[[1]](#footnote-1)

Continuous Speech Recognition Using Linear Dynamic Models

T. Ma, S. Srinivasan, G. Lazarou and J. Picone

*Abstract*— We propose a hybrid framework to use Linear Dynamic Models (LDMs) within the framework of hidden Markov model (HMM) for large vocabulary continuous speech recognition tasks. In speech recognition, HMMs typically assume a diagonal covariance matrix where 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 is complementary to existing HMM systems. Using proposed hybrid framework, we demonstrate 12.78% relative WER reduction on the Aurora-4 clean evaluation set, and a 13.24% relative WER reduction on babble noisy condition.

*Index Terms*— Linear dynamic models, speech recognition, acoustic modeling, nonlinear statistical modeling

# INTRODUCTION

O

ver the past several decades, Hidden Markov Models (HMMs) have been the most popular approach for acoustic modeling in automatic speech recognition (ASR) applications. 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 probabilities map hidden states to observations. HMM modeling techniques have relied on a standard assumption that speech features are temporally uncorrelated. Recent theoretical and experimental studies [1]-[3] suggest that exploiting frame-to-frame correlations in a speech signal further improves the performance of ASR systems. This is typically accomplished by developing an acoustic model which includes higher order statistics or trajectories [4].

Linear Dynamic Models (LDMs) have generated significant interest in recent years [3][6] 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 using a measurement equation to link the internal states to the observables. An autoregressive model is used to capture the time evolution of states [3][5]. An LDM models every word or phoneme segment as a non-separable unit which incorporates the dynamic evolution of the hidden states. Digalakis, *et al.* [2] presented a maximum likelihood approach using LDMs and a parameter estimation approach based on the Expectation-Maximization (EM) algorithm. In subsequent work by Frankel and King [3], LDMs were applied to an acoustic modeling problem to characterize articulatory dynamics. However, in both cases promising results were demonstrated on a limited recognition task such as TIMIT [9].

In this paper, we developed a hybrid framework to integrate LDM into public domain ISIP prototype decoder [8] and demonstrate significant improvement on difficult evaluation task: the Aurora-4 large vocabulary speech corpus [7]. This task includes clean and noisy speech data as well as conditions simulating mismatched training conditions. We show that the proposed hybrid recognizer provides 12.78% relative WER reduction on the Aurora-4 clean evaluation set, and a 13.24% relative WER reduction on babble noisy condition.

The outline of this paper is as follows. Section II presents a brief review of LDM modeling. Section III describes a hybrid HMM/LDM recognizer architecture that effectively integrates these two powerful technologies. Continuous speech recognition results on the Aurora-4 corpus are presented in Section IV. The paper concludes with a discussion of ongoing research on directly integrating LDM into HMMs system at the frame-level of speech signals.

# Linear Dynamic Models

LDM is an example of a Markovian state-space model, and in some sense, can be regarded as analogous to an HMM since LDMs use hidden state modeling. With LDMs, systems are described as underlying states and observables combined by a measurement equation [5]. Every observable has a corresponding hidden internal state as illustrated in Figure 1.

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



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

|  |  |  |
| --- | --- | --- |
|  |  | 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, *yt*, and the sequence of underlying states, *xt*, 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. 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 [1][3]. The state and noise variables can be combined into one single Gaussian random variable. Based on Figure 1, conditional density functions for the states and output can be written as follows:

|  |  |  |
| --- | --- | --- |
|  |  |  |

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

|  |  |  |
| --- | --- | --- |
|  |  |  |

The system’s states are hidden. We need to estimate the hidden state evolution given an *N*-length observation sequence *yt* 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 [3]. 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:



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

|  |  |  |
| --- | --- | --- |
|  |  |  |

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 a traditional Kalman filter. In Figure 3, the performance of the Kalman filter with RTS smoothing is shown. In both figures, the true state evolutions for our synthetic LDM model are compared to a scatter plot of the noisy observations of the LDM model and the RTS smoothed data. RTS smoothing produces significantly better prediction for the system’s internal states.

The EM algorithm [2] is used to find maximum likelihood estimates of parameters for a specific word or phone, where the model depends on unobserved latent variables. The relevant equations are:

|  |  |  |
| --- | --- | --- |
|  |  |  |

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 consists of evaluating the ML parameter estimates by replacing *xt* and *xtxtT* with their expectations.



