Latent Variable and Predictive Models of Dynamical Systems

Sajid M. Siddiqi

Thesis Proposal

December 2007

Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Abstract

The modeling of discrete-time dynamical systems under uncertainty is an important endeavor, with applications in a wide range of fields. We propose to investigate new models and algorithms for inference, structure and parameter learning that extend our capabilities of modeling such systems, with a focus on models for real-valued sequential data. Our work is grounded in existing generative models for dynamical systems that are based on latent variable representations. *Hidden Markov Models* (HMMs) and *Linear Dynamical Systems* (LDSs) are popular choices for modeling dynamical systems because of their balance of simplicity and expressive power, and because of the existence of efficient inference and learning algorithms for these models.

Recently proposed *predictive* models of dynamical systems, such as linear *Predictive State Representations* (PSRs) and *Predictive Linear Gaussians* (PLGs) have been shown to be equally powerful as (and often more compact than) HMMs and LDSs. Instead of modeling state by a latent variable, however, predictive models model the state of the dynamical system by a set of statistics defined on *future observable events*. This dependence on observable quantities and avoidance of latent variables makes it easier to learn consistent parameter settings and avoid local minima in predictive models, though they have other problems such as a paucity of well-developed learning algorithms. However, one interesting class of algorithms for learning models such as LDSs and PSRs is based on factoring matrices containing statistics about observations using techniques such as the *singular value decomposition* (SVD). This class of algorithms is especially popular in the control theory literature, under the name *system identification*.

A restriction of most current models is that they are restricted to either discrete, Gaussian, or mixture-of-Gaussian observation distributions. Recently, Wingate and Singh (2007) proposed a predictive model that aims to generalize PLGs to exponential family distributions. This allows us to exploit structure in the observations using exponential family graphical models. It also exposes us to problems of intractable inference, structure and parameter learning inherent in conventional algorithms for graphical models, necessitating the use of approximate inference techniques.

Our goal is to formulate models and algorithms that unify disparate elements of this set of tools. The ultimate aim of this thesis is to devise predictive models that unify HMM-style and LDS-style models in a way that captures the advantages of both and generalize them to exponential families, and to investigate efficient and stable structure and parameter learning algorithms for these models based on matrix decomposition techniques.

Contents

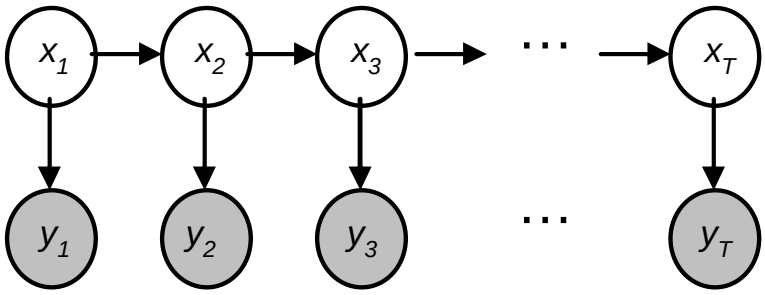
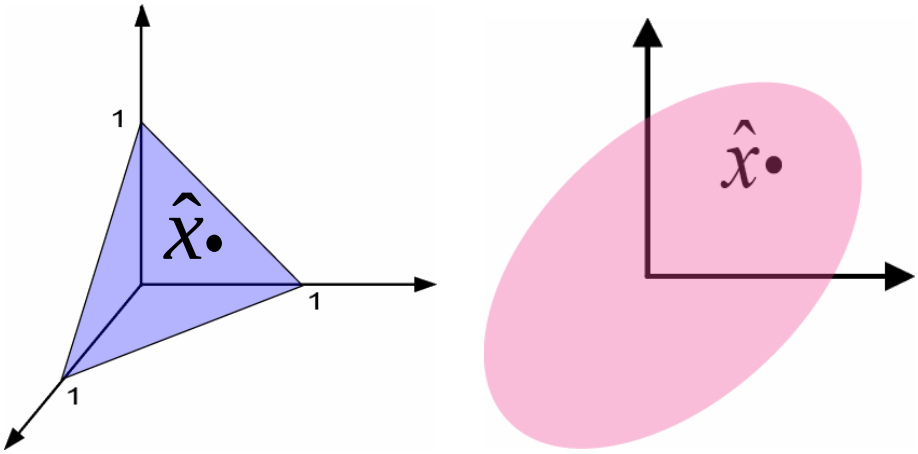


Figure 1: Directed graphical model for HMMs and LDSs. Shaded nodes are observed, others hidden.

1 Introduction

The modeling of discrete-time dynamical systems under uncertainty is an important endeavor, with applications in a wide range of fields such as engineering, econometrics and financial modeling, stock market prediction, speech and language modeling, bioinformatics and others. We propose to investigate new models and algorithms for inference, structure and parameter learning that extend our capabilities of modeling such systems, with a focus on models for real-valued sequential data. Our work is grounded in existing generative models for dynamical systems that are based on latent variable representations of the observed data, where the distribution of observations is assumed to depend on an underlying latent variable that evolves stochastically over time. This allows the model to represent the *state* of the system, effectively endowing it with infinite memory by allowing observations to have an effect arbitrarily far into the future via the latent state. *Hidden Markov Models* (HMMs) (defined in Section 2.1) and *Linear Dynamical Systems* (LDSs) (defined in Section 2.2) are popular choices for modeling dynamical systems because of their balance of simplicity and expressive power, and because of the existence of efficient inference and learning algorithms for these models. Both of them model the joint distribution of latent variables and observations by factoring it according to the directed graphical model shown in Figure 1. Different assumptions about the latent variable lead to either the HMM or the LDS, each with distinct characteristics, advantages and disadvantages. Roweis and Ghahramani (1999) reviews HMMs and LDSs in the larger context of *linear Gaussian models*. Relevant previous work done on these models is described in Sections 3.1 and 3.2.

Recently proposed *predictive* models of dynamical systems, such as linear *Predictive State Representations* (PSRs) and *Predictive Linear Gaussians* (PLGs) (defined in Section 2.3 and described in Section 3.5), have been shown to be equally powerful as (and often more compact than) HMMs and LDSs. Instead of modeling state by a latent variable, however, predictive models model the state of the dynamical system by a set of statistics defined on future observable events. This dependence on observable quantities and avoidance of latent variables makes it easier to learn consistent parameter settings and avoid local minima in predictive models, though they have other problems



(A) (B)

Figure 2: A. The belief state of an N-state HMM resides in the N-simplex (here N=3). B. The state vector of a stable N-dimensional LDS resides in some N-dimensional ellipsoid that its dynamics matrix maps to itself (here N = 2).

such as a paucity of well-developed learning algorithms. The state space where the PSR state vector (which is not a latent variable, but rather a vector of coefficients which yields the core test probabilities when combined with the PSR parameters) resides is also not well understood, unlike the HMM and LDS state spaces. PSRs generalize discrete-observation HMMs, and PLGs subsume the LDS model. Recently, Wingate and Singh (2007) proposed a predictive model that aims to generalize PLGs to exponential family distributions. Our goal is to formulate models and algorithms that unify disparate elements of this set of tools. The ultimate aim of this thesis is to devise predictive models that unify HMM-style and LDS-style models in a way that captures the advantages of both and generalize them to exponential families, and to investigate efficient inference, structure and stable parameter learning algorithms for these models (Section 5.3).

Though seemingly disparate beyond the graphical model they share, HMMs and LDSs are related very closely. The joint distribution of the latent variable and observation in both models is a member of the exponential family, with different choices of the link function. Choosing the N-dimensional generalization of the logit function leads to the multinomial distribution over discrete-valued latent and observable variables, as in HMMs. On the other hand, choosing the identity function leads to the Gaussian distribution over real-valued latent and observable variables as in LDSs. When performing inference in these models, the observation update and transition update can be viewed as being respectively *linear* in the natural parameters and expectation parameters of the corresponding distributions.

All other differences between these models stem from this difference in choice of link function. For an N-state HMM, since the latent variable distribution is multino-

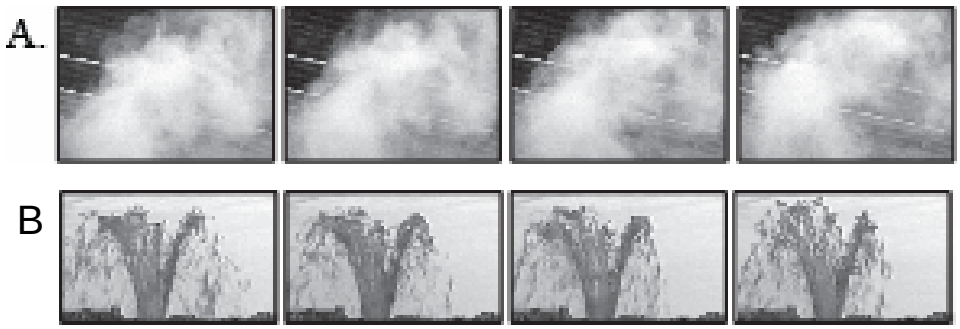


Figure 3: Frames from typical dynamic textures of steam rising (A) and a fountain flowing (B). Dynamic textures are videos that exhibit certain stationarity properties over time which can be captured by low-dimensional representations such as Linear Dynamical Systems.

mial, the belief state lies in the N-dimensional simplex (Figure 2(A)). The form of the *transition matrix* for HMMs must keep the belief state in the simplex in order to be stable (or even be a valid HMM), and hence is chosen to be a *stochastic* $N \times N$ matrix with each row encoding a probability distribution over states. Since the convex combination of two stochastic matrices is a stochastic matrix due to convexity of the simplex, the space of valid transition matrices is clearly convex. We have investigated HMM variants (Siddiqi & Moore, 2005) with additional constraints on the transition matrix (termed Dense-Mostly-Constant (DMC) transition matrices) that allow for more efficient inference and learning as well as larger state spaces while retaining much of the expressive power of unconstrained HMMs (Section 4.1). On the other hand, for stable LDSs, since the latent variable is Gaussian, the state vector lies in some N-dimensional ellipsoid containing the origin (Figure 2(B)) that gets mapped into itself, and hence the dynamics matrix must be a *stable* $N \times N$ matrix in the sense of its largest eigenvalue being at most unit magnitude. This ellipsoid is unknown *a priori*. This condition for a stable dynamics matrix is more difficult to enforce than the corresponding condition for valid transition matrices, since, though the space of matrices that map a particular ellipse into itself (and are hence stable) is convex, the space of *all* stable matrices, which is the union of all such convex spaces, is non-convex. We have proposed (Siddiqi et al., 2007a) a novel method for learning LDSs from data while enforcing stability in the dynamics matrix with application to modeling dynamic textures of video data (Section 4.3) of the kind in Figure 3.

