Eigenvoices: a compact representation of speakers in model space

Patrick Nguyen [1,2] (Patrick.Nguyen@eurecom.fr)
Roland Kuhn [1] (kuhn@stl.research.panasonic.com)
Jean-Claude Junqua [1] (jcj@stl.research.panasonic.com)
Nancy Niedzielski [1] (nn@stl.research.panasonic.com)
Christian Wellekens [2] (welleken@eurecom.fr)

[1] Speech Technology Laboratory
Panasonic Technologies
Suite #202
3888, State Street
Santa Barbara, CA 93105
USA

[2] Institut Eurecom
BP 193
2229, Route des Cretes
F-06904 Sophia Antipolis Cedex
FRANCE

Titre francais:
Voix propres: Vers une représentation compacte des locuteurs dans l'espace des modèles

Traduction du titre des figures:
Figure 1:
Schéma bloc d'un système de reconnaissance de la parole

Figure 2:
Schéma général du système de voix propres

Summary:
In this article, we present a new approach to modeling speaker-dependent systems. The approach was inspired by the eigenfaces techniques used in face recognition. We build a linear vector space of low dimensionality, called eigenspace, in which speakers are located. The basis vectors of this space are called eigenvoices. Each eigenvoice models a direction of inter-speaker variability. The eigenspace is built during the training phase. Then, any speaker model can be expressed as a linear combination of eigenvoices.
The benefits of this technique as set forth in this article reside in the reduction of the number of parameters that describe a model. Thereby we are able to reduce the number of parameters to estimate, as well as computation and/or storage costs. We apply the approach to speaker adaptation and speaker recognition. Some experimental results are supplied.

Résumé:
Cet article présente une nouvelle approche inspirée de la reconnaissance d'images, adaptée et appliquée à la parole. Nous construisons un espace vectoriel de dimension réduite, appelé espace propre (eigenspace), dans lequel les locuteurs se trouvent confinés. Cet espace est constitué de vecteurs caractéristiques appelés voix propres (eigenvoices). Chaque voix propre modélise une composante de variabilité inter-locuteur. L'espace propre est construit lors de la phase d'apprentissage classique pour des systèmes liés à la parole. Un modèle du locuteur est par la suite associé à une combinaison linéaire des vecteurs de l'espace réduit des locuteurs.
L'avantage de cette méthode, mis en avant dans l'article, est la réduction du nombre de paramètres caractéristiques d'un modèle. De ce fait nous réduisons le nombre de paramètres à estimer, ainsi que le temps de calcul et/ou de stockage. Cette technique est ici appliquée à l'adaptation du locuteur pour un système de reconnaissance automatique du locuteur et à la reconnaissance automatique du locuteur. Quelques résultats expérimentaux sont présentés à cette occasion.

1 Introduction

This paper describes the concept called eigenvoices in the context of coding. The eigenvoices technique, which is relatively new, is known best in the speaker adaptation community [KNJ99]. However, as it was first invented, the concept has broader ambitions and is potentially applicable to a wide variety of tasks. It can be viewed as the compression of models in a system that has to accommodate a large number of speaker-dependent subsystems.

The remainder of this section introduces Eigenfaces. Then we continue with a general overview of the eigenvoices approach.

1.1 Inspiration: eigenfaces

The initial inspiration as well as the name of the technique stem from the inventive brilliance of the image recognition community. Here, we briefly depict the technique invented by Turk and Pentland [TUP91] and provide the framework for later discussion. Simply put, the task of recognizing a face consists in selecting the most similar face in a database, given that faces are 2D pixel images. If we apply classical signal processing techniques, the problem rapidly becomes computationally intractable. Researchers soon understood that the problem of recognizing faces was in nature a lot more confined than that of recognizing arbitrary photographs. This simple observation leads us to apply a mechanism well known in statistical social sciences or biology, where the useful factors are hidden in a mass of data: we can make use of the Principal Component Analysis (PCA, [JOL86]).

Instead of working with the array of pixels itself, we work on a representation of the data that is much simpler, that is, much smaller in size. One approximates the 2D image of a face as a linear combination of base face images (eigenfaces). *Eigenfaces* are obtained by taking the components associated with the largest energy that result from a singular value decomposition of the autocorrelation of the face database, hence their name. Let X be the database, i.e., a matrix formed with faces as columns, and as many columns as faces. A face is a D dimensional vector, where $D=m \times n$, if the image is a black and white $m \times n$ photograph. Let N be an orthogonal $D \times D$ matrix containing the eigenfaces, and S be a diagonal matrix representing variational energies associated with each eigenvector. By definition, we have

$$XX^T = N^T SN$$

where $(\cdot)^T$ denotes transposition. We truncate N to yield M (take the first E vectors). E is said to be the dimensionality of the eigenspace. Each face y can hence be approximated with

$$y \cong MM^T y$$

The sub-dimensional representation of y is thus $w = M^T y$, an E-dimensional vector. Recognition of a new face, say y, now consists of selecting the index of $M^T X$ which is the closest in terms of Euclidean distance of $M^T y$, i.e. if S is the number of speakers in the database, then this yields S inner products of size E ($E << S << D$). The straightforward, canonical approach would consist in computing T inner products of size D. Note that a useful corollary of the reduction of dimension is an embedded process of noise reduction. In the context of face recognition, the approximation discards such unimportant features as (say) background landscape. Not only does PCA reduce costs in computation, it also cancels irrelevant features. Also, semantics can be associated with eigenfaces, for instance the presence or absence of a beard (or glasses, etc.) might be associated with a particular eigenface dimension.

