Reprinted from *IEEE Trans. on Speech and Audio Processing* 8(2):115-125 (2000).

Maximum Likelihood and Minimum Classification Error Factor Analysis for Automatic Speech Recognition

Lawrence Saul and Mazin Rahim
{lsaul,mazin}@research.att.com
AT&T Labs — Research
180 Park Ave E-171
Florham Park, NJ 07932

Abstract

Hidden Markov models (HMMs) for automatic speech recognition rely on high dimensional feature vectors to summarize the short-time properties of speech. Correlations between features can arise when the speech signal is non-stationary or corrupted by noise. We investigate how to model these correlations using factor analysis, a statistical method for dimensionality reduction. Factor analysis uses a small number of parameters to model the covariance structure of high dimensional data. These parameters can be chosen in two ways: (i) to maximize the likelihood of observed speech signals, or (ii) to minimize the number of classification errors. We derive an Expectation-Maximization (EM) algorithm for maximum likelihood estimation and a gradient descent algorithm for improved class discrimination. Speech recognizers are evaluated on two tasks, one small-sized vocabulary (connected alpha-digits) and one medium-sized vocabulary (New Jersey town names). We find that modeling feature correlations by factor analysis leads to significantly increased likelihoods and word accuracies. Moreover, the rate of improvement with model size often exceeds that observed in conventional HMMs.

1 Introduction

Hidden Markov models (HMMs) for automatic speech recognition[21] rely on high dimensional feature vectors to summarize the short-time, acoustic properties of speech. Though front-ends vary from recognizer to recognizer, the spectral information in each frame of speech is typically codified in a feature vector with thirty or more dimensions. In most systems, these vectors are conditionally modeled by mixtures of Gaussian probability density functions (PDFs). In this case, the correlations between different features are represented in two ways[16]: implicitly by the use of two or more mixture components, and explicitly by

the non-diagonal elements in each covariance matrix. Naturally, these strategies for modeling correlations—implicit versus explicit—involve tradeoffs in accuracy, speed, and memory. This paper examines these tradeoffs using the statistical method of factor analysis.

The present work is motivated by the following observation. Currently, many HMM-based recognizers do not include any explicit modeling of correlations; that is to say—conditioned on the hidden states, acoustic features are modeled by mixtures of Gaussian PDFs with *diagonal* covariance matrices. The reasons for this practice are well known. The use of full covariance matrices imposes a heavy computational burden, making it difficult to achieve real-time recognition. Moreover, one rarely has enough data to (reliably) estimate full covariance matrices. Some of these disadvantages can be overcome by parameter-tying[2]—e.g., sharing the covariance matrices across different states or models. But parameter-tying has its own drawbacks: it considerably complicates the training and decoding procedures, and it requires some artistry to design the tied HMMs.

Unconstrained and diagonal covariance matrices clearly represent two extreme choices for the hidden Markov modeling of speech. The statistical method of factor analysis[24, 7] represents a compromise between these two extremes. The idea behind factor analysis is to map systematic variations of the data into a lower dimensional subspace. This enables one to represent, in a very compact way, the covariance matrices for high dimensional data. These matrices are expressed in terms of a small number of parameters that model the most significant correlations without incurring much overhead in time or memory. For this reason, we shall often refer to factor analysis as a strategy for dimensionality reduction[18]. However, it should be emphasized that factor analysis does not merely project a high dimensional feature vector into a low dimensional one.

In this paper we investigate the combined use of mixture models and factor analysis in HMMs for automatic speech recognition. Applying factor analysis at the state and mixture component level[9, 11] results in a powerful form of dimensionality reduction, one tailored to the local properties of speech. This approach effectively combines two popular strategies in probabilistic modeling—clustering and dimensionality reduction. The combination of these two methods gives rise to a *local linear model*—local, because each mixture component models different facets of the data, and linear, because the dimensionality reduction is geared to Gaussian mixture components. Many variants on local linear models have been proposed in the machine learning literature. Successful applications include handwritten digit recognition[11], data compression[14], nonlinear image interpolation[3], and articulatory modeling of electropalatographic data[4].

Factor analysis uses a small number of parameters to model the covariance structure of high dimensional data. For automatic speech recognition, these parameters can be chosen in two ways: (i) to maximize the likelihood of observed speech signals, or (ii) to minimize the number of classification errors[1, 12, 13, 17]. In this paper, we derive both an Expectation-Maximization (EM) algorithm for maximum likelihood estimation and a gradient descent algorithm for improved class discrimination. To the best of our knowledge, neither form of factor analysis has been previously applied to hidden Markov models for automatic speech recognition.

Briefly, the organization of this paper is as follows. In section 2, we review the method of factor analysis and describe what makes it attractive for large problems in speech recognition.

In sections 3 and 4, we present the learning algorithms for maximum likelihood (ML) and minimum classification error (MCE) factor analysis. In section 5, we give results on two tasks in automatic speech recognition: connected alpha-digits and New Jersey town names. Finally, in section 6, we present our conclusions as well as ideas for future research.

2 Dimensionality reduction

Factor analysis[24, 7] is a *linear* method for the dimensionality reduction of Gaussian random variables. It is closely related to principal components analysis (PCA)[6] in which one extracts the subspace defined by the largest eigenvectors of the covariance matrix. In traditional PCA, the data vectors are projected into this subspace, resulting in a simple form of dimensionality reduction. Though straightforward, PCA has two important disadvantages: (i) it does not define a proper density model outside the subspace of principal components; and (ii) componentwise variations outside this subspace are modeled uniformly, even when the data does not warrant such an assumption. These disadvantages are serious drawbacks for HMM-based speech recognition. The first undermines the probabilistic formulation of HMMs, while the second precludes the use of composite data vectors (e.g., cepstra plus delta-cepstra) that combine features at different scales. Factor analysis does not suffer from either of these drawbacks, though it does contain traditional PCA (and also probabilistic versions of PCA[23, 27]) as a special limiting case.

Having described factor analysis as a linear method, we should emphasize that its application is not limited to strictly Gaussian PDFs. The combined use of mixture densities and factor analysis—resulting in a *non-linear* form of dimensionality reduction—was first applied by Hinton et al[11] to the modeling of handwritten digits. Essentially, a mixture of factor analyzers patches together the linear subspaces defined by its individual components to parameterize a globally non-linear (but low dimensional) manifold.

In this section, we first review the method of factor analysis for multivariate Gaussian PDFs. We then extend the method to mixture models and continuous density HMMs. The problem of parameter estimation is considered separately in sections 3 and 4.

2.1 Multivariate Gaussian PDF

Let $\boldsymbol{x} \in \mathcal{R}^D$ denote a Gaussian random variable with mean $\boldsymbol{\mu}$. If the number of dimensions, D, is very large, it may be prohibitively expensive to estimate, store, multiply, or invert a full covariance matrix. The idea behind factor analysis is to find a subspace of much lower dimension, $f \ll D$, that captures most of the variations in $\boldsymbol{x}$. To this end, let $\boldsymbol{z} \in \mathcal{R}^f$ denote a Gaussian random variable with zero mean and identity covariance matrix:

$$P(\boldsymbol{z}) = (2\pi)^{-f/2} \exp\left\{-\frac{1}{2}\boldsymbol{z}^T\boldsymbol{z}\right\}. \tag{1}$$

We now imagine that the variable $\boldsymbol{x}$ is generated by a random process in which $\boldsymbol{z}$ is a latent (or hidden) variable; the elements of $\boldsymbol{z}$ are known as the *factors*. Let Λ denote an arbitrary $D \times f$ matrix, and let Ψ denote a *diagonal*, positive-definite $D \times D$ matrix. We imagine that $\boldsymbol{x}$ is generated by sampling $\boldsymbol{z}$ from eq. (1), computing the D-dimensional vector $\boldsymbol{\mu} + \Lambda\boldsymbol{z}$, then

adding independent Gaussian noise with variances Ψ_{ii} to each component of this vector. The matrix Λ is known as the *factor loading* matrix. The relation between $\boldsymbol{x}$ and $\boldsymbol{z}$ is captured by the conditional distribution:

$$P(\boldsymbol{x}|\boldsymbol{z}) = \frac{|\Psi|^{-1/2}}{(2\pi)^{D/2}} \exp\left\{-\frac{1}{2}\left[\boldsymbol{x}-\boldsymbol{\mu}-\Lambda\boldsymbol{z}\right]^T \Psi^{-1}\left[\boldsymbol{x}-\boldsymbol{\mu}-\Lambda\boldsymbol{z}\right]\right\}, \tag{2}$$

