

# Continuous speech recognition using linear dynamic models

Tao Ma · Sundararajan Srinivasan · Georgios Lazarou · Joseph Picone

Received: 2 February 2013 / Accepted: 22 May 2013 / Published online: 6 June 2013  
© Springer Science+Business Media New York 2013

**Abstract** Hidden Markov models (HMMs) with Gaussian mixture distributions rely on an assumption that speech features are temporally uncorrelated, and often assume a diagonal covariance matrix where correlations between feature vectors for adjacent frames are ignored. A Linear Dynamic Model (LDM) is a Markovian state-space model that also relies on hidden state modeling, but explicitly models the evolution of these hidden states using an autoregressive process. An LDM is capable of modeling higher order statistics and can exploit correlations of features in an efficient and parsimonious manner. In this paper, we present a hybrid LDM/HMM decoder architecture that post-processes segmentations derived from the first pass of an

HMM-based recognition. This smoothed trajectory model is complementary to existing HMM systems. An Expectation-Maximization (EM) approach for parameter estimation is presented. We demonstrate a 13 % relative WER reduction on the Aurora-4 clean evaluation set, and a 13 % relative WER reduction on the babble noise condition.

**Keywords** Linear dynamic models · Nonlinear statistical modeling · Speech recognition · Acoustic modeling

## 1 Introduction

Over the past several decades, Hidden Markov Models (HMMs) that use Gaussian mixture models to model state observation distributions have been the most popular approach for acoustic modeling in automatic speech recognition (ASR) applications. We will refer to these as HMM/GMM. An HMM/GMM 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 (Frankel 2003; Frankel and King 2007; Digalakis et al. 1993) 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 that includes higher order statistics or parameter trajectories (Liang 2003).

Linear Dynamic Models (LDMs) have generated significant interest in recent years (Frankel 2003; Tsontzos et al. 2007) due to their ability to model higher order statistics. An LDM describes a linear dynamic system as underlying states

---

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.

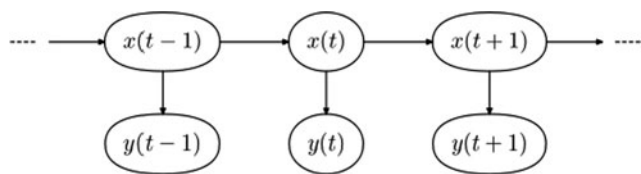
---

T. Ma  
Siri at Apple Inc, 2 Infinite Loop, mailstop 302-4APP, Cupertino,  
CA 95014, USA  
e-mail: [tma@apple.com](mailto:tma@apple.com)

S. Srinivasan  
Nuance Communications Inc., 1198 East Arques Avenue,  
Sunnyvale, CA 94085, USA  
e-mail: [sundararajan.srinivasan@gmail.com](mailto:sundararajan.srinivasan@gmail.com)

G. Lazarou  
The New York City Transit Authority, 30-74 38th Street, Apt 1A,  
Astoria, New York, NY 11103, USA  
e-mail: [glaz@ieee.org](mailto:glaz@ieee.org)

J. Picone (✉)  
Department of Electrical and Computer Engineering, Temple  
University, 1947 North 12th Street, Philadelphia, PA 19027, USA  
e-mail: [joseph.picone@isip.piconepress.com](mailto:joseph.picone@isip.piconepress.com)



**Fig. 1** The internal states and observations are shown for an LDM. Every observable has a corresponding hidden internal state

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. An LDM models every word or phoneme segment as a non-separable unit that incorporates the dynamic evolution of the hidden states. Digalakis et al. (1993) first applied LDMs to the speech recognition problem by developing a maximum likelihood approach based on an Expectation-Maximization (EM) parameter estimation algorithm. In subsequent work by Frankel and King (2007), LDMs were applied to an acoustic modeling problem to characterize articulatory dynamics. Promising results have been demonstrated on a limited recognition task based on TIMIT (Garofolo et al. 1993). More recently, LDMs have been applied to noisy speech recognition problems using Aurora-2 (Wöllmer et al. 2011), but not in a manner conducive to large vocabulary continuous speech recognition (LVCSR).

In this paper, we present a hybrid framework that integrates LDM into an LVCSR system, and demonstrate a significant improvement on a difficult evaluation task: the Aurora-4 Corpus (Parihar et al. 2004). This task includes clean and noisy speech data as well as conditions simulating mismatched training conditions. We show that the proposed hybrid recognizer provides 13 % relative WER reduction on the Aurora-4 clean evaluation set, and a 13 % relative WER reduction on a babble noise condition. In Sect. 2, we present a brief review of LDM. In Sect. 3, we describe a hybrid HMM/LDM recognizer architecture that effectively integrates these two technologies. Continuous speech recognition results on the Aurora-4 corpus are presented in Sect. 4. The paper concludes with a discussion of ongoing research on directly integrating LDM into HMMs system at the frame-level of speech signals.