Figure 2: *A Kalman Filter*

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 function of the number of iterations. This plot is referred as the EM evolution curve. We explored 1-, 4-, 6-, and 10-dimensions for each 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. The training procedure converges quickly, requiring no more than ten iterations.

# HYBRID HMM/LDM RECOGNIZER ARCHITECTURE

One significant drawback of LDMs is that, they are inherently static classifiers — they are not capable of implicitly modeling temporal evolution of speech signals. Static classifiers are not designed to find the optimal phonetic boundaries for a continuous speech utterance. On the other hand, HMMs have the advantage of being able to handle dynamic data with certain assumptions about stationarity and independence. Motivated by the success of HMM/SVM LVCSR recognizer [10], a two-pass hybrid HMM/LDM recognizer is developed which effectively integrates these two powerful technologies. This two-pass system leverages the temporal modeling and N-best list generation capabilities of the traditional HMM architecture in a first pass analysis. In the second pass, candidate sentence hypotheses are re-ranked using a phone-based LDM model. Figure 5 illustrates the N-best list rescoring architecture of hybrid HMM/LDM recognizer.



Figure 4: *EM evolution vs. state dimension*

The hybrid speech recognizer involves generating N-best lists using the HMM system and then post processing these lists using the LDM classifiers. Therefore, N-best list generation is critical for the performance of the hybrid architecture explored in this research. The N-best lists can be generated using different approaches. For ISIP Prototype Decoder, the word graph is generated and converted to N-best list using a stack-based word graph to N-best list converter. Word lattices or word graphs are a condensed representation of the search space during Viterbi decoding. It works as an intermediate representation for a multi-pass speech recognition system. Typically, a word graph contains word labels, start and stop times, a language model score and an acoustic score. To convert the word graph to N-best list, a stack is initialized with the start node of the graph. A recursive procedure is then processed to grow partial paths according to the word graph and re-rank the stack to find the best partial path. During this procedure, beam pruning is applied to only maintain the K best partial paths in the stack. In the end, N-best partial paths will be back traced to output N-best sentence hypothesis.

After getting the N-best utterance hypotheses and corresponding segmentation, in the second pass LDM classifiers are used to estimate the LDM likelihood score. In this work, transformation-based score combination scheme is applied for simplicity. The LDM likelihood scores are first normalized (transformed) to a common domain with HMM scores and then combined together. Choice of the normalization scheme and combination weight is data-dependent and requires empirical evaluation. Other score fusion techniques such as classifier-based score fusion and density-based score fusion could be alternatives but have not been investigated yet.

# Aurora-4 Experiments

In order to evaluate the hybrid HMM/LDM recognizer, Wall Street Journal (WSJ0) derived Aurora-4 speech corpus was chosen to do large vocabulary speech recognition experiments. The Aurora-4 Corpus is derived directly from WSJ0 and consists of the original WSJ0 data with digitally-added noise [7]. Aurora-4 is divided into two training sets and 14 evaluation sets . Training Set 1 and Training Set 2 include the complete WSJ0 training set known as SI-84. In Training Set 2, a subset of the training utterances contains various digitally-added noise conditions including six common ambient noise conditions. The 14 evaluation sets are derived from data defined by the November 1992 NIST evaluation set. Each evaluation set consists of a different microphone or noise combination. In this work, we use only TS1 dataset for training the acoustic models. This set consists of 7,138 training utterances spoken by 83 speakers. All utterances were recorded with a Sennheiser HMD-414 close-talking microphone. The data comes from WSJ0, but has a P.341 filter applied to simulate the frequency characteristics of a 16 kHz sample rate. The set totals approximately 14 hours of speech data with an average utterance length of 7.6 seconds and an average of 18 words per utterance. There are a total of 128,294 words spoken with 8,914 of these being unique words. Due to limited computational facilities available for these experiments ,only seven of the 14 evaluation sets were used. These sets include the original noise-free data recorded with the Sennheiser microphone and six versions with different types of digitally-added environmental noise at random levels between 5 and 15 dB. The environments include an airport, random babble, a car, restaurant, street, and a train.



Figure 5: *Architecture of hybrid HMM/LDM recognizer.*

Traditional 39 dimensional MFCC acoustic features (12 cepstral coefficients, absolute energy, and first and second order derivatives) were computed from each of the signal frames within the phoneme segments. Before extraction, each feature dimension was normalized to the range [-1,1] to improve the convergence property. A total of 40 phonemes are used for acoustic modeling, so there are 40 LDM classifiers in the hybrid decoder.