The marginal distribution for $\boldsymbol{x}$ is found by integrating out the hidden variable $\boldsymbol{z}$, or $P(\boldsymbol{x}) = \int \boldsymbol{dz}\ P(\boldsymbol{x}|\boldsymbol{z})P(\boldsymbol{z})$. The calculation is straightforward because both $P(\boldsymbol{z})$ and $P(\boldsymbol{x}|\boldsymbol{z})$ are Gaussian:

$$P(\boldsymbol{x}) = \frac{|\Psi+\Lambda\Lambda^T|^{-1/2}}{(2\pi)^{D/2}} \exp\left\{-\frac{1}{2}\left[\boldsymbol{x}-\boldsymbol{\mu}\right]^T \left(\Psi+\Lambda\Lambda^T\right)^{-1}\left[\boldsymbol{x}-\boldsymbol{\mu}\right]\right\}. \tag{3}$$

From eq. (3), we see that $\boldsymbol{x}$ is normally distributed with mean $\boldsymbol{\mu}$ and covariance matrix $\Psi+\Lambda\Lambda^T$. It follows that when the diagonal elements of Ψ are small, most of the variation in $\boldsymbol{x}$ occurs in the subspace $S(\Lambda)$ spanned by the columns of Λ. The variances Ψ_{ii} measure the typical size of componentwise fluctuations outside this subspace. In general, these variances may be quite different from one another. Hence, unlike PCA, eq. (3) can model a distribution in which two vectors with the same projection onto $S(\Lambda)$ have quite different likelihoods.

Covariance matrices of the form $\Psi+\Lambda\Lambda^T$ have a number of useful properties. Most importantly, they are expressed in terms of a small number of parameters, namely the $D(f+1)$ non-zero elements of Λ and Ψ. If $f \ll D$, then storing Λ and Ψ requires much less memory than storing a full covariance matrix. Likewise, estimating Λ and Ψ also requires much less data than estimating a full covariance matrix. Covariance matrices of this form can be efficiently inverted using the matrix inversion lemma[20],

$$(\Psi+\Lambda\Lambda^T)^{-1} = \Psi^{-1} - \Psi^{-1}\Lambda(I+\Lambda^T\Psi^{-1}\Lambda)^{-1}\Lambda^T\Psi^{-1} \tag{4}$$

where I is the $f \times f$ identity matrix. This decomposition also allows one to compute the probability $P(\boldsymbol{x})$ with only $O(fD)$ multiplies, as opposed to the $O(D^2)$ multiplies that are normally required when the covariance matrix is non-diagonal. We will refer to covariance matrices of the form $\Psi+\Lambda\Lambda^T$ as *factored* covariance matrices. As a theoretical aside, note that if we allow the number of factors to equal the number of dimensions (i.e., $f = D$), then we recover the ability to parameterize an arbitrary covariance matrix, M. This is done by choosing the columns of the factor loading matrix, Λ, to be parallel to the eigenvectors of M.

Viewing factor analysis as a latent variable model, we can also compute the posterior distribution, $P(\boldsymbol{z}|\boldsymbol{x})$. The statistics of this distribution are derived in appendix A. Computing the posterior statistics of the hidden variable $\boldsymbol{z}$ is an important step in the EM algorithm for ML factor analysis. We will return to the problem of parameter estimation in section 3.

2.2 Hidden Markov models

Nearly all speech recognizers rely on Gaussian PDFs to model the short-term properties of speech. These PDFs are associated with the state emission densities of continuous density

HMMs. Often, the dimensionality of acoustic feature vectors is too large for these PDFs to employ full covariance matrices. Instead, correlations are modeled implicitly by mixtures of Gaussians with diagonal covariance matrices. Intuitively, one might expect the mixture components to model discrete types of variability, such as the speaker's gender or local dialect. However, mixture models are not geared to modeling continuous types of variability, as might arise from coarticulation effects or background noise. Clearly, both types of variability—discrete and continuous—are important for building accurate models of speech. Factor analysis provides a way to model continuous variability without incurring the overhead associated with full covariance matrices.

Consider a continuous density HMM whose feature vectors, conditioned on the hidden states, are modeled by mixtures of Gaussian PDFs. A mixture of factor analyzers[9] is a mixture of Gaussian PDFs, each having the form of eq. (3). We can use these models as output distributions for each hidden state s:

$$P(\boldsymbol{x}|s) = \sum_c P(c|s)P(\boldsymbol{x}|s,c), \tag{5}$$

$$P(\boldsymbol{x}|s,c) = \frac{|\Psi_{sc} + \Lambda_{sc}\Lambda_{sc}^T|^{-1/2}}{(2\pi)^{D/2}} \exp\left\{-\frac{1}{2}\left[\boldsymbol{x} - \boldsymbol{\mu}_{sc}\right]^T \left(\Psi_{sc} + \Lambda_{sc}\Lambda_{sc}^T\right)^{-1} \left[\boldsymbol{x} - \boldsymbol{\mu}_{sc}\right]\right\}. \tag{6}$$

The mixture components in eqs. (5–6) are indexed by the subscript c. All of the recursive algorithms for computing statistics in HMMs can be applied to these models. The main difference—or advantage—is that here one can compute $P(\boldsymbol{x}|s)$ with fewer operations than is normally required for full covariance matrices. This is done by exploiting the special structure of factored covariance matrices, as shown by eq. (4). We will refer to these models as FA-HMMs and to HMMs with diagonal covariance matrices as DG-HMMs.

Clearly, an important consideration when applying factor analysis to speech recognition is the choice of acoustic features. In our experiments, we used a thirty-nine dimensional feature vector consisting of the first twelve cepstral coefficients (with first and second derivatives) and the normalized log-energy (with first and second derivatives). There are known to be correlations[16] between these features, especially between the different types of coefficients (e.g., cepstrum and delta-cepstrum). While these correlations have motivated our use of factor analysis, it is worth emphasizing that the method applies to arbitrary feature vectors. Indeed, whatever features are used to summarize the short-time properties of speech, one expects correlations to arise from coarticulation effects, background noise, speaker idiosyncrasies, etc.

3 Maximum likelihood factor analysis

The simplest learning criterion for continuous density HMMs is to maximize the likelihood of observed speech signals. The Expectation-Maximization (EM) algorithm[24] is a general iterative procedure for estimating the parameters of latent variable models. In this section, we begin by reviewing the EM algorithm for maximum likelihood factor analysis[24]. This is done for the multivariate Gaussian PDF of section 2.1. We then consider the extension to mixture models and continuous density HMMs. The EM algorithm for mixtures of factor analyzers was first derived by Ghahramani et al[9].

3.1 Multivariate Gaussian PDF

Consider the multivariate Gaussian PDF from eq. (3). Appealing to its representation as a latent variable model, we may derive an EM algorithm for obtaining maximum likelihood (ML) estimates of the parameters Λ, Ψ, and $\boldsymbol{\mu}$. For the parameter $\boldsymbol{\mu}$, an iterative procedure is not generally required, since one can simply set $\boldsymbol{\mu}$ equal to the sample mean (which is the ML estimate). However, by deriving the EM algorithm here in full generality, we will considerably simplify the extension (in section 3.2) to mixture models and continuous density HMMs. In this respect, our derivation of ML factor analysis will differ slightly from standard treatments.

To begin, let $\{\boldsymbol{x}_n\}_{n=1}^N$ denote a sample of N data points. The EM procedure is a two-step iterative procedure for maximizing the log-likelihood, $\sum_n \ln P(\boldsymbol{x}_n)$, with $P(\boldsymbol{x}_n)$ given by eq. (3). The E-step of this procedure is to compute the Q-function:

$$Q(\tilde{\boldsymbol{\mu}}, \tilde{\Lambda}, \tilde{\Psi}; \boldsymbol{\mu}, \Lambda, \Psi) = \sum_n \int d\boldsymbol{z}\ P(\boldsymbol{z}|\boldsymbol{x}_n, \boldsymbol{\mu}, \Lambda, \Psi) \ln P(\boldsymbol{z}, \boldsymbol{x}_n|\tilde{\boldsymbol{\mu}}, \tilde{\Lambda}, \tilde{\Psi}). \tag{7}$$

The right hand side of eq. (7) depends on $\boldsymbol{\mu}$, Λ and Ψ through the statistics of the posterior distribution, $P(\boldsymbol{z}|\boldsymbol{x}_n, \boldsymbol{\mu}, \Lambda, \Psi)$. Some useful properties of this distribution are summarized in appendix A. Of particular importance are the posterior statistics:

$$\begin{aligned} \mathrm{E}[\boldsymbol{z}|\boldsymbol{x}_n] &= [I + \Lambda^T \Psi^{-1} \Lambda]^{-1} \Lambda^T \Psi^{-1} (\boldsymbol{x}_n - \boldsymbol{\mu}) & (8) \\ \mathrm{E}[\boldsymbol{\delta z}\ \boldsymbol{\delta z}^T|\boldsymbol{x}_n] &= [I + \Lambda^T \Psi^{-1} \Lambda]^{-1}, & (9) \end{aligned}$$

where the shorthand $\boldsymbol{\delta z}$ in eq. (9) denotes the variation about the conditional mean, or $\boldsymbol{\delta z} = \boldsymbol{z} - \mathrm{E}[\boldsymbol{z}|\boldsymbol{x}_n]$. Note that the conditional covariance, $\mathrm{E}[\boldsymbol{\delta z}\ \boldsymbol{\delta z}^T|\boldsymbol{x}_n]$, does not depend on the particular value of $\boldsymbol{x}_n$; as a practical matter, this means that it does not have to be recomputed for each data point.