## 2 Linear dynamic models

An LDM is an example of a Markovian state-space model, and in some sense, can be regarded as analogous to an HMM/GMM since LDMs use hidden-state modeling. In an LDM, systems are described as underlying states and observables combined by a measurement equation (Digalakis et al. 1993). Every observable has a corresponding hidden internal state as illustrated in Fig. 1. The LDM formulation

is based on a state-space model:

$$x_{t+1} = Fx_t + \omega_t \quad (1)$$

$$y_t = Hx_t + v_t, \quad (2)$$

where  $x_t$  is a  $q$ -dimensional internal state vector,  $y_t$  is a  $p$ -dimensional observation vector,  $F$  is the state evolution matrix and  $H$  is the observation transformation matrix. The variables  $\omega_t$  and  $v_t$  are assumed to be uncorrelated white Gaussian noise with covariance matrices  $Q$  and  $R$ , respectively.

The sequence of observations,  $y_t$ , and underlying states,  $x_t$ , are finite dimensional and are assumed to follow multivariate Gaussian distributions for every time  $t$ . The first equation can be viewed as an autoregressive state process that 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,  $x_t$ , are the deterministic characteristic of an LDM that are also affected by random Gaussian noise. The state and noise variables can be combined into one single Gaussian random variable (Frankel and King 2007).

Based on Fig. 1, conditional density functions for the states and output can be written as follows:

$$P(y_t|x_t) = \exp\left\{-\frac{1}{2}[y_t - Hx_t]^T R^{-1}[y_t - Hx_t]\right\} \times (2\pi)^{-p/2} |R|^{-1/2} \quad (3)$$

$$P(x_t|x_{t-1}) = \exp\left\{-\frac{1}{2}[x_t - Fx_{t-1}]^T \times Q^{-1}[x_t - Fx_{t-1}]\right\} (2\pi)^{-k/2} |Q|^{-1/2}. \quad (4)$$

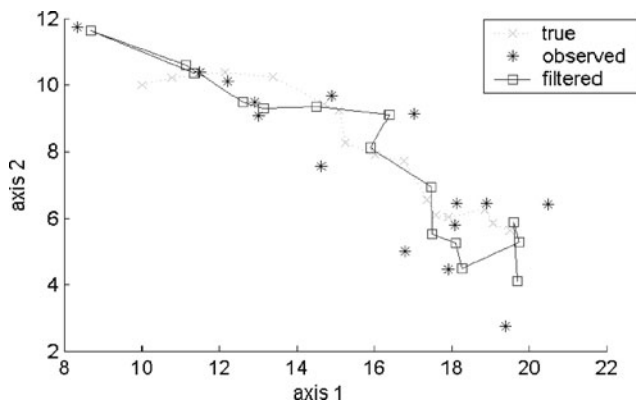
According to the Markovian assumption, the joint probability density function of the states and observations becomes:

$$P(\{x\}, \{y\}) = P(x_1) \prod_{t=2}^T P(x_t|x_{t-1}) \prod_{t=1}^T P(y_t|x_t). \quad (5)$$

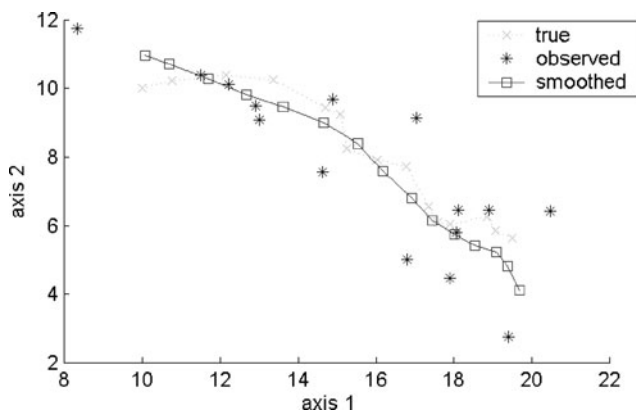
We need to estimate the hidden state evolution given  $y_t$  and the model parameters. This can be accomplished using a Kalman filter combined with a Rauch Tung Striebel (RTS) smoother (Frankel and King 2007). The Kalman filter provides an estimate of the state distribution at time  $t$  given the previous observations. 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. In addition, in both the forward and the backward pass, we need some additional recursions for the computation of the cross-covariance. The corresponding RTS equations are:

$$\hat{x}_{t-1/N} = \hat{x}_{t-1/t-1} + A_t(\hat{x}_{t/N} - \hat{x}_{t/t-1}) \quad (6)$$



