T-AS/38/2//32768

A Linear Predictive HMM for Vector-Valued Observations with Applications to Speech Recognition

Patrick Kenny

Matthew Lennig

Paul Mermelstein

Reprinted from
IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING
Vol. 38, No. 2, February 1990

A Linear Predictive HMM for Vector-Valued Observations with Applications to Speech Recognition

PATRICK KENNY, MATTHEW LENNIG, SENIOR MEMBER, IEEE, AND
PAUL MERMELSTEIN, SENIOR MEMBER, IEEE

Abstract—**In this paper we describe a new type of Markov model which we have developed to account for the correlations between successive frames of the speech signal. The idea is to treat the sequence of frames as a nonstationary autoregressive process whose parameters are controlled by a hidden Markov chain. We show that this type of model performs better than the standard multivariate Gaussian HMM when it is incorporated into a large-vocabulary isolated-word recognizer.**

I. Introduction

The technique of Markov modeling has aroused widespread interest in self-organizing pattern classifiers as a tool for speech recognition. Statistical methods have long been popular in other areas of pattern recognition, but it is only with the advent of Hidden Markov Models that it has become possible to use such methods on a large scale in speech recognition. The principal obstacle has been that the speech signal is manifestly nonstationary, whereas traditional stochastic models are only equipped to handle sequences of independent identically distributed (IID) observations or stationary time series.

The standard left-to-right HMM provides a technique for studying nonstationary time series (which may be vector or scalar valued). It is based on the assumption that the observations are *locally* IID: during its sojourn in a state of the Markov chain, the model generates observations by random sampling from the output distribution associated with the state. The only type of statistical dependence between observations allowed for by the model is due to the underlying Markov chain. Because of the Markov property, it decays rapidly in time, and in the degenerate case where all the output distributions coincide there is no dependence at all. A more flexible way of handling correlations in nonstationary time series is clearly desirable.

The general ARMA process is a stationary, linear model for sequences of identically distributed observations which are *not* independent. A natural way of studying the correlations between frames of the speech signal is to use a hidden Markov chain to incorporate nonstationarity into a vector-valued ARMA process in the same way that the standard HMM incorporates nonstationarity into an IID process. The purpose of this paper is to implement this for autoregressive processes. For a completely different approach to the problem, using neither AR processes nor HMM's, see the stochastic segment model of BBN [1].

Nonstationary autoregressive processes have received considerable attention in the past [2]–[4], both for their own sake and as a tool for speech analysis and recognition. The principal method for modeling nonstationarity has been to express the time-varying regression coefficients as a linear combination of a small number of basis functions, chosen arbitrarily. Poritz [5] was the first to use an HMM to capture nonstationarity in an AR process. Our model is formally very similar to the *Hidden Filter* model [6] (and, indeed, contains it as a special case) but it is designed to handle the speech signal at the frame level, where it is represented by feature vectors, rather than dealing with the signal directly.

We first describe our model and develop versions of the forward–backward and Baum–Welch algorithms for it. In order to use the model in speech recognition, it is necessary to find ways of reducing the number of parameters to be estimated. We have tried two methods of doing this and report the results of both series of experiments.

II. The General Linear Predictive Markov Model

We consider a vector-valued autoregressive process whose parameters are allowed to vary in time according to the evolution of a hidden Markov chain.

With each transition in the Markov chain, we associate a set of regression coefficients together with a mean vector and a covariance matrix which serve to characterize the distribution of the prediction error. To be precise, suppose that the Markov chain is in state s at time $t - 1$, and in state s' at time t, and let Y_t stand for the output of the process at time t. Then we assume that Y_t can be written in the form

$$Y_t = A(s, s') + B_1(s, s') Y_{t-1} + \cdots + B_p(s, s') Y_{t-p} + E_t. \quad (1)$$

Manuscript received February 24, 1988; revised February 17, 1989. This work was supported by the Natural Sciences and Engineering Research Council of Canada.