The M-step of the EM algorithm is to maximize the right hand side of eq. (7) with respect to $\tilde{\boldsymbol{\mu}}$, $\tilde{\Psi}$ and $\tilde{\Lambda}$. To this end, let us define the new vector quantities:

$$\begin{aligned} \boldsymbol{\Delta x}_n &= \boldsymbol{x}_n - \frac{1}{N} \sum_k \boldsymbol{x}_k, & (10) \\ \boldsymbol{\Delta z}_n &= \mathrm{E}[\boldsymbol{z}|\boldsymbol{x}_n] - \frac{1}{N} \sum_k \mathrm{E}[\boldsymbol{z}|\boldsymbol{x}_k]. & (11) \end{aligned}$$

These quantities measure the variation of each data point and its expected factors about the sample means. Note that $\boldsymbol{\Delta x}_n$ and $\boldsymbol{\Delta z}_n$ are measures of sample variance, as opposed to posterior averages such as $\mathrm{E}[\boldsymbol{\delta z}\, \boldsymbol{\delta z}^T|\boldsymbol{x}_n]$, which measure the uncertainty in $P(\boldsymbol{z}|\boldsymbol{x}_n)$ induced by a single data point. The definitions of $\boldsymbol{\Delta x}_n$ and $\boldsymbol{\Delta z}_n$ are useful insofar as they allow us to write the EM updates in an especially compact form. In appendix B, we show that maximizing eq. (7) leads to the iterative updates:

$$\begin{aligned} \tilde{\Lambda} &\leftarrow \left(\sum_n (\boldsymbol{\Delta x}_n)(\boldsymbol{\Delta z}_n)^T \right) \left(\sum_n \left[\mathrm{E}[\boldsymbol{\delta z}\ \boldsymbol{\delta z}^T|\boldsymbol{x}_n] + (\boldsymbol{\Delta z}_n)(\boldsymbol{\Delta z}_n)^T \right] \right)^{-1}, & (12) \\ \tilde{\boldsymbol{\mu}} &\leftarrow \frac{1}{N} \sum_n \left(\boldsymbol{x}_n - \tilde{\Lambda}\, \mathrm{E}[\boldsymbol{z}|\boldsymbol{x}_n] \right), & (13) \end{aligned}$$

$$\tilde{\Psi}_{ii} \;\leftarrow\; \frac{1}{N}\sum_n \left[\left(\boldsymbol{\Delta x}_n - \tilde{\Lambda}\boldsymbol{\Delta z}_n\right)_i^2 + \left(\tilde{\Lambda}\,\mathrm{E}[\boldsymbol{\delta z}\,\boldsymbol{\delta z}^T|\boldsymbol{x}_n]\,\tilde{\Lambda}^T\right)_{ii}\right], \tag{14}$$

where $\tilde{\Psi}_{ii}$ are the diagonal elements of $\tilde{\Psi}$. These updates are guaranteed to converge monotonically to a (possibly local) maximum of the log-likelihood. Note that it is important to perform the updates in the order shown, since (for example) the $\tilde{\boldsymbol{\mu}}$–update depends on the re-estimated value of $\tilde{\Lambda}$. Naturally, for $\Lambda = 0$ the updates in eqs. (13–14) reduce to the ML estimates for independent Gaussian random variables.

The re-estimation equations are supported by various intuitions. In eq. (12), for example, the factor loading matrix is re-estimated by finding the least-squares solution that projects the variation in $\boldsymbol{x}$ (about its mean) into a lower dimensional subspace. In eq. (13), the mean is used to account for the average offset between each observation and its reconstruction by the factors. Finally, in eq. (14), the variances are used to account for two types of variation in $\boldsymbol{x}$: those outside the dimensionality-reduced subspace (the first term on the right hand side), and those within the dimensionality-reduced subspace (the second term). All these intuitions extend to the more elaborate models considered in the next section.

3.2 Hidden Markov models

One can readily integrate the EM algorithm for factor analysis into the ML training of continuous density HMMs. The parameter updates for FA-HMMs have a similar structure to the Baum-Welch updates for conventional HMMs (i.e., those whose Gaussian PDFs employ diagonal or full covariance matrices). In fact, the main novelties—which arise from the special form of factored covariance matrices—were introduced in the previous section.

With that in mind, let us reconsider the HMM whose observations are conditionally modeled by eqs. (5–6). Suppose that $\boldsymbol{X} = \{\boldsymbol{x}_n\}$ represents a sequence of training data. The forward-backward procedure[21] enables one to compute the posterior probability, $\gamma_n^{sc} = P(s_n = s, c_n = c|\boldsymbol{X})$, that the HMM used state s and mixture component c to generate $\boldsymbol{x}_n$. Let $N_{sc} = \sum_n \gamma_n^{sc}$ denote the total probability mass (or *effective* number of feature vectors) attributed to state s and mixture component c. The parameter updates for FA-HMMs may be viewed as a specialization of eqs. (12–14) to each hidden state and mixture component, in which the observations $\boldsymbol{x}_n$ are weighted by their posterior probabilities, γ_n^{sc}. Analogous to eqs. (10–11), we define for each state and mixture component the vector quantities:

$$\boldsymbol{\Delta x}_n^{sc} \;=\; \boldsymbol{x}_n - \frac{1}{N^{sc}}\sum_k \gamma_k^{sc}\,\boldsymbol{x}_k, \tag{15}$$

$$\boldsymbol{\Delta z}_n^{sc} \;=\; \mathrm{E}_{sc}[\boldsymbol{z}|\boldsymbol{x}_n] - \frac{1}{N^{sc}}\sum_k \gamma_k^{sc}\,\mathrm{E}_{sc}[\boldsymbol{z}|\boldsymbol{x}_k], \tag{16}$$

where $E_{sc}[\boldsymbol{z}|\boldsymbol{x}_n]$ denotes an expectation with respect to the posterior distribution, $P(\boldsymbol{z}|\boldsymbol{x}_n, s_n = s, c_n = c)$. In terms of these vectors, the EM updates are given by:

$$\tilde{\Lambda}_{sc} \;\leftarrow\; \left(\sum_n \gamma_n^{sc}(\boldsymbol{\Delta x}_n^{sc})(\boldsymbol{\Delta z}_n^{sc})^T\right)\left(\sum_n \gamma_n^{sc}\left[\mathrm{E}_{sc}[\boldsymbol{\delta z}\boldsymbol{\delta z}^T|\boldsymbol{x}_n] + (\boldsymbol{\Delta z}_n^{sc})(\boldsymbol{\Delta z}_n^{sc})^T\right]\right)^{-1}, \tag{17}$$

$$\tilde{\boldsymbol{\mu}}_{sc} \leftarrow \frac{1}{N^{sc}} \sum_n \gamma_n^{sc} \left(\boldsymbol{x}_n - \tilde{\Lambda}_{sc} \, \mathrm{E}_{sc}[\boldsymbol{z}|\boldsymbol{x}_n] \right), \tag{18}$$

$$\left[\tilde{\Psi}_{sc}\right]_{ii} \leftarrow \frac{1}{N^{sc}} \sum_n \gamma_n^{sc} \left[\left(\boldsymbol{\Delta x}_n^{sc} - \tilde{\Lambda}_{sc} \boldsymbol{\Delta z}_n^{sc} \right)_i^2 + \left(\tilde{\Lambda}_{sc} \, \mathrm{E}_{sc}[\boldsymbol{\delta z} \boldsymbol{\delta z}^T | \boldsymbol{x}_n] \, \tilde{\Lambda}_{sc}^T \right)_{ii} \right]. \tag{19}$$

The reader should note the similarities in structure between eqs. (17–19) and eqs. (12–14). The FA-HMM updates have essentially the same form as eqs. (12-14), except that now each observation $\boldsymbol{x}_n$ is weighted by the posterior probability, γ_n^{sc}.

