# A COMPARATIVE STUDY OF ADAPTATION METHODS FOR SPEAKER VERIFICATION

*Johny Mariéthoz* *Samy Bengio*

Dalle Molle Institute for Perceptual Artificial Intelligence (IDIAP)
CP 592, rue du Simplon 4
1920, Martigny, Switzerland
{marietho,bengio}@idiap.ch

**ABSTRACT**

Real-life speaker verification systems are often implemented using client model adaptation methods, since the amount of data available for each client is often too low to consider plain Maximum Likelihood methods. While the *Bayesian Maximum A Posteriori* (MAP) adaptation method is commonly used in speaker verification, other methods have proven to be successful in related domains such as speech recognition. This paper reports on experimental comparison between three well-known adaptation methods, namely MAP, *Maximum Likelihood Linear Regression*, and finally *EigenVoices*. All three methods are compared to the more classical *Maximum Likelihood* method, and results are given for a subset of the *1999 NIST Speaker Recognition Evaluation* database.

## 1. INTRODUCTION

State-of-the-art speaker verification systems are based on statistical generative models such as Hidden Markov Models (HMMs) for text-dependent tasks or Gaussian Mixture Models (GMMs) for text-independent tasks. In both cases, using Bayes theorem, one needs to create a generative model for each client, as well as a generative model for a corresponding anti-client, often replaced by a global *world* model. Training the world model is done using the well-known EM algorithm in order to optimize the Maximum Likelihood criterion, over a set of pre-recorded sentences pronounced by people who will not be clients of the system. It is usually easy to find a large dataset of such sentences, hence create a well-estimated world model.

On the other hand, as it is less realistic to ask to a future client to stay hours in front of an acquisition system, it is hopeless to obtain a large dataset of sentences pronounced by the client. In order to overcome this lack of training material for each particular client, many researchers have proposed the use of *adaptation methods*, where one *adapts* the already trained world model to each client using his own material (hence, starting from the world model, and moving towards client information in some constrained way).

While the adaptation method most often used in speaker verification is *Bayesian Maximum A Posteriori* (MAP), other methods such as *Maximum Likelihood Linear Regression* (MLLR) and *EigenVoices* have been used with success in related domains such as speech recognition. In this paper, we propose to compare MAP to these two other methods on the task of text-independent speaker verification, using the benchmark database of the *1999 NIST Speaker Recognition Evaluation*.

In the following, we first review the classical Maximum Likelihood training method, then present the three adaptation methods, and finally the methodology of our comparative study as well as the obtained results.

Note that a similar comparison was published in [1] but was concerned about text-dependent applications, compared only two methods (MAP and MLLR), used a non-public database, and published performance using *a posteriori* selected hyper-parameters (such as thresholds) which might strongly bias the comparative results.

## 2. MAXIMUM LIKELIHOOD FOR GAUSSIAN MIXTURE MODELS

The most used model, in the context of text-independent speaker verification, is the Gaussian Mixture Model (GMM) with diagonal covariance matrix. In order to use such a model, we make the (often false) assumptions that the frames of the speech utterance are independent from each other and the features in each frame are uncorrelated: the probability of a sequence $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$ given a GMM with $N$ Gaussians is computed as follows

$$p(\mathbf{X}) = \prod_{t=1}^{T} p(\mathbf{x}_t) = \prod_{t=1}^{T} \sum_{n=1}^{N} w_n \cdot \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_n, \boldsymbol{\sigma}_n) \quad (1)$$

This research has been partially carried out in the framework of the European BANCA project, funded by the Swiss OFES project number 99-0563-1.

where $w_n$ is the weight of Gaussian $\mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_n, \boldsymbol{\sigma}_n)$ with mean $\boldsymbol{\mu}_n \in \mathbb{R}^d$ where $d$ is the number of features and with standard deviation $\boldsymbol{\sigma}_n \in \mathbb{R}^d$.

GMMs are usually trained using the EM algorithm [2] following the *Maximum Likelihood* (ML) principle which states that we should select the parameters $\theta$ that maximize the probability density of the observed data $\mathbf{X}$, that is

$$\hat{\theta} = \arg\max_{\theta} p(\mathbf{X}|\theta). \tag{2}$$

In the following sections, we present the three adaptation methods. Note that for all the methods, the only parameters that are adapted are the Gaussian means, while the weights and standard deviations are kept fixed and equal to their corresponding value in the world model. Thus, the total number of parameters $M$ per client is now equal to $N \cdot d$.

