.99	10.40	10.12	**9.72**	**9.42**
	(3.5)	(6.8)	(10.2)	(13.5)	(18.1)	(20.3)	(23.5)	(25.8)
SMLLR→*EV*	**12.26**	12.26	11.32	**10.55**	**10.08**	**9.83**	**9.74**	9.69
	(3.5)	(3.5)	(10.9)	(16.9)	(20.6)	(22.6)	(23.3)	(23.7)
SMLLR→*SEV*	**12.26**	12.07	11.60	11.24	10.67	10.49	10.50	10.30
	(3.5)	(5)	(8.7)	(11.5)	(16)	(17.4)	(17.3)	(18.9)

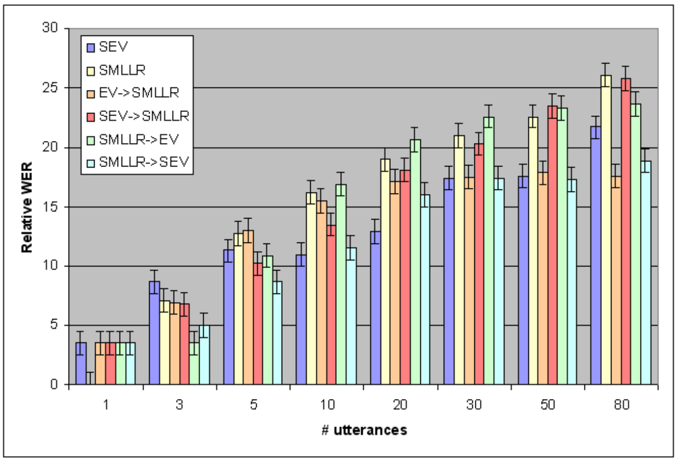

Fig. 10. Relative error reduction of the proposed approaches combining *SMLLR* and an EigenVoices-based scheme for an unsupervised incremental adaptation.

mode and in unsupervised incremental mode, with several adaptation utterances and higher values of N_S and N_{IT}. We hope that they will further improve the performance of the speech recognition systems.

Table 5
Absolute WER of the proposed genetic algorithms compared with SEV and EV with one adaptation utterance. Relative error reduction is specified inside parentheses.

	One utterance
SEV	12.14
	(4.4)
EV	12.14
	(4.4)
GA	**12.05**
	(5.1)
GA+EV	**11.98**
	(5.7)

5 Analysis in terms of complexity and memory requirements

The improvement of the recognition performance using *GA* or *GA+EV* goes along with an increase of the computational load and the memory needs. These main drawbacks can be explained by the fact that, unlike E.M.-based techniques (like *SMLLR* or *EigenVoices*), which estimate a single solution in N_{EM} iterations, a technique based on a genetic algorithm updates several initial solutions in N_{GA} iterations, with $N_{GA} > N_{EM}$. Furthermore, an iteration in *GA* takes more time than an iteration in *EV*, because the log-likelihood has to be computed in *GA* several times (one for each solution).
We give a detailed analysis of the complexity and memory requirements of *SMLLR*, EV and *GA* in the next subsections. Concerning the complexity analysis, we assume that only one iteration is carried out for each technique, and that the estimation of the adaptation parameters and the application of these parameters to gaussian means will merely be taken into account. Regarding the analysis of the memory requirements, we will only consider the storage of the gaussian means (those related to either the speaker-independent system or the speaker-dependent systems), assuming that the amount of memory used to store the other parameters is negligible.

For the sake of notation, let us recall that:

- N_G is the number of gaussians of the speaker-independent system and of each speaker-dependent system,
- N_D is the dimension of a gaussian mean vector,
- T is the number of the acoustic frames extracted from the adaptation utterances,

- N_S is the number of states of the acoustic models,
- M is the number of linear regressions estimated by *SMLLR*,
- K is the number of weights estimated by *EV*,
- N_I is the number of individuals in the initial population used by *GA*,
- p_m is the probability of mutation in *GA*.