As usual, in the case where multiple sequences are available as training data, the sums over n should be interpreted as sums over sequences as well[21]. Finally, we note that the updates for mixture weights and transition matrices in FA-HMMs are identical to those in conventional HMMs[21]. (This should be clear since FA-HMMs form a subset of HMMs whose Gaussian PDFs employ full covariance matrices.)

4 Minimum classification error factor analysis

Ultimately, we evaluate a speech recognizer by its accuracy, not by the likelihood scores of its component HMMs. Hence, maximizing the likelihood—while arguably the simplest criterion for parameter estimation—is not always the most desirable one. In fact, because continuous density HMMs represent only approximately correct models of speech, maximum likelihood estimates are not guaranteed to produce accurate recognizers, even in the limit of infinite training data. Moreover, empirically it is well known that maximizing likelihoods does not always translate into minimizing error rates. We will see an example of this—for connected alpha-digits—in section 5.1.

For all these reasons, many researchers have proposed alternative criteria for parameter estimation that more directly relate to the empirical error rate[1, 12, 13, 17]. In this paper, we consider the minimum classification error (MCE) criterion as described by Juang et al[12]. For each utterance, the MCE criterion relates the event of a recognition error to the log-likelihood ratio between correct and competing hypotheses, where each hypothesis represents one possible labeling of hidden states. In this section, we show that this type of discriminative training can be applied to FA-HMMs. To keep the paper self-contained, we begin by reviewing the basic framework for MCE parameter estimation. We then consider a simple example—how to train a classifier based on competing Gaussian PDFs. This example is analyzed for the special case of factored covariance matrices. Finally, in the last part of the section, we extend the method to continuous density HMMs.

4.1 Discriminative training

Let $\{\boldsymbol{x}_n, y_n\}_{n=1}^N$ denote a labeled sample of data points, where the label y_n indicates that $\boldsymbol{x}_n$ belongs to one of $\mathcal{Y}$ classes. If we have statistical models for the class-conditional distributions, $P(\boldsymbol{x}|y)$, then we can construct the maximum a posteriori (MAP) classifier that labels each point by its most probable assignment:

$$\mathcal{C}(\boldsymbol{x}) = \arg\max_y \left[\ln P(\boldsymbol{x}|y) P(y) \right]. \tag{20}$$

For simplicity, let us assume that the class labels have a priori equal probabilities, or $P(y) = 1/\mathcal{Y}$, so that the rightmost term in eq. (20) can be ignored. The empirical error rate on the training data, $\hat{J}$, is then given by:

$$\hat{J} = \frac{1}{N} \sum_n \Theta \left(\max_{y \neq y_n} \ln \left[\frac{P(\boldsymbol{x}_n|y)}{P(\boldsymbol{x}_n|y_n)} \right] \right), \tag{21}$$

where $\Theta(\cdot)$ denotes the unit threshold function, and the max operation ranges over all labels except the correct one. Note that eq. (21) computes the classification error for each data point in two steps. First, the max operation is used to find the most likely *incorrect* hypothesis. Second, the threshold function is used to register an error if this hypothesis is *more* likely than the correct one.

Suppose now that the statistical models, $P(\boldsymbol{x}|y)$, are expressed smoothly in terms of parameters, Φ, at our disposal. In what follows, we will assume this dependence implicitly, rather than writing out $P(\boldsymbol{x}|y;\Phi)$ wherever we encounter probabilities. The goal of discriminative training is to find the parameter values that minimize the empirical error rate. Note, however, that eq. (21) does not define a differentiable function of the log-likelihoods: both the step function and the max operation give rise to singular gradients. This makes it difficult to directly minimize the empirical error rate. Our strategy, following earlier work[12], is to construct a smooth objective function that mimics the behavior of eq. (21). This is done in two steps. First, we replace the max operation by a softmax: $\max_k z_k \approx \ln \sum_k e^{z_k}$. Second, we replace the threshold function by a sigmoid function: $\Theta(z) \approx [1 + e^{-z}]^{-1}$. In particular, for the nth data point, let

$$d(\boldsymbol{x}_n) = \ln \sum_{y \neq y_n} P(\boldsymbol{x}|y) - \ln P(\boldsymbol{x}_n|y_n), \tag{22}$$

denote the log-likelihood-ratio between the softmax–smoothed competing hypothesis[1] and the correct one. The function $d(\boldsymbol{x}_n)$ provides a smooth measure of the misclassification of $\boldsymbol{x}_n$. The MCE loss function is obtained by substituting this smoothed log-likelihood ratio into a sigmoid function:

$$J = \frac{1}{N} \sum_n \frac{1}{1 + \exp(-d(\boldsymbol{x}_n))}, \tag{23}$$

Minor variations[12] on this loss function can also be considered by generalizing the softmax and sigmoid functions. The important result is that eq. (23) is a smooth function of the log-likelihoods, $\ln P(\boldsymbol{x_n}|y)$.

The goal of discriminative training is to minimize the MCE loss function with respect to the parameters Φ. We can update these parameters by gradient descent, in which at each iteration

$$\Phi \leftarrow \Phi - \eta(\Phi) \frac{\partial J}{\partial \Phi}, \tag{24}$$

where $\eta(\Phi)$ is a positive learning rate, or more generally, a positive-definite matrix. Note that in the MCE loss function, eq. (23), all the dependence on Φ enters through the log-

[1] If the number of classes, $\mathcal{Y}$, is very large, one can restrict the sum over y in eq. (22) to the N–best alternatives, where $\text{N} \ll \mathcal{Y}$.

likelihoods, $\ln P(\boldsymbol{x}|y)$. We can therefore use the chain rule

$$\frac{\partial J}{\partial \Phi} = \sum_y \frac{\partial J}{\partial \ln P(\boldsymbol{x}|y)} \frac{\partial \ln P(\boldsymbol{x}|y)}{\partial \Phi} \tag{25}$$

to decompose the partial derivatives in eq. (24). Eqs. (22–25) provide the general framework for MCE parameter estimation. The specific form of the update equations necessarily depends on the parameterization of the class-conditional distributions, $P(\boldsymbol{x}|y)$. For a given parameterization, however, the only requirement is to compute the gradients with respect to Φ. In the rest of this section, we examine two cases of special interest.

4.2 Multivariate Gaussian PDFs

Suppose that the class-conditional distributions are multivariate Gaussian PDFs with factored covariance matrices. Discriminative training requires us to compute the gradients of $\ln P(\boldsymbol{x}|y)$, where for a particular class:

$$P(\boldsymbol{x}|y) = \frac{|\Psi + \Lambda\Lambda^T|^{-1/2}}{(2\pi)^{D/2}} \exp\left\{-\frac{1}{2}\left[\boldsymbol{x}-\boldsymbol{\mu}\right]^T \left(\Psi + \Lambda\Lambda^T\right)^{-1} \left[\boldsymbol{x}-\boldsymbol{\mu}\right]\right\}. \tag{26}$$

Let $M = \Psi + \Lambda\Lambda^T$ denote the covariance matrix in eq. (26). In appendix C, we show that the gradients of $\ln P(\boldsymbol{x}|y)$ are given by:

$$\frac{\partial}{\partial \boldsymbol{\mu}_i}\left[\ln P(\boldsymbol{x}|y)\right] = -\left[M^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right]_i, \tag{27}$$

$$\frac{\partial}{\partial \Lambda_{ij}}\left[\ln P(\boldsymbol{x}|y)\right] = \left[M^{-1}\Lambda\right]_{ij} - \left[M^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right]_i \left[(\boldsymbol{x}-\boldsymbol{\mu})^T M^{-1}\Lambda\right]_j, \tag{28}$$

$$\frac{\partial}{\partial \Psi_{ii}}\left[\ln P(\boldsymbol{x}|y)\right] = \frac{1}{2}\left\{M_{ii}^{-1} - \left[M^{-1}(\boldsymbol{x}-\boldsymbol{u})\right]_i^2\right\}, \tag{29}$$

where $[\cdot]_i$ denotes the ith vector element. The inverse covariance matrix M^{-1} that appears in eqs. (27–29) can be computed efficiently using the lemma in eq. (4). Naturally, these gradients also reduce to the correct form for diagonal covariance matrices, when no factors are present.