P. Kenny is with INRS-Télécommunications, 3 Place du Commerce, Montreal, P.Q., Canada H3E 1H6.

M. Lennig and P. Mermelstein are with INRS-Télécommunications, Montreal, P.Q., Canada, and Bell-Northern Research, Montreal, P.Q., Canada.

IEEE Log Number 8932768.

Here Y_t and $A(s, s')$ are $d \times 1$ vectors, the B's are $d \times d$ matrices, and the residual E_t is a $d \times 1$ random vector having a Gaussian distribution with mean 0 and covariance matrix $\Sigma(s, s')$; moreover, the residuals at different times are assumed to be independent. (It is natural to use matrix regression coefficients unless the components of the observation vectors are uncorrelated.)

It will be convenient to rewrite (1) in the short-hand form

$$Y_t = A(s, s') + B(s, s')X_t + E_t.$$

Here, X_t is the dp-dimensional column vector

$$\begin{pmatrix} Y_{t-1} \\ \vdots \\ Y_{t-p} \end{pmatrix}$$

and $B(s, s')$ is the $d \times dp$ matrix $(B_1(s, s') \mid \cdots \mid B_p(s, s'))$.

The Hidden Filter model is obtained by setting $d = 1$ and $A = 0$. Note that, for a stationary autoregressive process, the dc term A is necessarily 0; but for our nonstationary model there is no such restriction and we treat A as a parameter to be trained in the same way as the regression coefficients. The reason for doing so is to retain the descriptive power of the standard multivariate Gaussian HMM (MVGHMM), which can be obtained by setting $B = 0$. In fact, our model also contains a well-known variant of the MVGHMM as a special case: if we take $B = (0 \mid \cdots \mid 0 \mid I)$, we obtain the "dynamic parameters" model (without changing the parameter set).

In this section we show how to estimate the parameters of the model, namely, $A(s, s')$, $B(s, s')$, and $\Sigma(s, s')$ as well as the transition probabilities of the hidden Markov chain, from a sequence of observations $Y_{-p+1}, \cdots, Y_0, Y_1, \cdots, Y_T$.

A. The Forward–Backward Algorithm

If $\mathcal{S} = (s_0, \cdots, s_T)$ is a sequence of states, we denote by $P(Y, \mathcal{S} \mid X_1)$ the joint likelihood of the observation sequence $Y_1, \cdots, Y_T$ and the event that Y_1 is emitted on making the transition from s_0 to s_1, Y_2 on the transition from s_1 to s_2, etc., all conditioned on X_1; similarly, we use $P(Y \mid \mathcal{S}, X_1)$ to stand for the conditional likelihood.

The conditional likelihood is easily calculated. The residual E_t has density function

$$\frac{1}{(2\pi)^{d/2}\left|\Sigma(s_{t-1}, s_t)\right|^{1/2}} \exp\left\{-\frac{1}{2} E_t^* \Sigma(s_{t-1}, s_t)^{-1} E_t\right\}$$

and, because the residuals are independent, the joint density function of $E_1, \cdots, E_T$ conditioned on $\mathcal{S}$ is just the product

$$\prod_{t=1}^{T} \frac{1}{(2\pi)^{d/2}\left|\Sigma(s_{t-1}, s_t)\right|^{1/2}} \cdot \exp\left\{-\frac{1}{2} E_t^* \Sigma(s_{t-1}, s_t)^{-1} E_t\right\}.$$

(The asterisk indicates the transpose.) It follows that

$$P(Y \mid \mathcal{S}, X_1) = \prod_{t=1}^{T} D(X_t, Y_t, s_{t-1}, s_t)$$

where $D(X_t, Y_t, s_{t-1}, s_t)$ is being used to stand for the quantity

$$\frac{1}{(2\pi)^{d/2}\left|\Sigma(s_{t-1}, s_t)\right|^{1/2}} \exp\left\{-\frac{1}{2}(Y_t - A(s_{t-1}, s_t) - B(s_{t-1}, s_t)X_t)\, \Sigma(s_{t-1}, s_t)^{-1}(Y_t - A(s_{t-1}, s_t) - B(s_{t-1}, s_t)X_t)\right\}.$$

which is just the likelihood of the observation Y_t conditioned on the previous observations X_t and the transition $s_{t-1} \rightarrow s_t$.