5.1 Complexity and memory requirements of SMLLR

In *SMLLR*, the estimation of the parameters of a linear regression involves the inversion of a $((N_D + 1) \times (N_D + 1))$-dimensional matrix for each of the $N_D + 1$ lines of the linear regression. Thus, the estimation of the parameters of one linear regression can be done in $\mathcal{O}({N_D}^4)$ time, given that $\mathcal{O}({N_D}^3)$ time is required to inverse one $N_D \times N_D$-dimensional matrix. Once all linear regressions have been estimated, the adaptation of the gaussian means is carried out in $\mathcal{O}(N_G \times {N_D}^2)$ time. Consequently, the time τ^{SMLLR} required by one iteration of *SMLLR* is proportional to:

$$\tau^{SMLLR} \propto \mathcal{O}(M \times {N_D}^4 + N_G \times {N_D}^2) \tag{6}$$

SMLLR needs to store only the $N_G \times N_D$ gaussian means of the speaker-independent system.

5.2 Complexity and memory requirements of EV

In *EV*, the estimation of the $K+1$ weights involves the inversion of a $((K+1) \times (K+1))$-dimensional matrix, which can be done in $\mathcal{O}(K^3)$ time. Once all the weights have been estimated, the N_G gaussians are adapted in $\mathcal{O}(N_G \times N_D \times K)$ time. Thus, the time τ^{EV} required by one iteration of *EV* is proportional to:

$$\tau^{EV} \propto \mathcal{O}(K^3 + N_G \times N_D \times K) \tag{7}$$

EV needs to store $K + 1$ eigenvectors, each one being a $N_G \times N_D$-dimensional vector. Thus, $(K + 1) \times N_G \times N_D$ gaussian means are used and need to be stored by *EV*.

5.3 Complexity and memory requirements of GA

In *GA*, three steps are required to obtain the adapted models: the reproduction phase, the mutation phase and the selection phase.
The reproduction of the N_I parents generates N_I children. The generation of one children is done in $\mathcal{O}(N_G \times N_D)$ time, so that $\mathcal{O}(N_I \times N_G \times N_D)$ operations are needed to generate the population of children.
Mutation operator affects the genes of the N_I children. Statistically, $p_m \times N_G \times N_I$ genes are mutated, so that mutation phase is carried out in $\mathcal{O}(p_m \times N_G \times N_I \times N_D)$ time.
Finally, selection step requires the computing of the fitness function of each of the $2 \times N_I$ individuals (parents and children). Computing the fitness of one individual requires to compute the likelihood $p(O/\theta_i)$, which can be done in $\mathcal{O}(T \times N_S)$ time. Thus, selection operator can be executed in $\mathcal{O}(N_I \times T \times N_S)$ time.
Consequently, the time τ^{GA} required by one iteration of *GA* is proportional to:

$$\tau^{GA} \propto \mathcal{O}(N_G \times N_D \times (N_I + p_m \times N_I) + N_I \times T \times N_S) \tag{8}$$

GA uses a population of $2 \times N_I$ solutions (N_I parents and N_I children). Each solution is a $N_G \times N_D$-dimensional vector, so that $2 \times N_I \times N_G \times N_D$ gaussian means need to be stored.

5.4 Computing time for SMLLR, EV and GA

Table 6 shows the computing time in seconds for *SMLLR*, *EV* and *GA* on a Pentium 1.4 GHz with 512 MB RAM, under the conditions of the experiments described in section 4.1, such that:

- $N_G = 4352$,
- $N_D = 35$,
- $N_S = 136$,
- $K = 50$,
- $N_I = 73$ and
- $p_m = 0.001$.

As expected, we can notice that the computing time taken by *SMLLR* increases with the number of adaptation utterances used, and thus with the number of linear regressions estimated. Moreover, an adaptation using *EV* takes a constant time, irrespective of the available amount of adaptation utterances. Finally, like *SMLLR*, the computing time taken by *GA* increases with the available amount of adaptation data, but it is more than several hundred times higher than the one taken by an *E.-M.*-based technique like *SMLLR* or *EV*. For instance, *GA* is about 140 times longer than *EV*in the case of a single adaptation utterance. Nevertheless, one should keep in mind that genetic algorithms are intrinsically distributed and thus are not designed to run on single-processor architectures. For this reason, they would execute in linear time (according to the number of adaptation utterances) on a parallel architecture composed of $2 \times N_I$ processors, N_I being the number of individuals in the initial population.

Table 6
Comparison of the computing times (in seconds) of *SMLLR*, *EV* and *GA* on a Pentium 1.4 GHz with 512 MB RAM.

