

---

# Spectral Learning of Hidden Markov Models from Dynamic and Static Data

---

**Tzu-Kuo Huang**

Machine Learning Department, Carnegie Mellon University

TZUKUOH@CS.CMU.EDU

**Jeff Schneider**

Robotics Institute, Carnegie Mellon University

SCHNEIDE@CS.CMU.EDU

## Abstract

We develop spectral learning algorithms for Hidden Markov Models that learn not only from time series, or *dynamic data* but also *static data* drawn independently from the HMM's stationary distribution. This is motivated by the fact that static, orderless snapshots are usually easier to obtain than time series in quite a few dynamic modeling tasks. Building on existing spectral learning algorithms, our methods solve convex optimization problems minimizing squared loss on the dynamic data plus a regularization term on the static data. Experiments on synthetic and real human activities data demonstrate better prediction by the proposed method than existing spectral algorithms.

## 1. Introduction and Related Work

Hidden Markov Models (HMMs) (Rabiner, 1989) are a useful class of tools for analyzing time series data whose dynamic behavior depends on some unobserved variables, referred to as hidden states, and have found many applications. Due to the hidden states, the widely-used Expectation-Maximization (EM) based estimation has long suffered from ambiguities caused by highly multi-modal estimation objectives. Recently there has been an emerging line of work that proposes spectral algorithms for learning HMMs with discrete (Hsu et al., 2009; Siddiqi et al., 2010) and continuous observations (Siddiqi et al., 2010; Song et al., 2010). In contrast to EM, these algorithms give *unique* and, under mild conditions, provably consistent estimates of the full joint distribution of an observation

sequence, as well as the predictive distribution. They have also been shown to outperform EM-based learning methods in some challenging dynamic modelling tasks (Song et al., 2010).

While good models and learning algorithms play a crucial role in time series analysis, a major challenge in quite a few scientific dynamic modeling tasks, as pointed out by Huang and Schneider (2011), turns out to be collecting reliable time series data. In some situations, the dynamic process of interest may evolve slowly over time, such as the progression of Alzheimer's disease, and it may take months or even years to obtain enough time series data for analysis. In other situations, it may be very difficult to measure the dynamic process of interest repetitively, due to the destructive nature of the measurement technique. One such example is gene expression time series. Although obtaining reliable time series, or dynamic data, can be difficult, it is often easier to collect static, orderless snapshots of the dynamic process of interest. For example, doctors can collect many samples from a current pool of Alzheimer's patients in possibly different stages of the disease, and scientists can easily obtain large amounts of static gene expression data from multiple experiments. Huang and Schneider (2011) propose an estimator for the vector autoregressive (VAR) model that uses *both dynamic and static data*, and derive a simple gradient-descent algorithm to minimize its non-convex estimation objective. Through simulations and experiments on video data, they demonstrate that static data does help to improve estimation, especially when the amount of dynamic data is small.

We propose to incorporate static data into spectral learning algorithms for HMMs, following a similar principle: minimizing a squared *error* function on the *dynamic data* augmented with a *regularization* term based on *static data*. Somewhat surprisingly, the proposed optimization problems for estimation turn out to

be convex, thanks to the unique estimates from spectral algorithms. We conduct simulations and experiments on real Inertia-Measurement Unit recordings of human activities, and demonstrate that, as with VARs, static data also improves estimation of HMMs.

Unlike most spectral algorithms which rely only on Singular Value Decomposition (SVD), our method uses both SVD and convex optimization. Similar ideas have been proposed recently. Among others, Balle et al. (2012) solve a convex program in place of SVD, while Balle and Mohri (2012) use convex optimization to obtain input matrices to spectral algorithms.

We organize the paper as follows. Section 2 briefly reviews spectral learning algorithms, and Section 3 details the proposed algorithms, followed by experiments and results in Section 4 and conclusions in Section 5.