The probability of the state sequence $\mathcal{S} = (s_0, \cdots, s_T)$ is just the product of the transition probabilities

$$P(\mathcal{S}) = \prod_{t=1}^{T} P(s_t \mid s_{t-1})$$

(assuming that the starting distribution is concentrated on the state s_0) and the joint likelihood $P(Y, \mathcal{S} \mid X_1)$ can be found by combining the last two equations. The total likelihood of the observation sequence can now be obtained (in principle, if not in practice) by summing these joint likelihoods over all possible state sequences $\mathcal{S}$. However, the forward–backward algorithm [7] does this more efficiently.

For $t = 1, \cdots, T$, we define the forward probabilities by the equation

$$\alpha_t(s) = P(s_t = s, Y_1, \cdots, Y_t \mid X_1)$$

and for $t = 0, \cdots, T - 1$, we define the backward probabilities by

$$\beta_t(s) = P(Y_{t+1}, \cdots, Y_T \mid s_t = s, X_t)$$

for each state s in the Markov chain. The forward and backward probabilities can be calculated recursively from the formulas

$$\alpha_t(s') = \sum_{s} \alpha_{t-1}(s)\, P(s' \mid s)\, D(X_t, Y_t, s, s')$$

$$\beta_t(s) = \sum_{s'} P(s' \mid s)\, D(X_{t+1}, Y_{t+1}, s, s')\, \beta_{t+1}(s')$$

once we have starting values for α and terminal values for β. For the application we have in mind, we will be using a left-to-right model so we distinguish two states s_i (the initial state) and s_f (the final state) and constrain the state sequences $\mathcal{S} = (s_0, \cdots, s_T)$ to satisfy $s_0 = s_i$ and $s_T = s_f$.

The recursion formulas will give the correct values for α_1 and β_{T-1} if we adopt the conventions

$$\alpha_0(s) = \begin{cases} 1 & \text{if } s = s_i \\ 0 & \text{otherwise} \end{cases}$$

and

$$\beta_T(s) = \begin{cases} 1 & \text{if } s = s_f \\ 0 & \text{otherwise.} \end{cases}$$

The total likelihood of the observation sequence can now be obtained from the forward recursion by

$$P(Y_1, \cdots, Y_T | X_1) = \alpha_T(s_f).$$

One additional piece of notation will be useful. If s and s' are two states, we define $\gamma_t(s, s')$ to be the probability that the Markov chain is in state s at time $t - 1$ and in state s' at time t conditioned on the observation sequence. In terms of the forward and backward probabilities;

$$\gamma_t(s, s') = \frac{P(s' | s)\, \alpha_{t-1}(s)\, \beta_t(s')\, D(X_t, Y_t, s, s')}{P(Y_1, \cdots, Y_T | X_1)}.$$

B. The Reestimation Formulas: Single Token Case

As with other types of hidden Markov model, the EM algorithm provides an iterative solution to the problem of parameter estimation [6]. We define the auxiliary Q function as in [8]. Just as for the MVGHMM, the problem of optimizing the Q function turns out to be an exercise in least-squares so it is quite straightforward to find the critical point in closed form and we merely outline the derivation.

Let $Y = (Y_1, \cdots, Y_T)$ and suppose M_0 and M are two models corresponding to different choices for the parameter values. For each state sequence $\mathcal{S}$ of length $T + 1$, let $P_0(\mathcal{S})$ stand for the probability of $\mathcal{S}$ conditioned on the observation sequence, calculated using the parameters of the model M_0. Define

$$Q(M_0, M) = \sum_{\mathcal{S}} P_0(\mathcal{S}) \ln P(Y, \mathcal{S} | X_1, M).$$

The following lemma is a simple consequence of the convexity of the logarithmic function [8].