## 3. BAYESIAN MAXIMUM A POSTERIORI

The Bayesian *Maximum A Posteriori* (MAP) principle [3] differs from ML in that MAP assumes that the parameters $\theta$ of the distribution $p(\mathbf{X}|\theta)$ is also a random variable which has a prior distribution $p(\theta)$. The MAP principle states that one should select $\hat{\theta}$ such that it maximizes its posterior probability density, that is:

$$\begin{aligned} \hat{\theta} &= \arg\max_{\theta} p(\theta|\mathbf{X}) \\ &= \arg\max_{\theta} p(\mathbf{X}|\theta) \cdot p(\theta). \end{aligned} \tag{3}$$

Using MAP for client model adaptation usually means that the prior for the parameters of a client model will be represented by the world model parameters [4]. Moreover, one can simplify further without loss of performance by using a global parameter to tune the relative importance of the prior, as follows:

$$\hat{\boldsymbol{\mu}}_{n_c} = \alpha \boldsymbol{\mu}_{n_w} + (1-\alpha) \frac{\sum_{t=1}^{T} P(n|\mathbf{x}_t)\mathbf{x}_t}{\sum_{t=1}^{T} P(n|\mathbf{x}_t)} \tag{4}$$

where $\hat{\boldsymbol{\mu}}_{n_c}$ is the new mean of Gaussian $n$ for client $c$, $\boldsymbol{\mu}_{n_w}$ is the corresponding mean in the world model, $P(n|\mathbf{x}_t)$ is the posterior probability of Gaussian $n$, and $\alpha$ is chosen by cross-validation.

## 4. MAXIMUM LIKELIHOOD LINEAR REGRESSION

Maximum Likelihood Linear Regression (MLLR) [5] is an adaptation method that proposes to constrain the means of the Gaussians of a given client GMM to be linear combinations of the means of the corresponding Gaussians of the world model:

$$\hat{\boldsymbol{\mu}}_{n_c} = \mathbf{A}_n \boldsymbol{\mu}_{n_w} + \mathbf{b}_n \tag{5}$$

where the matrix $\mathbf{A}_n$ and the vector $\mathbf{b}_n$ are parameters to be found by maximizing the likelihood of the client data. This can be done using a modified version of the EM algorithm presented in [5].

The main idea behind adaptation methods is to constrain the client models in a small appropriate space (hence with only a few parameters to adjust), given the small amount of data available for each client. Unfortunately, if MLLR is applied as is, the number of parameters to be updated becomes bigger than with standard ML, since for each mean vector $\boldsymbol{\mu}_{n_c}$ of size $d$, one now have a matrix $\mathbf{A}_n$ of size $d \cdot d$ and a vector $\mathbf{b}_n$ of size $d$ to adjust, which is apparently not a good idea.

Hence, in order to keep the number of parameters low, it is recommended to *tie* or *cluster* some Gaussians together in order to force them to share the same matrix $\mathbf{A}$ and vector $\mathbf{b}$, using for instance the method of *regression class trees* [6]. This method grows dynamically a binary tree using a heuristic that tries to cluster similar Gaussians (in the Euclidean sense) together while keeping the number of observations in each cluster above a minimum limit.

## 5. EIGENVOICES

The idea of *EigenVoices* [7] has been inspired by a similar idea often used in face recognition and initially introduced in [8]: the *eigenfaces*. The underlying hypothesis of *eigenfaces* is that all faces represented in a given space of dimension $M$ could in fact be represented in a much smaller space of dimension $K \ll M$. The most commonly used tool to select this smaller space is the well-known PCA, which generates an orthogonal basis derived from the first $K$ eigenvectors of the covariance matrix of some available examples already represented in the original space.

More formally, given a set of $T$ client models already estimated (in this paper, we used the speakers of the world model and adapted a specific model for each of them using MAP), one can represent each client model as its underlying parameter vector (of dimension $M$).

PCA is then used to compute the first $K$ eigenvectors $\mathbf{e}_k$ of the covariance matrix of the $T$ parameter vectors. Afterward, for each new client, one can train a new model for which the parameters are constrained to be a linear combination of these eigenvectors,

$$\hat{\boldsymbol{\mu}}_c = \mathbf{E}\mathbf{w}_c \tag{6}$$

where $\mathbf{E}$ is the matrix containing the $K$ eigenvectors $\mathbf{e}_k$ as rows and $\mathbf{w}_c$ is the parameter vector of client $c$ in the eigenspace. This parameter vector can be learned using a modified EM algorithm as described in [7].