The parameter updates for class-conditional Gaussian PDFs are computed from eqs. (24–25). In practice, one chooses a positive-definite matrix, $\eta(\Phi)$, to ensure that each parameter update has roughly the same magnitude effect on the MCE loss function. This is achieved by scaling the $(\frac{\partial}{\partial \mu}, \frac{\partial}{\partial \Lambda})$ gradients by the variance, Ψ. Also, as is standard practice[12], the variance parameters are updated in the log domain to enforce the constraint of non-negativity.

4.3 Hidden Markov models

In this section, we consider how discriminative training can be applied to FA-HMMs for automatic speech recognition. Suppose that FA-HMMs are being used to model sub-word

units, such as context-dependent phones. Training data in this setting consists of variable-length utterances and their segmentations into strings of phones or sub-phones. In particular, the segmentations label each frame of speech by the state of a particular HMM. We can naturally view these labels as *targets* for the Viterbi decoding of the recognizer. Learning to match these targets is a form of MCE parameter estimation, where the class-conditional distributions of eq. (21) are the state-emission densities of HMMs, and the hypotheses in eq. (22) range over the possible segmentations of each recognized utterance. Here we consider the special case where the feature vectors, conditioned on the hidden states, are modeled by mixtures of Gaussian PDFs with factored covariance matrices.

For this type of discriminative training, we must compute the gradients of the state-emission densities with respect to their parameters. In FA-HMMs, these distributions are given by eqs. (5–6). The gradients for mixtures of factor analyzers are a straightforward extension of our previous results. In particular, consider a segmented utterance in which the feature vector $\boldsymbol{x}$ is assigned to state s of an FA-HMM. Also, let $P(c|\boldsymbol{x}, s)$ denote the posterior probability that within state s, the feature vector $\boldsymbol{x}$ is attributed to mixture component c. Then the required gradients are:

$$\left(\frac{\partial}{\partial \boldsymbol{\mu}_{sc}}, \frac{\partial}{\partial \Lambda_{sc}}, \frac{\partial}{\partial \Psi_{sc}}\right) [\ln P(\boldsymbol{x}|s)] = P(c|\boldsymbol{x}, s) \left(\frac{\partial}{\partial \boldsymbol{\mu}_{sc}}, \frac{\partial}{\partial \Lambda_{sc}}, \frac{\partial}{\partial \Psi_{sc}}\right) [\ln P(\boldsymbol{x}|s, c)] \,. \tag{30}$$

The gradients on the right hand side of these equations are given by eqs. (27–29) from the previous section. Besides these parameters, one may also adapt the mixture weights, $P(c|s)$, by gradient descent. In practice, one adapts the transformed parameters w_{sc}, where $P(c|s) = e^{w_{sc}} / \sum_k e^{w_{sk}}$; this is done to enforce the constraints $P(c|s) \geq 0$ and $\sum_c P(c|s) = 1$. In this case, the necessary gradients are given by:

$$\frac{\partial}{\partial w_{sc}} [\ln P(\boldsymbol{x}|s)] = P(c|\boldsymbol{x}, s) - P(c|s). \tag{31}$$

Given these gradients, the parameter updates are again computed from eqs. (24–25). For each state and mixture component, one considers a different positive-definite matrix, $\eta(\Phi)$, that multiplies the gradients in the learning rule. In particular, this matrix is used to scale the $(\frac{\partial}{\partial \mu_{sc}}, \frac{\partial}{\partial \Lambda_{sc}})$ gradients by the variances, Ψ_{sc}. In the experiments described below, we computed the MCE loss function for each utterance, interpreting the sum over n in eq. (23) as a sum over the sequence of feature vectors. The parameters were updated on an utterance-by-utterance basis. In addition, the softmax in eq. (22) was approximated by considering only the four best alternatives to the correct segmentation. In these respects and others, we followed the procedure outlined in earlier work[12].

5 Experiments

Continuous density HMMs were evaluated on two tasks in automatic speech recognition, one small-sized vocabulary (36 words) and one medium-sized vocabulary (1219 words). The same front end was applied to both tasks. For signal processing, waveforms were pre-emphasized and blocked into 30ms frames at every 10ms interval. For feature extraction, frames were

Hamming windowed, autocorrelated, and processed by LPC cepstral analysis to produce a vector of twelve liftered cepstral coefficients. The feature vector was then augmented by its normalized log energy value, as well as temporal derivatives of first and second order. Overall, each frame of speech was described by thirty nine features. These input features were fed directly to the HMMs described below.

5.1 Connected alpha-digits

As a small vocabulary task, we considered the recognition of alphanumeric strings (e.g., N Z 3 V J 4 E 3 U 2). Easily confused letters such as B/V, C/Z, and M/N make this a challenging problem in speech recognition. The training and test data[25] for this task were recorded over a telephone network and consisted of 14622 and 7255 utterances, respectively. The speech recognizers were built using 285 left-to-right HMMs, each of which modeled a context-dependent sub-word unit. Non-zero transition probabilities were fixed to uniform values. Testing was done with a free grammar network (i.e., no grammar constraints).

We trained continuous density HMMs with both diagonal and factored covariance matrices. Our first experiments considered the maximum likelihood (ML) training procedure described in section 3.2. We ran several experiments, varying both the number of mixture components and the number of factors allocated to each state in the HMMs. The goal was to determine the best model of acoustic feature correlations.

Table 1 summarizes the results of these experiments. The columns from left to right show the number of mixture components (C), the number of factors (f), the word error rates (including insertion, deletion, and substitution errors) on the test set, the average log-likelihood per frame of speech on the test set, and the CPU time to recognize twenty test utterances (on an SGI R4000). Not surprisingly—given the large amount of training data—both word accuracies and likelihood scores increase with the number of modeling parameters. Nevertheless, the table shows that adding factors leads to significant improvements in performance, at roughly the same rate as adding mixture components.

Perhaps the most interesting comparisons are between models with the same number of parameters. The overall number of parameters is proportional to $C(f+2)$. A useful rule of thumb is that FA-HMMs with two factors have the same number of parameters as DG-HMMs with twice as many mixture components. The left graph in figure 1 shows a plot of the average log-likelihood versus the number of parameters, or $C(f+2)$; the stars and circles in this plot indicate recognizers with DG-HMMs and FA-HMMs, respectively. One sees quite clearly from this plot that given a *fixed* number of parameters, models with factored covariance matrices tend to have significantly higher likelihoods. The right graph in figure 1 shows a similar plot of the word error rates versus the number of parameters. Here one does not see much difference; presumably, because HMMs represent only approximately correct models of speech, higher likelihoods do not necessarily translate into lower error rates. We will return to this point later.

It is worth noting that the above experiments used a fixed number of factors per mixture component. In fact, because the variability of speech is highly context-dependent, it makes sense to vary the number of factors, even across states within the same HMM. A simple heuristic is to adjust the number of factors depending on the amount of training data for

C	f	word error (%)	log-likelihood	CPU time (sec)
1	0	16.2	32.9	25
1	1	14.6	34.2	30
1	2	13.7	34.9	30
1	3	13.0	35.3	38
1	4	12.5	35.8	39
2	0	13.4	34.0	30
2	1	12.0	35.1	44
2	2	11.4	35.8	48
2	3	10.9	36.2	61
2	4	10.8	36.6	67
4	0	11.5	34.9	46
4	1	10.4	35.9	80
4	2	10.1	36.5	93
4	3	10.0	36.9	132
4	4	9.8	37.3	153
8	0	10.2	35.6	93
8	1	9.7	36.5	179
8	2	9.6	37.0	226
16	0	9.5	36.2	222

Table 1: Results for different ML recognizers on connected alphadigits. The columns indicate the number of mixture components (C), the number of factors (f), the word error rates and average log-likelihood scores on the test set, and the CPU time to recognize twenty utterances.

each state (as determined by an initial segmentation of the training utterances). We found that this heuristic led to more pronounced differences in likelihood scores and error rates. In particular, substantial improvements were observed for three recognizers whose HMMs employed an *average* of two factors per mixture component. Table 2 summarizes these results. The reader will notice that these recognizers are extremely competitive in all aspects of performance—accuracy, memory, and speed—with the baseline recognizers in table 1. The likelihoods and error rates of these recognizers are indicated by the dashed lines in figure 1.

