Hilbert Space Embeddings of Hidden Markov Models

Le Song LESONG@CS.CMU.EDU
Byron Boots BEB@CS.CMU.EDU
School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA

Sajid M. Siddiqi SIDDIQI@GOOGLE.COM
Google, Pittsburgh, PA 15213, USA

Geoffrey Gordon GGORDON@CS.CMU.EDU
School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA

Alex Smola ALEX@SMOLA.ORG
Yahoo! Research, Santa Clara, CA 95051, USA

Abstract

Hidden Markov Models (HMMs) are important tools for modeling sequence data. However, they are restricted to discrete latent states, and are largely restricted to Gaussian and discrete observations. And, learning algorithms for HMMs have predominantly relied on local search heuristics, with the exception of spectral methods such as those described below. We propose a nonparametric HMM that extends traditional HMMs to structured and non-Gaussian continuous distributions. Furthermore, we derive a local-minimum-free kernel spectral algorithm for learning these HMMs. We apply our method to robot vision data, slot car inertial sensor data and audio event classification data, and show that in these applications, embedded HMMs exceed the previous state-of-the-art performance.

1. Introduction

Hidden Markov Models (HMMs) have successfully modeled sequence data in a wide range of applications including speech recognition, analysis of genomic sequences, and analysis of time series. HMMs are *latent variable models* of dynamical systems: they assume a latent state which evolves according to Markovian dynamics, as well as observations which depend only on the hidden state at a particular time.

Appearing in *Proceedings of the 27th International Conference on Machine Learning*, Haifa, Israel, 2010.

Despite their simplicity and wide applicability, HMMs are limited in two major respects: first, they are usually restricted to discrete or Gaussian observations, and second, the latent state variable is usually restricted to have only moderate cardinality. For non-Gaussian continuous observations, and for structured observations with large cardinalities, standard inference algorithms for HMMs run into trouble: we need huge numbers of latent states to capture such observation distributions accurately, and marginalizing out these states during inference can be very computationally intensive. Furthermore, standard HMM learning algorithms are not able to fit the required transition and observation distributions accurately: local search heuristics, such as the EM algorithm, lead to bad local optima, and standard approaches to regularization result in under- or overfitting.

Recently, Hsu et al. (2009) proposed a spectral algorithm for learning HMMs with discrete observations and hidden states. At its core, the algorithm performs a singular value decomposition of a matrix of joint probabilities of past and future observations, and then uses the result, along with additional matrices of joint probabilities, to recover parameters which allow tracking or filtering. The algorithm employs an observable representation of a HMM, and avoids explicitly recovering the HMM transition and observation matrices. This implicit representation enables the algorithm to find a consistent estimate of the distribution of observation sequences, without resorting to local search.

Unfortunately, this spectral algorithm is only formulated for HMMs with discrete observations. In contrast, many sources of sequential data are continous

or structured; the spectral algorithm does not apply to such data without discretization and flattening. So, the goal of the current paper is to provide a new kernel-based representation and kernelized spectral learning algorithm for HMMs; this new representation and algorithm will allow us to learn HMMs in any domain where we can define a kernel. Furthermore, our algorithm is free of local minima and admits finite-sample generalization guarantees.

In particular, we will represent HMMs using a recent concept called Hilbert space embedding (Smola et al., 2007; Sriperumbudur et al., 2008). The essence of Hilbert space embedding is to represent probability measures (in our case, corresponding to distributions over observations and latent states in a HMM) as points in Hilbert spaces. We can then perform inference in the HMM by updating these points, entirely in their Hilbert spaces, using covariance operators (Baker, 1973) and conditional embedding operators (Song et al., 2009). By making use of the Hilbert space's metric structure, our method works naturally with continous and structured random variables, without the need for discretization.

In addition to generalizing HMMs to arbitary domains where kernels are defined, our learning algorithm contributes to the theory of Hilbert space embeddings with hidden variables. Previously, Song et al. (2009) derived a kernel algorithm for HMMs; however, they only provided results for fully observable models, where the training data includes labels for the true latent states. By contrast, our algorithm only requires access to an (unlabeled) sequence of observations.

We provide experimental results comparing embedded HMMs learned by our spectral algorithm to several other well-known approaches to learning models of time series data. The results demonstrate that our novel algorithm exceeds the previous state-of-the-art performance, often beating the next best algorithm by a substantial margin.

2. Preliminaries

In this paper, we follow the convention that uppercase letters denote random variables (*e.g.* X_t, H_t) and lowercase letters their instantiations (*e.g.* x_t, h_t). We will use $\mathbb{P}$ to denote probability distribution in the discrete cases and density in the continuous cases. For matrices and vectors, we will use notation $u = (u_i)_i$ and $C = (C_{ij})_{ij}$ to list their entries. Following (Hsu et al., 2009), we abbreviate a sequence $(x_1, \ldots, x_t)$ by $x_{1:t}$, and its reverse $(x_t, \ldots, x_1)$ by $x_{t:1}$. When we use a sequence as a subscript, we mean the product of quantities indexed by the sequence elements (*e.g.* $A_{x_{t:1}} = A_{x_t} \ldots \mathcal{A}_{x_1}$). We use $\mathbf{1}_m$ to denote an $m \times 1$ column of ones.

A *discrete* HMM defines a probability distribution over sequences of hidden states, $H_t \in \{1, \ldots, N\}$, and observations, $X_t \in \{1, \ldots, M\}$. We assume $N \ll M$, and let $T \in \mathbb{R}^{N \times N}$ be the state transition probability matrix with $T_{ij} = \mathbb{P}(H_{t+1} = i | H_t = j)$, $O \in \mathbb{R}^{M \times N}$ be the observation probability matrix with $O_{ij} = \mathbb{P}(X_t = i | H_t = j)$, and $\pi \in \mathbb{R}^N$ be the stationary state distribution with $\pi_i = \mathbb{P}(H_t = i)$. The conditional independence properties of the HMM imply that T, O and π fully characterize the probability distribution of any sequence of states and observations.