The resulting difference in state spaces endows HMMs and LDSs with their relative advantages and disadvantages. Consider the state space of an LDS learned from data. The real-valued latent variable in an LDS allows smooth evolution of the state variable over the entire state space. This is often crucial, as in the case of dynamic texture modeling, where smooth variations in the generated video can only be modeled by smooth evolution in the latent variable. However, conventional LDSs can only model unimodal observation distributions. In particular, the log of the PDF must be concave

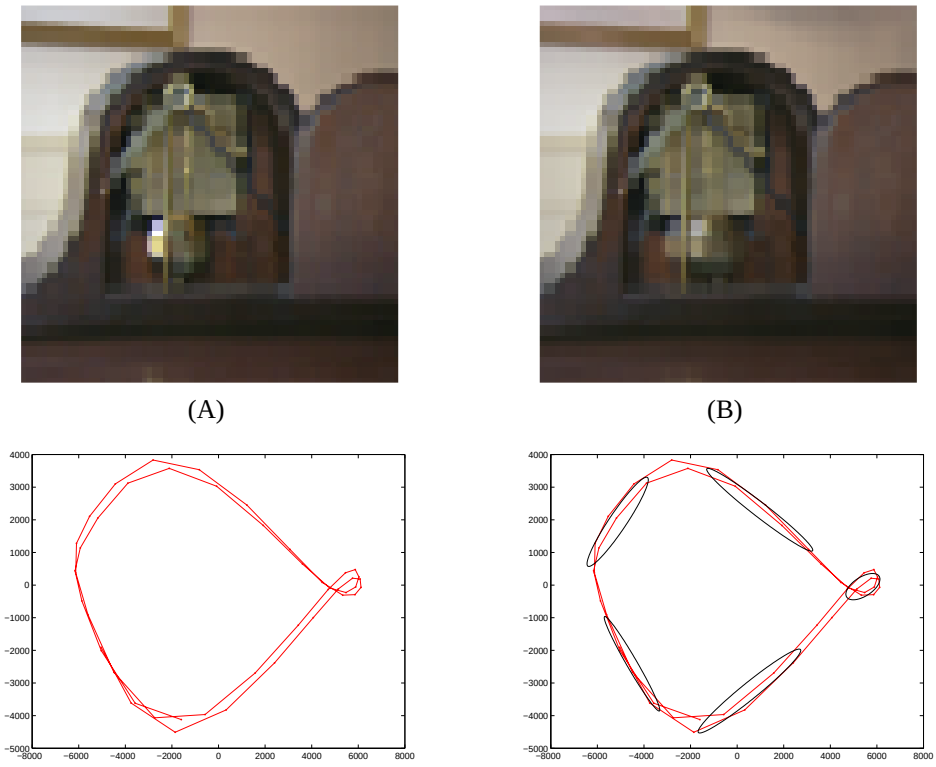


Figure 4: A. A frame from the clock pendulum video. B. A frame from the video resulting from simulating from a stable LDS which falls into invalid state configurations, resulting in multiple superimposed pendulum images. C. The state sequence for this dynamic texture clearly evolves in a non-convex state space. D. Unlike LDSs, HMMs trained on the state sequence can represent non-convex state spaces by tiling it with Gaussian densities, however they suffer from non-smooth evolution of the state variable which is essential to model smoothly varying observations such as those in dynamic texture videos.

as a function of the observations. For example, if an LDS predicts that image A is a likely observation at step t, and that image B is a likely observation at step t, then it must also predict that the image $(A + B)/2$ is a likely observation at step t. This is a limiting drawback, since many real systems do not have this property: for example, if A is an image of a clock with its pendulum in one position, and B is an image of a clock with its pendulum in another position, $(A + B)/2$ will be an inconsistent image with a half-intensity pendulum in both positions. Figure 4(B) shows an example of an invalid image produced by simulation from a conventional LDS, with the pendulum

blurred across multiple positions.

Given the LDS's linear observation model, the only way to produce a non-unimodal observation distribution is to start from a non-unimodal latent variable distribution, as shown in Figure 4(C). One way to model such a non-unimodal latent variable distribution is with a mixture of Gaussians, as shown in Figure 4(D). Unlike the original LDS, the mixture shown in Figure 4(D) excludes most of the latent states in the interior of the region bounded by the solid curve. (These latent states correspond to invalid images like the one in Figure 4(B).)

If we pick a fixed set of centers and covariances for the latent-state mixture distribution, and vary only the mixture weights, the resulting model is essentially an HMM: each discrete hidden state corresponds to a single ellipse. This HMM correctly predicts that the average of two likely images can be unlikely. Though conventional HMM learning algorithms are prone to local minima and are unlikely to discover the optimal state configuration even if a suitable value for N were known, there exist algorithms for choosing N while learning parameters and avoiding local minima. We devised a novel method for structure and parameter learning in Gaussian-based HMMs called STACS (Siddiqi et al., 2007b) that outperforms previous methods in efficiency and accuracy (Section 4.2). STACS simultaneously searches the space of structure parameters (i.e. the number of states) and model parameters in a greedy fashion using an EM-style inner loop to efficiently learn a model of appropriate dimensionality along with locally optimal parameter settings.

However, the disadvantage of HMMs is their inability to model smooth evolution of the latent variable, which can be essential for some dynamical systems. In the case of the clock pendulum video, HMMs would result in a model whose simulated videos contain valid pendulum images but abrupt, jerky transitions.

A natural question at this point, which this thesis aims to explore, is whether it is possible to formulate a model that can represent a smoothly evolving state variable like LDSs but can still handle non-Gaussian distributions over latent and observable variables like HMMs. This would effectively unify HMMs and LDSs and thus obviate the need for choosing between these models for the task of modeling dynamical systems.

Hybrid models unifying HMMs and LDSs have been investigated before. We briefly review this literature in Section 3.4. The final model we propose will differ from existing models in being easier to learn via decomposition-based methods (described below), and more general in allowing arbitrary exponential family distributions over the data.

A notable difficulty with learning latent variable models from data is local minima problems inherent in learning algorithms such as EM which are designed to maximize log-likelihood. In addition, there is the problem of model unidentifiability inherent in latent variable models. Finally, there is also the issue of model selection. An alternative set of methods for learning LDSs as well as PSR-style models is based on matrix decomposition using techniques such as the *Singular Value Decomposition (SVD)* on matrices containing the observations or statistics about them. The matrices are designed so as to be inherently low-rank in the noiseless case, with rank equal to dimensionality of the system. With enough data at hand, this allows us to simultaneously learn parameters and perform model selection by choosing the dimensionality of the model based on singular values of the matrix. The global convergence and lack

of local minima in SVD and related decomposition algorithms makes them attractive alternatives to EM-based algorithms for parameter learning. For LDSs this approach is termed *subspace identification* (reviewed in Section 3.3) and is well known in the controls and systems literature. The matrix chosen for decomposition is the *Hankel matrix* of stacked observations. For PSRs, the conventional model represents a set of probabilities over a large set of core tests as the sufficient statistics of the system. In fact, one conceptual representation of a PSR is the *System-Dynamics matrix* (Singh et al., 2004), a doubly-infinite matrix consisting of conditional probabilities $p(t_i \mid h_i)$ of tests t_i (futures) given histories h_i (pasts) for all possible histories (rows) and tests (columns). The rank of a dynamical system is shown to be equal to the rank of its corresponding system-dynamics matrix. Subspace-ID-style matrix decomposition can be used to learn low-dimensional PSRs (Rosencrantz & Gordon, 2004). However, the learning algorithms proposed for TPSRs are inefficient in their use of data, and do not attempt to learn stable parameters, both of which we will address (Section 5.1). A logical next step is to formulate a decomposition-based algorithm to learn low-dimensional PLGs (Section 5.2) as well as decomposition-based algorithms for learning the unified model we are proposing to develop (Section 5.4).

2 Notation and Definitions

Let $X = \{x_t\}_{t=1}^T$ denote the set of latent states over a sequence of T timesteps, and $Y = \{y_t\}_{t=1}^T$ be the set of real-valued observations. We assume a single training sequence for ease of presentation, though all algorithms described can be easily generalized to the multiple-sequence case.

2.1 Hidden Markov Models

Hidden Markov Models are a popular tool for modeling the statistical properties of observation sequences in domains where the observations can be assumed to be indicative of an underlying latent state sequence. In HMMs, the latent variables $\{x_t\}_{t=1}^T$ are discrete-valued, taking on one of N values where N is the number of states. An HMM is characterized by the following parameters:

1. N, the dimensionality of the latent variable.
2. M, the dimensionality of the real-valued observation.
3. $A = \{a_{ij}\}$, the $N \times N$ state transition matrix, where $a_{ij} = P(x_{t+1} = j \mid x_t = i)$. A is a *stochastic matrix*, with each of its rows constituting a probability distribution over the set of N states.
4. $B = \{b_j(a)\}_{a=1}^M$, the observation probability distribution in each state j, where $b_j(a) = P(y_t = a \mid y_t = j)$. B can be considered a $T \times N$ matrix. In practice, multinomials (for discrete-valued data), Gaussians and mixtures of Gaussians (for real-valued data) are commonly used as observation models.
5. $\pi = \langle \pi_i \rangle$, the initial state probability distribution, where $\pi_i = P(y_1 = i)$.

The complete set of parameters is indicated by $\lambda = (A, B, \pi)$.