All the recognizers in tables 1 and 2 were trained by maximum likelihood (ML) estimation. The plots in figure 1, however, show that increased log-likelihoods in FA-HMMs do not translate directly into lower error rates. This observation motivated us to train a number of recognizers using the minimum classification error (MCE) criterion described in section 4. This was done for the three DG-HMM recognizers in table 1 and the three FA-HMM recognizers in table 2. In each case, the parameters of the MCE recognizer were initialized by those of its ML counterpart. Training consisted of five iterations of gradient descent, where each iteration involved a pass through all the utterances in the training set. The test set error rates of these recognizers, before and after discriminative training, are shown in figure 2.

C	f	word error (%)	log-likelihood	CPU time (sec)
1	2	12.3	35.4	32
2	2	10.5	36.3	53
4	2	9.6	37.0	108

Table 2: Results for ML recognizers with variable numbers of factors; f denotes the *average* number of factors per mixture component.

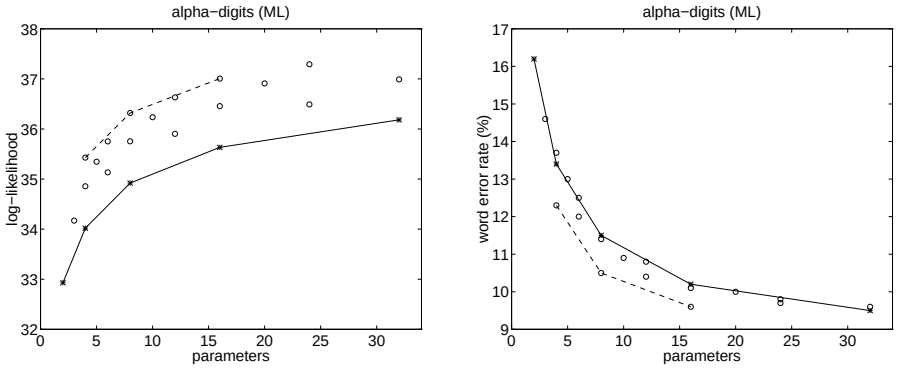

Figure 1: Plots of log-likelihood scores and word error rates on the test set versus the number of parameters. The stars, connected by solid lines, indicate DG-HMMs; the circles indicate FA-HMMs. The dashed lines connect the recognizers in table 2.

The figure shows that all the recognizers had significantly lower error rates as a result of MCE training. At the same time, the recognizers using factor analysis retained their edge in accuracy. This shows that MCE training and factor analysis are complementary methods for improving overall performance. Figure 2 also shows the results of a further training procedure—hierarchical signal bias removal (BR)[22]—designed to reduce the acoustic mismatch between training and testing environments. This method led to further improvements in recognition accuracy.

5.2 New Jersey town names

As a medium vocabulary task, we considered the recognition of New Jersey town names (e.g., Cranbury). Telephone speech from two different collections[25] was used in this experiment. The training data consisted of 12100 short phrases, each 2-4 words long, spoken in the seven major dialects of American English. This collection of phrases was carefully designed to obtain a fairly even coverage over triphone units. The test data consisted of 2426 isolated utterances of New Jersey town names, collected from nearly 100 speakers. The number of town names was 1219. All the speech data was digitized at the caller's local switch and transmitted in this form to the recognizers. The recognizers were built using 43 left-to-right HMMs, each of which modeled a context-independent English phone. Phones were modeled by three-state HMMs; the only exception to this rule was that silence and background noise

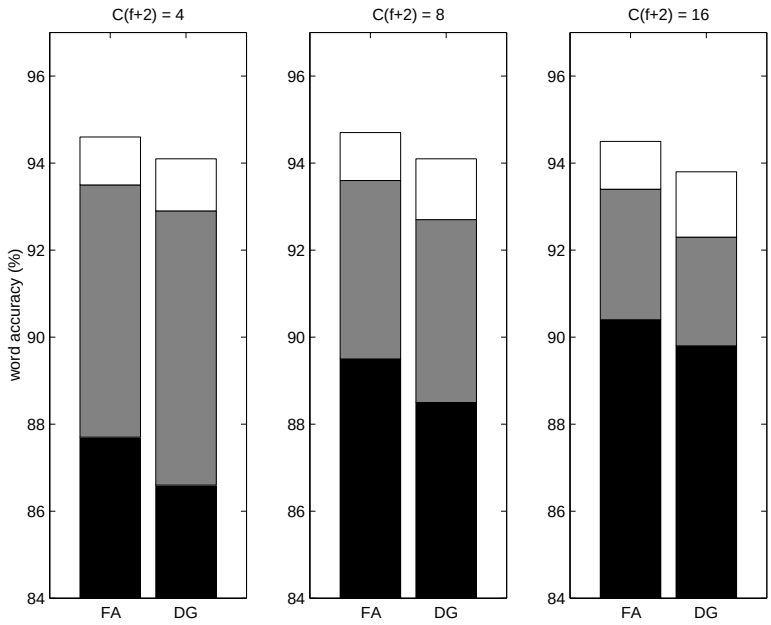

Figure 2: Word accuracies of ML, MCE, and MCE+BR recognizers shown in black, gray, and white. Each subplot compares recognizers with the same overall number of parameters, but with different types of covariance matrices (FA vs DG).

were modeled by a single state. All non-zero transition probabilities were fixed to uniform values.

Again, we trained continuous density HMMs with both diagonal and factored covariance matrices. Since in the alphadigit experiments, the largest gains (per parameter) occurred with small numbers of factors, here we only considered FA-HMMs with exactly one factor per mixture component. The results of ML and MCE training for several recognizers are shown in table 3. Again we see that parameters devoted to the explicit modeling of correlations lead to sizable reductions in the error rate. Also, both the DG-HMM and FA-HMM recognizers are substantially improved by discriminative training. Figure 3 shows plots of the classification error rate versus the number of modeling parameters (i.e., $C(f+2)$) for the ML and MCE recognizers. The rate of improvement in performance (as a function of model size) is roughly comparable for DG-HMMs and FA-HMMs.

6 Summary

In this paper we have studied the combined use of mixture densities and factor analysis for speech recognition. This was done in the framework of hidden Markov modeling, where acoustic features are conditionally modeled by mixtures of Gaussian PDFs. On two tasks—connected alpha-digits and New Jersey town names—we have shown that mixture densities and factor analysis are complementary means of modeling acoustic correlations. Moreover, when used together, they can lead to smaller, faster, and more accurate recognizers than either method on its own. For example, comparing the last lines of tables 1 and 2 reveals a

C	f	ML error (%)	MCE error (%)
4	0	19.0	17.4
4	1	17.6	15.0
8	0	16.8	14.5
8	1	15.0	12.5
16	0	14.6	12.6
16	1	13.8	12.8
32	0	13.6	12.0
32	1	12.7	11.7
64	0	12.6	11.7

Table 3: Results on New Jersey town names. The columns indicate the number of mixture components, the number of factors, and the error rates for ML and MCE recognizers.

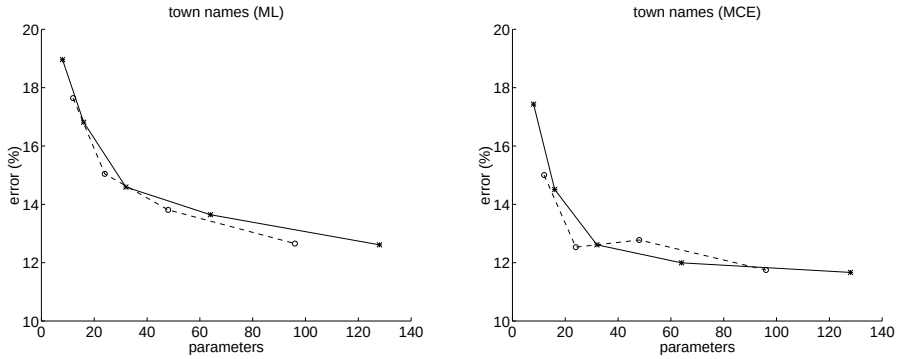

Figure 3: Plots of town name error rates versus the number of parameters for ML and MCE recognizers. The solid and dashed lines indicate recognizers with DG-HMMs and FA-HMMs, respectively.

factor of two improvement in speed and memory with hardly any loss in accuracy.

Both mixture models and factor analysis may be understood as latent variable methods—the former for clustering, the latter for dimensionality reduction. Just as mixture densities have proved indispensable in automatic speech recognition, we believe that factor analysis can make a similarly important contribution. Both methods have a sound probabilistic framework that enables them to be fully exploited in statistical pattern recognition.