2.1. Observable representation for HMMs

Jaeger (2000) demonstrated that discrete HMMs can be formulated in terms of 'observation operators' A_{x_t}. Each A_{x_t} is a matrix of size $N \times N$ with its ij-th entry defined as $\mathbb{P}(H_{t+1} = i | H_t = j)\mathbb{P}(X_t = x_t | H_t = j)$, or in matrix notation, $A_{x_t} = T \operatorname{diag}(O_{x_t,1}, \ldots, O_{x_t,m})$. Then the probability of a sequence of observations, $x_{1:t}$, can be written as matrix operations,

$$\mathbb{P}(x_{1:t}) = \mathbf{1}_N^\top A_{x_t} \ldots A_{x_1} \pi = \mathbf{1}_N^\top A_{x_{t:1}} \pi. \qquad (1)$$

Essentially, each A_{x_t} incorporates information about one-step observation likelihoods and one-step hidden state transitions. The sequence of matrix multiplications in equation (1) effectively implements the marginalization steps for the sequence of hidden variables, $H_{t+1:1}$. Likewise, the predictive distribution for one-step future X_{t+1} given a history of observations can be written as a sequence of matrix multiplications,

$$(\mathbb{P}(X_{t+1} = i | x_{1:t}))_{i=1}^M \propto O A_{x_{t:1}} \pi \qquad (2)$$

The drawback of the representations in (1) and (2) is that they requires the exact knowledge of the transition matrix T and observation matrix O, and neither quantity is available during training (since the latent states are usually not observable).

A key observation concerning Equations (1) and (2) is that if we are only interested in the final quantity $\mathbf{1}_N^\top A_{x_{t:1}} \pi$ and $O A_{x_{t:1}} \pi$, we may not *need* to recover the A_{x_t}s exactly. Instead, it will suffice to recover them up to some invertible transformation. More specifically, suppose that matrix S is invertible, we can define a set of new quantities,

$$b_1 := S\pi, \quad b_\infty := OS^{-1}, \quad B_x := SA_xS^{-1} \qquad (3)$$

and equivalently compute $O A_{x_{t:1}} \pi$ by cancelling out all S during matrix multiplications, resulting in

$$\begin{aligned} O A_{x_{t:1}} \pi &= \left(OS^{-1}\right)\left(SA_{x_t}S^{-1}\right) \ldots \left(SA_{x_1}S^{-1}\right)(S\pi) \\ &= b_\infty B_{x_{t:1}} b_1 \end{aligned} \qquad (4)$$

The natural question is how to choose S such that b_1, b_∞ and B_x can be computed based purely on observation sequences, $x_{1:t}$.

Hsu et al. (2009) show that $S = U^\top O$ works, where U is the top N left singular vectors of the joint probability matrix (assuming stationarity of the distribution):

$$C_{2,1} := (\mathbb{P}(X_{t+1} = i, X_t = j))_{i,j=1}^{M}. \quad (5)$$

Furthermore, b_1, b_∞ and B_x can also be computed from observable quantities (assuming stationarity),

$$u_1 := (\mathbb{P}(X_t = i))_{i=1}^{M}, \quad (6)$$

$$C_{3,x,1} := (\mathbb{P}(X_{t+2} = i, X_{t+1} = x, X_t = j))_{i,j=1}^{M} \quad (7)$$

which are the marginal probability vector of sequence singletons, and *one slice* of the joint probability matrix of sequence triples (*i.e.* a slice indexed by x from a 3-dimensional matrix). Hsu et al. (2009) showed

$$b_1 = U^\top u, \quad b_\infty = C_{2,1}(U^\top C_{2,1})^\dagger \quad (8)$$

$$B_x = (U^\top C_{3,x,1})(U^\top C_{2,1})^\dagger. \quad (9)$$

2.2. A spectral algorithm for learning HMMs

The spectral algorithm for learning HMMs proceeds by first estimating u_1, $C_{2,1}$ and $C_{3,x,1}$. Given a dataset of m *i.i.d.* triples $\{(x_1^l, x_2^l, x_3^l)\}_{l=1}^{m}$ from a HMM (superscripts index training examples), we estimate

$$\hat{u}_1 = \tfrac{1}{m} \sum\nolimits_{l=1:m} \varphi(x_1^l) \quad (10)$$

$$\hat{C}_{2,1} = \tfrac{1}{m} \sum\nolimits_{l=1:m} \varphi(x_2)\varphi(x_1)^\top \quad (11)$$

$$\hat{C}_{3,x,1} = \tfrac{1}{m} \sum\nolimits_{l=1:m} \mathbb{I}[x_2^l = x]\varphi(x_3^l)\varphi(x_1^l)^\top \quad (12)$$

where the delta function (or delta kernel) is defined as $\mathbb{I}[x_2^l = x] = 1$ if $x_2^l = x$ and 0 otherwise; and we have used 1-of-M representation for discrete variables. In this representation, $\varphi(x = i)$ is a vector of length M with all entries equal to zero except 1 at i-th position. For instance, if $x = 2$, then $\varphi(x) = (0, 1, 0, \ldots, 0)^\top$. Furthermore, we note that $\hat{C}_{3,x,1}$ is not a single but a collection of matrices each indexed by an x. Effectively, the delta function $\mathbb{I}[x_2^l = x]$ partition the observation triples according to x, and each $\hat{C}_{3,x,1}$ only gets a fraction of the data for the estimation.

Next, a 'thin' SVD is computed for $\hat{C}_{2,1}$. Let its top N left singular vectors be $\hat{U}$, then the observable representation for the HMM ($\hat{b}_1$, $\hat{b}_\infty$ and $\hat{B}_x$) can be estimated by replacing the population quantities with their corresponding finite sample counterparts.

A key feature of the algorithm is that it does not explicitly estimate the transition and observation models; instead it estimates a set of observable quantities that differ by an invertible transformation. The core part of the algorithm is a SVD which is local minimum free. Furthermore, (Hsu et al., 2009) also prove that under suitable conditions this spectral algorithm for HMMs efficiently estimates both the marginal and predictive distributions.

3. Hilbert Space Embeddings of HMMs

The spectral algorithm for HMMs derived by Hsu et al. (2009) is only formulated for discrete random variables. Based on their formulation, it is not clear how one can apply this algorithm to general cases with continuous and structured variables. For instance, a difficulty lies in estimating $\hat{C}_{3,x,1}$. As we mentioned earlier, to estimate each $\hat{C}_{3,x,1}$, we need to partition the observation triples according to x, and each $\hat{C}_{3,x,1}$ only gets a fraction of the data for the estimation. For continous observations, x can take infinite number of possibile values, which makes the partition estimator impractical. Alternatively, one can perform a Parzen window density estimation for continuous variables. However, further approximations are needed in order to make Parzen window compatible with this spectral algorithm (Siddiqi et al., 2009).

