Robust Speech Recognition Using Linear Dynamic Models

Tao Ma[1], Sundar Srinivasan[1], Daniel May[1], Georgios Lazarou[2] and Joseph Picone[1]

[1] Department of Electrical and Computer Eng., Mississippi State University, MS State, MS, USA
[2] New York City Transit Authority, New York, New York, USA
{tm334, ss754, dom5, glaz, picone}@ece.msstate.edu

Abstract

We propose Linear Dynamic Models (LDMs) as an alternative to hidden Markov models (HMMs) for robust speech recognition in noisy environments. HMMs in speech recognition typically utilize a diagonal covariance matrix assumption in which correlations between feature vectors for adjacent frames are ignored. LDMs use a state space-like formulation that explicitly models the evolution of hidden states using an autoregressive process. This smoothed trajectory model allows the system to better track speech dynamics in noisy environments. We demonstrate that LDMs provide a 4.9% relative improvement on the Aurora-4 clean evaluation set, and a 6.5% relative improvement on the noisy evaluation set.

Index Terms: linear dynamic models, speech recognition, acoustic modeling, nonlinear statistical modeling

1. Introduction

Over the past several decades, hidden Markov models (HMMs) have been the most popular approach for acoustic modeling in speech recognition. An HMM can be regarded as a finite state machine in which the states of the system evolve in accordance with an inherent deterministic mechanism and the emission probability function maps the hidden states to observation domain. HMM modeling techniques for speech recognition have relied on the standard assumption that speech features are temporally uncorrelated. Recent theoretical and experimental evidence [1][2][3] has suggested that exploiting frame to frame correlations in the speech signal will further improve performance of speech recognition systems by developing an acoustic model which represents higher order statistics in the signal. The phone level in the speech recognition modeling hierarchy is a good level to explore exploiting such statistics [4].

Linear Dynamic Models (LDMs) have generated significant interest in recent years [4][5] due to their ability to model higher order statistics. The fundamental idea behind an LDM is to describe a linear dynamic system as underlying states and observables with a measurement equation to link the internal states to the observables, and an autoregressive model to capture the time-evolution of the states [2][4]. An LDM models every word or phoneme segment as a nonseparable unit which incorporates the dynamic evolution of the hidden states. Digalakis *et al*. [3] present both an LDM maximum likelihood approach and a derivation of the EM algorithm [3]. In subsequent work by Frankel and King [4], LDMs were applied to the acoustic modeling problem to model articulatory dynamics.

In this paper, we began with Digalakis and Frankel's work as a starting point, and refined these approaches for a more difficult evaluation task – the Aurora-4 large vocabulary evaluation task [6] that includes clean and noisy speech data as well as conditions simulating mismatched training conditions. We show that LDM provides improved performance in noisy environments.

The outline of this paper is as follows. In Section 2 we briefly review the underlying concept behind LDMs including model assumptions and equations, state inference and smoothing, and EM training. In Section 3, we describe the preliminary sustained phoneme classification experiments that demonstrate the ability of LDM to model acoustics better than traditional HMM. We also present phonetic recognition results of the Aurora-4 corpus which demonstrates LDM as a good acoustic modeling technique for noise robust speech recognition. We conclude with a discussion of our ongoing research on extending these approaches to more challenging and complex speech recognition tasks.

2. Linear Dynamic Models

Linear Dynamic Models (LDMs) are an example of a Markovian state space model, and in some sense can be regarded as analogous to an HMM since LDMs do use hidden state modeling. With LDMs, systems are described as underlying states and observables combined together by a measurement equation. Every observable will have a corresponding hidden internal state. This is illustrated in Figure 1.

Suppose y_t is a p-dimension observation vector and x_t is a q-dimension internal state vector. The LDM formulation is based on a state-space model:

$$\begin{aligned} x_{t+1} &= Fx_t + \omega_t \\ y_t &= Hx_t + \nu_t \end{aligned} \tag{1}$$

where F is the state evolution matrix and H is the observation transformation matrix. The variables ω_t and ν_t are assumed to be uncorrelated white Gaussian noise with covariance matrices Q and R, respectively, which drive the linear stochastic system. The sequence of observations, y_t, and the sequence of underlying states, x_t, are finite dimensional and follow multivariate Gaussian distributions for every time t. The first equation is an autoregressive state process which describes how states evolve from one time frame to the next [3]. The second equation maps the output observations to the internal states.