How is this interpreted in the context of model reduction? As a pattern matching task, our face recognition problem can be viewed as

1. enrollment of people
2. pattern matching (selecting the most similar person)

With the addition of PCA, it becomes:

1. enrollment

2. discovery of eigenfaces
3. pattern matching in the eigenspace

Models (faces) are now reduced in size and we can work in a simpler, cleaner setup.

1.2 Eigenvoices: conceptual overview

Speech recognition systems are in nature more convoluted than our previous example. Whereas face recognition involves models that allow for simple distance measure in the form of the canonical (possibly weighted) Euclidean distance, statistical-based speech recognition systems make use of Hidden Markov Models (HMMs) that complicate the process in terms of mathematical and computational tractability. In speech, there is a clear distinction between features and models.

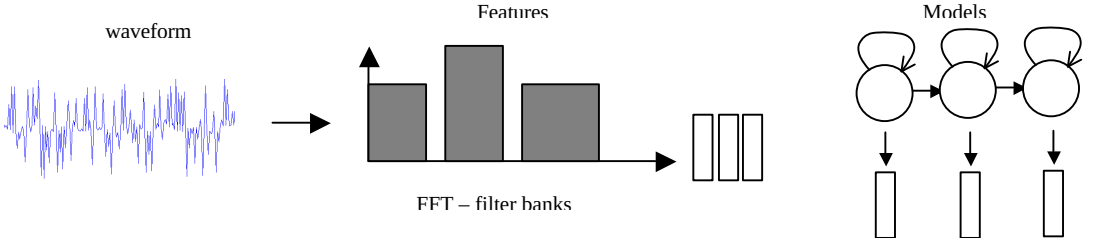

Figure 1. Block-diagram of a speech recognition system

Figure 1 represents components of a speech recognition system. First, speech is recorded through a microphone. The waveform is then transformed into a sequence of feature vectors. Models are a statistical explanation for the sequences of feature vectors.

The next question is: Where do we apply dimensionality reduction?

Influences of speaker characteristics occur at all points in time and therefore propagate from the production of speech itself to models. For instance, a speaker with disorders in the laryngeal area may tend to speak slower, knowing his disability and the implied

decrease in intelligibility, will produce sounds with a different glottal pulse, and will be modeled with different linear prediction filters, etc. One generally distinguishes between three levels where it is possible to apply dimensionality reduction. We can work within the featurization process itself, or as a post processor to features, or finally on the models: we refer to those as subfeature, feature, or model domains respectively. Let us review these domains briefly.

The subfeature domain largely depends on the type of feature vectors in use: each feature vector system implies different assumptions or approximations. Features can be such entities as energies in certain frequency bands, representing speech at a certain point in time as supposedly heard by a human ear as opposed to a microphone. Popular features include Mel-Frequency Cepstral Coefficients (MFCCs) or Perceptual Linear Prediction (PLP [HER90]). A source for inter-speaker variability, for instance, is known to be length of the vocal tract. One can compensate for it using frequency warping. Subfeature transformations can take almost any analytical or parametric form and any number of parameters (but typically from 1 to 5).

Feature transformations have also been investigated. In fact, a large set of speaker recognition techniques make use of features only and discard linguistic information (which is coded by models). Bi-linear transformation of the feature vectors is a good example of a non-linear normalization for speaker variability. A typical dimensionality of the feature space ranges from 10 to 50.

Lastly, models are HMMs in statistical-based speech recognition. They have an intrinsically large number of parameters (typically 2k-400k). Maximum-likelihood linear regression (MLLR [LEW95]) is a paramount example of model-based transformation.

The eigenvoices approach operates in the model domain. Given the dimensionality of the domains, it is where we can achieve the most drastic gain in dimensionality. Surprisingly, previous research on the application of dimensionality reduction techniques to speech recognition focused on dimensionality reduction in the feature space rather than model space, even though much greater cardinality of model space offers much more scope for dimensionality reduction. The cardinality and topology of the true speaker space is unknown.

We have discussed eigenvoices mainly for speech recognition. However, speech coding systems could also benefit from a dimensionality reduction. For instance, if we can transmit reference glottal pulses in a Code Excited Linear Prediction (CELP) vocoder prior to communication we can improve on the quality of the system.

2 Eigenvoices

This section is devoted to a more detailed description of the so-called eigenvoices approach. First, we describe the general eigenvoices concept for any speech system. Then, we show two applications of the idea, namely speaker adaptation and speaker recognition. We develop the mathematical specialization of eigenvoices in those contexts. Experimental results are also provided.