In the following, we will derive a new presentation and a kernel spectral algorithm for HMMs using a recent concept called Hilbert space embeddings of distributions (Smola et al., 2007; Sriperumbudur et al., 2008). The essence of our method is to represent distributions as points in Hilbert spaces, and update these points entirely in the Hilbert spaces using operators (Song et al., 2009). This new approach avoids the need for partitioning the data making it applicable to any domain where kernels can be defined.

3.1. Hilbert space embeddings

Let $\mathcal{F}$ be a reproducing kernel Hilbert space (RKHS) associated with kernel $k(x, x') := \langle \varphi(x), \varphi(x') \rangle_{\mathcal{F}}$. Then for all functions $f \in \mathcal{F}$ and $x \in \mathcal{X}$ we have the reproducing property: $\langle f, \varphi(x) \rangle_{\mathcal{F}} = f(x)$, *i.e.* the evaluation of function f at x can be written as an inner product. Examples of kernels include the Gaussian RBF kernel $k(x, x') = \exp(-s \left\| x - x' \right\|^2)$, however kernel functions have also been defined on strings, graphs, and other structured objects.

Let $\mathcal{P}$ be the set of probability distributions on $\mathcal{X}$, and X the random variable with distribution $\mathbb{P} \in \mathcal{P}$. Following Smola et al. (2007), we define the mapping of $\mathbb{P} \in \mathcal{P}$ to RKHS $\mathcal{F}$, $\mu_X := \mathbb{E}_{X \sim \mathbb{P}}[\varphi(X)]$, as the Hilbert space embedding of $\mathbb{P}$ or simply mean map. For all $f \in \mathcal{F}$, $\mathbb{E}_{X \sim \mathbb{P}}[f(X)] = \langle f, \mu_X \rangle_{\mathcal{F}}$ by the reproducing property. A characteristic RKHS is one for which the mean map is injective: that is, each distribution has a unique embedding (Sriperumbudur et al., 2008). This

property holds for many commonly used kernels (eg. the Gaussian and Laplace kernels when $\mathcal{X} = \mathbb{R}^d$).

As a special case of the mean map, the marginal probability vector of a discrete variable X is a Hilbert space embedding, *i.e.* $(\mathbb{P}(X = i))_{i=1}^M = \mu_X$. Here the kernel is the delta function $k(x, x') = \mathbb{I}[x = x']$, and the feature map is the 1-of-M representation for discrete variables (see section 2.2).

Given m *i.i.d.* observations $\{x^l\}_{l=1}^m$, an estimate of the mean map is straightforward: $\hat{\mu}_X := \frac{1}{m}\sum_{l=1}^m \varphi(x^l) = \frac{1}{m}\Upsilon \mathbf{1}_m$, where $\Upsilon := (\varphi(x^1), \ldots, \varphi(x^m))$ is a conceptual arrangement of feature maps into columns. Furthermore, this estimate computes an approximation within an error of $O_p(m^{-1/2})$ (Smola et al., 2007).

3.2. Covariance operators

The covariance operator is a generalization of the covariance matrix. Given a joint distribution $\mathbb{P}(X, Y)$ over two variables X on $\mathcal{X}$ and Y on $\mathcal{Y}$[1], the *uncentered* covariance operator $\mathcal{C}_{XY}$ is (Baker, 1973)

$$\mathcal{C}_{XY} := \mathbb{E}_{XY}[\varphi(X) \otimes \phi(Y)], \tag{13}$$

where $\otimes$ denotes tensor product. Alternatively, $\mathcal{C}_{XY}$ can simply be viewed as an embedding of joint distribution $\mathbb{P}(X, Y)$ using joint feature map $\psi(x, y) := \varphi(x) \otimes \phi(y)$ (in tensor product RKHS $\mathcal{G} \otimes \mathcal{F}$). For discrete variables X and Y with delta kernels on both domains, the covariance operator will coincide with the joint probability table, *i.e.* $(\mathbb{P}(X = i, Y = j)_{i,j=1}^M = \mathcal{C}_{XY}$ (also see section 2.2).

Given m pairs of *i.i.d.* observations $\{(x^l, y^l)\}_{l=1}^m$, we denote by $\Upsilon = \big(\varphi(x^1), \ldots, \varphi(x^m)\big)$ and $\Phi = \big(\phi(y^1), \ldots, \phi(y^m)\big)$. Conceptually, the covariance operator $\mathcal{C}_{XY}$ can then be estimated as $\hat{\mathcal{C}}_{XY} = \frac{1}{m}\Upsilon\Phi^\top$. This estimate also computes an approximation within an error of $O_p(m^{-1/2})$ (Smola et al., 2007).

3.3. Conditional embedding operators

By analogy with the embedding of marginal distributions, the conditional density $\mathbb{P}(Y|x)$ can also be represented as an RKHS element, $\mu_{Y|x} := \mathbb{E}_{Y|x}[\phi(Y)]$. We emphasize that $\mu_{Y|x}$ now traces out a family of embeddings in $\mathcal{G}$, with each element corresponding to a particular value of x. These conditional embeddings can be defined via a conditional embedding operator $\mathcal{C}_{Y|X} : \mathcal{F} \mapsto \mathcal{G}$ (Song et al., 2009),

$$\mu_{Y|x} = \mathcal{C}_{Y|X}\varphi(x) := \mathcal{C}_{YX}\mathcal{C}_{XX}^{-1}\varphi(x). \tag{14}$$

For discrete variables with delta kernels, conditional embedding operators correspond exactly to conditional probability tables (CPT), *i.e.* $(\mathbb{P}(Y = i|X = j))_{i,j=1}^M = \mathcal{C}_{Y|X}$, and each individual conditional embedding corresponds to one column of the CPT, *i.e.* $(\mathbb{P}(Y = i|X = x))_{i=1}^M = \mu_{Y|x}$.

Given m *i.i.d.* pairs $\{(x^l, y^l)\}_{l=1}^m$ from $\mathbb{P}(X, Y)$, the conditional embedding operator can be estimated as

$$\hat{\mathcal{C}}_{Y|X} = \tfrac{\Phi\Upsilon^\top}{m}(\tfrac{\Upsilon\Upsilon^\top}{m} + \lambda I)^{-1} = \Phi(K + \lambda m I)^{-1}\Upsilon^\top \tag{15}$$

where we have defined the kernel matrix $K := \Upsilon^\top\Upsilon$ with (i, j)th entry $k(x_i, x_j)$. The regularization parameter λ is to avoid overfitting. Song et al. (2009) also showed $\big\|\hat{\mu}_{Y|x} - \mu_{Y|x}\big\|_{\mathcal{G}} = O_p(\lambda^{1/2} + (\lambda m)^{-1/2})$.