The system's hidden states are the deterministic characteristic of an LDM which are also affected by random Gaussian noise [7]. The state and noise variables can be combined into one single Gaussian random variable. Based on Figure 1, the conditional density functions for the states and

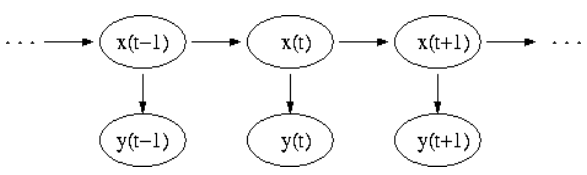

Figure 1: *Internal states and observations in a LDM.*

output can be written as follows:

$$
\begin{aligned}
P(y_t \mid x_t) &= \exp\{-0.5[y_t - Hx_t]' R^{-1} \\
&\quad [y_t - Hx_t]\}(2\pi)^{-p/2} \mid R \mid^{-0.5} \\
P(x_t \mid x_{t-1}) &= \exp\{-0.5[x_t - Fx_{t-1}]' Q^{-1} \\
&\quad [x_t - Fx_{t-1}]\}(2\pi)^{-k/2} \mid Q \mid^{-0.5}
\end{aligned}
\tag{2}
$$

According to the Markovian assumption, the joint probability density function of the states and observations becomes:

$$
P(\{x\},\{y\}) = P(x_1)\prod_{t=2}^{T} P(x_t \mid x_{t-1})\prod_{t=1}^{T} P(y_t \mid x_t)\,. \tag{3}
$$

The system's states are hidden. We need to estimate the hidden state evolution given an N-length observation sequence y_t and the model parameters. This can be accomplished using a Kalman filter combined with a Rauch-Tung-Striebel (RTS) smoother. The Kalman filter provides an estimate of the state distribution at time t given all the observations up to and including that time. The RTS smoother gives a corresponding estimate of the underlying state conditions over the entire observation sequence. For the smoothing part, a fixed interval RTS smoother is used to compute the required statistics once all data has been observed.

The RTS smoother adds a backward pass that follows the standard Kalman filter forward recursion [2]. In addition, in both the forward and the backward pass, we need some additional recursions for the computation of the cross-covariance. The RTS equations are:

$$
\begin{aligned}
&\hat{x}_{t-1/N} = \hat{x}_{t-1/t-1} + A_t(\hat{x}_{t/N} - \hat{x}_{t/t-1}) \\
&\Sigma_{t-1/N} = \Sigma_{t-1/t-1} + A_t(\Sigma_{t/N} - \Sigma_{t/t-1})A_t^T \\
&A_t = \Sigma_{t-1/t-1} F^T \Sigma_{t/t-1}^{-1} \\
&\Sigma_{t,t-1/N} = \Sigma_{t,t-1/t} + (\Sigma_{t/N} - \Sigma_{t/t})\Sigma_{t/t}^{-1}\Sigma_{t,t-1/t}
\end{aligned}
\tag{4}
$$

A synthetic LDM model with two-dimensional states and one-dimensional observations was created to demonstrate the contribution of RTS smoothing. In Figure 2 we show the state predictions of this LDM model using traditional Kalman filter. In Figure 3, the performance of the Kalman filter with RTS smoothing is shown. In both figures, the green lines represent the trajectories of the two-dimensional true state evolution for our synthetic LDM model. The blue points are the scatter plot of the noisy observations of the LDM model.

We can see the predicted results roughly simulate the true state evolution. After adding RTS smoothing into the Kalman filtering process, we observe significantly better prediction for the system internal states.

The Expectation-maximization (EM) algorithm [7] is used to find the maximum likelihood estimates of parameters for a specific word or phone, where the model depends on unobserved latent variables. The relevant equations are:

$$
\begin{aligned}
&E\left[x_t \middle/ y, \theta^{(i)}\right] = \hat{x}_{t/N} \\
&E\left[x_t x_t^T \middle/ y, \theta^{(i)} g\right] = \Sigma_{t/N} + \hat{x}_{t/N}\hat{x}_{t/N}^T \\
&E\left[x_t x_{t-1}^T \middle/ y, \theta^{(i)}\right] = \Sigma_{t,t-1/N} + \hat{x}_{t/N}\hat{x}_{t-1/N}^T
\end{aligned}
\tag{5}
$$

The E step algorithm consists of computing the conditional expectations of the complete-data sufficient statistics for standard ML parameter estimation. Therefore, the E step involves computing the expectations conditioned on observations and model parameters. The RTS smoother described previously can be used to compute the complete-data estimates of the state statistics. EM for LDM then