2.1 Speech systems

As with face recognition systems, we have two separate steps when working with speech

recognition systems. First, we gather information to construct HMM models and prior information. This is referred to as the *training* phase. Work accomplished here is said to be *offline*, because it is done once, and not when the system is deployed. Second, we use the system with the goal of recognizing either speech or a person. Reduction of the dimensionality of the system is performed during the first step, so that the second step becomes easier. Reduction of the number of parameters not only saves space and time, it also improves on the quality of their estimation, given a finite (and small) amount of data. The eigenvoices approach consists of conjecturing that the space spanned by speaker models is a simple vector space. If l is the model of a speaker, then we have

$$l = \sum_{e=1}^{E} w(e)\,\bar{l}\,(e), \qquad e = 1,...,E$$

where $\bar{l}$ is the eigenvoice, and $w(e)$ is a value specific to our speaker in that direction. The speaker space *M* is thence given by the set of $\bar{l}$'s and each speaker is depicted by a vector of *characteristics* $w = [w(1), w(2),..., w(E)]^T$. The eigenvoice assumption is therefore equivalent to stating that any speaker model can be written as a linear combination of eigenvoices.

Let us now quickly review the process of applying eigenvoices. As stated previously, we proceed in two phases:

1. Train the system: build speaker-independent systems and the eigenspace. For reasons that will become obvious later the eigenspace is a subset of what is called prior information or set of hyperparameters.
2. Deployment of the system: this corresponds to the test adaptation phase of a recognition system. Here, we use our prior knowledge to ease or improve the

execution of our task.

In the training phase, we observe the distribution of speakers in the model space, $\Re^D$: we observe a large number of speakers (typically 100-500), each of whom is associated with a vector of dimension *D* describing the model for that speaker. If we consider means adaptation, the vector is the concatenation of all means of all gaussians of the set of HMM models of that speaker. As *D* increases the speaker-dependent system has more and more degrees of freedom, and can thus model speech more accurately. *D* ranges between about 2k-400k. Thus, we see a collection of points scattered in a *D*-dimensional space, but which we contend are confined in a small space.

Before we apply such an algorithm as PCA, in most cases it is useful to apply a so-called *whitener*. This procedure pre-processes our data to remove mean and inter-correlations of parameters under the Gaussian assumption, and hence produces data that are as close as possible to being white noise, hence its name. Since PCA works with Euclidean distances, this serves intuitively as a renormalization of features. After this step we could possibly apply an algorithm for detecting and removing outliers.

Then, we apply a dimension reduction algorithm, such as PCA, ICA (Independent Component Analysis [COM94]), or LDA (Linear Disciminant Analysis [FUK72]) to obtain the basis vectors of our linear subspace. ICA yields a possibly non-orthogonal basis (with the canonical inner product) for which eigenvalues are not correlated. In other words, the information that we have given one eigenvalue gives absolutely no information or constraint on any other eigenvalue. It is regarded as a very good theoretical solution albeit computationally heavy and sensitive to pre-processing in practice. LDA

returns the set of eigenvoices that has the best discriminative properties, that is, eigenvoices for which we can differentiate cleary from each speaker. ICA finds a space optimal in the sense of mutual information, and LDA in the sense of discrimination. Another popular optimality criterion in speech recognition is that of Maximum-Likelihood (ML). This leads to the algorithm called MLES that will be described further down. Once we have performed that reduction, we are left with *E eigenvoices* that form the eigenspace. The eigenspace *M* is a matrix of the form *D* x *E*. The eigendimension *E* is set in practice to about 10 to 50. This is a heursitic compromise dictated by complexity and performance. As we increase the number of degrees of freedom, we increase computational complexity. Additional degrees of freedom allow for more precision and therefore better recognition results, to the expense of an increase in requirements of adaptation data.

Now that we have constructed our prior knowledge, we can proceed to deploy our system and recognize speakers. In this phase, we are now reduced to the localization of a speaker in eigenvoice space, that is, estimating the eigenvalues vector *w*, which has only *E* parameters. If so desired, we can create the corresponding *D*-dimensional model.

Figure 2 summarizes the steps involved in eigenvoices.

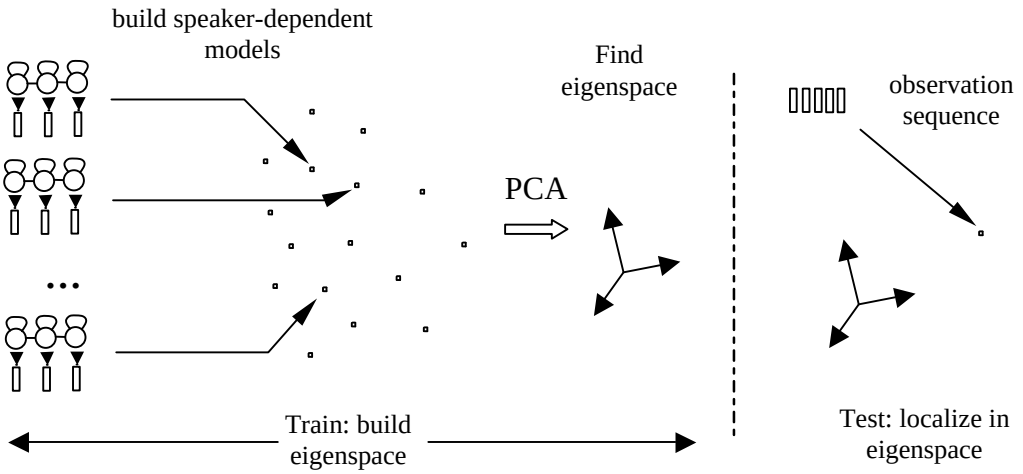

Figure 2. Overview of eigenvoices

In the next subsection we apply the concept to speaker adaptation and HMM-based speaker recognition.