**Fig. 2** State predictions for an LDM model using a Kalman filter are shown



**Fig. 3** A Kalman filter with RTS smoothing produces smoother state trajectories

$$\sum_{t-1/N} = \sum_{t-1/t-1} + A_t \left( \sum_{t/N} - \sum_{t/t-1} \right) A_t^T \quad (7)$$

$$A_t = \sum_{t-1/t-1} F^T \sum_{t/t-1}^{-1} \quad (8)$$

$$\sum_{t,t-1/N} = \sum_{t,t-1/t} + \left( \sum_{t/N} - \sum_{t/t} \right) \sum_{t/t}^{-1} \sum_{t,t-1/t} \quad (9)$$

A synthetic LDM model with two-dimensional states and one-dimensional observations was created to demonstrate the contribution of RTS smoothing. In Fig. 2, we show the state predictions of this LDM model using a traditional Kalman filter. In Fig. 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 Expectation-Maximization (EM) algorithm (Ding et al. 1993) is used to find maximum likelihood esti-

mates of parameters for a specific word or phone, where the model depends on unobserved latent variables. The relevant equations are:

$$E[x_t/y, \theta^{(i)}] = \hat{x}_{t/N} \quad (10)$$

$$E[x_t x_t^T / y, \theta^{(i)}] = \sum_{t/N} + \hat{x}_{t/N} \hat{x}_{t/N}^T \quad (11)$$

$$E[x_t x_{t-1}^T / y, \theta^{(i)}] = \sum_{t,t-1/N} + \hat{x}_{t/N} \hat{x}_{t-1/N}^T \quad (12)$$

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  $x_t$  and  $x_t x_t^T$  with their expectations.

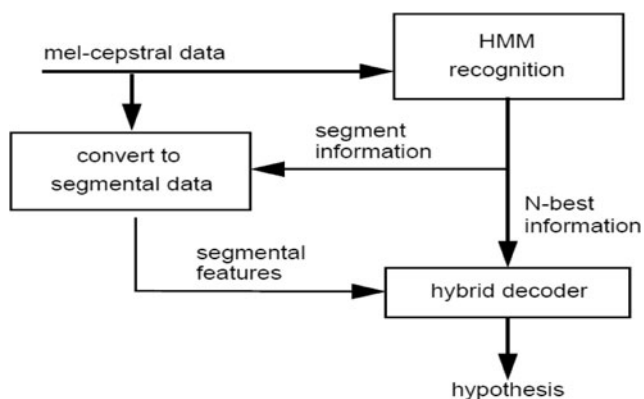
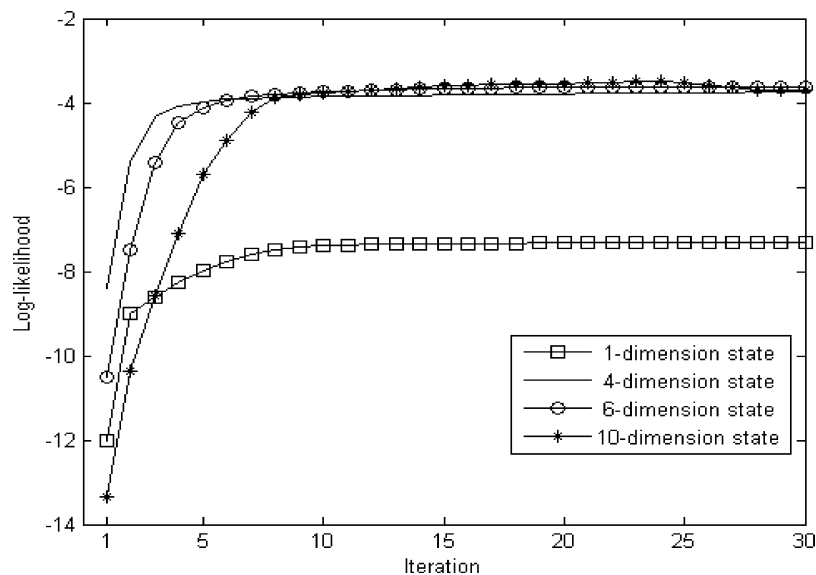
The EM algorithm converges quickly and is stable for our synthetic LDM model of two-dimensional states and one-dimensional observations. After initializing this LDM model with an identity state transition matrix and random observation matrix, the first iteration of ML parameter estimation was applied 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 all iterations of LDM were plotted as a function of the number of iterations. This plot is referred to 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 Fig. 4, the EM evolution curve is shown as a function of the state dimension. The training procedure converges quickly, requiring no more than 10 iterations.