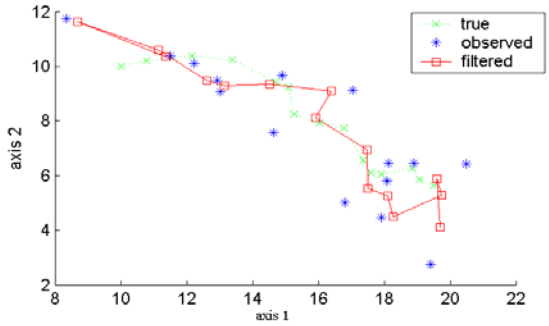

Figure 2: *A Kalman Filter*

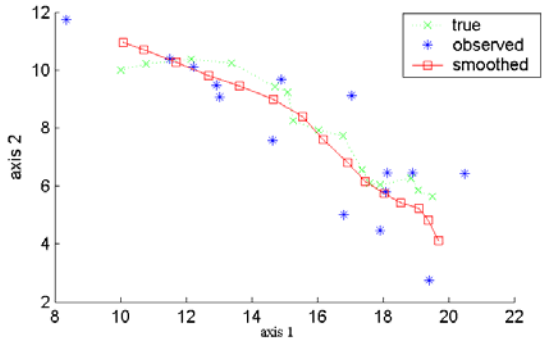

Figure 3: *A Kalman filter with an RTS smoother*

consists of evaluating the ML parameter estimates by replacing x_t and $x_t x_t^T$ with their expectations.

The EM algorithm converges quickly and is stable for our synthetic LDM model of two-dimensional states and one-dimensional observations. After initilizing this LDM model with an identity state transition matrix and random observation matrix, the first iteration of ML parameter estimation was applied to update the model parameters. Log-likelihood scores of observation vectors were calculated and saved in order to perform further analysis.

EM training was applied for 30 iterations. After the training recursion, intermediate log-likelihood scores of observation vectors for each iteration of LDM were plotted as a funtion of the number of iterations. This plot is is refered as the EM evolution curve. We explored 1-, 4-, 7-, and 10-dimensions for a state in the LDM approach, and applied EM training for each specified dimension. In Figure 4, the EM evolution curve is shown as a function of the state dimension.

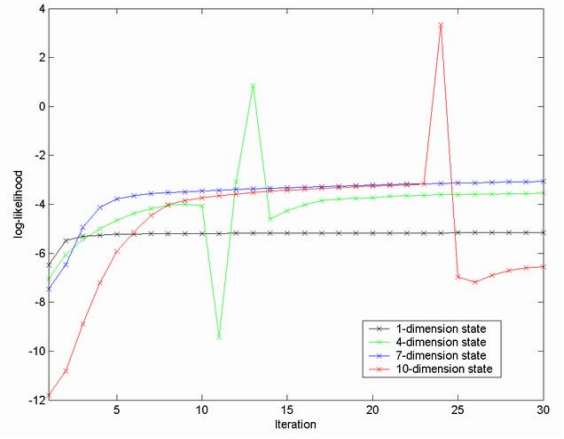

Figure 4: *EM evolution vs. state dimension*

One important practical issue about our EM implementation is that the linear transformation matrix F might lead the ML parameter estimation to produce erroneous parameters when $|F| > 1$. The reason for this is that the LDM state evolution would grow exponentially if the matrix F is not a decaying transformation [4]. Such behavior may not be apparent over a small numbers of frames, but it appears quite often when the training dataset gets large, especially in the situation where the state is not reset between models.

In this case, the most common solution is to use Singular Value Decomposition (SVD) to force $|F| < 1$ after each iteration of EM training. SVD provides a pair of orthonormal bases U and V, and a diagonal matrix of singular values S such that:

$$F = USV^T. \tag{6}$$

Every element of S greater than $1\text{-}\varepsilon$ will be replaced by $1\text{-}\varepsilon$ for a small number of ε (usually $\varepsilon = 0.005$). By adding the SVD component, we attain good model stability for LDM training, as was described in [2].

For a given speech segment, the likelihood that this segment was generated from a specific LDM can be calculated from Kalman filter equations. For a standard Kalman Filter, the state estimation error at time t can be represented as:

$$\begin{aligned} e_t &= y_t - \hat{y}_t \\ &= y_t - H\hat{y}_{t|t-1}. \end{aligned} \tag{7}$$

After replacing y_t with the observation equation, the error term becomes:

$$\begin{aligned} y_t - H\hat{x}_{t|t-1} &= H(x_t - \hat{x}_{t|t-1}) + v_t \\ e_t &= H(x_t - \hat{x}_{t|t-1}) + v_t \end{aligned}. \tag{8}$$

The associated covariance is:

$$\begin{aligned} \Sigma e_t &= E[e_t e_t^T] \\ &= H \Sigma_{t|t-1} H^T + R. \end{aligned} \tag{9}$$

Since errors are assumed uncorrelated and Gaussian, the log-likelihood of an N-length observation sequence y_t given the model parameters can be calculated as:

$$\begin{aligned} \log(p_1^N \mid \theta) = -\frac{1}{2}\sum_1^N \{\log \mid \Sigma e_t \mid + e_t^T \Sigma^{-1} e_t\} \\ - \frac{Np}{2}\log(2\pi) \end{aligned} \tag{10}$$

where e_t and Σe_t are computed as part of the standard Kalman filter recursions. In classification applications, the latter normalization term can be omitted because it is constant [2].