Lemma:

$$\ln \left(\frac{P(Y | X_1, M)}{P(Y | X_1, M_0)} \right) \geq Q(M_0, M) - Q(M_0, M_0).$$

The point of this inequality is that if M_0 is the model corresponding to an initial estimate of the parameters, the likelihood of the observation sequence can be increased by choosing the parameters of the new model M so as to maximize $Q(M_0, M)$.

A manageable description of Q can be obtained by straightforward manipulation:

$$Q = \sum_{s,s'} \sum_{t=1}^{T} \gamma_t(s, s' | M_0) (\ln D(X_t, Y_t, s, s') + \ln P(s' | s))$$

where the outer sum extends over all pairs of states s, s' in the Markov chain.

The reestimation formulas for the transition probabilities are obtained by maximizing the term $\Sigma_{s,s'} \Sigma_{t=1}^{T} \gamma_t(s, s' | M_0) \ln P(s' | s)$; the new estimates are

$$P(s' | s) = \frac{\Sigma_{t=1}^{T} \gamma_t(s, s' | M_0)}{\Sigma_{s_1} \Sigma_{t=1}^{T} \gamma_t(s, s_1 | M_0)}.$$

Since we are assuming that there are no constraints on the model relating the regression parameters associated with different transitions,[1] the reestimation formulas for the regression parameters are obtained by maximizing $\Sigma_t \gamma_t(s, s' | M_0) \ln D(X_t, Y_t, s, s')$ for each pair s, s'; dropping the reference to s, s' and ignoring the constant term $\ln (2\pi)^{d/2}$, the objective function is

$$M(A, B, \Sigma) = \sum_{t=1}^{T} \gamma_t \{ -\tfrac{1}{2} \ln |\Sigma| - \tfrac{1}{2} (Y_t - A - BX_t)^* \Sigma^{-1} \cdot (Y_t - A - BX_t) \}.$$

Setting the derivatives of M with respect to A in all directions equal to 0 gives the vector equation

$$S_Y = NA + BS_X$$

where

$$S_Y = \sum_t \gamma_t Y_t$$

$$N = \sum_t \gamma_t.$$

Likewise, setting the derivatives of M with respect to B equal to 0 gives the matrix equation

$$S_{YX} = AS_X^* + BS_{XX}$$

where

$$S_X = \sum_{t=1}^{T} \gamma_t X_t$$

$$S_{XX} = \sum_{t=1}^{T} \gamma_t X_t X_t^*$$

$$S_{YX} = \sum_{t=1}^{T} \gamma_t Y_t X_t^*.$$

The new estimates for A and B can now be obtained by solving the pair of simultaneous equations

$$\begin{cases} S_Y = NA + BS_X \\ S_{YX} = AS_X^* + BS_{XX} \end{cases}$$

which, incidentally, bear an interesting resemblance to the equations for the slope and intercept in simple linear regression.

The reestimation formula for Σ is obtained as in the case of the MVGHMM [9]:

$$\Sigma = \frac{1}{N} \sum_{t=1}^{T} \gamma_t (Y_t - A - BX_t)(Y_t - A - BX_t)^*$$

[1] The Appendix outlines the changes that have to be made when such constraints are imposed.

where, as before, $N = \Sigma_t \gamma_t$. An efficient formula for implementation can be obtained by multiplying this out and using the equation

$$S_Y = NA + BS_X$$

to simplify the result. This gives

$$\Sigma = \frac{1}{N}(S_{YY} - BS_{YX}^* - S_{YX}B^* + BS_{XX}B^* - NAA^*)$$

where

$$S_{YY} = \sum_{t=1}^{T} \gamma_t Y_t Y_t^*.$$

C. Multiple Tokens

A left-to-right model. cannot, in general, be reliably trained with a single token, so suppose that we have L tokens $Y^{(1)}, \cdots, Y^{(L)}$ of lengths $T^{(1)}, \cdots, T^{(L)}$ at our disposal. For each token $Y^{(l)}$, calculate