The Viterbi algorithm (Viterbi, 1967) is a dynamic programming method for inferring the globally optimal path through state space given a sequence of observations. Parameter learning is most commonly carried out using Baum-Welch (Baum, 1972) which is an EM algorithm (Dempster et al., 1977) for finding a local optimum of the incomplete data likelihood.

2.2 Linear Dynamical Systems

Uncontrolled, discrete-time *Linear Dynamical Systems* can be described by the following two equations:

$$\begin{aligned} x_{t+1} &= Ax_t + w_t \\ y_{t+1} &= Cx_t + v_t \end{aligned} \tag{1}$$

Time is indexed by the discrete index t. Here x_t denotes the real-valued latent variable in $\mathbb{R}^n$, y_t the observations in $\mathbb{R}^m$, and w_t and v_t are zero-mean Normally distributed state and observation noise variables. Assume the initial state $x(0) = x_0$. The parameters of the system are the dynamics matrix $A \in \mathbb{R}^{n \times n}$, the observation model $C \in \mathbb{R}^{m \times n}$, and the noise covariance matrices Q and R. Note that we are learning *uncontrolled* linear dynamical systems though, as in previous work, control inputs can easily be incorporated into the objective function and convex program.

Let $\{\lambda_i(M)\}_{i=1}^M$ denote the eigenvalues of a square matrix M in decreasing order of magnitude, $\{\nu_i(M)\}_{i=1}^M$ the corresponding unit-length eigenvectors, and define its *spectral radius* $\rho(M) \equiv |\lambda_1(M)|$. An LDS with dynamics matrix A is *stable* if all of A's eigenvalues have magnitude at most 1, i.e. $\rho(A) \leq 1$.

Linear dynamical systems can also be viewed as probabilistic graphical models. The standard LDS filtering and smoothing inference algorithms (Kalman, 1960; Rauch, 1963) are instantiations of the junction tree algorithm for Bayesian Networks (see, for example, (Murphy, 2002)). Standard algorithms for learning LDS parameters learn locally optimal values by gradient descent (Ljung, 1999), Expectation Maximization (EM) (Ghahramani & Hinton, 1996) or least squares on a state sequence estimate obtained by subspace identification methods (Van Overschee & De Moor, 1996).

2.3 Predictive Models

Predictive State Representations (Littman et al., 2002; Singh et al., 2004) represent state in dynamical systems by tracking a set of predictions about future outcomes. Define a *test* to be a sequence of one or more observations in the future, and a *history* to be a sequence of observations since the beginning of time until the current timestep. N-dimensional PSRs for discrete-valued data define their state to be the probabilities of a set of N *core tests*. The core tests are chosen such that their probabilities constitute a *sufficient statistic* for the state of the system, in the sense that the probabilities of all possible tests are linear combinations of core test probabilities. The general PSR definition does not specify what kind of function of the sufficient statistic yields other test probabilities, and nonlinear PSRs have been examined to some extent (Rudary &

Singh, 2003). However, in *linear* PSRs, which most of the literature focuses on and which we will limit ourselves to, the probability of any test is a *linear* function of the core test probabilities at any timestep. We also restrict ourselves to the uncontrolled case, though controlled PSRs can be used for planning as an alternative to POMDPs. Tracking the core test belief in linear PSRs is an analogous operation to tracking the belief state in discrete HMMs; the fact that linear PSRs can provably model a larger class of dynamical systems than HMMs is because PSR model parameters need not be non-negative, unlike discrete HMM parameters (Singh et al., 2004). Jaeger (2000) presents an example of a 3-dimensional linear PSR that cannot be modeled by any finite HMM, in a model analogous to PSRs called the *Observable Operator Model* (OOM).

Predictive Linear Gaussians (Rudary et al., 2005) generalize the idea of PSRs to representing state using the *parameters of a distribution* over future outcomes. PLGs are predictive models of real-valued data that assume a joint Gaussian distribution over the next N observations, and assume this distribution to be a sufficient statistic for the distribution of all future observations. In addition, the $(N+1)^{st}$ future observation is assumed to be a noisy linear combination of the N before it. The state is tracked over time by *extending* to the $(N+1)^{st}$ future observation and *conditioning* on the 1^{st} future observation. PLGs provably subsume LDSs, are as compact as them and have fewer parameters. Like PSRs, PLGs also allow a consistent parameter estimation algorithm, though one which is not guaranteed to return valid parameters.

3 Background and Related Work

In this section we review past and current work from the literature that is related to the work proposed in this thesis, and puts it in context. The field is very large and it is impossible to mention all significant results in the space at hand; hence what follows is a selection of the most important and relevant lines of research. There are several survey and overview works on sequential data modeling using the models we discuss. Roweis and Ghahramani (1999) presents HMMs and LDSs as being variants of the same linear Gaussian model, along with several other models. Ghahramani (1998) derive HMMs and LDSs as different kinds of DBNs. A more detailed and in-depth survey of the DBN view of sequential data modeling can be found in Murphy (2002).

3.1 Hidden Markov Models

Introduced in the late 1960s, HMMs have been used most extensively in speech recognition (Rabiner, 1989; Bahl et al., 1983), language modeling and bioinformatics (Krogh et al., 1994) but also in diverse application areas such as computer vision (Sunderesan et al., 2003) and information extraction (Seymore et al., 1999). A classic tutorial on HMMs can be found in the work of Rabiner (1989). More recently, HMMs and their algorithms have been re-examined in light of their connections to Bayesian Networks, such as in Ghahramani (2001). Many variations on the basic HMM model have also been proposed, such as coupled HMMs (Brand et al., 1997) for modeling multiple interacting processes, Input-Output HMMs (Bengio & Frasconi, 1995) which incorporate inputs into the model, hierarchical HMMs (Fine et al., 1998) for modeling hierarchi-

cally structured state spaces, and factorial HMMs (Ghahramani & Jordan, 1995) that model the state space in a distributed fashion.

Felzenszwalb et al. (2003) recently proposed fast algorithms for a class of HMMs where the states can be embedded in an underlying parameter space and the transition probabilities can be expressed in terms of distances in this space. In comparison, our work on fast inference and learning in HMMs (Siddiqi & Moore, 2005) introduces a class of transition models that are applicable to arbitrary state spaces. Assuming a sparse transition matrix is another way to speed up these algorithms, and is an underlying assumption in many cases. This has two drawbacks. Firstly, this is an overly restrictive assumption for many problem domains. Secondly, it requires the sparse structure to be known or extracted in advance. Other approaches for fast HMM algorithms include the work by Murphy and Paskin (2002), which treats HMMs as a special kind of Dynamic Bayesian Network and proposes faster inference algorithms for hierarchical HMMs, as well as Salakhutdinov et al. (2003) which derives an alternative learning algorithm for HMM parameters under conditions when EM is slow.

There has been extensive work on HMM model selection. However, most of this work is either tailored to a specific application or is not scalable to learning topologies with more than a few states. The most successful approaches are greedy algorithms that are either bottom-up (i.e., starting with an overly large number of states) or top-down (i.e., starting with a small or single-state model). One disadvantage of bottom-up approaches is having to know an upper bound on the number of states beforehand. In some cases, one also faces the problem of deciding which of N^2 pairs of states to merge. We therefore favor top-down methods for HMM model selection, especially when the number of states may be large. Among these methods, Li and Biswas (1999) and Ostendorf and Singer (1997) are notable examples, which we compare to in our empirical evaluations. Another top-down algorithm is the entropic prior method of Brand (1999).

3.2 Linear Dynamical Systems

Linear dynamical systems are also known as *Kalman Filters* (Kalman, 1960) and *state-space models*. LDSs are an important tool for modeling time series in engineering, controls and economics as well as the physical and social sciences. Nonlinear dynamical systems and their learning algorithms have also been studied, such as the extended Kalman filter (Kopp & Orford, 1963; Cox, 1964) which linearizes the nonlinear system around the state estimate at every step, allowing the approximate state distribution to remain Gaussian. Ghahramani and Roweis (1999) presents an EM algorithm for learning nonlinear dynamical systems. However, we restrict our attention to linear systems.

Learning the dimensionality and parameters of LDSs is known as *linear system identification*, and is a well-studied subject (Ljung, 1999). Within this area, *subspace identification methods* (Van Overschee & De Moor, 1996) have been very successful. These techniques first estimate the model dimensionality and the underlying state sequence, and then derive parameter estimates using least squares. Within subspace methods, techniques were developed to enforce stability by augmenting the extended observability matrix with zeros (Chui & Maciejowski, 1996) or adding a regularization term to the least squares objective (Van Gestel et al., 2001).

All previous methods were outperformed by Lacy and Bernstein (2002), henceforth referred to as LB-1. They formulate the problem as a semidefinite program (SDP) whose objective minimizes the state sequence reconstruction error, and whose constraint bounds the largest singular value by 1. This convex constraint is obtained by rewriting the nonlinear matrix inequality $I_n - AA^T \succeq 0$ as a linear matrix inequality[1], where I_n is the $n \times n$ identity matrix. Here, $\succ$ ($\succeq$) denotes positive (semi-) definiteness. The existence of this constraint also proves the convexity of the $\sigma_1 \leq 1$ region. This condition is *sufficient* but not *necessary*, since a matrix that violates this condition may still be stable.

A follow-up to this work by the same authors (Lacy & Bernstein, 2003), which we will call LB-2, attempts to overcome the conservativeness of LB-1 by approximating the Lyapunov inequalities[2] $P - APA^T \succ 0$, $P \succ 0$ with the inequalities $P - APA^T - \delta I_n \succeq 0$, $P - \delta I_n \succeq 0$, $\delta > 0$. However, the approximation is achieved only at the cost of inducing a nonlinear distortion of the objective function by a problem-dependent reweighting matrix involving P, which is a variable to be optimized. In our experience, this causes LB-2 to perform worse than LB-1 (for any δ) in terms of the state sequence reconstruction error, even while obtaining solutions outside the feasible region of LB-1.

3.3 Subspace Identification

Subspace methods calculate the LDS parameters by first decomposing a matrix of observations to yield an estimate of the underlying state sequence. The most straightforward such technique is used here, see Van Overschee and De Moor (1996) for variations.