## 6. EXPERIMENTAL COMPARISON

The goal of this paper is to compare experimentally many adaptation methods applied to a text-independent speaker verification task. In this section, we first present the database, then review the general methodology used for the comparison, and finally give results comparing ML, MAP, MLLR and EigenVoices.

### 6.1. Database and Protocol

All the algorithms were tested on a subset of the database that was used for the *1999 NIST Speaker Recognition Evaluation*, which comes from the Switchboard-2 Phase 3 Corpus collected by the Linguistic Data Consortium. This corpus consists of 2728 conversations of 5 minute length free speech involving 640 speakers. While in the original database two different handsets were used (carbon and electret), in the subset selected for this paper, we only used data from electret handsets.

The database was separated into three subsets: a training set for the world model, and both a development set and an evaluation set of clients. Furthermore, for each client, there was training material and test accesses.

As it was done during the contest, we separated the data into male and female data, in order to create two different world models. The male world model was trained on 137 speakers for a total of 1.5 hours of speech, while the female world model was trained on 218 speakers for a total of 3 hours of speech.

For both development and evaluation clients, there was about 2 minutes of telephone speech used to train the models and each test access was less than 1 minute long. Each population consisted of 45 males and 45 females. The total number of accesses for each population was around 5000 with a proportion of 10% of true accesses.

### 6.2. General Methodology

All the experiments described here have followed the same methodology. First of all, the original waveforms have been sampled every 10ms and then parameterized into 16 MFCC coefficients and their first derivative, as well as the energy together with its first derivative, for a total of 34 features.

Afterward, a *bi-Gaussian method* has been used in order to remove the silence frames from the data. We trained a Gaussian Mixture Model (GMM) with two Gaussians in an unsupervised mode, with the hope that one Gaussian would capture the speech frames while the second would capture the silence frames, since they have quite different characteristics. We then simply removed the frames for which the maximum likelihood was given by the Gaussian corresponding to the silence frames.

While the energy and its first derivative were important in order to remove the silence frames, they were not adapted to the task of discrimination between clients and impostors, and they were thus removed from the features after silence removal. Hence, the world and client models were trained with 32 features (instead of 34).

In order to find the optimal capacity of the models, we used a K-fold cross-validation method on the training set to select the size of the GMM as well as other potential *hyper-parameters* such as the *v-floor* which represents the minimal proportion of the global variance that a Gaussian can take. The hyper-parameters of the clients also included: for ML, the number of Gaussians in the client models; for MAP, the $\alpha$ factor between the world and the client model; for MLLR, the clustering factor that forced the Gaussians to share their linear regression parameters $\mathbf{A}$ and $\mathbf{b}$; for EigenVoices, the optimal number of eigenvectors $K$ kept in the transformation matrix $\mathbf{E}$. To train the prior model used for MAP, $\mathbf{A}$ and $\mathbf{b}$ used for MLLR and $\mathbf{E}$ used for EigenVoices, we use data from the world model subset.

In any case, we used the same methodology: using the development set, for each value of the hyper-parameter to tune, we trained the client models using the training data available for each client. We then selected the value of the hyper-parameter that optimized the Equal Error Rate (EER) on the test accesses of the development set. Finally, we trained the models of the evaluation set using these hyper-parameters and report the results obtained on the test accesses of the evaluation set. Hence, these results are unbiased as the corresponding data have not been used for any purpose during the development of the models.

### 6.3. Results

The values of the hyper-parameters found on the development set were the following: 128 Gaussians in the world model, 70 Gaussians for client models trained by ML, $\alpha = 0.5$ for MAP, $K = 350$ out of 355 eigenvectors selected for EigenVoices (the male and female speakers were merged to obtain more potential eigenvectors), and finally 1000 observations minimum in each node of the tree used in MLLR.

The results of the experiments are given in Table 1. FAR represents the *false acceptance rate* (number of false acceptances divided by number of impostor accesses), FRR is the *false rejection rate* (number of false rejections divided by number of client accesses), while HTER is the *half total error rate* (the average of FAR and FRR). The corresponding DET curves of the four methods are also shown in Figure 1.

In the first part of Table 1, we compared the four methods used alone. As it can be seen along the HTER column, ML gave the worst result (22.93), while MLLR and EigenVoices were only slightly better (18.42 and 20.57). MAP was in fact statistically significantly better (15.81) than all the other methods (with more than 99% confidence).