$$\gamma_t^{(l)}(s, s' | M_0) = P(s_{t-1} = s, s_t = s' \,|\, Y^{(l)}, M_0)$$

for $t = 1, \cdots, T^{(l)}$. For each pair of states s, s', the matrices $S_X(s, s')$, $S_Y(s, s')$, $S_{XX}(s, s')$, $S_{YX}(s, s')$, $S_{YY}(s, s')$ and the scalar $N(s, s')$ are now defined by

$$S_X(s, s') = \sum_{l=1}^{L} \sum_{t=1}^{T^{(l)}} \gamma_t^{(l)}(s, s' | M_0) X_t^{(l)}$$

$$S_Y(s, s') = \sum_{l=1}^{L} \sum_{t=1}^{T^{(l)}} \gamma_t^{(l)}(s, s' | M_0) Y_t^{(l)}$$

$$S_{XX}(s, s') = \sum_{l=1}^{L} \sum_{t=1}^{T^{(l)}} \gamma_t^{(l)}(s, s' | M_0) X_t^{(l)} X_t^{(l)*}$$

$$S_{YX}(s, s') = \sum_{l=1}^{L} \sum_{t=1}^{T^{(l)}} \gamma_t^{(l)}(s, s' | M_0) Y_t^{(l)} X_t^{(l)*}$$

$$S_{YY}(s, s') = \sum_{l=1}^{L} \sum_{t=1}^{T^{(l)}} \gamma_t^{(l)}(s, s' | M_0) Y_t^{(l)} T_t^{(l)*}$$

$$N(s, s') = \sum_{l=1}^{L} \sum_{t=1}^{T^{(l)}} \gamma_t^{(l)}(s, s' | M_0)$$

and the reestimation equations take the form

$$S_Y(s, s') = N(s, s') A(s, s') + B(s, s') S_X(s, s')$$

$$S_{YX}(s, s') = A(s, s') S_X^*(s, s') + B(s, s') S_{XX}(s, s')$$

$$\begin{aligned}\Sigma(s, s') = \frac{1}{N(s, s')} \big(&S_{YY}(s, s') - B(s, s') S_{YX}^*(s, s') \\ &- S_{YX}(s, s') B^*(s, s') \\ &+ B(s, s') S_{XX}(s, s') B^*(s, s') \\ &- N(s, s') A(s, s') A^*(s, s')\big)\end{aligned}$$

$$P(s' | s) = \frac{\Sigma_{l=1}^{L} \Sigma_{t=1}^{T^{(l)}} \gamma_t^{(l)}(s, s' | M_0)}{\Sigma_{s_1} \Sigma_{l=1}^{L} \Sigma_{t=1}^{T^{(l)}} \gamma_t^{(l)}(s, s_1 | M_0)}$$

The derivation is similar to the single token case.

III. Experimental Results

We have tried implementing several variants of the linear predictive model in our experimental large-vocabulary speaker-dependent isolated-word recognizer (described in detail in [10]).

The recognizer is phoneme-based with a set of 44 phonemes each represented by a left-to-right HMM. Our best results have been obtained using an MVGHMM constrained so that all the output distributions associated with the transitions in a phoneme model have the same covariance matrix. The number of transitions in a model can be as large as 30 (in the case of diphthongs) so pooling the covariance matrices is necessary to ensure robust estimation; it also has the advantage of reducing the amount of computer time needed in training and recognition.

Using a window of length 25 ms, we calculate a set of eight mel-based cepstral coefficients ($c_0, \cdots, c_7$) every 10 ms; c_0 is the loudness. As our feature vector for the MVGHMM we take ($c_1, \cdots, c_7, \Delta c_0, \cdots, \Delta c_7$) where Δc_0 is calculated by taking the difference between the loudness over an interval of length 40 ms; likewise for $\Delta c_1, \cdots, \Delta c_7$.