	1	5	10	30	50	100	300	600
SMLLR	0	9	10	11	12	15	17	18
EV	13	13	13	13	13	13	13	13
GA	1800	8400	-	-	-	-	-	-

6 Conclusions

We have proposed in this paper several original speaker adaptation techniques of Hidden Markov Models for speech recognition: *SEV*, *EV→SMLLR*, *SMLLR→EV*, *SEV→SMLLR*, *SMLLR→SEV*, *GA* and *GA+EV*. All of them were implemented on the same speech recognition system (ESPERE) and they were experimentally evaluated on the same speech corpus (namely *Resource Management*). They were compared to the well-known adaptation techniques: *SMLLR* and *EV*.

We presented in the first part a structural version of *EigenVoices* and four methods combining *SMLLR* and *EigenVoices*-based techniques (EV or SEV). It has been shown experimentally that Structural *EigenVoices* can push back the early saturation in performance encountered by *EigenVoices*, and this for both supervised batch mode and unsupervised incremental mode. Moreover, for a supervised batch adaptation, *SEV→SMLLR* provides the best global results compared to the other evaluated methods, irrespective of the number of adaptation utterances used.

In the second part of this work, two pioneering approaches, both based on a genetic algorithm, were proposed for a rapid speaker adaptation of acoustic models in supervised batch mode. It has been shown experimentally that the *GA* technique, which makes use of a genetic algorithm to directly update

the gaussian means of a speaker-independent system, provides results similar to the ones of *EigenVoices*. Furthermore, the *GA+EV* method outperforms *EigenVoices* by providing a speaker space that is better adapted to the new speaker. This implies that the estimation of the weights for the *EigenVoices* approach can be carried out more precisely.
Despite the increase of complexity involved by the use of genetic algorithms for speaker adaptation, the results remain very encouraging. We believe that the concept of directly updating the gaussian means as a function of the likelihood of the adaptation data, instead of first estimating a set of adaptation parameters which are then applied to the gaussian means, constitutes a promising novel approach in the field of speaker adaptation. Moreover, one should keep in mind that these techniques are relatively new in the framework of speech adaptation. Thus, they are still subject to some optimizations in terms of computation time and memory needs.

Our future work will deal with the application of genetic algorithms in the field of speaker adaptation of acoustic models. One interesting approach would be to reverse the roles of *GA* and *EV* in our algorithm *GA+EV*, i.e. introduce the adapted models based on *EV* as one of the individuals in the initial population to which *GA* is applied. Another approach would consist of estimating the weights of *EV* or *SEV* with the help of genetic algorithms. The weights in *EigenVoices* are currently estimated by the *Maximum Likelihood Eigen-Decomposition* procedure [10]. They represent a linear combination of acoustic models. The simplicity and versatility of genetic algorithms might be used to estimate weights which represent a *polynomial combination of acoustic models.* We anticipate that such a polynomial combination would produce more accurate adapted models than the adapted models built from a linear combination of acoustic models. Moreover, such a technique would require less amounts of memory and a reduced computationally load compared to *GA* and *GA+EV*.

7 Acknowledgment

The authors would like to thank Filipp Korkmazsky, who gave us precious advice to design our genetic algorithms.

[10] This procedure is based on the E.-M. algorithm and hence can only find a local solution.

A Experimental evaluation of *SMLLR*, *SMAP* and *EV*

We have also studied, implemented and experimentally evaluated *SMAP* [19,20]. Figures A.1 and A.2 present the results we obtained with it on our speech recognition system ESPERE by using the experimental conditions described in section 4.1.

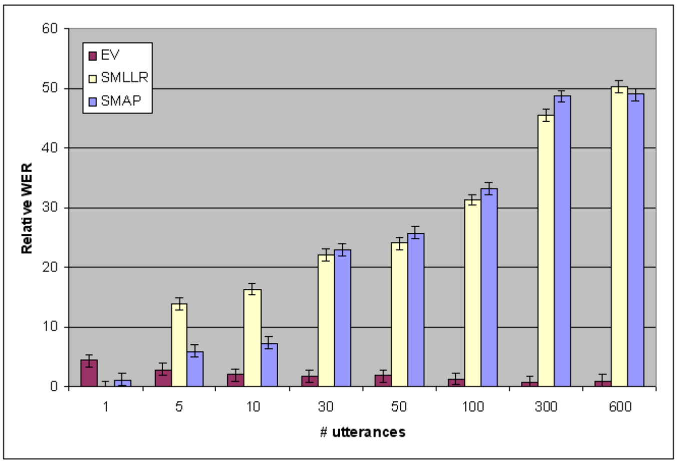

Fig. A.1. Relative error reduction of *SMAP*, *SMLLR* and *EV* for a supervised batch adaptation.