2.2 Adaptation

The first and foremost application of the eigenvoices approach is speaker adaptation. We will explore the benefits and drawbacks of the eigenvoices approach in depth. Remember that the goal of eigenvoices is to build speaker-dependent models with as few parameters as possible. In the previous sections we have talked about how to recognize a speaker amongst other speakers in the database. Adaptation can be regarded as a speaker recognition task where we have no discriminative constraint.

2.2.1 What is speaker adaptation?

As stated previously there are different stages in a speech recognition system where we can apply adaptation. In this paper, we will only deal with adaptation of models.

Basically, speaker adaptation of model consists of modifying HMM parameters given a speaker-specific utterance so that future or current utterances of the speaker will be recognized more accurately.

There are four types of parameters that we can alter in an HMM model: mean vectors, covariance matrices, mixture weights, and transition probabilities. Other parameters regarding topology are supposed to be fixed in advance. In this paper we only deal with adaptation of the means. These parameters are believed to have the most important impact on the performance of the recognition task. Variances are second in importance after means but due to mathematical tractability we will suppose that they are known and constant throught the adaptation process.

If the text is known, then the adaptation is said to be supervised. When we have a very small amount of adaptation data, we designate the modification process as fast adaptation. The most difficult (and also most useful) case is unsupervised fast adaptation.

2.2.2 Prior art

Roughly stated, there are two kinds of adaptation techniques: smoothing techniques and constrained estimation. Smoothing techniques include variations of the deleted (linear) interpolation or Maximum A Posteriori (MAP [GAL92]). In such techniques, programmatically, we are reduced to combine statistics of the Baum-Welch algorithm for each parameter. Generally, they converge. Constrained adaptation makes use of indirect parameters: we estimate a smaller set of parameters (for instance coefficients of a linear transformation), and apply it to HMM parameters in an additional step. Such methods are typically more complex and do not converge, but require less adaptation data. Maximum

likelihood linear regression and its bayesian variants (MLLR [LEW95] and MAPLR[CHO99, GOK99]), speaker clustering [AHA96, SAM98], Reference Speaker Weighting (RSW, [HAZ98]) and Eigenvoices are paramount examples of such techniques. Best results are usually obtained by using smoothing after constrained estimation. The resulting algorithm gets quickly to a reasonable estimate and is further ameliorated by smoothing.

2.2.3 Optimal estimators for eigenvoices

As noted earlier, in the eigenfaces approach we use simple Euclidean distances to perform pattern matching. Due to the mathematical expression of HMMs in speech recognition, our optimality criterion is no longer the minimum distance but the maximum-likelihood (ML) or maximum a posteriori (MAP).For that purpose, we will enhance the discovery of the eigenspace and localization in the eigenspace. PCA and canonical projection perform these tasks using Euclidean distances. In the following subsection we develop ML versions, named MLES and MLED respectively. Discovery of the eigenspace takes places during training. MLED is performed during testing.

2.2.3.1 An optimal estimator for the location in the eigenspace: MLED

In this section, we derive the optimal estimator for eigenvoices with regards to the maximum-likelihood criterion. This estimator is called MLED, for Maximum-Likelihood Eigen Decomposition [KNJ99]. For that purpose we can use the well-known theory of HMMs. Due to the hidden nature of HMM parameters, we have to apply the Expectation-

Maximization (EM, [DLR77]). Our goal is to maximize the likelihood of the observation sequence *O* given our model parameters θ:

$$\hat{q} = \arg\max_{q} L(O \mid q)$$

with L(.) the likelihood function. The EM algorithm tells us to maximize the auxiliary function, which is the expectation of the likelihood from our current estimate θ_0

$$Q(q, q_0) = \mathrm{E}\left[\log L(O, z \mid q) \mid O, q_0\right]$$

and ζ the hidden data estimated with θ_0. It can be shown that, for the adaptation of the means, this is equivalent to optimizing

$$Q_b(m, m_0) = -\tfrac{1}{2} L(O \mid m_0) \sum_m \sum_t g_m(t) \left[(o_t - m_m)^T C_m^{-1} (o_t - m_m) \right]$$

with the following definitions:

μ_m	mean parameters
o_t	observation vector at time *t*
C_m^{-1}	inverse covariance (precision) matrix
$\gamma_m(t)$	posterior probability of seeing o_t at that time with that mixture

And *m* denotes a Gaussian component.