Note that the model we wish to learn is *unidentifiable*, i.e., several possible sets of parameters and initial conditions can give rise to a particular observation sequence. For any matrices A, C, Q that represent a model, an infinite number of equivalent models can be obtained by substituting A with SAS^{-1}, C with CS^{-1}, Q with SQS^T, and x_0 with Sx_0, for any invertible $n \times n$ matrix S. To identify a unique model (up to permutation of rows and sign changes) from a sequence $\{y_t\}$, we choose the *canonical model*, which has a C with orthonormal columns and an asymptotically diagonal state covariance.

Let $Y_{1:\tau} = [y_1\ y_2\ \ldots\ y_\tau] \in \mathbb{R}^{m \times \tau}$ and $X_{1:\tau} = [x_1\ x_2\ \ldots\ x_\tau] \in \mathbb{R}^{n \times \tau}$. We use $\mathcal{D}$ to denote the matrix of observations which is the input to SVD. One typical choice for $\mathcal{D}$ is $\mathcal{D} = Y_{1:\tau}$; we will discuss others below. (For now we assume $m \gg n$; other choices for $\mathcal{D}$ allow us to relax this assumption.) SVD yields

$$\mathcal{D} \approx U \Sigma V^T \tag{2}$$

where $U \in \mathbb{R}^{m \times n}$ and $V \in \mathbb{R}^{\tau \times n}$ have orthonormal columns $\{u_i\}$ and $\{v_i\}$, and $\Sigma = \text{diag}\{\sigma_1, \ldots, \sigma_n\}$ contains the singular values. We assume the desired hidden state dimensionality n is determined using standard methods; for example, we could

[1] This bounds the top singular value by 1 since it implies $\forall x\, x^T(I_n - fAA^T)x \geq 0 \Rightarrow \forall x\, x^T AA^T x \leq x^T x \Rightarrow$ for $\nu = \nu_1(AA^T)$ and $\lambda = \lambda_1(AA^T)$, $\nu^T AA^T \nu \leq \nu^T \nu \Rightarrow \nu^T \lambda \nu \leq 1 \Rightarrow \sigma_1^2(A) \leq 1$ since $\nu^T \nu = 1$ and $\sigma_1^2(M) = \lambda_1(MM^T)$ for any square matrix M.

[2] These inequalities hold iff the spectral radius is less than 1. For a proof sketch, see (Horn & Johnson, 1985) pg. 410.

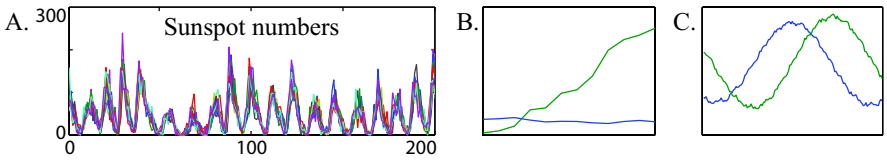


Figure 5: A. Sunspot data, sampled monthly for 200 years. Each curve is a month, the x-axis is over years. B. First two principal components of a 1-observation Hankel matrix. C. First two principal components of a 12-observation Hankel matrix, which better reflect temporal patterns in the data.

choose n by keeping all singular values of $\mathcal{D}$ above a certain threshold. In accordance with the canonical model assumptions above, we obtain estimates of C and X:

$$\begin{aligned} \hat{C} &= U \\ \hat{X} &= \Sigma V^T \end{aligned} \tag{3}$$

See (Soatto et al., 2001) for an explanation of why these estimates satisfy the canonical model assumptions. $\hat{X}$ is referred to as the *extended observability matrix* in the control systems literature; the t^{th} column of $\hat{X}$ represents an estimate of the hidden state of our LDS at time t. A least squares estimate of A is obtained by solving

$$\hat{A} = \arg\min_A \left\| AX_{0:\tau-1} - X_{1:\tau} \right\|_F^2 \tag{4}$$

where $\| \cdot \|_F$ denotes the Frobenius norm. Eq. (4) asks A to minimize the error in predicting the state at time $t+1$ from the state at time t. Given the above estimates $\hat{A}$ and $\hat{C}$, the covariance matrices $\hat{Q}$ and $\hat{R}$ can be estimated directly from residuals.

3.3.1 Hankel Matrices

In the decomposition above, we chose each column of $\mathcal{D}$ to be the observation vector for a single time step. Suppose that instead we set $\mathcal{D}$ to be a matrix of the form

$$\mathcal{D} = \begin{bmatrix} y_1 & y_2 & y_3 \\ y_2 & y_3 & y_4 \\ y_3 & y_4 & y_5 \end{bmatrix} \tag{5}$$

A matrix of this form is called a *block Hankel* matrix (Ljung, 1999). We say "d-observation Hankel matrix of size τ" to mean the data matrix $\mathcal{D} \in \mathbb{R}^{md \times \tau}$ with d length-m observations per column. Stacking observations causes each state to incorporate more information about the future, since $\hat{x}_t$ now represents coefficients reconstructing y_t *as well as* other observations in the future. However the observation model must now be $\hat{C} = U(:, 1:m)$, i.e., the submatrix consisting of the first m columns of U, because $U(:, 1:m)\hat{x}_t = \hat{y}_t$ for any t, where $\hat{y}_t$ denotes a reconstructed observation. Having multiple observations per column in $\mathcal{D}$ is particularly helpful when the underlying dynamical system is known to have periodicity. For example, see Figure 5.

3.4 Hybrid Models

Since shortly after the advent of LDSs, there have been attempts to combine the discrete transitions of HMMs with the linear dynamics of LDSs. We perform a brief review of the literature on hybrid models; see Ghahramani and Hinton (2000) for a more thorough review. Ackerson and Fu (1970) formulates a switching LDS variant where both the state and observation variable noise models are mixture of Gaussians with the mixture switching variable evolving according to Markovian dynamics, and derives the (intractable) optimal filtering equations where the number of Gaussians needed to represent the belief increases exponentially over time. They also propose an approximate filtering algorithm for this model based on a single Gaussian. Shumway and Stoffer (1993) propose learning algorithms for an LDS with switching observation matrices. Bar-Shalom and Li (1993) review models where both the observations and state variable switch according to a discrete variable with Markov transitions. *Hidden Filter HMMs* (HFHMMs) (Fraser & Dimitriadis, 1993) combine discrete and real-valued state variables and outputs that depend on both. The real-valued state is deterministically dependent on previous observations in a known manner, and only the discrete variable is hidden. This allows exact inference in this model to be tractable. Chen and Liu (2000) formulates the *Mixture Kalman Filter* (MKF) model along with a filtering algorithm, similar to Ackerson and Fu (1970) except that the filtering algorithm is based on sequential Monte-Carlo sampling.

Switching State-Space Models (SSSMs) (Ghahramani & Hinton, 2000) posit the existence of several real-valued hidden state variables that evolve linearly, with a single Markovian discrete-valued switching variable selecting the state which explains the real-valued observation at every timestep. Since exact inference and learning are intractable in this model, the authors derive a structured variational approximation that decouples the state space and switching variable chains, effectively resulting in Kalman smoothing on the state space variables and HMM forward-backward on the switching variable. In their experiments, the authors find SSSMs to perform better than regular LDSs on physiological data modeling task with multiple distinct underlying dynamical models. HMMs performed comparably well in terms of log-likelihood, indicating their ability to model nonlinear dynamics though the resulting model was less interpretable than the best SSSM.

3.5 Predictive Models

Several learning algorithms for PSRs have been proposed (Singh et al., 2003; James & Singh, 2004; Wolfe et al., 2005). It is easier for PSR learning algorithms to return *consistent* parameter estimates because the parameters are based on observable quantities. Rosencrantz and Gordon (2004) develops an SVD-based method for finding a low-dimensional variant of PSRs, called *Transformed PSRs* (TPSRs). Instead of tracking the probabilities of a small number of tests, TPSRs track a small number of linear combinations of a larger number of tests. This allows more compact representations, as well as dimensionality selection based on examining the singular values of the decomposed matrix, as in subspace identification methods. Note that nonlinearity can be encoded into the design of core tests. Rudary and Singh (2003) introduced the concept

of *e-tests* in PSRs that are indicator functions of aggregate sets of future outcomes, e.g. all sequence of observations in the immediate future that end with a particular observation after k timesteps. In general, tests in discrete PSRs can be indicator functions of arbitrary statistics of future events, thus encoding nonlinearities that might be essential for modeling some dynamical systems.

Kernelized versions of PLGs have been developed as well (Wingate & Singh, 2006a; Wingate & Singh, 2006b), which are analogous to nonlinear tests in discrete PSRs by tracking expected values of arbitrary nonlinear functions of future observations. Recently, Exponential Family PSRs (EFPSRs) (Wingate & Singh, 2007) were introduced as an attempt to generalize the PLG model to allow general exponential family distributions over the next N observations. In the EFPSR, state is represented by modeling the parameters of a time-varying exponential family distribution over the next N timesteps. This allows graphical structure to be encoded in the distribution, by choosing the parameters accordingly. The justification for choosing an exponential family comes from maximum entropy modeling. Though inference and parameter learning are difficult in graphical models of non-trivial structure, approximate inference methods can be utilized to make these problems tractable. Like PLGs, the dynamical component of EFPSRs is modeled by *extending* and *conditioning* the distribution over time. However, the method presented Wingate and Singh (2007) has some drawbacks, e.g. the extend-and-condition method is inconsistent with respect to marginals over individual timesteps between the extended and un-extended distributions.

4 Technical Contributions

In this section we describe work which has already been completed, resulting in algorithms for efficient inference, parameter and structure learning in latent variable models of dynamical systems of different kinds.

4.1 Fast Inference and Learning in Large State Space HMMs

For HMMs with fully connected transition models, the three fundamental problems of evaluating the likelihood of an observation sequence, estimating an optimal state sequence for the observations, and learning the model parameters, all have quadratic time complexity in the number of states. In Siddiqi and Moore (2005), we introduced a novel class of non-sparse Markov transition matrices called Dense-Mostly-Constant (DMC) transition matrices that allow us to derive new algorithms for solving the basic HMM problems in sub-quadratic time. We describe the DMC HMM model and algorithms and attempt to convey some intuition for their usage. Empirical results for these algorithms show dramatic speedups for all three problems. In terms of accuracy, the DMC model yields strong results and outperforms the baseline algorithms even in domains known to violate the DMC assumption.