It is worth comparing factor analysis to other approaches[10, 26, 28, 5] proposed for dimensionality reduction and feature space analysis in automatic speech recognition. We believe that factor analysis has two distinguishing features. First, unlike many strategies for dimensionality reduction, factor analysis does not merely project a high dimensional feature vector into a low dimensional one; variations outside the reduced-dimensionality subspace are also modeled by the variance parameters, Ψ. Second, ML parameter estimation in factor analysis is supported by an efficient, batch-update EM algorithm; this is of considerable value even when the parameter estimates are ultimately refined by gradient-based discriminative

training.

In this paper, we used factor analysis to model correlations between cepstra, delta-cepstra, and delta-delta-cepstra. It is worth emphasizing, however, that the method applies to arbitrary features. Indeed, the ability to model correlations efficiently should enable researchers to consider other features besides cepstra. While cepstra have the advantage of being only weakly correlated, it may be that other features (e.g., narrow-band statistics) actually convey more information about the speech signal.

We believe that factor analysis has many potential applications to automatic speech recognition. For example, in segmental HMMs[19], where feature vectors encode multiple (successive) frames of speech, factored covariance matrices provide a natural way to model correlations over time. In model adaptation[8, 15, 29], where parameters are re-estimated to match new testing conditions, they provide a more robust alternative to adapting full covariance matrices. These are only two examples; in principle, factor analysis can be applied wherever one introduces Gaussian PDFs.

Factor analysis should be evaluated in light of other methods for modeling correlations. One popular method is to employ full covariance matrices through some form of parameter-tying. While we have presented factor analysis as an alternative to this approach, more generally it should be viewed as a complement. Certainly, the clever tying of factor loading matrices across units, states, and/or mixture components would lead to further improvements in overall performance (as a function of model size). Thus, whatever gains have been achieved by tying full covariance matrices, one would expect *additional* gains to be achieved from tying factored ones.

Overall performance in automatic speech recognition is measured in terms of speed, memory, and accuracy. Factor analysis can help in all three respects. Compared to full covariance matrices, factored ones are easier to manipulate and more robust to overfitting. In general, factor analysis represents a useful compromise between diagonal and full covariance matrices, one that promises improvements over both extremes.

Acknowledgments

We are grateful to A. Ljolje (AT&T Labs) and Z. Ghahramani (University College London) for useful discussions. We also thank P. Modi (AT&T Labs) for providing an initial segmentation of the training utterances.

A Posterior distribution

In this appendix, we consider the properties of the posterior distribution, $P(\boldsymbol{z}|\boldsymbol{x})$, for the multivariate Gaussian PDF mentioned at the end of section 2.1. The posterior distribution is computed from Bayes' rule as:

$$P(\boldsymbol{z}|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|\boldsymbol{z})P(\boldsymbol{z})}{P(\boldsymbol{x})}. \tag{32}$$

The three terms on the right hand side are given explicitly by eqs. (1–3). Because $P(\boldsymbol{z})$ and $P(\boldsymbol{z}|\boldsymbol{x})$ are Gaussian, it follows from the form of Bayes' rule that $P(\boldsymbol{z}|\boldsymbol{x})$ is also Gaussian.

In particular, up to a normalization constant, the posterior distribution is given by:

$$P(\boldsymbol{z}|\boldsymbol{x}) \sim \exp\left\{-\frac{1}{2}\boldsymbol{z}^T(I+\Lambda^T\Psi^{-1}\Lambda)\boldsymbol{z} + (\boldsymbol{x}-\boldsymbol{\mu})^T\Psi^{-1}\Lambda\boldsymbol{z}\right\}. \tag{33}$$

The statistics of the hidden variable $\boldsymbol{z}$, conditioned on the observation $\boldsymbol{x}$, are determined by the linear and quadratic terms in the exponent of eq. (33). In particular, the conditional mean and covariance are given by:

$$\begin{aligned} \mathrm{E}[\boldsymbol{z}|\boldsymbol{x}] &= [I+\Lambda^T\Psi^{-1}\Lambda]^{-1}\Lambda^T\Psi^{-1}(\boldsymbol{x}-\boldsymbol{\mu}), & (34)\\ \mathrm{E}[\boldsymbol{\delta z}\ \boldsymbol{\delta z}^T|\boldsymbol{x}] &= [I+\Lambda^T\Psi^{-1}\Lambda]^{-1}, & (35) \end{aligned}$$

where $\mathrm{E}[\cdot|\boldsymbol{x}]$ denotes an average with respect to the posterior distribution, $P(\boldsymbol{z}|\boldsymbol{x})$, and $\boldsymbol{\delta z} = \boldsymbol{z} - \mathrm{E}[\boldsymbol{z}|\boldsymbol{x}]$ denotes the variation about the conditional mean. Note that the right hand side of eq. (35) does not depend on the value of $\boldsymbol{x}$. As a practical matter, this means that the conditional covariance matrix $\mathrm{E}[\boldsymbol{\delta z}\ \boldsymbol{\delta z}^T|\boldsymbol{x}]$ does not have to be recomputed for each data point.

B EM algorithm

In this appendix, we provide a more detailed derivation of the EM algorithm in section 3.1. We begin by evaluating the Q-function, eq. (7). This is done by averaging $\ln P(\boldsymbol{z}, \boldsymbol{x}_n|\tilde{\boldsymbol{\mu}}, \tilde{\Lambda}, \tilde{\Psi})$ over the posterior distribution, $P(\boldsymbol{z}|\boldsymbol{x}_n, \boldsymbol{\mu}, \Lambda, \Psi)$, then taking the sum over all data points. Up to a constant term (which does not depend on $\tilde{\boldsymbol{\mu}}$, $\tilde{\Lambda}$, or $\tilde{\Psi}$), the Q-function is given by:

$$\begin{aligned} Q(\tilde{\boldsymbol{\mu}}, \tilde{\Lambda}, \tilde{\Psi}; \boldsymbol{\mu}, \Lambda, \Psi) &= -\frac{1}{2}\sum_n \mathrm{E}\left[\left(\boldsymbol{x}_n - \tilde{\boldsymbol{\mu}} - \tilde{\Lambda}\boldsymbol{z}\right)^T \tilde{\Psi}^{-1}\left(\boldsymbol{x}_n - \tilde{\boldsymbol{\mu}} - \tilde{\Lambda}\boldsymbol{z}\right)\Big|\,\boldsymbol{x}_n\right] \\ &\quad - \frac{N}{2}\ln\left|\tilde{\Psi}\right| + \text{const}, \end{aligned} \tag{36}$$

where $\mathrm{E}[\cdot|\boldsymbol{x}_n]$ denotes an expectation over the posterior distribution, $P(\boldsymbol{z}|\boldsymbol{x}_n, \boldsymbol{\mu}, \Lambda, \Psi)$. Let $\boldsymbol{\delta z} = \boldsymbol{z} - \mathrm{E}[\boldsymbol{z}|\boldsymbol{x}_n]$ denote the variation about the conditional mean of the hidden variable. We can decompose the terms inside the expectation as:

$$\begin{aligned} Q(\tilde{\boldsymbol{\mu}}, \tilde{\Lambda}, \tilde{\Psi}; \boldsymbol{\mu}, \Lambda, \Psi) &= -\frac{1}{2}\sum_n \left(\boldsymbol{x}_n - \tilde{\boldsymbol{\mu}} - \tilde{\Lambda}\,\mathrm{E}[\boldsymbol{z}|\boldsymbol{x}_n]\right)^T \tilde{\Psi}^{-1}\left(\boldsymbol{x}_n - \tilde{\boldsymbol{\mu}} - \tilde{\Lambda}\,\mathrm{E}[\boldsymbol{z}|\boldsymbol{x}_n]\right) \\ &\quad - \frac{1}{2}\sum_n \mathrm{E}\left[(\tilde{\Lambda}\boldsymbol{\delta z})^T\tilde{\Psi}^{-1}(\tilde{\Lambda}\boldsymbol{\delta z})\Big|\,\boldsymbol{x}_n\right] - \frac{N}{2}\ln\left|\tilde{\Psi}\right| + \text{const}. \end{aligned} \tag{37}$$

The M-step of the EM algorithm is to maximize this expression with respect to $\tilde{\boldsymbol{\mu}}$, $\tilde{\Lambda}$, and $\tilde{\Psi}$. Setting $\partial Q/\partial\tilde{\boldsymbol{\mu}} = 0$ gives at once the re-estimation formulae for the mean, eq. (13). Setting $\partial Q/\partial\tilde{\Lambda} = 0$ gives:

$$\tilde{\Lambda}\left(\sum_n \mathrm{E}\left[\boldsymbol{\delta z}\ \boldsymbol{\delta z}^T\Big|\,\boldsymbol{x}_n\right]\right) = \sum_n \left(\boldsymbol{x}_n - \tilde{\boldsymbol{\mu}} - \tilde{\Lambda}\,\mathrm{E}[\boldsymbol{z}|\boldsymbol{x}_n]\right)^T \mathrm{E}[\boldsymbol{z}|\boldsymbol{x}_n]. \tag{38}$$