In the second part of Table 1, we combined MAP with ML, MLLR and EigenVoices. For MLLR and EigenVoices, the resulting update equation is similar to the MAP update equation (4), but replacing the term multiplying $(1-\alpha)$ by the right side of the other model's equation. A new $\alpha$ was selected afterward (but still on the development set). We also show how ML performed when only the means were allowed to be modified while keeping the other parameters fixed to the world model (which corresponds in fact to MAP with $\alpha = 0$). These combinations did improve the results, but they remained worse that MAP alone.

| Method | FAR | FRR | HTER |
|---|---|---|---|
| ML | 25.50 | 20.35 | 22.93 |
| MAP $\alpha = 0.5$ | 15.69 | 15.93 | 15.81 |
| MLLR | 20.24 | 16.59 | 18.42 |
| EigenVoices | 21.66 | 19.47 | 20.57 |
| ML+MAP $\alpha = 0$ | 20.48 | 15.70 | 18.10 |
| MLLR+MAP $\alpha = 0.5$ | 16.82 | 15.04 | 15.93 |
| EigenVoices+MAP $\alpha = 0.3$ | 20.87 | 19.69 | 20.28 |

**Table 1**. Performance of different adaptation methods on the evaluation set of the NIST database.

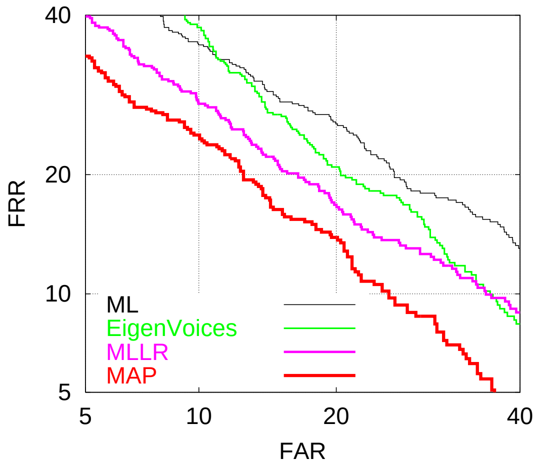


**Fig. 1**. DET curves on the NIST evaluation set.

## 7. DISCUSSION

The goal of the paper was to compare the most used method, MAP, to other adaptation methods which had already given good performance on related tasks. It appears that, without specific modifications, MLLR and EigenVoices do not perform as well as MAP on text-independent speaker verification. One tentative explanation of the poor results of MLLR and EigenVoices might be that both these methods are intended to force the parameters of the clients to live in a smaller space (due to the lack of data) containing only clients, which can be a good idea for discriminating clients from everything else, but not necessarily for discriminating clients between each others.

While all three methods maximize the likelihood of the client data under different constraints controlled by hyper-parameters, it seems that the MAP hyper-parameter has a discriminant impact on the model while MLLR and Eigen-Voices hyper-parameters do not have this discriminant property. Hence, the choice of their hyper-parameter is always towards the weakest constraint, which brings the model near the classical maximum likelihood model (with weights and standard deviation fixed).

## 8. CONCLUSION

In this paper, we have presented a comparative study of several client model adaptation methods for text-independent speaker verification tasks. All methods were compared to the more traditional Maximum Likelihood method on a well-known benchmark database. It appears that the Bayesian Maximum A Posteriori method, which was already the most used method, is currently the best one for such a task. Further studies should be conducted in order to modify the other methods in order to force the parameters to live in a more discriminant space.

## 9. REFERENCES

[1] S. Ahn, S. Kang, and H. Ko, "Effective speaker adaptation for speaker verification," in *IEEE International Conference on Acoustic, Speech, and Signal Processing, ICASSP*, 2000, pp. 1081–1084.

[2] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum-likelihood from incomplete data via the EM algorithm," *Journal of Royal Statistical Society B*, vol. 39, pp. 1–38, 1977.

[3] J. L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observation of markov chains," in *IEEE Transactions on Speech Audio Processing*, April 1994, vol. 2, pp. 291–298.

[4] Douglas A. Reynolds, Thomas F. Quatieri, and Robert B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1–3, 2000.

[5] C. Leggetter and P. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density HMMs," *Computer Speech and Language*, vol. 9, pp. 171–185, 1995.

[6] M. Gales, "The generation and use of regression class trees for MLLR adaptation," Technical Report TR 263, Cambridge University Engineering Department, 1996.

[7] R. Kuhn, P. Nguyen, J.C. Junqua, L. Goldwasser, N. Niedzielski, S. Fincke, K. Field, and M. Contolini, "Eigenvoices for speaker adaptation," in *Proceedings of the International Conference on Speech and Language Processing, ICSLP*, 1998.

[8] M. Kirby and L. Sirovich, "Application of the Karhunen-Loève procedure for the characterization of human faces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp. 103–108, 1990.