Some researchers report that the state's contribution to the error covariance Σe_t is detrimental to classification performance [2]. During EM training, the resulting fluctuations in the likelihoods computed during the segment-initial frames have the most effect on the overall likelihood of shorter phone segments. For shorter speech segments, it is recommended to replace the error covariance calculation

$$\Sigma e_t = H \Sigma_{t|t-1} H^T + R. \tag{11}$$

with

$$\Sigma e_t' = R. \tag{12}$$

However, our experimental results did not show a performance improvement for shorter speech segments by using this approach. Hence, in the following experiments, the LDM implementations used the traditional error covariance form.

3. Pilot Classification Experiments

Since LDM has proven to be effective on simulated data, a logical next step was to apply it to the classification of phonetic segments in speech. Our first experiment involved evaluating LDM as a classifier on a simple database consisting of a few phones clearly articulated by a small group of speakers. This data was used to gain a better understanding of key algorithm parameters and their impact on convergence. We refer to this data as the sustained phones database.

The sustained phone database is composed of 2 speakers with 3 phones recorded for each speaker. Each speaker produced 0.5 second utterances of the following phonemes: one vowel 'aa', one nasal 'm' and one fricative 'sh' at a sampling rate of 16 kHz. Feature vectors were generated by computing 12 mel-scaled cepstral coefficients and absolute energy. A frame duration of 10 milliseconds and a window duration of 25 milliseconds was used for feature extraction. The training set consisted of 210 examples (70% of the sustained phone database) of 3 phones from two speakers and the test set consisted of 90 examples (30% of the sustained phone database).

After the data recording and feature extraction, we initialized 3 LDMs (phonemes 'aa', 'm', and "sh") using the following strategy: state transition matrix as identity matrix; observation matrix as random entries; observation noise covariance as identity matrix; state transition matrix as identity matrix multiplied by a factor 0.1. The EM algorithm was used for training. We observed that EM training converges after approximately 5 iterations. Different dimensionalities of the state-space were examined and we found 13 dimensions were adequate. Increasing the dimensionality of the state-space to 40 did not improve the classification accuracy in this case.

An HMM system with GMMs was built as the benchmark to evaluate LDM as a phoneme classifier. Table 1 summarizes the relative difference in classification accuracy between LDMs and HMMs. We see that the classification accuracy of the LDM system is 98.9%, which outperforms the best HMM baseline classification accuracy of 91.1% (8-mixture). In the next section, we will further assess LDMs on a large vocabulary evaluation corpus Aurora-4.

Table 1. *Classification (% accuracy) results for sustained phone database.*

model	**vowel *aa***	**nasal *m***	**fricative *sh***	**total**
HMM (2-mixt)	66.67	70.0	96.77	77.8
HMM (4-mixt)	90.0	70.0	100	86.7
HMM (8-mixt)	100	73.33	100	91.1
LDM	100	96.67	100	98.9

4. Aurora Experiments

Motivated by the encouraging results on the sustained phone classification experiment, we continued to evaluate LDMs on the Aurora-4 large vocabulary evaluation corpus [6]. This corpus is a well-established LVCSR benchmark that does not require extensive computational resources. The data was generated from a machine readable corpus of Wall Street Journal news text. The corpus is divided into a training set and an evaluation set. The training set consists of 7,138 utterances from 83 speakers totaling in 14 hours of speech. The evaluation set consists of 330 utterances from 8 speakers. All utterances were generated at 16 kHz.

The HMM system is used to generate alignments at the phone level. Each phone instance is treated as one segment. A total of 40 LDM phone models, one classifier per model, were used to cover the pronunciations. Each classifier was trained using the segmental features derived from 13-dimensional frame-level feature vectors comprised of 12 cepstral coefficients and absolute energy. The full training set has as many as 30k training examples per classifier. Each phone-level classifier is trained as a one-vs-all classifier. The classifiers are used to predict the probability of an acoustic segment.