We had a speaker record a set of 1203 words (consisting of several short texts and a number of words chosen to represent various phoneme clusters) which was then hand-segmented into phonemes. Using the 15-dimensional feature vector ($c_1, \cdots, c_7, \Delta c_0, \cdots, \Delta c_7$) the MVGHMM gives a recognition rate of 86.7 percent. Performance drops to 79.4 percent when the parameter set is restricted to ($c_1, \cdots, c_7, \Delta c_0$); this was the parameter vector used for the linear predictive models.[2] We used a test set of 399 words of text in all experiments; the dictionary consisted of 60 000 words and a uniform[3] language model was used.

A major issue in implementing the linear predictive model is to decide which lags to use in the regression. In a preliminary experiment, we took $p = 1$ and obtained a poor recognition rate of 73 percent. (Paradoxically, the likelihoods obtained on the training and test data were much higher than for any other model we have worked with.) We decided to exclude lag 1 which made it necessary to change the estimation formulas slightly. Quite generally, the reestimation formulas can be used to train an autoregressive model of the form

$$Y_t = A + B_1 Y_{t-l_1} + \cdots + B_p Y_{t-l_p} + E_t$$

if we define X_t to be

$$\begin{pmatrix} Y_{t-l_1} \\ \vdots \\ Y_{t-l_p} \end{pmatrix}$$

and proceed as before.

[2] Δc_0 is a better choice of parameter than c_0 for recognizers based on standard Markov models since it is independent of the overall energy level. However, it might be appropriate to use c_0 itself for the linear predictive model and rely on the regression terms to track the variation of c_0 from one frame to the next. We did not explore this possibility.

[3] All words in the vocabulary are considered *a priori* equally likely, regardless of context.

Also, there is the problem of robust estimation. In our first implementation (reported in Table I), we constrained the models so that the regression and covariance matrices were the same for all transitions in each of the phoneme models. This complicates the reestimation formulas somewhat (see the Appendix). In this framework, the recognizer uses the same prediction mechanism for all transitions in a phone model, and the uncertainty as to which transition corresponds to a given frame is rather high. This may explain why the improvement over the standard Markov model is small.

In the other implementation that we tried, the regression and covariance matrices were constrained to be diagonal (Table II). This amounts to treating the components of the frame vector as if they were independent of each other (a reasonable assumption in the case of cepstral coefficients). The reestimation formulas for the parameters of the model corresponding to each component can be obtained by simply putting $d = 1$ in the above derivation. In this case, we pooled the regression and covariance matrices for all transitions from each of the states but imposed no tying across states.

When the MVGHMM is trained with analogous constraints on the covariance matrices and the full 15-dimensional feature vector is used, the recognition rate is 85.5 percent.

As a final experiment, we tried combining the two methods by taking the 15-dimensional feature vector as input to the diagonal linear predictive model with lag 4. The performance degraded to 78.7 percent, presumably because the number of parameters to be estimated was too great.

IV. Discussion

The general problem of decorrelating the frames of the speech signal requires a global computational model of speech dynamics, i.e., a good theory. Our work is merely experimental. We chose to work with the HMM formalism because it is the simplest global model available, and the only reason for choosing AR rather than MA processes is that they have worked before in speech analysis and the mathematics is tidier; there are undoubtedly many other linear statisticɑl models that can be easily integrated with the EM algorithm, some of which may be better able to handle the correlations between frames.

We found that our linear predictive model outperforms the MVGHMM in our large-vocabulary isolated-word recognition task when the parameter set $(c_1, \cdots, c_7, \Delta c_0)$ is used, but that our best results are obtained with the MVGHMM and the parameter set $(c_1, \cdots, c_7, \Delta c_0, \cdots, \Delta c_7)$.

As explained in Section II, the linear predictive model contains both the "static" and "dynamic" MVGHMM's as special cases, so it is somewhat surprising that the "static + dynamic" MVGHMM should give better performance than any of the versions of the linear predictive model that we implemented. (Since we used the same training set for all models, irrespective of the number of parameters to be estimated, this could be just a matter of undertraining.)