3.4. Hilbert space observable representation

We will focus on the embedding $\mu_{X_{t+1}|x_{1:t}}$ for the predictive density $\mathbb{P}(X_{t+1}|x_{1:t})$ of a HMM. Analogue to the discrete case, we first express $\mu_{X_{t+1}|x_{1:t}}$ as a set of Hilbert space 'observable operators' $\mathcal{A}_x$. Specifically, let the kernels on the observations and hidden states be $k(x, x') = \langle\varphi(x), \varphi(x')\rangle_{\mathcal{F}}$ and $l(h, h') = \langle\phi(h), \phi(h')\rangle_{\mathcal{G}}$ respectively. For rich RKHSs, we define a linear operator $\mathcal{A}_x : \mathcal{G} \mapsto \mathcal{G}$ such that

$$\mathcal{A}_x\phi(h_t) = \mathbb{P}(X_t = x|h_t)\ \mathbb{E}_{H_{t+1}|h_t}[\phi(H_{t+1})]. \tag{16}$$

Then, by applying variable elimination, we have

$$\begin{aligned}
&\mu_{X_{t+1}|x_{1:t}} = \mathbb{E}_{H_{t+1}|x_{1:t}}\mathbb{E}_{X_{t+1}|H_{t+1}}[\varphi(X_{t+1})] \\
&= \mathcal{C}_{X_{t+1}|H_{t+1}}\mathbb{E}_{H_{t+1}|x_{1:t}}[\phi(H_{t+1})] \\
&= \mathcal{C}_{X_{t+1}|H_{t+1}}\mathcal{A}_{x_t}\mathbb{E}_{H_t|x_{1:t-1}}[\phi(H_t)] \\
&= \mathcal{C}_{X_{t+1}|H_{t+1}}\Big(\prod\nolimits_{\tau=1}^{t}\mathcal{A}_{x_\tau}\Big)\mu_{H_1}.
\end{aligned} \tag{17}$$

where we used the following recursive relation

$$\begin{aligned}
&\mathbb{E}_{H_{t+1}|x_{1:t}}[\phi(H_{t+1})] \\
&= \mathbb{E}_{H_t|x_{1:t-1}}\big[\mathbb{P}(X_t = x_t|H_t)\ \mathbb{E}_{H_{t+1}|H_t}[\phi(H_{t+1})]\big] \\
&= \mathcal{A}_{x_t}\mathbb{E}_{H_t|x_{1:t-1}}[\phi(H_t)]\,.
\end{aligned} \tag{18}$$

If we let $\mathcal{T} := \mathcal{C}_{X_t|H_t}$, $\mathcal{O} := \mathcal{C}_{X_{t+1}|H_{t+1}}$ and $\pi := \mu_{H_1}$, we obtain a form $\mu_{X_{t+1}|x_{1:t}} = \mathcal{O}\mathcal{A}_{x_{t:1}}\pi$ analogous to the discrete case (Equation (2)). The key difference is that Hilbert space representations are applicable to general domains with kernels defined.

Similar to the discrete case, the operators $\mathcal{A}_x$ *cannot* be directly estimated from the data since the hidden states are not provided. Therefore we derive a representation for $\mu_{X_{t+1}|x_{1:t}}$ based only on observable quantities (assuming stationarity of the distribution):

$$\mu_1 := \mathbb{E}_{X_t}[\varphi(X_t)] = \mu_{X_t} \tag{19}$$

$$\mathcal{C}_{2,1} := \mathbb{E}_{X_{t+1}X_t}[\varphi(X_{t+1}) \otimes \varphi(X_t)] = \mathcal{C}_{X_{t+1}X_t} \tag{20}$$

$$\begin{aligned}
\mathcal{C}_{3,x,1} &:= \mathbb{E}_{X_{t+2}(X_{t+1}=x)X_t}[\varphi(X_{t+2}) \otimes \varphi(X_t)] \\
&= \mathbb{P}(X_{t+1} = x)\mathcal{C}_{3,1|2}\varphi(x).
\end{aligned} \tag{21}$$

[1] a kernel $l(y, y') = \langle\phi(y), \phi(y')\rangle_{\mathcal{G}}$ is define on $\mathcal{Y}$ with associated RKHS $\mathcal{G}$.

where we have defined $\mathcal{C}_{3,1|2} := \mathcal{C}_{X_{t+2}X_t|X_{t+1}}$. First, we examine the relation between these observable quantities and the unobserved $\mathcal{O}$, $\mathcal{T}$ and π:

$$\begin{aligned}\mu_1 &= \mathbb{E}_{H_t}\mathbb{E}_{X_t|H_t}[\varphi(X_t)] = \mathcal{C}_{X_t|H_t}\mathbb{E}_{H_t}[\phi(H_t)] \\ &= \mathcal{O}\pi \qquad (22)\\ \mathcal{C}_{2,1} &= \mathbb{E}_{H_t}\left[\mathbb{E}_{X_{t+1}H_{t+1}|H_t}[\varphi(X_{t+1})] \otimes \mathbb{E}_{X_t|H_t}[\varphi(X_t)]\right] \\ &= \mathcal{C}_{X_{t+1}|H_{t+1}}\mathcal{C}_{H_{t+1}|H_t}\mathcal{C}_{H_tH_t}\mathcal{C}_{X_t|H_t}^\top \\ &= \mathcal{O}\mathcal{T}\mathcal{C}_{H_tH_t}\mathcal{O}^\top \qquad (23)\\ \mathcal{C}_{3,x,1} &= \mathbb{E}_{H_t}\left[\mathcal{O}\mathcal{A}_x\mathcal{T}\phi(H_t) \otimes \mathbb{E}_{X_t|H_t}[\varphi(X_t)]\right] \\ &= \mathcal{O}\mathcal{A}_x\mathcal{T}\mathcal{C}_{H_tH_t}\mathcal{O}^\top \qquad (24)\end{aligned}$$

In (24), we plugged in the following expansion

$$\begin{aligned}&\mathbb{E}_{X_{t+2}H_{t+2}H_{t+1}(X_{t+1}=x)|H_t}[\varphi(X_{t+2})] \\ &= \mathbb{E}_{H_{t+1}|H_t}\left[\mathbb{P}(x|H_{t+1})\mathbb{E}_{H_{t+2}|H_{t+1}}\mathbb{E}_{X_{t+2}|H_{t+2}}[\varphi(X_{t+2})]\right] \\ &= \mathcal{O}\mathcal{A}_x\mathcal{T}\phi(H_t) \qquad (25)\end{aligned}$$