By definition, the central conjecture of eigenvoices leads us to express model means μ as a linear combination of eigenvoices $\overline{m}(e)$,

$$m_m = \sum_{e=1}^{E} w_e \overline{m}_m(e)$$

There are *E* eigenvoices. Thus, we need to maximize the auxiliary function *Q* with our

eigenvalues *w*. Replacing into the auxiliary function and differentiating with each eigenvalue, one obtains the following system of equations

$$\sum_m \sum_t g_m(t) \bar{m}_m^T(e) C_m^{-1} o_t = \sum_m \sum_t g_m(t) \sum_{j=1}^{E} w_j \bar{m}_m^T(e) C_m^{-1} \bar{m}_m(j), \qquad e = 1,...,E$$

to solve for the eigenvalues w_e. It is a linear system of equations that one can solve with Singular Value Decomposition or Gaussian elimination.

2.2.3.2 An optimal estimator for the eigenvoices: MLES

In the previous developments, we assumed that eigenvoices were derived with PCA. However, this approach is suboptimal. PCA minimizes the Euclidean distance between two models, and therefore does not maximize likelihood and requires gaussians to be aligned, i.e. Gaussian 1 of gaussian mixture of state 1 of model 1 is aligned with its peer in the other model. In the terminology of estimation theory, PCA yields the least squares estimator whereas we would need a maximum-likelihood estimator. Also, applying PCA to a large set of speakers requires large amounts of memory. Lastly, we would like to integrate a priori knowledge that we have about the database into our estimation. This arises in particular when properties of the training database do not match with those of the deployment conditions.

The Maximum-Likelihood EigenSpace or MLES [NWJ99] for short overcomes these limitations. Again, we use EM, but this time extend the hidden data ζ with *w*. The parameters θ that we need to estimate are the eigenvoices. Let the eigenspace M be $M = \left[\bar{m}^T(1),...,\bar{m}^T(E)\right]^T$, a D x E matrix. Replacing into the EM formulation we get:

$$\hat{M} = \arg\max_{M} \sum_{q=1}^{T} \int \log L(O, w \mid M) P_0(w, q) \mathrm{d}w$$

where q is an index that denotes the speaker. $P_0(.)$ is a weighting function that accounts for a priori information about the speaker given his characteristics. For instance, if there is a large bias in the favor of the number of male speakers in the database as is often the case, we would tend to model males better. In the deployment phase we would need the same proportion of male vs female speakers. With this probability we can cancel the bias. Since EM is an iterative algorithm, we need to have an initial estimate. It can be obtained with PCA, ICA or LDA. When eigenvalues are not fully set by a priori knowledge, we can use MLED to estimate the most likely values. The mathematical derivation of the eigenvoices is very similar to the derivation of Baum-Welch formulae and yields

$$\overline{m}_m(e) = \frac{\sum_q L_q w_q(e) \sum_t g_m(t) \left[o_t - \tilde{m}_m^{(q)}(e) \right]}{\sum_q L_q w_q^2(e) \sum_t g_m(t)}$$

with $L_q = L(O^{(q)} \mid w_q(e)) P_0(w, q)$ the posterior probability of the speech utterances of the speaker. We define the complement of the eigenvoice with respect to the observation

$$\tilde{m}_m^{(q)}(e) = \sum_{k=1, k \neq e}^{E} w_q(k) \overline{m}_m(e)$$

Intuitively, it represents the residual error not modelled by other dimensions. Retraining of the eigenvoices is similar to a Baum-Welch training pass, but we need E accumulators instead of one, and for each observation sequence we have an embedded EM step for MLED. Practically, MLES needs approximately twice as many iterations as training a speaker-independent model.

2.2.4 Experiments

In our experiments we have used the TIMIT database. It is divided into two sets: a training set and a test set. The training set comprises of 462 speakers. For each of the speakers, we have recordings of 8 sentences. The approximate duration of a sentence is about 2-7 seconds. We used 30 speakers for the adaptation experiments. The eigenspace was built using all 462 speakers of the training set and tested on the 30 speakers of the test set, using one supervised adaptation sentence and recognition on the 7 remaining sentences.

The speech recognition system uses 18 features (9 static PLP parameters including energy and 9 delta) for phoneme recognition. The sampling rate was set to 16 kHz. There were 48 context-independent models of phonemes, with 3 states each and 16 gaussians per mixture. The baseline recognition score is 60.94% accuracy. The next table summarizes the results. We have tested several eigenspaces: PCA, and MLES trained with different dimensions (E=10, 20, 50). Increasing *E* beyond 50 yields only marginal improvements in performance. These spaces were tested with different dimensions (E=5, 10, 20, 50). Understandably, the number of dimensions with which an eigenspace is tested cannot exceed the number of dimensions for which was trained. The more dimensions we allow eigenvoices to use, the more precision we have and the better the recognition scores. Moreover, to obtain best performance with MLES, we must know the dimensionality of the eigenspace in advance.

A dimension of the eigenspace of about 10 yields decent results.