### 3 A hybrid HMM/LDM architecture

One significant drawback of LDMs is that, they are inherently static classifiers—they are not capable of implicitly modeling the temporal evolution of a speech signal. Static classifiers are not designed to find the optimal start and stop times for a phone hypothesis. HMMs, on the other hand, are very good at optimizing segmentations while performing classification. Based on our previous work integrating a Support Vector Machine into a speech recognition system (Ganapathiraju et al. 2004), we employed a similar two-pass hybrid HMM/LDM recognizer. This system, shown in Fig. 5, leverages the temporal modeling and  $N$  best list generation capabilities of the traditional HMM architecture in a

**Fig. 4** The EM evolution as a function of iteration is shown for a variety of state dimensions. EM training procedure converges quickly, requiring no more than 10 iterations



**Fig. 5** A hybrid HMM/LDM architecture is shown in which LDM is used to postprocess phone hypotheses using HMM segmentations

firstpass analysis, and uses a second pass to re-rank candidate sentence hypotheses with a phone-based LDM model. A more thorough analysis of alternate strategies for integrating LDMs into an HMM framework was explored in Ma (2010). The hybrid system  $N$ -best rescoring approach was found to be the most promising.

Since the hybrid architecture postprocesses  $N$ -best lists, high performance  $N$ -best list generation is critical to achieving good performance. In our research, a word graph is generated and converted to an  $N$  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. Word graphs are an intermediate representation commonly used in 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 an  $N$ -best list, a stack is initialized with the start node of the graph. A recursive procedure is then used to grow partial paths according to the word graph and to

re-rank the stack to find the best partial path. During this procedure, beam pruning is applied to maintain the  $K$  best partial paths in the stack. Upon completion, the  $N$  best partial paths ( $N < K$ ) are traced to produce the final  $N$ -best sentence hypotheses.

Once this list is produced, along with the corresponding segmentations for the acoustic units, LDM classifiers are used in a second pass to estimate the likelihood scores. In this work, a transformation-based score combination scheme is applied for simplicity. The LDM likelihood scores are first normalized (transformed) to match the range of the HMM scores, and then a weighted combination of these two scores is used:

*Likelihood*

$$= HMM\_Score + LDM\_Scale * LDM\_Score \quad (13)$$

Ma (2010) explored methods of combining these two scores and determined that a weighted sum of the two scores provided a small gain in performance over using only the LDM score in the rescoring process. Choice of the normalization scheme and combination weight is data-dependent and requires empirical evaluation. Alternate approaches such as classifier-based score fusion and density-based score fusion could be used, but our experience was that the overall results are not sensitive to the type of score fusion used.

#### 4 Aurora-4 experiments

In order to evaluate the hybrid HMM/LDM recognizer, the Aurora-4 Corpus (Parihar et al. 2004) was chosen because it contains mismatched training and evaluation conditions, which is a fundamental problem addressed in this work.

**Table 1** Experimental tuning of the scale factor combining LDM and HMM scores resulted in small improvements in performance

Normalization Factor: LDM_SCALE	Hybrid Decoder WER
0.100	12.3 %
0.050	12.1 %
<b>0.010</b>	<b>11.8 %</b>
0.005	11.9 %
0.001	11.9 %

The Aurora-4 Corpus consists of the original WSJ0 data with digitally-added noise and is divided into two training sets and 14 evaluation sets. Training Set 1 (TS1) and Training Set 2 (TS2) include the complete WSJ0 training set known as SI-84. TS1 consists of the original WSJ recordings, while TS2 contains various digitally-added 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 and use the 14 evaluation sets for performance analysis.

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 behavior of our LDM training. A total of 40 phonemes are used for acoustic modeling, so there are 40 LDM classifiers in the hybrid decoder.

The scale factor for combining HMM and LDM scores is shown in Table 1. We see that performance varies slightly with changes in LDM\_Scale. It is not surprising that the variation is small, since the HMM-based segmentations play an important role in the overall combination of the two scores. Segmentation is an important part of the recognition process, and generally if segmentation is correct, recognition performance is high. The  $N$ -best rescoring process is intimately dependent on the HMM process. A better alternative would be to embed LDM in the first pass of the recognition system, but that is the subject of future research.