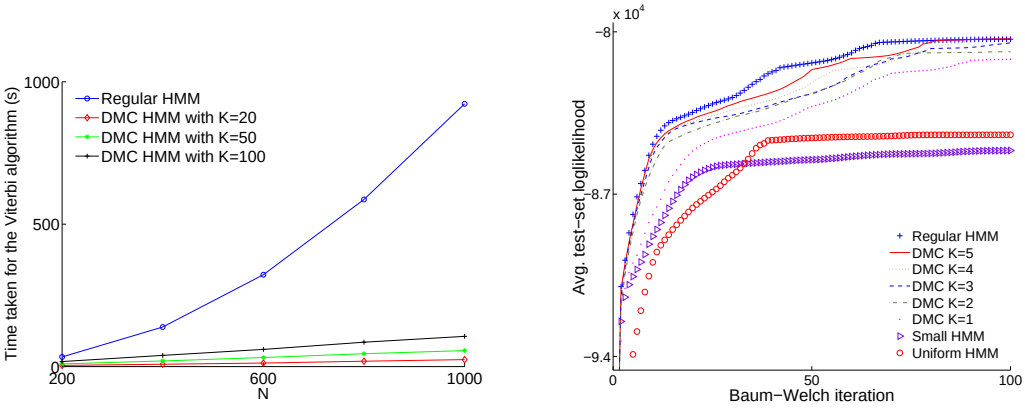


Figure 6: A. Running times (in seconds) for the Viterbi algorithm in regular and DMC HMMs. The synthetic data set being used here consists of 2000 rows of two-dimensional observations. B. Learning curves for 20-state DMC HMMs with different K values on the Motionlogger data set, compared to a 20-state regular HMM and two baseline models.

4.1.1 The Model

Specifically, a DMC transition matrix can be decomposed into the sum of a sparse matrix with K non-zero entries per row, and a rank-one matrix. For example, for a 3×3 transition matrix with $K = 1$, an example of a DMC transition matrix is:

$$\begin{pmatrix} \mathbf{0.5} & 0.25 & 0.25 \\ 0.2 & 0.2 & \mathbf{0.6} \\ 0.1 & \mathbf{0.8} & 0.1 \end{pmatrix} = \begin{pmatrix} 0.25 & 0 & 0 \\ 0 & 0 & 0.4 \\ 0 & 0.7 & 0 \end{pmatrix} + \begin{pmatrix} 0.25 & 0.25 & 0.25 \\ 0.2 & 0.2 & 0.2 \\ 0.1 & 0.1 & 0.1 \end{pmatrix}$$

The intuition behind DMC transition matrices is the assumption that there are only K transitions per state that are important enough to model exactly, and otherwise the state transition happens with some uniform probability. Another way of looking at it is as a mixture of a sparse transition model and a uniform one.

The DMC HMM model allows us to derive algorithms for likelihood evaluation, inference and learning (with fixed set of important transitions) that are sub-quadratic in the number of states, which is useful in scaling up to large state spaces. However, we would ideally like to update the set of important transitions at each iteration of EM as well. We describe an algorithm for simultaneous learning of DMC HMM structure (i.e. the set of important transitions) and parameters, which has a small quadratic component. The advantages of DMC HMMs over sparse transition matrices, which also allow sub-quadratic algorithms, are the greater expressiveness of allowing all possible transitions, and the ability to learn the set of important transitions from data. In contrast, HMMs with sparse transition matrices disallow certain transitions entirely, which can make the model brittle on noisy data. Furthermore, since transition parameters

initialized to zero during Baum-Welch will stay at zero, the set of non-zero transitions in such HMMs must be decided *a priori*. Importantly, the DMC assumption on the transition model does not imply any particular structure in the belief state; the model is still free to take on any valid belief from the N-dimensional simplex just as in regular HMMs.

4.1.2 Algorithms

Let $NC = \{NC_i\}_{i=1}^{N}$ be a collection of K-sized lists denoting the set of K important transitions for each state. Consider the Viterbi algorithm for computing the optimal path through state space. Given an observation sequence Y, the Viterbi algorithm is used to calculate the state sequence X that maximizes $P(X|Y, \lambda)$. Define $\delta_t(i)$ as

$$\delta_t(i) = \max_{x_1,\cdots,x_{t-1}} P[x_1 x_2 \cdots x_t = i, y_1 y_2 \cdots y_t | \lambda] \tag{6}$$

The inductive formula for $\delta_t(j)$ used in the Viterbi algorithm is

$$\delta_{t+1}(j) = [\max_i \delta_t(i) a_{ij}] b_j(y_{t+1}) \tag{7}$$

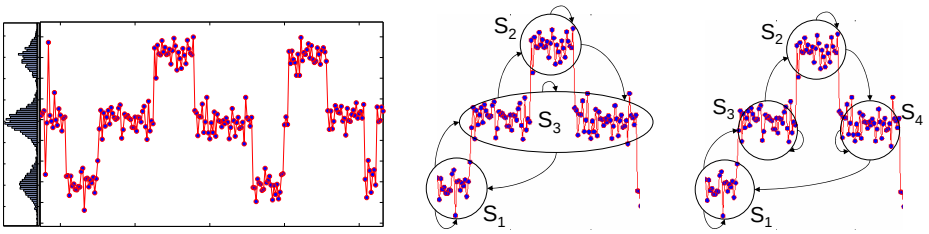
Since there is a maximization over N terms carried out for each state per timestep, and there are $N \times T$ $\delta_t(i)$ values to be calculated, the total running time of the Viterbi algorithm is $O(TN^2)$. Under the DMC condition, however, we can calculate this maximization more efficiently.

$$\begin{aligned} \delta_{t+1}(i) &= \big[\max_i \delta_t(i) a_{ij}\big] b_j(y_{t+1}) \\ &= \big[\max\big\{\max_i \delta_t(i) c_i, \max_{i:j \in NC_i} \delta_t(i) a_{ij}\big\}\big] b_j(y_{t+1}) \end{aligned} \tag{8}$$

We can split the $O(N)$ maximization into two terms: a maximization over N terms that is common to the entire timestep, and a maximization over an average of K terms per state. The first maximization adds an amortized cost of $O(1)$ over the N states, and the second one adds a $O(K)$ cost. Overall, this results in a time complexity of $O(TNK)$ for computing the optimal state sequence. An analogous trick with summations instead of maximization allows fast computation of the alpha and beta variables in the forward-backward algorithm. If we fix the set of important outgoing transitions over iterations of EM, parameter learning is strictly $O(TNK)$ as well.

4.1.3 Experimental Results

Figure 6 shows results comparing DMC HMMs to regular HMMs and baseline models. For any fixed value of K, the speedup factor for inference is significant as the number of states increases, as seen in Figure 6(A). The speedup for structure and parameter learning is also large but not as much as for inference, since some amount of quadratic-time overhead is spent in structure learning. Figure 6(B) plots mean test-set log-likelihood results over EM iterations from learning a 20-state HMM on the Motion-logger data set (see paper for details). We see that the DMC assumption does not hurt the results significantly in comparison to the baseline models of a uniform-transitions



(A) (B) (C)

Figure 7: A. A time series from a 4-state HMM. Observations from the two states *down-middle-up* and *up-middle-down* overlap and are indistinguishable without temporal information. B. The HMM topology learned by ML-SSS and Li-Biswas on the data in A. C. The correct HMM topology, successfully learned by the STACS algorithm.

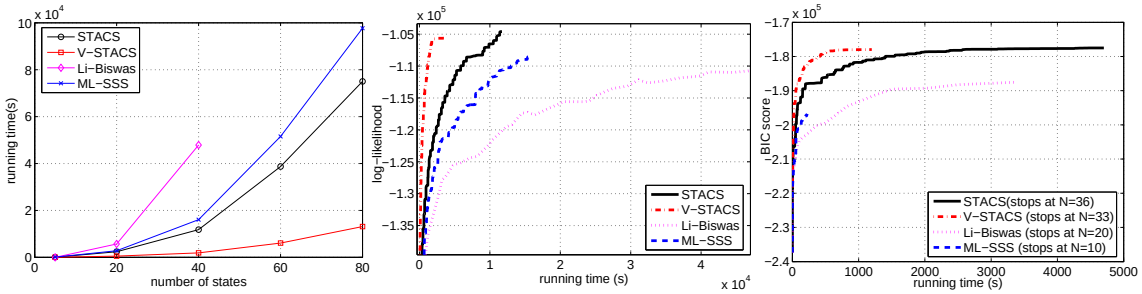
HMM (i.e. a mixture of Gaussians) and a regular 5-state HMM. Indeed, as we increase K from 1 upwards, we find that the results for $K = 4$ and $K = 5$ are equivalent to the unconstrained HMM, indicating that we've reached some 'intrinsic' DMC K-value for this data sequence.

4.2 Fast State Discovery for HMM Model Selection and Learning

Choosing the number of hidden states and their topology (model selection) and estimating model parameters (learning) are important problems for HMMs. In Siddiqi et al. (2007b), we presented a new state-splitting algorithm called Simultaneous Temporal and Contextual Splitting (STACS) that addresses both these problems. The algorithm models more information about the dynamic context of a state during a split, enabling it to discover underlying states more effectively than previous splitting methods, which fail on overlapping densities such as in Figure 7 where STACS succeeds. Compared to previous top-down methods, STACS also touches a smaller fraction of the data per split, leading to faster model search and selection. Because of its efficiency and ability to avoid local minima, the state-splitting approach is a good way to learn HMMs even if the desired number of states is known beforehand. We compare our approach to previous work on synthetic data as well as several real-world data sets from the literature, revealing significant improvements in efficiency and test-set likelihoods. We also compare to previous algorithms on a sign-language recognition task, with positive results. A more efficient, coarser variation called Viterbi STACS (V-STACS) is also developed, and is shown to perform better or nearly as well in most circumstances.