It is straightforward to eliminate $\tilde{\boldsymbol{\mu}}$ from this equation using eq. (13). This yields the re-estimation formula for the factor loading matrix, eq. (12). Finally, setting $\partial Q / \partial \tilde{\Psi}_{(ii)} = 0$ gives:

$$\tilde{\Psi}_{ii} = \frac{1}{N} \sum_n \left[\left(\boldsymbol{x}_n - \tilde{\boldsymbol{\mu}} - \tilde{\Lambda}\, \mathrm{E}[\boldsymbol{z}|\boldsymbol{x}_n] \right)_i^2 + \left(\tilde{\Lambda}\, \mathrm{E}[\boldsymbol{\delta z}\, \boldsymbol{\delta z}^T|\boldsymbol{x}_n] \tilde{\Lambda}^T \right)_{ii} \right]. \tag{39}$$

Here again, eliminating $\tilde{\boldsymbol{\mu}}$ via eq. (13) leads to the desired result—in this case, the re-estimation formula for the variance, eq. (14).

C Gradients

In this appendix, we derive the gradients required for discriminative training of FA-HMMs, namely eqs. (27–29). To this end, consider a multivariate Gaussian PDF with mean $\boldsymbol{\mu}$ and factored covariance matrix $\Psi + \Lambda\Lambda^T$. If we denote the covariance matrix by M, then the log-likelihood is given by:

$$\ln P(\boldsymbol{x}) = \frac{1}{2} \left[\ln\det M - (\boldsymbol{x} - \boldsymbol{\mu})^T M^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) \right]. \tag{40}$$

For discriminative training, we must compute the gradients of $\ln P(\boldsymbol{x})$ with respect to the parameters $\boldsymbol{\mu}$, Ψ, and Λ. The first of these is straightforward and gives eq. (27). For the others, we can use the chain rule—differentiating the right hand side of eq. (40) with respect to M, then differentiating M with respect to Ψ and Λ. The first step is facilitated by the identities:

$$\frac{\partial}{\partial M_{ij}} \left[\ln\det M \right] = M^{-1}_{ji}, \tag{41}$$

$$\frac{\partial}{\partial M_{ij}} \left[M^{-1}_{kl} \right] = -M^{-1}_{ki} M^{-1}_{jl}. \tag{42}$$

Using these identities to differentiate eq. (40), we obtain:

$$\frac{\partial}{\partial M_{ij}} \left[\ln P(\boldsymbol{x}) \right] = \frac{1}{2} \left\{ M^{-1}_{ji} - \left[M^{-1}(\boldsymbol{x} - \boldsymbol{\mu}) \right]_i \left[M^{-1}(\boldsymbol{x} - \boldsymbol{\mu}) \right]_j \right\}. \tag{43}$$

From the definition $M = \Psi + \Lambda\Lambda^T$, it is straightforward to compute the gradients $\partial M / \partial \Psi$ and $\partial M / \partial \Lambda$. Finally, inserting these gradients into the chain rule gives eqs. (28–29).

References

[1] Bahl, L., Brown, P., deSouza, P., and Mercer, L. (1986) Maximum mutual information estimation of hidden Markov model parameters for speech recognition. In *Proceedings of ICASSP 86*: 49–52.

[2] Bellegarda, J., and Nahamoo, D. (1990) Tied mixture continuous parameter modeling for speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing* **38**:2033-2045.

[3] Bregler, C. and Omohundro, S. (1995) Nonlinear image interpolation using manifold learning. In G. Tesauro, D. Touretzky, and T. Leen, eds. *Advances in Neural Information Processing Systems* **7**:971–980. Cambridge, MA: MIT Press.

[4] Carreira-Perpiñán, M. Á. and Renals, S. J. (1998) Dimensionality reduction of electropalatographic data using latent variable models. *Speech Communication* **26**:259–282.

[5] Chengalvarayan, R., and Deng, L. (1997) HMM-based speech recognition using state-dependent, discriminatively-derived transforms on Mel-warped DFT features. IEEE Transactions on Speech and Audio Processing **5**: 243–256.

[6] Duda, R.O. and Hart, P.E. (1973) *Pattern Classification and Scene Analysis*. New York: John Wiley.

[7] Everitt, B. (1984) *An introduction to latent variable models.* London: Chapman and Hall.

[8] Gauvain., J. and Lee, C. (1994) Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Transactions on Acoustics, Speech, and Signal Processing* **2**:291–298.

[9] Ghahramani, Z. and Hinton, G. (1996) The EM algorithm for mixtures of factor analyzers. *University of Toronto Technical Report* CRG-TR-96-1.

[10] Haeb-Umbach, H. and Ney, H. (1992) Linear discriminant analysis for improved large vocabulary continuous speech recognition. In *Proceedings of ICASSP 92*: 13–16.

[11] Hinton, G., Dayan, P., and Revow, M. (1997) Modeling the manifolds of images of handwritten digits. *IEEE Transactions on Neural Networks* **8**:65–74.

[12] Juang, B.H., Chou, W., and Lee, C.H. (1997) Minimum classification error rate methods for speech recognition. *IEEE Transactions on Speech and Audio Processing* **5**:266–277.

[13] Juang, B. and Katagiri, S. (1992) Discriminative learning for minimum error classification. *IEEE Transactions on Acoustics, Speech, and Signal Processing* **40**: 3043–3054.

[14] Kambhatla, N, and Leen, T. (1994) Fast non-linear dimension reduction. In J. Cowan, G. Tesauro, and J. Alspector, eds. *Advances in Neural Information Processing Systems* **6**:152–159. San Francisco: Morgan Kauffman.

[15] Leggetter, C. and Woodland, P. (1995) Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. *Computer speech and language* 9:171–185.

[16] Ljolje, A. (1994) The importance of cepstral parameter correlations in speech recognition. *Computer Speech and Language* **8**:223-232.

[17] Ljolje, A., Ephraim, Y., and Rabiner, L. (1990) Estimation of hidden Markov model parameters by minimizing empirical error rate. In *Proceedings of ICASSP 1990*: 709–712.

[18] Oja, E. (1983) *Subspace Methods of Pattern Recognition*. Research Studies Press.

[19] Ostendorf, M., Digalakis, V., and Kimball, O. (1996) From HMMs to segment models: a unified view of stochastic modeling for speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing* **4**: 360–378.

[20] Press, W., Teukolsky, S., Vetterling, W., and Flannery, B. (1992) *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge: Cambridge University Press.

[21] Rabiner, L., and Juang, B. (1993) *Fundamentals of Speech Recognition*. Englewood Cliffs: Prentice Hall.

[22] Rahim, M., Juang, B., Chou, W., and Buhrke, E. (1996) Signal conditioning techniques for robust speech recognition. *IEEE Signal Processing Letters* **3**:107–109.

[23] Roweis, S. (1998) EM algorithms for PCA and SPCA. In Jordan, M., Kearns, M., and Solla, S. eds. *Advances in Neural Information Processing Systems* **10**:626–632. Cambridge, MA: MIT Press.

[24] Rubin, D., and Thayer, D. (1982) EM algorithms for factor analysis. *Psychometrika* **47**:69–76.

[25] Sachs, R., Tikijian, M., and Roskos, E. (1994) United States English subword speech data. *AT&T Unpublished report*.

[26] Schukat-Talamazzini, E., Hornegger, J., and Niemann, H. (1995) Optimal linear feature transformations for semi-continuous hidden Markov models. In *Proceedings of ICASSP 95*: 369–372.

[27] Tipping, M. E., and Bishop, C. M. (1997) Mixtures of principal component analysers. In *Proceedings of the IEEE Fifth International Conference on Artificial Neural Networks*.

[28] Watanabe, H., Yamaguchi, T., and Katagiri, S. (1997) Discriminative metric design for robust pattern recognition. *IEEE Transactions on Signal Processing* **45**: 2655–2662.

[29] Zavaliagkos, G., Schwartz, R., and Makhoul, R. (1995) Batch, incremental and instantaneous adaptation techniques for speech recognition. In *Proceedings of ICASSP 1995*: 676–679.