Second, analogous to the discrete case, we perform a 'thin' SVD of the covariance operator $\mathcal{C}_{2,1}$, and take its top N left singular vectors $\mathcal{U}$, such that the operator $\mathcal{U}^\top\mathcal{O}$ is invertible. Some simple algebraic manipulations establish the relation between observable and unobservable quantities

$$\beta_1 := \mathcal{U}^\top\mu_1 = (\mathcal{U}^\top\mathcal{O})\pi \qquad (26)$$

$$\beta_\infty := \mathcal{C}_{2,1}(\mathcal{U}^\top\mathcal{C}_{2,1})^\dagger = \mathcal{O}(\mathcal{U}^\top\mathcal{O})^{-1} \qquad (27)$$

$$\mathcal{B}_x := (\mathcal{U}^\top\mathcal{C}_{3,x,1})(\mathcal{U}^\top\mathcal{C}_{2,1})^\dagger = (\mathcal{U}\mathcal{O})\mathcal{A}_x(\mathcal{U}\mathcal{O})^{-1}. \qquad (28)$$

With β_1, β_∞ and $\mathcal{B}_{x_{t:1}}$, $\mu_{X_{t+1}|x_{1:t}}$ can be expressed as the multiplication of observable quantities

$$\mu_{X_{t+1}|x_{1:t}} = \beta_\infty\mathcal{B}_{x_{t:1}}\beta_1 \qquad (29)$$

In practice, $\mathcal{C}_{3,x,1}$ (in equation (24)) is difficult to estimate, since it requires partitioning the training samples according to $X_{t+1} = x$. Intead, we use $\mathcal{C}_{3,1|2}\varphi(x)$ which does not require such partitioning, and is only a fixed multiplicative scalar $\mathbb{P}(x)$ away from $\mathcal{C}_{3,x,1}$. We define $\bar{\mathcal{B}}_x := (\mathcal{U}^\top(\mathcal{C}_{3,1|2}\varphi(x)))(\mathcal{U}^\top\mathcal{C}_{2,1})^\dagger$, and we have $\mu_{X_{t+1}|x_{1:t}} \propto \beta_\infty\bar{\mathcal{B}}_{x_{t:1}}\beta_1$.

We may want to predict i steps into future, *i.e.* obtain embeddings $\mu_{X_{t+i}|x_{t:1}}$ instead of $\mu_{X_{t+1}|x_{t:1}}$. This can be achieved by defining an i-step covariance operator $\mathcal{C}_{i+1,1} := \mathbb{E}_{X_{t+i}X_t}[\varphi(X_{t+i}) \otimes \varphi(X_t)]$ and replacing $\mathcal{C}_{2,1}$ in β_∞ (equation (27)) by $\mathcal{C}_{i+1,1}$. We then obtain the embedding $\mu_{X_{t+i}|x_{t:1}} \propto \beta_\infty^i\bar{\mathcal{B}}_{x_{t:1}}\beta_1$ where we use β_∞^i to denote $\mathcal{C}_{i+1,1}(\mathcal{U}^\top\mathcal{C}_{2,1})^\dagger$.

3.5. Kernel spectral algorithm for HMMs

Given a sample of m *i.i.d.* triplets $\left\{(x_1^l, x_2^l, x_3^l)\right\}_{l=1}^m$ from a HMM, the kernel spectral algorithm for HMMs proceeds by first performing a 'thin' SVD of the sample covariance $\hat{\mathcal{C}}_{2,1}$. Specifically, we denote feature matrices $\Upsilon = (\varphi(x_1^1), \ldots, \varphi(x_1^m))$ and $\Phi = (\varphi(x_2^1), \ldots, \varphi(x_2^m))$, and estimate $\hat{\mathcal{C}}_{2,1} = \frac{1}{m}\Phi\Upsilon^\top$. Then the left singular vector $v = \Phi\alpha$ ($\alpha \in \mathbb{R}^m$) can be estimated as follows

$$\begin{aligned}\Phi\Upsilon^\top\Upsilon\Phi^\top v = \omega v \quad &\Leftrightarrow \quad \Phi KLA\alpha = \omega\Phi\alpha \Leftrightarrow \\ LKL\alpha = \omega L\alpha, \quad &(\alpha \in \mathbb{R}^m,\ \omega \in \mathbb{R}) \qquad (30)\end{aligned}$$

where $K = \Upsilon^\top\Upsilon$ and $L = \Phi^\top\Phi$ are the kernel matrices, and α is the generalized eigenvector. After normalization, we have $v = \frac{1}{\sqrt{\alpha^\top L\alpha}}\Phi\alpha$. Then the $\mathcal{U}$ operator in equation (26), (27) and (28) is the column concatenation of the N top left singular vectors, *i.e.* $\hat{\mathcal{U}} = (v_1, \ldots, v_N)$. If we let $A := (\alpha_1, \ldots, \alpha_N) \in \mathbb{R}^{m\times N}$ be the column concatenation of the N top α_i, and $D := \operatorname{diag}\left((\alpha_1^\top L\alpha_1)^{-1/2}, \ldots, (\alpha_N^\top L\alpha_N)^{-1/2}\right) \in \mathbb{R}^{N\times N}$, we can concisely express $\hat{\mathcal{U}} = \Phi AD$.

Next we estimate $\hat{\mu}_1 = \frac{1}{m}\Upsilon\mathbf{1}_m$, and according to (26) $\hat{\beta}_1 = \frac{1}{m}D^\top A^\top\Phi^\top\Upsilon\mathbf{1}_m$. Similarly, according to (27) $\hat{\beta}_\infty = \frac{1}{m}\Phi\Upsilon^\top\left(D^\top A^\top\Phi^\top\frac{1}{m}\Phi\Upsilon^\top\right)^\dagger = \Phi KLAD\Omega^{-1}$, where we have defined $\Omega := \operatorname{diag}(\omega_1, \ldots, \omega_N)$, and used the relation $LKLA = LA\Omega$ and $A^\top LA = D^{-2}$. Last denote $\Psi = \left(\varphi(x_3^1), \ldots, \varphi(x_3^m)\right)$, then $\hat{\mathcal{C}}_{3,1|2}(\cdot) = \Psi\operatorname{diag}\left((L+\lambda I)^{-1}\Phi^\top(\cdot)\right)KLAD\Omega^{-1}$ in (21).