Method	E=5	E=10	E=20	E=50
PCA	60.67	60.58	61.29	61.56
MLES (E=10)	62.53	65.10	-	-
MLES (E=20)	63.06	65.01	65.37	-
MLES (E=50)	61.74	63.77	64.84	66.96

Note that in that range of amount of adaptation data (2-7 seconds), MLLR followed by MAP performs badly: it yields a decrease in performance (59.64%). The reason is that we do not have enough data to estimate MLLR parameters reliably. To the best of our knowledge eigenvoices is the most rapid adaptation technique to date. In our problem, MLLR was used with a full matrix shared amongst all phonemes, and has 18x19 parameters. Eigenvoices has E=5, ... 50 parameters. MAP has the same dimensionality as ML, which is the number of parameters of the system (about 40k). As a result, MLLR needs about 3-4 sentences before it can be used efficiently whereas MAP by itself would require a minute or more.

2.2.5 Summary

In this section, we have presented how eigenvoices enabled improvements in speaker adaptation. In particular, dimensionality reduction involves the following benefits:

- Reliable parameters: since we have fewer parameters to estimate, we have more data per parameter

- Reduction of noise: when reducing the degrees of freedom of the system, we also delete noisy dimensions (i.e. dimension associated with intra-speaker variability as opposed to inter-speaker variability)
- Correct adaptation to unseen data: ML methods do not update unseen parameters. However we observe indirect parameters and can therefore see all parameters of the model.
- The eigenvoices store the mapping from low- to high-dimensional problems, and thereby model the internal consistency of the speaker. Since an eigenvoice can be associated with a characteristic, it means that if we have models for the phoneme ‘aa’ and ‘ae’, hearing an ‘aa’ from a female, we can update ‘ae’ with female characteristics.
- The simple linear formulate yields simpler (computationally easier) update formulae than MLLR
- The softness of the approach allows us to choose from an infinite set of speakers, as opposed to speaker clustering
- An intuitive interpretation of eigenvoices as directions of variability modeling speaker characteristics can be useful. For instance, the first eigenvoice has been identified as modeling gender

2.3 Speaker Recognition

The most obvious task to which the eigenvoices is speaker identification. Basically, speaker identification consists of finding the identity of a speaker given some utterance. It

can be viewed as a discretized adaptation process. It is a subset of problems that are classified under the name of speaker recognition. For instance, we may postulate some speaker identity and the system has to find out (given first and second order probabilities) whether the speaker corresponds to that identity or not. This is called speaker verification. A somewhat hybrid problem occurs when we must identify a speaker in an open set of speakers: the person may or may not be in the database, we must identify him correctly or reject as unknown. There are a number of names designating speakers given the reaction of the system, such as goat, sheeps, etc. They may fool the system unintentionally, or may get rejected for no apparent reasons, etc.

Applications of speaker recognition include biometric authentification or speaker segmentation. Speaker recognition may depend on a specific text or not, in which case it is said to be text-dependent or text-independent.

As with speaker adaptation, there are a number of stages where differences between speakers have an impact: by looking at features [BMM95], or by using statistical models such as HMMs. An HMM is called a Gaussian Mixture Model (GMM) when there is just one state. GMMs are popular [REY95] in speaker verification. HMMs [FOR95, RLS90] have also been used for speaker recognition but tend to be text-dependent.

In this paper we choose an HMM-based approach in a text-dependent context.

2.3.1 How do we measure performance?

In speaker identification, performance can be measured in a simple way: we count the percentage of speaker who are correctly recognized. Speaker verification, however, is a

hypothesis testing problem and as such is associated with two error probabilities: the false acceptance (FA) and false rejection (FR) rates. FA occurs when a speaker is wrongfully accepted into the system, FR occurs when the speaker corresponds to his claimed ID but gets rejected by the system. A high FA means that the system is too permissive, i.e., would let anybody enter. On the other hand, a high FR has a tendency to reject legitimate attempts too often. To combine the two in one single performance figure, the equal error rate (EER) is defined in the litterature (e.g. [FOR95]), which is the error rate for which FA equals FR. FA and FR are monotonically decreasing functions of each other.

2.3.2 Prior art

Rose and Reynolds [ROR90] introduced GMMs for speaker identification. They build a GMM l_s for each speaker. Speaker identification is then equivalent to selecting the speaker model with highest likelihood:

$$\hat{s} = \arg\max_{s} L(O \mid l_s)$$

This operation requires the computation of the likelihood for all speakers in the database (exhaustive search), which can be prohibitive in certain contexts.

For speaker verification, we face the problem of normalizing likelihoods. We introduce the concept of background speakers [REY95] or cohort [RLJ92]. While identifying a speaker, we compute the self match score against the cohort match score:

likelihood ratio = P(*X* is the claimed speaker) / P(*X* is not the claimed speaker)

Choosing the set of speakers that will form the cohort is difficult. The size of that set is user-defined, but again we have to perform a direct match with a set of speakers.

HMM-based systems are essentially similar to what we just described. They also require the introduction of likelihood ratios to normalize scores. Discriminative Observation Probabilities (DOPs) [FOR95] are an interesting alternative to likelihood ratios: we use an explicitly discriminative measure instead of a ratio. Also note that speaker selection (or cluster selection), that was presented before as an adaptation technique, is also a speaker identification technique. Interestingly, we build a cluster tree when building clusters. Thus, it might be desirable to apply hierarchical classifier theory to reduce computation [SAM98]. Instead of computing the match with each and every other speaker in the database, we search down the tree. Unfortunately, this approach is suboptimal.