The evaluation results for the clean dataset and six noisy evaluation sets are presented in Table 2. The results for the hybrid HMM/LDM decoder for the condition labeled “Clean,” which represents matched training and testing in a noise-free environment, are encouraging. The hybrid HMM/LDM system achieves an 11.6 % WER which represents a 12.8 % relative WER reduction compared to a comparably configured HMM baseline. The hybrid decoder also achieves 13.2 % relative WER reduction for the babble noise evaluation dataset, and smaller improvements for a majority of the other conditions, which represent mismatched training and evaluation conditions. The overall results are

**Table 2** Experimental results for the hybrid HMM/LDM system are compared to a conventional HMM system. Substantial improvements were obtained on the clean and babble noise conditions

Condition	HMM Baseline	Hybrid LDM	Relative Reduction
Clean	13.3	11.6	<b>12.8 %</b>
Airport	53.0	50.3	5.09 %
Babble	55.9	48.5	<b>13.2 %</b>
Car	57.3	59.8	−4.4 %
Restaurant	53.4	50.6	5.2 %
Street	61.5	59.4	3.4 %
Train	66.1	63.4	4.1 %

promising given that the segmentations have not been optimized for the LDM system and confirms LDM’s capability to model speech dynamics in a manner that is complementary to a traditional HMM.

## 5 Summary

In this paper, we proposed a hybrid framework to integrate LDMs within the framework of an HMM for large vocabulary continuous speech recognition tasks. The theoretical foundation of the linear dynamic model is discussed and an EM-based training paradigm is introduced. The hybrid decoder architecture is an off-line processing mechanism and is bootstrapped using a baseline HMM system. Several issues related to applying an LDM in a hybrid system have been addressed: modifications to the HMM system; implementation of the  $N$ -best list generation; and development of an  $N$ -best rescoring paradigm using HMM and LDM score fusion. Results on the Aurora-4 Corpus are encouraging.

In this work, the LDM postprocesses segmentations derived from the first pass of an HMM-based recognition. It is well known that segmentation plays a major role in high performance speech recognition systems. Future work will be focused on closely integrating the LDM into the core search loop of a speech recognizer, providing acoustic scores at the frame level that can be directly integrated into the Viterbi search, alleviating the need to do  $N$ -best rescoring. This would allow a deeper analysis of the utterance and improve performance beyond that achievable with  $N$ -best rescoring and fixed segmentations.

## References

- Digalakis, V., Rohlicek, J., & Ostendorf, M. (1993). ML estimation of a stochastic linear system with the EM algorithm and its application to speech recognition. *IEEE Transactions on Speech and Audio Processing*, 1(4), 431–442.
- Frankel, J. (2003). Linear dynamic models for automatic speech recognition. Retrieved from [http://homepages.inf.ed.ac.uk/joe/pubs/2003/Frankel\\_thesis2003.pdf](http://homepages.inf.ed.ac.uk/joe/pubs/2003/Frankel_thesis2003.pdf).

- Frankel, J., & King, S. (2007). Speech recognition using linear dynamic models. *IEEE Transactions on Speech and Audio Processing*, 15(1), 246–256.
- Ganapathiraju, A., Hamaker, J., & Picone, J. (2004). Applications of support vector machines to speech recognition. *IEEE Transactions on Signal Processing*, 52(8), 2348–2355.
- Garofolo, J., Lamel, L., Fisher, W., Fiscus, J., Pallet, D., Dahlgren, N., & Zue, V. (1993). *TIMIT acoustic-phonetic continuous speech corpus. The linguistic data consortium catalog*. Philadelphia: The Linguistic Data Consortium. ISBN:1-58563-019-5.
- Liang, F. (2003). An effective Bayesian neural network classifier with a comparison study to support vector machine. *Neural Computation*, 15(8), 1959–1989.
- Ma, T. (2010). *Linear dynamic model for continuous speech recognition*. Starkville: Mississippi State University.
- Parihar, N., Picone, J., Pearce, D., & Hirsch, H.-G. (2004). Performance analysis of the Aurora large vocabulary baseline system. In *Proceedings of the European signal processing conference*, Vienna, Austria (pp. 553–556).
- Tsontzos, G., Diakouloukas, V., Koniaris, C., & Digalakis, V. (2007). Estimation of general identifiable linear dynamic models with an application in speech recognition. In *Proceedings of the IEEE international conference on acoustics, speech, and signal processing* (Vol. 4, pp. IV-453–IV-456).
- Wöllmer, M., Klebert, N., & Schuller, B. (2011). *Switching linear dynamic models for recognition of emotionally colored and noisy speech. Sprachkommunikation 2010. ITG-FB* (Vol. 225, pp. 1–4). Bochum: Springer.