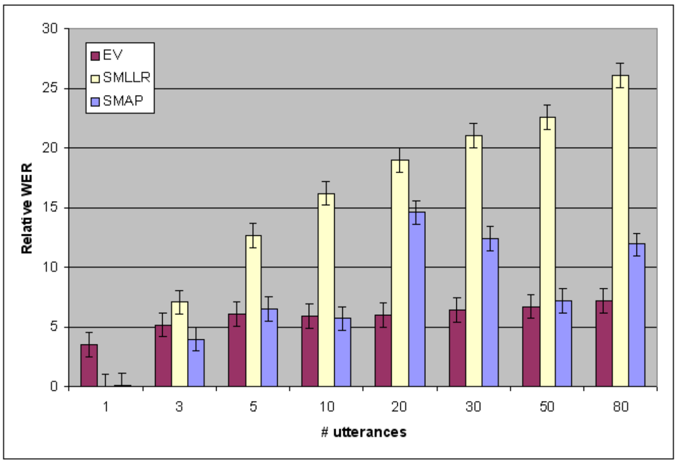

Fig. A.2. Relative error reduction of *SMAP*, *SMLLR* and *EV* for an unsupervised incremental adaptation.

As already mentioned in the introduction, *SMAP* is not as efficient as *EV* when few adaptation utterances are available, and it is only as efficient as *SMLLR* when enough adaptation data becomes available. Moreover, *SMAP* gives worse results than *SMLLR* in incremental mode, irrespective of the amount of adaptation data.

B Experimental comparison results of ***SEV→SMLLR***, ***SMLLR***, ***EV***, ***GA*** and ***GA+EV*** using the McNemar's test

The McNemar's test [6] stipulate that the joint performance of two speech recognition algorithms A_1 and A_2 working on the same corpus can be summarized in a 2×2 table, as shown in table B.1.

Table B.1
McNemar's test.

		A_2	
		Correct	Incorrect
A_1	Correct	N_{00}	N_{01}
	Incorrect	N_{10}	N_{11}

N_{00} is the number of utterances A_1 classifies **correctly**, A_2 classifies **correctly**.

N_{01} is the number of utterances A_1 classifies **correctly**, A_2 classifies **incorrectly**.

N_{10} is the number of utterances A_1 classifies **incorrectly**, A_2 classifies **correctly**.

N_{11} is the number of utterances A_1 classifies **incorrectly**, A_2 classifies **incorrectly**.

We let the interested readers to consult [6] for deciding whether the difference in error rates between A_1 and A_2 is statistically significant.

Table B.2 shows the comparison of errors made by *SEV→SMLLR* and *SMLLR* when between 50 and 100 utterances are used.

Table B.2
McNemar's test of *SMLLR* and *SEV→SMLLR*. A_1 is *SMLLR*. A_2 is *SEV→SMLLR*.

	50	70	80	90	100
N_{00}	12812	12874	12883	12921	12918
N_{01}	162	161	162	154	147
N_{10}	214	204	206	181	181
N_{11}	1048	990	985	980	990
	Significant	Significant	Significant	NOT Significant	NOT Significant

Table B.3 shows the comparison of errors made by *EV* and *GA* on the one hand, and *EV* and *GA+EV* on the other hand, when one adaptation utterance is used.

Table B.3
McNemar's test of EV and GA, and EV and $GA+EV$. For all tests, A_1 represents EV.

	EV and GA	EV and $GA+EV$
N_{00}	12406	12414
N_{01}	204	186
N_{10}	183	198
N_{11}	1443	1438
	NOT Significant	NOT Significant

References

[1] X. L. Aubert. Eigen-MLLRs applied to Unsupervised Speaker Enrollment for Large Vocabulary Continuous Speech Recognition. *International Conference on Acoustics, Speech, and Signal Processing*, pages 349–352, 2004.