2.3.3 Eigenvoices

Eigenvoices will enable us to reduce computations and model complexity. Following the eigenfaces example, we consider the sets of model parameters as simple random variables. As with speaker adaptation, we locate an unknown speaker in the eigenvoice space. Then, eigenvoices-based speaker identification consists of selecting the closest speaker in that low dimensional linear vector space (a simple geometrical problem). For speaker verification we reject or accept the speaker based on how close the person is to the claimed speaker. If we are in the hypersphere of a predefined radius, we accept. The larger the radius, the larger the FA.

Note that proximity in the eigenspace does not imply high likelihood score. In other words, the approach is suboptimal in the sense that choosing the closest speaker in

eigenspace may not yield the speaker with highest likelihood score.

To get a clearer understanding of the process in use, we describe a closed-set speaker recognition system utilizing eigenvoices.

1. Building the eigenspace
    - enrollment
    - dimensionality reduction
2. Speaker recognition: locate most similar speaker in database or determine if the speaker is close enough to the reference of his claimed identity

In the test phase, the system also requires an exhaustive search though the entire database, but the cost of computing the match is that of a simple inner product in the eigenspace. For classic systems, the most advantageous example is GMM where we would need as many inner products in the feature space as the number of mixtures per model times the number of observations. For HMM systems it would be an entire Baum-Welch per speaker.

The theoretically correct similarity measure is divergence: when we observe a model, we want to select the model that explains the observation as closely as possible. However, for HMMs and GMMs divergence is parametric but there exists no closed form expression. Thence, if we have to recompute likelihoods for each model using a Viterbi or Baum-Welch algorithm, we lose our advantage. We need to find alternative ways of computing distances (e.g. [BMS98]). Therefore, we use simpler closed-form similarity measures in

the eigenspace. The canonical inner product is chosen. As a first approach, we used the Euclidean distance [OLS98] between speakers *r* and *q*:

$$d(r,q) = \left\| w_r - w_q \right\|$$

where || . || denotes the 2-norm, ie $\left\| x \right\| = < x, x >^{1/2}$, with $< \cdot, \cdot >$ the inner product.

A better similarity measure happened to be the normalized correlation. Normalizing is interesting in that we become independent of the norm of the eigenvalues vectors.

$$d(r,q) = \frac{< w_r, w_q >}{\left\| w_r \right\| \left\| w_q \right\|}$$

2.3.4 Experiments

Experiments for Speaker Verification and Speaker Identification were carried out on the speech recognition system that we have described previously in our speaker adaptation experiments. The eigenspace was trained on all 462 speakers of the training database. Then we used 150 speakers of the TIMIT test set to evaluate the system. Therefore, the eigenspace was trained on one set and tested on another non-overlapping set. Thus we need not rebuild the eigenspace when a new speaker is enrolled. The dimension of the eigenspace was set to 20 arbitrarily. Note the reduction in storage space: the system that does not use eigenvoices stores all HMM parameters (40k) for all speakers (150). The eigenvoices-based system, on the other hand, stores the eigenspace (40k x 20) and an eigenvalues vector for each speaker (20 x 150).

For speaker identification results, we obtained 100% identification score, that is, all speakers were recognized correctly amongst a base of 150 speakers. This figure is

attained quite frequently in the litterature, especially given our test conditions. For speaker verification, all speakers acted as impostors: each of them claimed every 150 identities. In that experiment we obtained a 2% EER. Given that we have high-quality speech these results are relatively modest but encouraging.

3 Conclusion

In this paper, we have explained the concept of eigenvoices as a means of representing speakers compactly in model space. Furthermore, we describe applications to speaker adaptation and speaker recognition with specialization of the formulae for HMMs. Thanks to the reduction in dimensionality of speaker-dependent models, we improve alternatively on several aspects of the process. We decrease the complexity and the number of parameters involved. With fewer parameters, given small amount of data, we can build more robust models, since we have fewer parameters to estimate. As with image recognition, fewer parameters also means fewer computations and reduction of memory requirements for speaker identification. We also believe that decomposing parameters into speaker characteristics yields a more elegant and intuitive approach. Speaker adaptation was the most thoroughly explored application. We hope that researchers in the speaker recognition and speech coding will adopt the idea.

References

[AHA96] Ahadi-Sarkani S., "Bayesian and Predictive Techniques for Speaker Adaptation", Ph.D. Thesis, 1996, Cambridge University.

[BMM95] Bimbot F., Magrin-Chagnolleau I., Mathan L., "Second-Order Statistical Measures for Text-Independent Speaker Identification", *Speech Communication*, 1995, vol 17., pp. 177-192.

[BMS98] Beigi H. S. M., Maes S. H., Sorensen J. S., "A Distance Measure Between Collections of Distributions and Its Applications to Speaker Recognition", Proceedings of the *International Conference on Acoustics, Speech and Signal Processing* (ICASSP), 1998, V. 2, pp. 753-757.

[CDS97] Chen S. and De Souza P., "Speaker Adaptation by Correlation (ABC)", Proceedings of *Eurospeech*, 1997, pp.2111-2114.