4.2.1 The Algorithm

The primary idea behind STACS is to trade off the amount of contextual information considered per split in return for increased modeling of temporal information. Here, contextual information is quantified by the number of data points considered during



(A) (B) (C)

Figure 8: A. Running time vs. number of final states on the ROBOT data set. B. Log-Likelihood vs. running time for learning a 40-state model on the ROBOT data. C. BIC score vs. running time on the MOCAP data when allowed to stop splitting autonomously. The plots are representative.

split design, and temporal information by the number of transition parameters optimized.

Specifically, we model the posterior belief at each timestep by a delta function at the Viterbi-optimal state. Each split is optimized over data points with non-zero belief in this approximated posterior, which number T/N on average. This is in contrast to ML-SSS where each split could consider $O(T)$ timesteps. Let T^s denote the set of timesteps owned by state s in the Viterbi path. We choose the split that locally optimizes the *partially observed likelihood* $P(Y, X^*_{\setminus T^s} \mid \lambda)$ at each model selection step, with all timesteps fixed at their optimal path states except those in T^s. Variants of Baum-Welch and Viterbi Training are developed for the task of optimizing the partially observed likelihood. Care is taken to ensure that the overall algorithm is asymptotically as efficient as conventional inference and learning, i.e. $O(TN^2)$.

4.2.2 Experimental Results

We evaluated STACS and V-STACS on a number of real-world data sets. Experiments show that our algorithms are more efficient and scalable in N than previous state-splitting methods (Figure 8(A)). They also learn better models with higher test-set likelihood scores when learning models of predetermined size (Figure 8(B)). When they're allowed to stop splitting autonomously, the improved split design mechanism of STACS and V-STACS leads them to find more good splits and thus larger final models that have better test-set likelihoods than those discovered by other methods (Figure 8(C)).

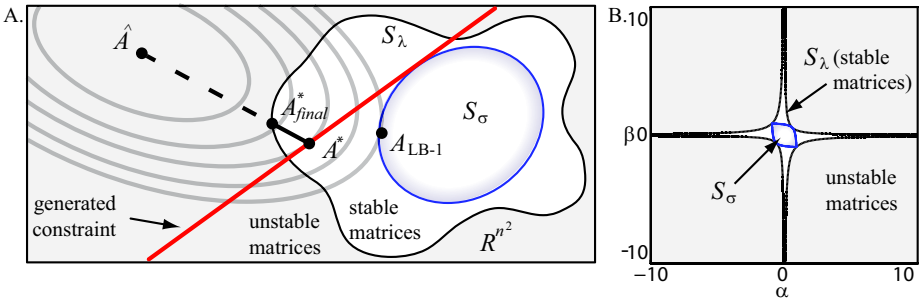


Figure 9: (A): Conceptual depiction of the space of $n \times n$ matrices. The region of stability (S_λ) is non-convex while the smaller region of matrices with $\sigma_1 \leq 1$ (S_σ) is convex. The elliptical contours indicate level sets of the quadratic objective function of the QP. $\hat{A}$ is the unconstrained least-squares solution to this objective. $A_{\text{LB-1}}$ is the solution found by LB-1 (Lacy & Bernstein, 2002). One iteration of constraint generation yields the constraint indicated by the red line, and (in this case) leads to a stable solution A^*. The final step of our algorithm improves on this solution by interpolating A^* with the previous solution (in this case, $\hat{A}$) to obtain A^*_{final}. (B): The actual stable and unstable regions for the space of 2×2 matrices $E_{\alpha,\beta} = [\, 0.3 \;\; \alpha \,;\, \beta \;\; 0.3 \,]$, with $\alpha, \beta \in [-10, 10]$. Constraint generation is able to learn a nearly optimal model from a noisy state sequence of length 7 simulated from $E_{0,10}$, with better state reconstruction error than either LB-1 or LB-2.

4.3 Learning Stable Linear Dynamical Systems with Constraint Generation

Stability is a desirable characteristic for LDSs, but it is often ignored by algorithms that learn these systems from data. We proposed (Siddiqi et al., 2007a) a novel method for learning stable LDSs: we formulate an approximation of the problem as a convex program, start with a solution to a relaxed version of the program, and incrementally add constraints to improve stability. Rather than continuing to generate constraints until we reach a feasible solution, we test stability at each step; because the convex program is only an approximation of the desired problem, this early stopping rule can yield a higher-quality solution. We apply our algorithm to the task of learning dynamic textures from image sequences as well as to modeling biosurveillance drug-sales data. The constraint generation approach leads to noticeable improvement in the quality of simulated sequences. We compare our method to those of Lacy and Bernstein (Lacy & Bernstein, 2002; Lacy & Bernstein, 2003), with positive results in terms of accuracy, quality of simulated sequences, and efficiency.

4.3.1 The Algorithm

An estimate of the underlying state sequence is first obtained using subspace identification. We then formulate the least-squares minimization problem for the dynamics

matrix as a quadratic program (QP) (Boyd & Vandenberghe, 2004), initially without constraints. When this QP is solved, the estimate $\hat{A}$ obtained may be unstable. However, any unstable solution allows us to derive a linear constraint which we then add to our original QP and re-solve. The above two steps are iterated until we reach a stable solution, which is then refined by a simple interpolation to obtain the best possible stable estimate.

Our method can be viewed as *constraint generation* (Horst & Pardalos, 1995) for an underlying convex program with a feasible set of all matrices with singular values at most 1, similar to work in control systems such as (Lacy & Bernstein, 2002). However, we terminate *before* reaching feasibility in the convex program, by checking for matrix stability after each new constraint. This makes our algorithm less conservative than previous methods for enforcing stability since it chooses the best of a larger set of stable dynamics matrices. The difference in the resulting stable systems is noticeable when simulating data. The constraint generation approach also implies much greater efficiency than previous methods in nearly all cases.

Figure 9(A) illustrates the space of dynamics matrices along with the quadratic objective, spaces of stable and $\sigma_1 \leq 1$ matrices, and linear constraints as well as the naive least-squares solution $\hat{A}$, the solution returned by our algorithm, and that found by previous methods. Figure 9(B) plots a 2-D slice view of the space of stable 2×2 matrices for a particular class of matrices, as an example of how highly non-convex the space of stable matrices can be, making it a difficult optimization problem.

4.3.2 Experimental Results

One application of LDSs in computer vision is learning *dynamic textures* from video data (Soatto et al., 2001). An advantage of learning dynamic textures is the ability to play back a realistic-looking generated sequence of desired duration. In practice, however, videos synthesized from dynamic texture models can quickly become degenerate because of instability in the underlying LDS. In contrast, sequences generated from dynamic textures learned by our method remain "sane" even after arbitrarily long durations. We also apply our algorithm to learning baseline dynamic models of over-the-counter (OTC) drug sales for biosurveillance, and sunspot numbers from the UCR archive (Keogh & Folias, 2002). Comparison to the best alternative methods (Lacy & Bernstein, 2002; Lacy & Bernstein, 2003) on these problems yields positive results, some of which are presented in Figure 10.

We examine daily counts of OTC drug sales in pharmacies, obtained from a biosurveillance laboratory. The counts are divided into 23 different categories and are tracked separately for each zipcode in the country. We focus on zipcodes from a particular American city. The data exhibits 7-day periodicity due to differential buying patterns during weekdays and weekends. We isolate a 60-day subsequence where the data dynamics remain relatively stationary, and attempt to learn LDS parameters to be able to simulate sequences of baseline values for use in detecting anomalies.

We perform two experiments on different aggregations of the OTC data, with parameter values $n = 7, d = 7$ and $\tau = 14$. Figure 4.3.2(A) plots 22 different drug categories aggregated over all zipcodes, and Figure 4.3.2(B) plots a single drug category (cough/cold) in 29 different zipcodes separately. In both cases, constraint generation is

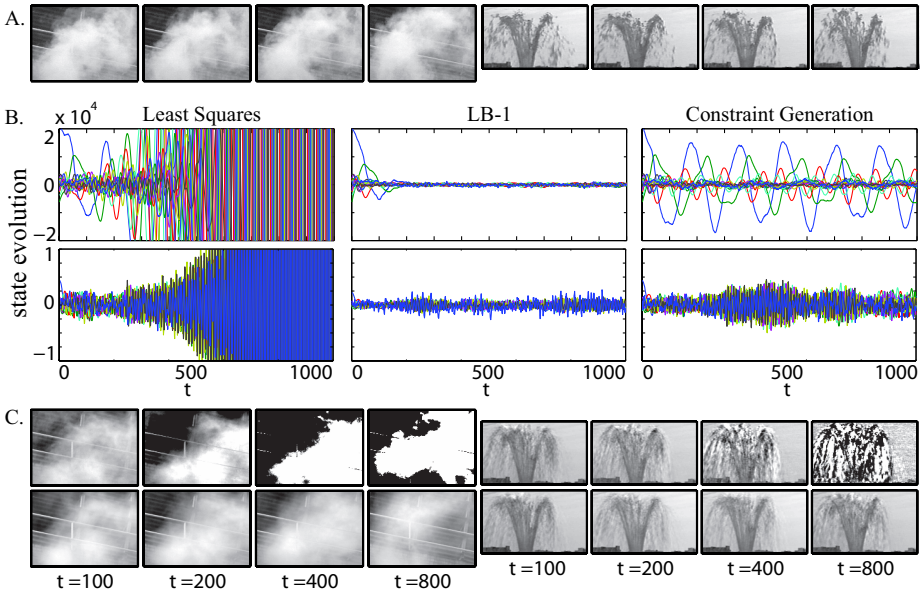


Figure 10: Dynamic textures. A. Samples from the original `steam` sequence and the `fountain` sequence. B. State evolution of synthesized sequences over 1000 frames (`steam` top, `fountain` bottom). The least squares solutions display instability as time progresses. The solutions obtained using LB-1 remain stable for the full 1000 frame image sequence. The constraint generation solutions, however, yield state sequences that are stable over the full 1000 frame image sequence without significant dampening. Finally, the constraint generation synthesized `steam` sequence is qualitatively better looking than the `steam` sequence generated by LB-1 although there is little qualitative difference between the two synthesized `fountain` sequences. C. Samples drawn from 1000 image least squares synthesized dynamic texture sequences (top), and samples from the constraint generation model drawn from a similarly synthesized 1000 image dynamic texture sequence (bottom).

able to use very little training data to learn a stable model that captures the periodicity in the data, while the least squares model is unstable and its observations diverge over time. LB-1 learns a model that is stable but overconstrained, and the simulated observations quickly drift from the correct magnitudes. We also tested the algorithms on the sunspots data (Figure 4.3.2A) with parameters $n = 7, d = 18$ and $\tau = 50$, with similar results.