The evaluation results for the clean dataset and six noisy evaluation sets are presented in Table 1. The recognition result of hybrid HMM/LDM decoder for the noise-free data are encouraging. It achieves 11.6% WER which represents 12.78% relative WER reduction comparing to HMM baseline. The performance for noise-free data varies and not as significant as clean speech data. The hybrid decoder achieves 13.24% relative WER reduction for Babble noise evaluation dataset. Marginal performance improvement is observed for airport, restaurant, street, and train noise conditions. However, it increases the recognition WER for car noise condition by 4.36%. Overall, hybrid HMM/LDM decoder shows promising result over the HMM baseline especially for clean speech and babble noise condition. It confirms LDM’s good ability to model speech dynamics which is complementary to traditional HMM.

# Conclusions and Future Work

In this paper, we proposed a hybrid framework to integrate LDMs within the framework of HMM for large vocabulary continuous speech recognition tasks. The theoretical foundation of linear dynamic model is discussed along with the EM-based model training algorithms. The hybrid decoder architecture is an off-line processing mechanism and is boot-strapped using a baseline HMM system. Different aspects have been addressed that are specific to the use of LDMs in a hybrid system: modifications to the HMM system; implementation of N-best list generation; development of an N-best rescoring paradigm using HMM and LDM score fushion. For Aurora-4 large vocabulary speech corpus, the proposed hybrid approach shows significant improvement over the HMM baseline especially for clean speech and babble noise condition.

In this work, LDM operates at the speech segment level with segment alignment information derived from first-pass HMM recognition. In future work, LDMs could be closely integrated into HMMs system at the frame-level of speech signals. For example, the HMM state sequence characteristics might can be modeled using LDM and the related acoustic score can be integrated into search path score directly in Viterbi search. With careful implementation, this approach would untie the N-best list rescoring limitation and further improve the performance.

References

1. Frankel, J., “Linear Dynamic Models for Automatic Speech Recognition,” *Ph.D. Dissertation*, The Centre for Speech Technology Research, University of Edinburgh, Edinburgh, UK, 2003.
2. 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.
3. 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.
4. Liang, F, “An effective Bayesian neural network classifier with a comparison study to support vector machine,” *Neural Computation*, vol. 15, no. 8, pp. 1959-1989, August 2003.
5. Digalakis, V., “Segment-based Stochastic Models of Spectral Dynamics for Continuous Speech Recognition,” Ph.D*. Dissertation*, Boston University, Boston, Massachusetts, USA, 1992.
6. 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.

Table 1: *Aurora4 experimental results of hybrid decoder for clean and noisy data.*

|  |  |  |  |
| --- | --- | --- | --- |
| WER (%) | HMM Baseline | Hybrid HMM/LDM | Relative Reduction |
| Clean | 13.3 | 11.6 | **12.78%** |
| Airport | 53.0 | 50.3 | 5.09% |
| Babble | 55.9 | 48.5 | **13.24%** |
| Car | 57.3 | 59.8 | -4.36% |
| Restaurant | 53.4 | 50.6 | 5.24% |
| Street | 61.5 | 59.4 | 3.41% |
| Train | 66.1 | 63.4 | 4.08% |

1. 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.
2. Huang, K. and Picone, J., “Internet-Accessible Speech Recognition Technology,” presented at *the IEEE Midwest Symposium on Circuits and Systems*, Tulsa, Oklahoma, USA, August 2002.
3. Lamel L., Kassel R., and Seneff S., “Speech database development: design and analysis of the acoustic-phonetic corpus.” in *Proc. Speech Recognition Workshop*, pp. 100–109, Palo Alto, CA, February 1986.
4. Ganapathiraju A., Hamaker J., and Picone, J., “Hybrid HMM/SVM Architectures for Speech Recognition,” *Proceedings of the Department of Defense Hub 5 Workshop*, College Park, Maryland, USA, May 2000.
1. Manuscript received November 8, 2010. 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.

T. Ma and S. Srinivasan are with Department of Electrical and Computer Engineering, Mississippi State University, Mississippi State, Mississippi, USA (e-mail: {tm334, ss754}@ece.msstate.edu).

J. Picone is with Department of Electrical and Computer Engineering, Temple University, Philadelphia, Pennsylvania, USA (e-mail: picone@temple.edu).

G. Lazarou is with New York City Transit Authority, New York, New York, USA (e-mail: glaz@ieee.org). [↑](#footnote-ref-1)