Algorithm 1 Kernel Spectral Algorithm for HMMs

In: m *i.i.d.* triples $\left\{(x_1^l, x_2^l, x_3^l)\right\}_{l=1}^m$, a sequence $x_{1:t}$.
Out: $\hat{\mu}_{X_{t+1}|x_{t:1}}$

1: Denote feature matrices $\Upsilon = (\varphi(x_1^1), \ldots, \varphi(x_1^m))$, $\Phi = (\varphi(x_2^1)\ldots\varphi(x_2^m))$ and $\Psi = (\varphi(x_3^1)\ldots\varphi(x_3^m))$.
2: Compute kernel matrices $K = \Upsilon^\top\Upsilon$, $L = \Phi^\top\Phi$, $G = \Phi^\top\Upsilon$ and $F = \Phi^\top\Psi$.
3: Compute top N generalized eigenvectors α_i using $LKL\alpha_i = \omega_i L\alpha_i$ ($\omega_i \in \mathbb{R}$ and $\alpha_i \in \mathbb{R}^m$).
4: Denote $A = (\alpha_1, \ldots, \alpha_N)$, $\Omega = \operatorname{diag}(\omega_1, \ldots, \omega_N)$ and $D = \operatorname{diag}\left((\alpha_1^\top L\alpha_1)^{-1/2}, \ldots, (\alpha_N^\top L\alpha_N)^{-1/2}\right)$.
5: $\hat{\beta}_1 = \frac{1}{m}D^\top A^\top G\mathbf{1}_m$
6: $\hat{\beta}_\infty = \Phi Q$ where $Q = KLAD\Omega^{-1}$
7: $\hat{\mathcal{B}}_{x_\tau} = \frac{\mathbb{P}(x_\tau)}{m}D^\top A^\top F\operatorname{diag}\left((L+\lambda I)^{-1}\Phi^\top\varphi(x_\tau)\right)Q$, for $\tau = 1, \ldots, t$.
8: $\hat{\mu}_{X_{t+1}|x_{t:1}} = \hat{\beta}_\infty\hat{\mathcal{B}}_{x_{t:1}}\hat{\beta}_1$

The kernel spectral algorithm for HMMs can be summarized in Algorithm 1. Note that in the algorithm, we assume that the marginal probability $\mathbb{P}(x_\tau)$ ($\tau = 1\ldots t$) is provided to the algorithm. In practice, this quantity is never explicitly estimated. Therefore, the algorithm returns $\hat{\beta}_\infty\bar{\mathcal{B}}_{x_{t:1}}\hat{\beta}_1$ which is just a constant scaling away from $\mu_{X_{t+1}|x_{t:1}}$ (note $\bar{\mathcal{B}}_x := \mathcal{B}_x/\mathbb{P}(x)$).

3.6. Sample complexity

In this section, we analyze the sample complexity of our kernel spectral algorithm for HMMs. In particu-

lar, we want to investigate how the difference between the estimated embedding $\hat{\mu}_{X_{t+1}|x_{1:t}}$ and its population counterpart scales with respect to the number m of training samples and the length t of the sequence $x_{1:t}$ in the conditioning. We use Hilbert space distances as our error measure and obtain the following result (the proof follows the template of Hsu et al. (2009), and it can be found in the appendix):

Theorem 1 *Assume* $\|\varphi(x)\|_{\mathcal{F}} \leq 1$, $\|\phi(h)\|_{\mathcal{G}} \leq 1$, $\max_x \|\mathcal{A}_x\|_2 \leq 1$. *Then* $\left\|\mu_{X_{t+1}|x_{1:t}} - \hat{\mu}_{X_{t+1}|x_{1:t}}\right\|_{\mathcal{F}} = O_p(t(\lambda^{1/2} + (\lambda m)^{-1/2}))$.

We expect that Theorem 1 can be further improved. Currently it suggests that given a sequence of length t, in order to obtain an unbiased estimator of $\mu_{X_{t+1}|x_{1:t}}$, we need to decrease λ with a schedule of $O_p(m^{-1/2})$ and obtain an overall convergence rate of $O_p(tm^{-1/4})$. Second, the assumption, $\max_x \|\mathcal{A}_x\|_2 \leq 1$, imposes smoothness constrants on the likelihood function $\mathbb{P}(x|H_t)$ for the theorem to hold. Finally, the current bound depends on the length t of the conditioning sequence. Hsu et al. (2009) provide a result that is independent of t using the KL-divergence as the error measure. For Hilbert space embeddings, it remains an open question as to how to estimate the KL-divergence and obtain a bound independent of t.

3.7. Predicting future observations

We have shown how to maintain the Hilbert space embeddings $\mu_{X_{t+1}|x_{1:t}}$ for the predictive distribution $\mathbb{P}(X_{t+1}|x_{1:t})$. The goal here is to determine the most probable future observations based on $\mu_{X_{t+1}|x_{1:t}}$. We note that in general we cannot directly obtain the probability of the future observation based on the embedding presentation of the distribution.

However, for a Gaussian RBF kernel defined over a compact subset of a real vector space, the embedding $\mu_{X_{t+1}|x_{1:t}}$ can be viewed as a nonparametric density estimator after proper normalization. In particular, let f be a constant function in the RKHS such that $\langle f, \varphi(X_{t+1})\rangle_{\mathcal{F}} = 1$, then the normalization constant Z can be estimated as $\hat{Z} = \left\langle f, \hat{\mu}_{X_{t+1}|x_{1:t}}\right\rangle_{\mathcal{F}}$. Since $\hat{\mu}_{X_{t+1}|x_{1:t}}$ is represented as $\sum_{l=1}^{m} \gamma_i \varphi(x_3^l)$, $\hat{Z}$ is simply $\sum_{l=1}^{m} \gamma_i$. We can then find the maximum a posteri (MAP) future observation by

$$\hat{x}_{t+1} = \text{argmax}_{x_{t+1}} \left\langle \hat{\mu}_{X_{t+1}|x_{1:t}}, \varphi(x_{t+1})\right\rangle_{\mathcal{F}} / \hat{Z} \quad (31)$$

Since $\|\varphi(x)\|_{\mathcal{F}} = 1$ for a Gaussian RBF kernel, a geometric interpretation of the above MAP estimate is to find a delta distribution $\delta_{x_{t+1}}$ such that its embedding $\varphi(x_{t+1})$ is closest to $\hat{\mu}_{X_{t+1}|x_{1:t}}$, *i.e.* $\hat{x}_{t+1} = \text{argmin}_{x_{t+1}} \|\varphi(x_{t+1}) - \hat{\mu}_{X_{t+1}|x_{1:t}}\|_{\mathcal{F}}$. The optimization in (31) may be a hard problem in general. In some cases, however, it is possible to define the feature map $\varphi(x)$ in such a way that an efficient algorithm for solving the optimization can be obtained, *e.g.* Cortes et al. (2005). In practice, we can also decode $\hat{x}_{t+1}$ by choosing the best one from existing training examples.