5 Directions for Future Work

The research we have carried out thus far explores several aspects of efficient and accurate inference, structure and parameter learning in models of dynamical systems.

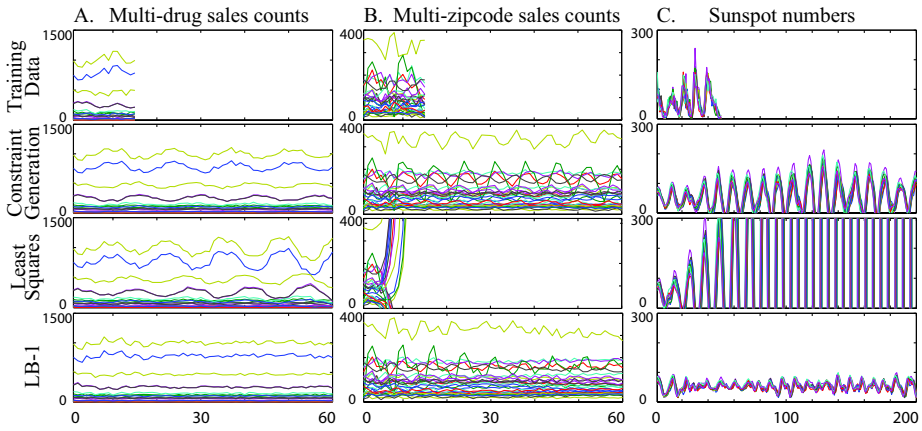


Figure 11: (A): 60 days of data for 22 drug categories aggregated over all zipcodes in the city. (B): 60 days of data for a single drug category (cough/cold) for all 29 zipcodes in the city. (C): Sunspot numbers for 200 years separately for each of the 12 months. The training data (top), simulated output from constraint generation, output from the unstable least squares model, and output from the over-dampened LB-1 model (bottom).

DMC HMMs (Siddiqi & Moore, 2005) offer efficiency and scalability while allowing a principled tradeoff between fully unconstrained and sparse transition models in HMMs. STACS and V-STACS (Siddiqi et al., 2007b) are fast, scalable algorithms for simultaneously learning the topology of HMMs while determining locally optimal parameters, while avoiding many of the local minima problems inherent in other methods for HMM model selection and learning. Siddiqi et al. (2007a) uses subspace identification to learn LDS parameters while enforcing stability in the dynamics matrix through a constraint generation algorithm, in a way that is more efficient and accurate than previous methods. All of these models and algorithms increase our understanding of dynamical systems and latent variable models, and provide us a set of tools for efficient operations on these models.

This research draws our attention to some recurring themes that motivate our proposed work. Efficiently and accurately learning latent variable models while avoiding instability and local minima is a difficult problem. Model selection and dimensionality reduction are also challenging. Finally, different latent variable models offer different benefits and drawbacks. LDSs possess a smoothly evolving state variable that is effective at modeling systems with smooth state trajectories, but fail at the task of modeling non-convex observation likelihoods such as required by some systems. HMMs handle non-convexity better due to the discrete state, but are only suitable for systems with non-continuous state transitions and not for systems where smooth evolution of the state is required. In this thesis we aim to build better models for dynamical systems by: (a) extending our learning algorithms to the predictive modeling framework of PSRs in order to leverage advantages of that framework such as the absence of identifiabil-

ity problems and the existence of *consistent* learning algorithms, (b) using the matrix decomposition-based learning framework of subspace identification to perform model selection and dimensionality reduction and avoid the intractability and local minima problems of latent variable model algorithms, and (c) generalizing to the larger set of exponential family models for greater flexibility and representative power. We believe the result of this work will be a novel model that combines the benefits of models such as HMMs and PLGs. Our model will differ from existing hybrid models such as SSSMs by being more general and by avoiding some of the parameter learning difficulties inherent in these models. Another goal is to formulate better algorithms for existing predictive models.

5.1 Better Learning Algorithms for Low Dimensional PSRs

Conventional PSRs represent the state of a discrete-observation dynamical system as a vector of N probabilities of a set of observable future outcomes called *core tests* that form a sufficient statistic for the system at any given time, in the sense that the probability of *any other future event* can be represented linearly in terms of the core test probabilities. As mentioned above, note that tests need not be simply the occurrence of future observations. they could also be some arbitrary nonlinear statistic about one or more future observations. While this makes the state highly interpretable, it poses the problem of discovering the dimensionality of a system given a data set as well as discovering the correct set of core tests. In contrast, low dimensional PSR models such as TPSRs represent the state of a PSR not as a set of probabilities of N core tests, but as N linear combinations of a larger number of test probabilities. Though the state itself no longer consists of quantities as easily interpretable as probabilities, an advantage of the TPSR representation is that it removes the need of discovering an exact set of core tests. Rather, an overly large set of simple tests can be initially specified since we will be storing a small number of linear combinations of their probabilities. Another advantage of the TPSR representation is that it allows us to estimate the dimensionality of the model using matrix decomposition algorithms such as SVD and thresholding the singular values to obtain a low-rank approximation (Rosencrantz & Gordon, 2004). TPSRs have been shown to perform well in comparison to HMMs in a robot tracking task.

We propose to improve on two shortcomings of TPSR learning algorithms. Firstly, the algorithms are currently based on the original definition of $histories$ as a sequence of observable events from the beginning of time until the current timestep. Instead, *suffix-histories* (Wolfe et al., 2005) specify a sequence of observable events in the last few timesteps and use that to find core test probabilities, which is implicitly an average over rows of the system-dynamics matrix. Suffix-histories are similar to the *indicative events* (Jaeger, 2000) of OOMs. Using suffix-histories to represent TPSR states will make them more efficient and compact. Secondly, TPSR learning algorithms are not guaranteed to return *stable* parameters. This causes TPSRs to eventually generate invalid states (i.e. invalid probabilities) when we simulate data from them. We aim to investigate properties of the PSR state space and address this shortcoming by formulating a parameter learning algorithm for TPSRs that guarantees stability. One step in this direction is to reparameterize the TPSR state space into a form which is easier

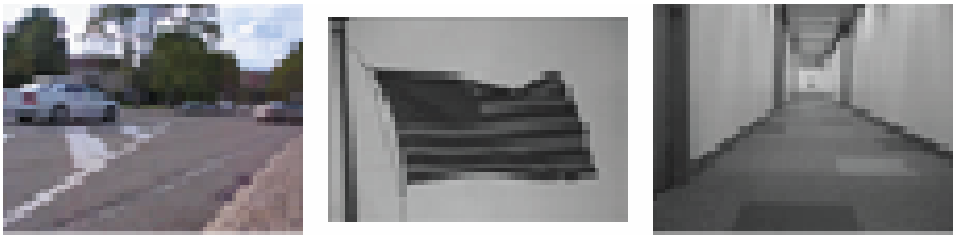


Figure 12: Frames from more structured dynamic textures with non-convex state spaces (traffic moving, flag waving, robot traversing hallway) that we aim to be able to model using low-dimensional exponential family PSRs.

to represent and renormalize at every step, similar to the simplex for HMMs and the ellipse for LDSs.

5.2 Learning Low-Dimensional PLGs

Predictive Linear Gaussians represent dynamical systems of real-valued observations with a Gaussian distribution over the next N future observations. The parameters of this distribution form a sufficient statistic for the state of the system, and inference is performed by *extending* and *conditioning* this window of N observations. PLGs subsume LDSs and have the advantage, like other predictive models, of allowing *consistent* learning algorithms. However, what if the state of the system depends on some longer-range future observations that extend far beyond N timesteps in the future? PLGs can only model such systems by extending their window out to those future observations, which may make N so large as to render the model impractical, especially in cases where each individual observation is high-dimensional (e.g. images). The current PLG model does not easily allow us to represent such systems nor to detect their dimensionality in the first place.

We propose to incorporate the benefits of transformed PSRs into PLGs by formulating a low dimensional PLG or transformed PLG (TPLG) model that maintains a distribution over N linear combinations over a much larger set of future observations. In addition to allowing us to model linear Gaussian systems with a small set of long-range dependencies in its state, this model allows us to use matrix decomposition to determine the order of the system, much as in TPSRs.

5.3 A Low-Dimensional Exponential Family PSR

The EFPSR model of Wingate and Singh (2007) defines the state of a dynamical system as the time-varying parameters of an exponential family distribution over the next N future observations. This allows us to model structure in the observations using graphical exponential family models, lending EFPSRs both the power of state-of-the-art probabilistic modeling and the problems of intractable inference and learning in the general case, which are tackled using approximate inference techniques. Exponential

Family PSRs offer the potential to generalize predictive models such as PSRs and PLGs (which in turn subsume latent variable models such as HMMs and LDSs) by allowing us to model statistics of future observations with arbitrary graphical structure as random variables with possibly non-Gaussian distributions that are still well-behaved in some sense due to convenient properties of the exponential family.

We propose to investigate generalizations and alternative formulations of exponential family PSRs and their algorithms for both real-valued and discrete observations. One possible generalization is to devise low-dimensional exponential family PSRs which model N statistics over a large number of future observations along with their dependencies using graphical exponential family models over arbitrary features of future observations. We believe that exponential family PSRs will allow us to combine benefits of models we have seen so far and model systems such as the clock pendulum video (Figure 4) and others that exhibit non-convex state spaces (Figure 12) which requires the ability to model smoothly evolving state variables in non-convex observation likelihood spaces. The challenge is to formulate a model general enough to unify these different trends while still allowing tractable inference and learning algorithms.