Table 2 summarizes the results of the Aurora-4 phoneme classification experiments. The baseline system is composed of 3-state HMMs with varying numbers of mixtures. We show results only for 4-mixture GMMs since the performance increase for larger mixtures was only marginal. The HMM system achieves up to 46.9% and 36.8% accuracy for the clean evaluation data and noisy evaluation data respectively. For the noisy evaluation data, six different kinds of noise (Airport, Babble, Car, Restaurant, Street, and Train) were added randomly to better simulate the real world noisy environment.

From Table 2, we can see that the LDM classifiers achieve superior performance to the HMM classifiers with a classification accuracy of 49.2% for the clean evaluation data and 39.2% for the noisy evaluation data. This represents a 4.9% relative and a 6.5% relative increase in performance over a comparable HMM system with 3-state models. We claim that the LDM model generalizes better than HMM across different channel conditions, which makes LDM a noise robust speech recognition technique.

5. Conclusions and Future Work

In this paper, we proposed LDMs as a noise robust acoustic modeling technique for speech recognition. EM-based training algorithms and other related issues such as model initialization and dimensionality were investigated. We presented results on two tasks: a sustained phone classification experiment and a phone classification experiment on the Aurora-4 large vocabulary corpus. In these two experiments, LDMs outperformed baseline HMMs, particularly for the noisy evaluation data set. We believe that the state transition noise component and measurement noise component in LDM equations are the major contribution for the noise robust characteristic of LDMs. Theoretical verification is being investigated and will be reported later.

We are currently developing a HMM/LDM hybrid decoder architecture to model the frame correlation using LDMs as well as utilizing HMMs techniques for phone segment alignment. Preliminary experiments will be presented on the Alphadigits (AD) and Resource Management (RM) speech corpora. This HMM/LDM hybrid decoder architecture will be a good evaluation of LDMs on continuous speech recognition tasks, and can be compared to other hybrid decoders we have developed that utilize other nonlinear statistical models (e.g., support vector machines and relevance vector machines).

Table 2. *Classification (% accuracy) results for the Aurora-4 large vocabulary corpus (the relative improvements are shown in parentheses).*

model	**clean dataset**	**noisy dataset**
HMM (4-mixt)	46.9(-)	36.8(-)
LDM	49.2 (4.9%)	39.2 (6.5%)

6. Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. IIS-0414450. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

7. References

[1] Digalakis, V., "Segment-based Stochastic Models of Spectral Dynamics for Continuous Speech Recognition," *Ph.D. Dissertation*, Boston University, Boston, Massachusetts, USA, 1992.

[2] Frankel, J., "Linear Dynamic Models for Automatic Speech Recognition," *Ph.D. Dissertation*, The Centre for Speech Technology Research, University of Edinburgh, Edinburgh, UK, 2003.

[3] Digalakis, V., Rohlicek, J. and Ostendorf, M., "ML Estimation of a Stochastic Linear System with the EM Algorithm and Its Application to Speech Recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 1, no. 4, pp. 431–442, October 1993.

[4] Frankel, J. and King, S., "Speech Recognition Using Linear Dynamic Models," *IEEE Transactions on Speech and Audio Processing*, vol. 15, no. 1, pp. 246–256, January 2007.

[5] Tsontzos, G., Diakoloukas, V., Koniaris, C., and Digalakis, V., "Estimation of General Identifiable Linear Dynamic Models with an Application in Speech Recognition," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. IV-453– IV-456, Honolulu, Hawaii, USA, April 2007.

[6] Parihar, N. and Picone, J., "An Analysis of the Aurora Large Vocabulary Evaluation," *Proceedings of the European Conference on Speech Communication and Technology*, pp. 337-340, Geneva, Switzerland, September 2003.

[7] Roweis, S. and Ghahramani, Z., "A Unifying Review of Linear Gaussian Models," *Neural Computation*, vol. 11, no. 2, February 1999.

[8] Ostendorf, M., Digalakis, V. and Kimball, O., "From HMMs to Segment Models: A Unified View of Stochastic Modeling for Speech Recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 5, pp. 360–378, September 1996.

[9] Picone, J., Pike, S., Regan, R., Kamm, T., Bridle, J., Deng, L., Ma, Z., Richards, H. and Schuster, M., "Initial Evaluation of Hidden Dynamic Models on Conversational Speech," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 109-112, Phoenix, Arizona, USA, May 1999.

[10] Rosti, A. and Gales, M., "Generalized Linear Gaussian Models," Cambridge University Engineering, Technical Report, CUED/F-INFENG/TR.420, 2001.

[11] Ghahramani, Z. and Hinton, G. E., "Parameter Estimation for Linear Dynamical Systems," Technical Report CRG-TR-96-2, University of Toronto, Toronto, Canada, 1996.