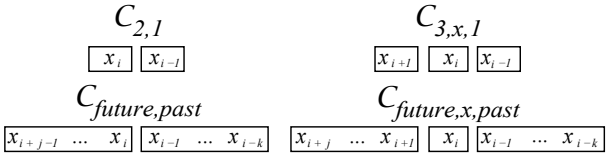

Figure 1. Operators $\mathcal{C}_{future,past}$ and $\mathcal{C}_{future,x,past}$ capture the dependence of *sequences* of k past and j future observations instead of single past and future observations.

3.8. Learning with sequences of observations

In the learning algorithm formulated above, each variable X_t corresponds to a single observation x_t from a data sequence. In this case, the operator $\mathcal{C}_{2,1}$ only captures the dependence between a single past observation and a single future observation (similarly for $\mathcal{C}_{3,x,1}$). In system identification theory, this corresponds to assuming 1-step observability (Van Overschee & De Moor, 1996) which is unduly restrictive for many partially observable real-world dynamical systems of interest. More complex sufficient statistics of past and future may need to be modeled, such as the *block Hankel matrix* formulations for subspace methods (Van Overschee & De Moor, 1996), to identify linear systems that are not 1-step observable. To overcome this limitation one can consider *sequences* of observations in the past and future and estimate operators $\mathcal{C}_{future,past}$ and $\mathcal{C}_{future,x,past}$ accordingly (Figure 1). As long as past and future sequences never overlap, these matrices have rank equal to that of the dynamics model and the theoretical properties of the learning algorithm continue to hold (see (Siddiqi et al., 2009) for details).

4. Experimental Results

We designed 3 sets of experiments to evaluate the effectiveness of learning embedded HMMs for difficult real-world filtering and prediction tasks. In each case we compare the learned embedded HMM to several alternative time series models including (I) linear dynamical systems (LDS) learned by Subspace Identification (Subspace ID) (Van Overschee & De Moor, 1996) with stability constraints (Siddiqi et al., 2008), (II) discrete HMMs learned by EM, and (III) the Reduced-rank HMM (RR-HMM) learned by spectral methods (Siddiqi et al., 2009). In these experiments we demonstrate that the kernel spectral learning algorithm for embedded HMMs achieves the state-of-the-art performance.

Robot Vision. In this experiment, a video of 2000 frames was collected at 6 Hz from a Point Grey Bumblebee2 stereo camera mounted on a Botrics Obot d100 mobile robot platform circling a stationary obstacle

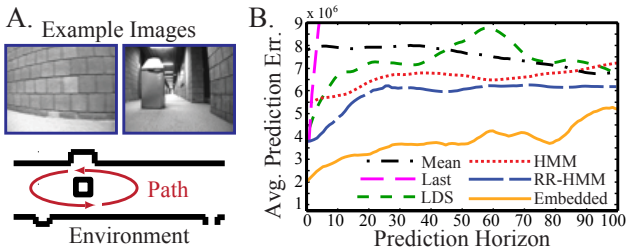

Figure 2. Robot vision data. (A) Sample images from the robot's camera. The figure below depicts the hallway environment with a central obstacle (black) and the path that the robot took through the environment (the red counter-clockwise ellipse). (B) Squared error for prediction with different estimated models and baselines.

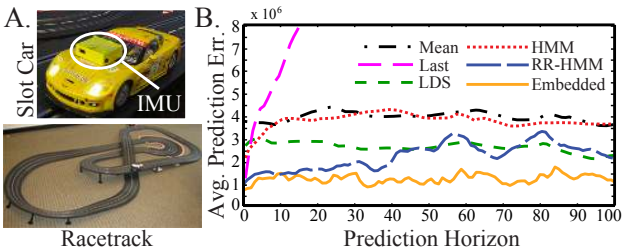

Figure 3. Slot car inertial measurement data. (A) The slot car platform and the IMU (top) and the racetrack (bottom). (B) Squared error for prediction with different estimated models and baselines.

(under imperfect human control) (Figure 2(A)) and 1500 frames were used as training data for each model. Each frame from the training data was reduced to 100 dimensions via SVD on single observations. The goal of this experiment was to learn a model of the noisy video, and, after filtering, to predict future image observations.

We trained a 50-dimensional[2] embedded HMM using Algorithm 1 with sequences of 20 consecutive observations (Section 3.8). Gaussian RBF kernels are used and the bandwidth parameter is set with the median of squared distance between training points (median trick). The regularization parameter λ is set of 10^{-4}. For comparison, a 50-dimensional RR-HMM with Parzen windows is also learned with sequences of 20 observations (Siddiqi et al., 2009); a 50-dimensional LDS is learned using Subspace ID with Hankel matrices of 20 time steps; and finally a 50-state discrete HMM and axis-aligned Gaussian observation models is learned using EM algorithm run until convergence.

For each model, we performed filtering[3] for different extents $t_1 = 100, 101, \ldots, 250$, then predicted an image which was a further t_2 steps in the future, for $t_2 = 1, 2..., 100$. The squared error of this prediction in pixel space was recorded, and averaged over all the different filtering extents t_1 to obtain means which are plotted in Figure 2(B). As baselines, we also plot the error obtained by using the mean of filtered data as a predictor (Mean), and the error obtained by using the last filtered observation (Last).

Any of the more complex algorithms perform better than the baselines (though as expected, the 'Last' predictor is a good one-step predictor), indicating that this is a nontrivial prediction problem. The embedded HMM learned by the kernel spectral algorithm yields significantly lower prediction error compared to each of the alternatives (including the recently published RR-HMM) consistently for the duration of the prediction horizon (100 timesteps, *i.e.* 16 seconds).

Slot Car Inertial Measurement. In a second experiment, the setup consisted of a track and a miniature car (1:32 scale model) guided by a slot cut into the track. Figure 3(A) shows the car and the attached IMU (an Intel Inertiadot) in the upper panel, and the 14m track which contains elevation changes and banked curves. At each time step we extracted the estimated 3-D acceleration of the car and the estimated difference between the 3-D orientation of the car from the previous time step at a rate of 10Hz. We collected 3000 successive measurements of this data while the slot car circled the track controlled by a constant policy. The goal was to learn a model of the noisy IMU data, and, after filtering, to predict future readings.