5.4 Algorithms for Low-Dimensional Exponential Family PSRs

We also propose to devise efficient inference, parameter learning and structure learning algorithm for the low-dimensional exponential family PSR, again utilizing the global convergence and lack of local minima inherent in matrix decomposition techniques such as the SVD. In addition to learning parameters, this will allow us to select the dimensionality of the models. One drawback of of the existing EFPSR inference algorithm is its inconsistency in marginal distribution over future observations during its extension and conditioning phases, which arises due to ignoring the need for backward message-passing from the future in the model. We aim to address this drawback and ensure consistency in our inference algorithms.

An important benefit of the learning algorithms we propose is its avoidance of the need for approximate inference and learning in other hybrid models such as SSSMs (Ghahramani & Hinton, 2000) by using the subspace identification paradigm.

5.5 Experimental Evaluation

We will evaluate our models and algorithms on domains consistent with previous work in the respective areas, using sequential data sets well known in the literature as in (Siddiqi & Moore, 2005; Siddiqi et al., 2007b). An important application domain is dynamic textures of video data, as in our previous work (Siddiqi et al., 2007a), where we aim to be able to model interesting textures with more structure and complexity than previously possible. We will also apply our algorithms to data from the biosurveillance domain, as done in the past to model stable baseline models for outbreak detection (Siddiqi et al., 2007a).

5.6 Timeline

Work on the ideas and goals outlined in this thesis proposal is already in progress. The research on low-dimensional PSRs will be carried out by the end of March 2008, and the work on low-dimensional PLGs by May 2008. Work on the exponential family PSR model and algorithms is projected to conclude by September 2008, with thesis writing and defense scheduled to complete by December 2008.

References

Ackerson, G. A., & Fu, K. S. (1970). On state estimation in switching environments. *IEEE Transactions on Automatic Control, 15(1)*, 10–17.

Bahl, L. R., Jelinek, F., & Mercer, R. L. (1983). A maximum likelihood approach to continouous speech recognition. *IEEE Trans Pattern Anal Machine Intell.* (pp. 179–190).

Bar-Shalom, Y., & Li, X. R. (1993). *Estimation and tracking*. Artech House.

Baum, L. (1972). An inequality and associated maximization technique in statistical estimation of probabilistic functions of a Markov process. *Inequalities, 3*, 1–8.

Bengio, Y., & Frasconi, P. (1995). An input output HMM architecture. *Proc. NIPS*.

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Brand, M. (1999). Structure learning in conditional probability models via an entropic prior and parameter extinction. *Neural Computation, 11*, 1155–1182.

Brand, M., Oliver, N., & Pentland, A. (1997). Coupled hidden Markov models for complex action recognition. *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Chen, R., & Liu, J. (2000). Mixture kalman filters. *Journal of the Royal Statistical Society B, 62*, 493–508.

Chui, N. L. C., & Maciejowski, J. M. (1996). Realization of stable models with subspace methods. *Automatica, 32(100)*, 1587–1595.

Cox, H. (1964). On the estimation of state variables and parameters for noisy dynamic systems. *IEEE Transactions on Automatic Control, 9*, 5–12.

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal Royal Statistical Society, Series B, 39*, 1–38.

Felzenszwalb, P., Huttenlocher, D., & Kleinberg, J. (2003). Fast Algorithms for Large State Space HMMs with Applications to Web Usage Analysis. *Advances in Neural Information Processing Systems (NIPS)*.

Fine, S., Singer, Y., & Tishby, N. (1998). The Hierarchical Hidden Markov Model: Analysis and Applications. *Machine Learning, 32*, 41–62.

Fraser, A. M., & Dimitriadis, A. (1993). Forecasting probability densities by using hidden markov models with mixed states.

Ghahramani, Z. (1998). Learning dynamic Bayesian networks. *Lecture Notes in Computer Science*, *1387*, 168–197.

Ghahramani, Z. (2001). An introduction to hidden markov models and bayesian networks. *International Journal of Pattern Recognition and Artificial Intelligence*, *15*, 9–42.

Ghahramani, Z., & Hinton, G. E. (1996). *Parameter estimation for Linear Dynamical Systems* (Technical Report CRG-TR-96-2). U. of Toronto, Department of Comp. Sci.

Ghahramani, Z., & Hinton, G. E. (2000). Variational learning for switching state-space models. *Neural Computation*, *12*, 831–864.

Ghahramani, Z., & Jordan, M. (1995). Factorial hidden Markov models. *Proc. Conf. Advances in Neural Information Processing Systems, NIPS* (pp. 472–478). MIT Press.

Ghahramani, Z., & Roweis, S. (1999). Learning nonlinear dynamical systems using an EM algorithm. *Proc. NIPS*.

Horn, R., & Johnson, C. R. (1985). *Matrix analysis*. Cambridge University Press.

Horst, R., & Pardalos, P. M. (Eds.). (1995). *Handbook of global optimization*. Kluwer.

Jaeger, H. (2000). Observable operator models for discrete stochastic time series. *Neural Computation*, *12*, 1371–1398.

James, M., & Singh, S. (2004). Learning and discovery of predictive state representations in dynamical systems with reset. *Proc. ICML*.

Kalman, R. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*.

Keogh, E., & Folias, T. (2002). The UCR Time Series Data Mining Archive.

Kopp, R. E., & Orford, R. J. (1963). Linear regression applied to system identification and adaptive control systems. *Journal of the American Institute of Aeronautics and Astronautics*, *1*, 2300–2306.

Krogh, A., Mian, I., & Haussler, D. (1994). A hidden Markov model that finds genes in E. coli DNA. *Nucleic Acids Research*, *22*, 4768–4778.

Lacy, S. L., & Bernstein, D. S. (2002). Subspace identification with guaranteed stability using constrained optimization. *Proc. American Control Conference*.

Lacy, S. L., & Bernstein, D. S. (2003). Subspace identification with guaranteed stability using constrained optimization. *IEEE Transactions on Automatic Control, 48(7)*, 1259–1263.

Li, C., & Biswas, G. (1999). Temporal pattern generation using hidden markov model based unsupervised classification (pp. 245–256.).

Littman, M., Sutton, R., & Singh, S. (2002). Predictive representations of state. *Advances in Neural Information Processing Systems (NIPS)*.

Ljung, L. (1999). *System Identification: Theory for the user*. Prentice Hall. 2nd edition.

Murphy, K. (2002). *Dynamic Bayesian Networks: Representation, Inference and Learning*. Doctoral dissertation, UC Berkeley.

Murphy, K., & Paskin, M. (2002). Linear Time Inference in Hierarchical HMMs. *Advances in Neural Information Processing Systems (NIPS)*.

Ostendorf, M., & Singer, H. (1997). Hmm topology design using maximum likelihood successive state splitting. *Computer Speech and Language, 11*, 17–41.

Rabiner, L. R. (1989). A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proc. IEEE, 77*, 257–285.

Rauch, H. (1963). Solutions to the linear smoothing problem. *IEEE Transactions on Automatic Control*.

Rosencrantz, M., & Gordon, G. J. (2004). Learning low dimensional predictive representations. *Proc. ICML*.

Roweis, S., & Ghahramani, Z. (1999). A unified view of linear gaussian models. *Neural Computation, 11*, 305–345.

Rudary, M., & Singh, S. (2003). A nonlinear predictive state representation. *Proc. NIPS*.

Rudary, M., Singh, S., & Wingate, D. (2005). Predictive linear-gaussian models of stochastic dynamical systems. *Proc. UAI*.

Salakhutdinov, R., Roweis, S., & Ghahramani, Z. (2003). Optimization with EM and Expectation-Conjugate-Gradient. *Proc. ICML*.

Seymore, K., McCallum, A., & Rosenfeld, R. (1999). Learning hidden Markov model structure for information extraction. *AAAI'99 Wkshp Machine Learning for Information Extraction*.

Shumway, R. H., & Stoffer, D. S. (1993). Dynamic linear models with switching. *J. Amer. Stat. Assoc., 86*, 763–769.

Siddiqi, S., Boots, B., & Gordon, G. J. (2007a). A constraint generation approach to learning stable linear dynamical systems. *Proc. NIPS*.

Siddiqi, S., Gordon, G. J., & Moore, A. (2007b). Fast state discovery for HMM model selection and learning. *Proceedings of the* 11^{th} *International Conference on Artificial Intelligence and Statistics (AI-STATS)*.

Siddiqi, S. M., & Moore, A. W. (2005). Fast inference and learning in large-state-space HMMs. *Proc. ICML*.

Singh, S., James, M., & Rudary, M. (2004). Predictive state representations: A new theory for modeling dynamical systems. *Proc. UAI*.

Singh, S., Littman, M., Jong, N., Pardoe, D., & Stone, P. (2003). Learning predictive state representations. *Proc. ICML*.

Soatto, S., Doretto, G., & Wu, Y. (2001). Dynamic Textures. *Intl. Conf. on Computer Vision*.

Sunderesan, A., Chowhdury, A. K. R., & Chellappa, R. (2003). A Hidden Markov Model Based Framework for Recognition of Humans from Gait Sequences. *Proc. Intl. Conf. on Image Processing*.

Van Gestel, T., Suykens, J. A. K., Van Dooren, P., & De Moor, B. (2001). Identification of stable models in subspace identification by using regularization. *IEEE Transactions on Automatic Control*.

Van Overschee, P., & De Moor, B. (1996). *Subspace identification for linear systems: Theory, implementation, applications*. Kluwer.

Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, *IT-13*, 260–267.

Wingate, D., & Singh, S. (2006a). Kernel predictive linear gaussian models for nonlinear stochastic dynamical systems. *Proc. ICML*.

Wingate, D., & Singh, S. (2006b). Mixtures of predictive linear gaussian models for nonlinear stochastic dynamical systems. *Proc. AAAI*.

Wingate, D., & Singh, S. (2007). Exponential family predictive representations of state. *Proc. NIPS*.

Wolfe, B., James, M., & Singh, S. (2005). Learning predictive state representations in dynamical systems without reset. *Proc. ICML*.