TABLE I
Pooled Regression and Covariance Matrices

Model	Recognition Rate
$p = 1, l_1 = 2$	78.9 percent
$p = 1, l_1 = 4$	81.0 percent
$p = 1, l_1 = 6$	80.2 percent
$p = 1, l_1 = 8$	81.0 percent
$p = 2, l_1 = 3, l_2 = 6$	81.0 percent

TABLE II
Diagonal Regression and Covariance Matrices

Model	Recognition Rate
$p = 1, l_1 = 4$	82.7 percent
$p = 2, l_1 = 4, l_2 = 8$	83.0 percent
$p = 3, l_1 = 2, l_2 = 4, l_3 = 6$	81.4 percent
$p = 3, l_1 = 4, l_2 = 6, l_3 = 8$	83.0 percent

The "static + dynamic" MVGHMM is in fact very similar to the linear predictive model with $p = 1$. Wellekens [11] points out that when the dynamic parameter vector for each frame is constructed by adjoining the static parameters from the previous frame, the lag 1 correlation matrices between the frames occur as submatrices of the covariance matrices of the Markov model, and the correlation matrix is essentially the regression coefficient $\boldsymbol{B}$. The major difference is that the mean vectors in the "static + dynamic" model contain more information than those of the linear predictive model (being of twice the dimensionality). Both models account for short-term dependencies between frames and ignore long-term dependencies between phonemes. Other work that we have done [12] leads us to believe that the latter type of dependency is probably the more important. Needless to say, it is also much more difficult to model mathematically.

Appendix
Pooling the Estimates of the Regression and Covariance Matrices

Suppose we are given a linear predictive Markov model and a partition of the transitions into classes such that the transitions in each class have common regression and covariance matrices. The transitions within each class $\mathcal{C}$ can no longer be treated independently of each other in deriving the reestimation formulas but the argument is very similar—in this case, one maximizes

$$\sum_{(s,s')} \sum_{t} p_t(s, s' | M_0) \ln D(X_t, Y_t, s, s')$$

where the outer sum extends over all transitions $s \rightarrow s'$ in $\mathcal{C}$. We merely state the result here.

Calculate $S_X(s, s')$, $S_Y(s, s')$, $S_{XX}(s, s')$, $S_{YX}(s, s')$, $S_{YY}(s, s')$, and $N(s, s')$ as above for each $(s, s') \in \mathcal{C}$. Let

$$S_{XX} = \sum_{(s,s')\in\mathcal{C}} S_{XX}(s, s')$$

$$S_{YX} = \sum_{(s,s')\in\mathcal{C}} S_{YX}(s, s')$$

$$S_{YY} = \sum_{(s,s')\in\mathcal{C}} S_{YY}(s, s')$$

$$N = \sum_{(s,s')\in\mathcal{C}} N(s, s').$$

The reestimation equations are

$$S_Y(s, s') = N(s, s')\, A(s, s') + BS_X(s, s') \qquad ((s, s') \in \mathcal{C})$$

$$S_{YX} = \sum_{(s,s')\in\mathcal{C}} A(s, s')\, S_X(s, s')^* + BS_{XX}$$

$$\Sigma = \frac{1}{N}\Bigg(S_{YY} - BS_{YX}^* - S_{YX}B^* + BS_{XX}B^* - \sum_{(s,s')\in\mathcal{C}} N(s, s')\, A(s, s')\, A(s, s')^* \Bigg).$$

REFERENCES

[1] S. Roucos, M. Ostendorf, H. Gish, and A. Derr, "Stochastic segment modelling using the estimate-maximize algorithm," in *Proc. ICASSP*, vol. 1, 1988, pp. 127-130.

[2] L. A. Liporace, "Linear estimation of nonstationary signals," *J. Acoust. Soc. Amer.*, vol. 58, pp. 1288-1295, 1975.

[3] Y. Grenier, "Time-dependent ARMA modeling of nonstationary signals," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 899-911, 1986.

[4] R. Charbonnier, M. Barlaud, G. Allengrin, and J. Menez, "Results on AR-modelling of nonstationary signals," *Signal Processing*, vol. 12, pp. 143-151, 1987.