We trained a 20-dimensional embedded HMM using Algorithm 1 with sequences of 150 consecutive observations (Section 3.8). The bandwidth parameter of the Gaussian RBF kernels is set with 'median trick'. The regularization parameter λ is 10^{-4}. For comparison, a 20-dimensional RR-HMM with Parzen windows is learned also with sequences of 150 observations; a 20-dimensional LDS is learned using Subspace ID with Hankel matrices of 150 time steps; and finally, a 20-state discrete HMM (with 400 level of discretization for observations) is learned using EM algorithm.

For each model, we performed filtering for different extents $t_1 = 100, 101, \ldots, 250$, then predicted an image which was a further t_2 steps in the future, for $t_2 = 1, 2..., 100$. The squared error of this prediction in the IMU's measurement space was recorded, and averaged over all the different filtering extents t_1 to obtain means which are plotted in Figure 3(B). Again the embedded HMM yields lower prediction error compared to each of the alternatives consistently for the duration of the prediction horizon.

Audio Event Classification. Our final experiment concerns an audio classification task. The data, recently presented in Ramos et al. (2010), consisted of sequences of 13-dimensional Mel-Frequency Cepstral

[2]Set $N = 50$ in Algorithm 1.

[3]Update models online with incoming observations.

Coefficients (MFCC) obtained from short clips of raw audio data recorded using a portable sensor device. Six classes of labeled audio clips were present in the data, one being Human speech. For this experiment we grouped the latter five classes into a single class of Non-human sounds to formulate a binary Human vs. Non-human classification task. Since the original data had a disproportionately large amount of Human Speech samples, this grouping resulted in a more balanced dataset with 40 minutes 11 seconds of Human and 28 minutes 43 seconds of Non-human audio data. To reduce noise and training time we averaged the data every 100 timesteps (equivalent to 1 second).

For each of the two classes, we trained embedded HMMs with $10, 20, \ldots, 50$ latent dimensions using spectral learning and Gaussian RBF kernels with bandwidth set with the 'median trick'. The regularization parameter λ is 10^{-1}. For comparison, regular HMMs with axis-aligned Gaussian observation models, LDSs and RR-HMMs were trained using multi-restart EM (to avoid local minima), stable Subspace ID and the spectral algorithm of (Siddiqi et al., 2009) respectively, also with $10, \ldots, 50$ latent dimensions.

For RR-HMMs, regular HMMs and LDSs, the class-conditional data sequence likelihood is the scoring function for classification. For embedded HMMs, the scoring function for a test sequence $x_{1:t}$ is the log of the product of the compatibility scores for each observation, *i.e.* $\sum_{\tau=1}^{t} \log\left(\left\langle \varphi(x_\tau), \hat{\mu}_{X_\tau|x_{1:\tau-1}} \right\rangle_{\mathcal{F}}\right)$.

For each model size, we performed 50 random 2:1 partitions of data from each class and used the resulting datasets for training and testing respectively. The mean accuracy and 95% confidence intervals over these 50 randomizations are reported in Figure 4. The graph indicates that embedded HMMs have higher accuracy and lower variance than other standard alternatives at every model size. Though other learning algorithms for HMMs and LDSs exist, our experiment shows this to be a non-trivial sequence classification problem where embedded HMMs significantly outperform commonly used sequential models trained using typical learning and model selection methods.

5. Conclusion

We proposed a Hilbert space embedding of HMMs that extends traditional HMMs to structured and non-Gaussian continuous observation distributions. The essence of this new approach is to represent distributions as elements in Hilbert spaces, and update these elements entirely in the Hilbert spaces using operators. This allows us to derive a local-minimum-free kernel spectral algorithm for learning the embedded HMMs, which exceeds previous state-of-the-art in real world challenging problems. We believe that this new way of combining kernel methods and graphical models can potentially solve many other difficult problems in graphical models and advance kernel methods to more structured territory.

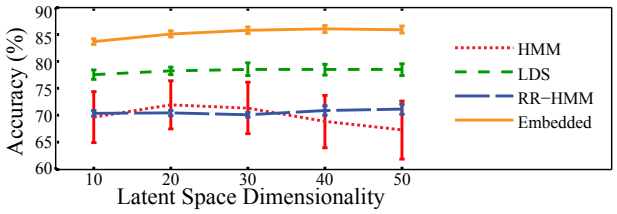

Figure 4. Accuracies and 95% confidence intervals for Human vs. Non-human audio event classification, comparing embedded HMMs to other common sequential models at different latent state space sizes.

Acknowledgement

LS is supported by a Ray and Stephenie Lane fellowship. SMS was supported by the NSF under grant number 0000164, by the USAF under grant number FA8650-05-C-7264, by the USDA under grant number 4400161514, and by a project with MobileFusion/TTC. BB was supported by the NSF under grant number EEEC-0540865. BB and GJG were supported by ONR MURI grant number N00014-09-1-1052.

References

Baker, C. (1973). Joint measures and cross-covariance operators. *Trans. A.M.S.*, *186*, 273–289.

Cortes, C., Mohri, M., & Weston, J. (2005). A general regression techinque for learning transductions. In *ICML*.

Hsu, D., Kakade, S., & Zhang, T. (2009). A spectral algorithm for learning hidden markov models. In *COLT*.

Jaeger, H. (2000). Observable operator models for discrete stochastic time series. *Neural Computation*, *12*(6), 1371–1398.

Ramos, J., Siddiqi, S., Dubrawski, A., Gordon, G., & Sharma, A. (2010). Automatic state discovery for unstructured audio scene classification. In *ICASSP*.

Siddiqi, S., Boots, B., & Gordon, G. (2008). A constraint generation approach to learning stable linear dynamical systems. In *NIPS*.

Siddiqi, S., Boots, B., & Gordon, G. (2009). Reduced-rank hidden markov models. *http://arxiv.org/abs/0910.0902*.

Smola, A., Gretton, A., Song, L., & Schölkopf, B. (2007). A Hilbert space embedding for distributions. In *ALT*.

Song, L., Huang, J., Smola, A., & Fukumizu, K. (2009). Hilbert space embeddings of conditional distributions. In *ICML*.

Sriperumbudur, B., Gretton, A., Fukumizu, K., Lanckriet, G., & Schölkopf, B. (2008). Injective Hilbert space embeddings of probability measures. In *COLT*.

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