[CHO99] Chou W., "Maximum a Posterior Linear Regression with Elliptically Symmetric Matrix Variate Priors", Proceedings of *Eurospeech*, 1999, V. 1, pp. 1-4.

[COM94] Comon P., "Independent Component Analysis, a new concept?", Signal Processing, 1994, V. 36, No. 3, pp. 287-314.

[DLR77] Dempster A. P., Laird N.M. and Rubin D. P., "Maximum-Likelihood from Incomplete Data via the EM algorithm", Journal of the Royal Statistical Society, 1977, Vol. B, pp. 1-38.

[FOR95] Forsyth M., "Hidden Markov Models for Automatic Speaker Verification", PhD thesis, University of Edinburgh, 1995.

[FUK72] Fukunaga K., "Introduction to Statistical Patter Recognition", 1972, Academic Press, New York and London.

[GAL92] Gauvain J.-L. and Lee C.-H., "Bayesian Learning for Hidden Markov Model with Gaussian Mixture State Observation Densities", Speech Communications, 1992, V. 11, pp. 205-213.

[GAL97] Gales M. F. J, "Transformation Smoothing for Speaker and Environmental Adaptation", Proceedings of *Eurospech*, 1997, pp. 2067-2071.

[GAL98] Gales M.F.J., "Cluster Adaptive Training for Speech Recognition", Proceedings of the *International Conference on Speech and Language Processing* (ICSLP), 1998, V. 5, pp. 1783-1786.

[GAW96] Gales M. F. J. and Woodland P., "Mean and Variance Adaptation within the MLLR Framework", Computer Speech and Language, 1996, V. 10, N. 4, pp. 250-264.

[GOK99] Goronzy S. and Kompe R., "A MAP-Like Weighting Scheme for MLLR Speaker Adaptation", Proceedings of *Eurospeech*, 1999, V. 1, pp. 5-8.

[HAZ98] Hazen T., "The Use of Speaker Correlation Information for Automatic Speech Recognition", PhD Thesis, 1998, MIT.

[HER90] Hermansky H., "Perceptual Linear Predictive (PLP) Analysis of Speech", Journal of the American Society of Acoustics (JASA), 1990, Vol. 87, N. 4, pp. 1738-1752.

[JOL86] Jolliffe I. T., "Principal Component Analysis", Springer-Verlag, 1986.

[KAO97] Kannan A. and Ostendorf M. "Modeling Dependency in Adaptation of Acoustic Models Using Multiscale Tree Processes", Proceedings of *Eurospech*, 1997, pp. 1863-1867.

[KNJ99] Kuhn R., Nguyen P., Junqua J.-C., Boman R., Niedzielski N., Fincke S., Field K. and Contolini M., "Fast Speaker Adaptation in Eigenvoice Space", Proceedings of the

International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 1999, V. 2., pp 749-752.

[NWJ99] Nguyen P., Wellekens C. and Junqua J.-C., "Maximum-Likelihood Eigenspace and MLLR for Speech Recognition in Noisy Environments", Proceedings of *Eurospeech*, 1999, V. 6, pp. 2519-2522.

[]LEW95] Legetter C. J., and Woodland P. C., "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models", Computer Speech and Language, 1995, V. 9, pp. 171-185

[OLS98] Olsen J., "Speaker Recognition Based On Discriminative Projection Models", Proceedings of the *International Conference on Speech and Language Processing* (ICSLP), 1998, pp. 1919-1922.

[REY95] Reynolds D. A., "Speaker Identification and Verification Using Gaussian Mixture Speaker Models", Speech Communication, vol 17, 1995, pp. 91-108

[RLJ92] Rosenberg A.E., Lee C.-H., Juang B.-H. and Song F.K., "The Use of Cohort Normalized Scores for Speaker Verification", Proceedings of the *International Conference on Acoustics, Speech, and Signal Processing* (ICASSP), 1992, pp. 262-272.

[RLS90] Rosenberg A.E., Lee C.-H., Song F.K. and McGee A., "Experiments in Automatic Talker Verification using Sub-Word Unit Hidden Markov Models", Proceedings of the *International Conference on Speech and Language Processing* (ICSLP), 1990, pp. 141 -144

[ROR90] Rose R.C. and Reynolds D.A., "Text-Independent Speaker Identification Using Automatic Acoustic Segmentation", Proceedings of the *International Conference on Acoustics, Speech and Signal Processing* (ICASSP), 1990, pp. 293-296.

[SAM98] Suzuki M., Abe T., Mori H., Makino S. and Aso H., "High-Speed Speaker Adaptation Using Phoneme-Dependent Tree-Structured Speaker Clustering", Proceedings of the *International Conference on Speech and Language Processing* (ICSLP), 1998, pp. 2299-2302.

[TUP91] Turk M. and Pentland A., "Eigenfaces for Recognition", Journal of Cognitive Neuroscience, 1991, V.3, no 1, pp. 71-86.

[VIL98] Viikki O. and Laurila K., "Incremental Online Speaker Adaptation In Adverse Conditions", Proceedings of the *International Conference on Speech and Language Processing* (ICSLP), 1998, V. 5, pp. 1779-1782.