[5] A. B. Poritz, "Linear predictive hidden Markov models and the speech signal," in *Proc. ICASSP*, 1982, pp. 1291-1294.

[6] —, "Hidden Markov models: A guided tour," in *Proc. ICASSP*, vol. 1, 1988, pp. 7-13.

[7] L. E. Baum, "An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes," *Inequalities*, vol. 3, pp. 1-8, 1972.

[8] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc.*, Ser. 39, pp. 1-38, 1979.

[9] L. A. Liporace, "Maximum likelihood estimation for multivariate observations of Markov sources," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 729-734, 1982.

[10] V. Gupta, M. Lennig, and P. Mermelstein, "Fast search strategy in a large vocabulary word recognizer," *J. Acoust. Soc. Amer.*, vol. 84, no. 6, 1988.

[11] C. Wellekens, "Explicit time correlation in hidden Markov models for speech recognition," in *Proc. ICASSP*, 1987, pp. 384-386.

[12] L. Deng, V. N. Gupta, M. Lennig, and P. Mermelstein, "Modelling acoustic-phonetic detail in an HMM-based large vocabulary speech recognizer," in *Proc. ICASSP*, 1988, pp. 509-512.

Patrick Kenny was born in Montreal, Canada, in 1955. He graduated from Trinity College, Dublin, Ireland, with first class honors in mathematics in 1976 and received the M.Sc. and Ph.D. degrees, also in mathematics, from McGill University.

He has been working in speech recognition at INRS-Télécommunications, Université du Québec, since 1986, and he is particularly interested in stochastic modeling of the speech signal.

Matthew Lennig (M'79-SM'85) received the A.B. degree (summa cum laude) from Princeton University in 1974 in theoretical linguistics; he received the Ph.D. degree in sociolinguistics in 1978 from the University of Pennsylvania, which he attended as a National Science Foundation Fellow; and the M.Eng. degree in electrical engineering from McGill University in 1984.

He joined Bell-Northern Research in 1978 and is currently the Manager of Interactive Voice Systems, with responsibility for the development of the interactive voice technology component of Northern Telecom's AABS product, which uses speech recognition in the telephone network to automate collect and third-number billed calls. Prior to 1988 he was the Manager of Speech Systems, with responsibility for development of Bell Canada's speech-recognition-based 976 Directory and for algorithmic research in the areas of speech recognition, speaker verification, and speech synthesis. Since 1981 he has been a Visiting Professor at l'Institut National de la Recherche Scientifique en Télécommunications (Université du Québec) where he currently directs a research project on very large (86 000-word) vocabulary speech recognition.

Paul Mermelstein (S'58-M'63-SM'77) was born in Czechoslovakia in 1939. He received the B.Eng. degree in engineering physics from McGill University, Montreal, P.Q., Canada, in 1959, and the S.M., E.E., and D.Sc. degrees from the Massachusetts Institute of Technology, Cambridge, in 1960, 1963, and 1964, respectively.

From 1964 to 1973 he was a member of the Technical Staff in the Speech and Communications Research Department of Bell Laboratories, Murray Hill, NJ. From 1973 to 1977 he was a member of the Research Staff of Haskins Laboratories conducting research in speech analysis, perception, and recognition. Over the years 1977-1986 he has been Manager of Speech Communication Systems at Bell-Northern Research in Montreal. He currently serves as Manager of Man-Machine Systems at BNR. He served as Associate Editor for Speech Processing for the *Journal of the Acoustical Society of America* in 1983-1987. He has also served on numerous expert groups on speech coding of CCITT WP8 recommending standards on speech coding for telecommunication applications. Since 1977 he has held appointments as Visiting Professor at INRS-Télécommunications (Université du Québec) and Auxiliary Professor of Electrical Engineering at McGill University.

Dr. Mermelstein is Editor for Speech Communication of the IEEE TRANSACTIONS ON COMMUNICATIONS.