[2] H. Botterweck. Very Fast Adaptation for Large Vocabulary Continuous Speech Recognition using Eigenvoices. *International Conference on Spoken Language Processing*, pages 354–357, 2000.

[3] K.-T. Chen, W.-W. Liau, H.-M. Wang, and L.-S. Lee. Fast Speaker Adaptation using Eigenspace-based Maximum Likelihood Linear Regression. *International Conference on Spoken Language Processing*, pages 742–745, 2000.

[4] V.V. Digalakis. Online Adaptation Hidden Markov Models using Incremental Estimation Algorithms. *Transactions on Speech and Audio Processing*, 7(3):253–261, 1999.

[5] D. Fohr, O. Mella, and C. Antoine. The Automatic Speech Recognition Engine ESPERE : Experiments on Telephone Speech. *International Conference on Spoken Language Processing*, pages 246–249, 2000.

[6] L. Gillick and S. Cox. Some Statistical Issues in the comparison of Speech Recognition Algorithms. *International Conference on Acoustics, Speech, and Signal Processing*, pages 532–535, 1989.

[7] R. Kuhn, J.-C. Junqua, P. Nguyen, and N. Niedzielski. Rapid Speaker Adaptation in Eigenvoice Space. *Transactions on Speech and Audio Processing*, 8(6):695–707, 2000.

[8] R. Kuhn, P. Nguyen, J.-C. Junqua, and al. Eigenvoices for Speaker Adaptation. *International Conference on Spoken Language Processing*, pages 1771–1774, 1998.

[9] R. Kuhn, P. Nguyen, J.-C. Junqua, R. Boman, N. Niedzielski, S. Fincke, K. Field, and M. Contolini. Fast Speaker Adaptation using A Priori Knowledge.

International Conference on Acoustics, Speech, and Signal Processing, pages 1587–1590, 1999.

[10] F. Lauri, I. Illina, and D. Fohr. Combining Eigenvoices and Structural MLLR for Speaker Adaptation. *International Conference on Acoustics, Speech, and Signal Processing*, 1:580–583, 2003.

[11] F. Lauri, I. Illina, and D. Fohr. Using Genetic Algorithms for Rapid Speaker Adaptation. *Eurospeech*, pages 1497–1500, 2003.

[12] C.J. Leggetter and P.C. Woodland. Flexible Speaker Adaptation using Maximum Likelihood Linear Regression. *Eurospeech*, pages 1155–1158, 1995.

[13] C.J. Leggetter and P.C. Woodland. Maximum Likelihood Linear Regression for Speaker Adaptation of Continous Density Hidden Markov Models. *Computer Speech and Language*, 9:171–185, 1995.

[14] B. Mak, S. Ho, and J. T. Kwok. Speedup of Kernel Eigenvoice Speaker Adaptation by Embedded Kernel PCA. *International Conference on Spoken Language Processing*, 4:2913–2916, 2004.

[15] B. Mak and R. Hsiao. Improving Eigenspace-based MLLR Adaptation by Kernel PCA. *International Conference on Spoken Language Processing*, 1:13–16, 2004.

[16] B. Mak, J. T. Kwok, and S. Ho. A Study of Various Composite Kernels for Kernel Eigenvoices Speaker Adaptation. *International Conference on Acoustics, Speech, and Signal Processing*, 1:325–328, 2004.

[17] Z. Michalewicz. *Genetic Algorithm + Data Structures = Evolution Programs.* Springer-Verlag, 1996.

[18] P. Nguyen. *Fast Speaker Adaptation.* Rapport de these professionnelle, Institut Eurcom, 1998.

[19] K. Shinoda and C.-H. Lee. Unsupervised Adaptation using Structural Bayes Approach. *International Conference on Acoustics, Speech, and Signal Processing*, pages 793–796, 1998.

[20] K. Shinoda and C.-H. Lee. A Structural Bayes Approach to Speaker Adaptation. *Transactions on Speech and Audio Processing*, 9(3):276–287, 2001.

[21] N.J.-C. Wang, S. S.-M. Lee, F. Seide, and L.-H. Lee. Rapid Speaker Adaptation using A Priori Knowledge by Eigenspace Analysis of MLLR Parameters. *International Conference on Acoustics, Speech, and Signal Processing*, 1:345–348, 2001.