## 2. Spectral Learning of HMMs

We begin with discrete observations, and mainly follow the exposition by Siddiqi et al. (2010). Instead of learning the original model parameters, i.e., initial state probabilities, state transition probabilities, and state-conditioned observation probabilities, the spectral algorithm learns an *observable representation* of the HMM, which consists of the following parameters:

$$\mathbf{b}_1 := U^\top \mathbf{p}, \quad (1)$$

$$\mathbf{b}_\infty := (P_{2,1}^\top U)^\dagger \mathbf{p}, \quad (2)$$

$$B_x := (U^\top P_{3,x,1})(U^\top P_{2,1})^\dagger, \quad 1 \leq x \leq N, \quad (3)$$

where  $\dagger$  denotes the pseudo inverse,  $N$  is the number of observation symbols,  $\mathbf{p}$  is the stationary distribution of observations, and  $P_{2,1}$  and  $P_{3,x,1}$  are joint observation probability matrices such that for  $1 \leq i, x, j \leq N$ ,

$$(P_{2,1})_{ij} := \text{Prob}(x_{t+1} = i, x_t = j), \quad (4)$$

$$(P_{3,x,1})_{ij} := \text{Prob}(x_{t+1} = i, x_t = x, x_{t-1} = j),$$

$x_t$  being the observation symbol at time  $t$ , and  $U \in \mathbb{R}^{N \times k}$  is column concatenation of the top  $k$  left singular vectors of  $P_{2,1}$ . As the name suggests, the observable representation parameters (1) to (3) only depend on observable quantities, leading naturally to the estimates  $\widehat{\mathbf{b}}_1, \widehat{\mathbf{b}}_\infty$ , and  $\widehat{B}_x$  based on empirical averages  $\widehat{\mathbf{p}}, \widehat{P}_{2,1}, \widehat{P}_{3,x,1}$ , and  $\widehat{U}$ , the top- $k$  left singular vectors of  $\widehat{P}_{2,1}$ . These estimates allow us to perform inferences on a new sequence of observations  $y_1, \dots, y_t$ :

- Predict whole sequence probability:

$$\widehat{\text{Prob}}(y_1, \dots, y_t) = \widehat{\mathbf{b}}_\infty^\top \widehat{B}_{y_t} \cdots \widehat{B}_{y_1} \widehat{\mathbf{b}}_1. \quad (5)$$

- Internal state update:  $\widehat{\mathbf{b}}_{t+1} := \widehat{B}_{y_t} \widehat{\mathbf{b}}_t / (\widehat{\mathbf{b}}_\infty^\top \widehat{B}_{y_t} \widehat{\mathbf{b}}_t)$ .

- Conditional probability of  $y_t$  given  $y_1, \dots, y_{t-1}$ :

$$\widehat{\text{Prob}}(y_t | y_1, \dots, y_{t-1}) := \frac{\widehat{\mathbf{b}}_\infty^\top \widehat{B}_{y_t} \widehat{\mathbf{b}}_t}{\sum_x \widehat{\mathbf{b}}_\infty^\top \widehat{B}_x \widehat{\mathbf{b}}_t}. \quad (6)$$

Under some mild conditions, of which the most critical being that both the state transition and state-conditioned observation probability matrices are of rank  $k$ , Siddiqi et al. (2010) showed that the whole sequence probability estimate (5) is consistent (with high probability) and gives a finite-sample bound on the estimation error.

Based on the same idea, Song et al. (2010) developed a spectral algorithm for learning HMMs with continuous observations. Instead of operating on probability distributions directly, their algorithm operates on *Hilbert space embeddings* of distributions of observable quantities (assuming stationarity of the HMM):

$$\mu_1 := \mathbb{E}_{\mathbf{x}_t}[\phi(\mathbf{x}_t)], \quad (7)$$

$$\mathcal{C}_{2,1} := \mathbb{E}_{\mathbf{x}_{t+1}\mathbf{x}_t}[\phi(\mathbf{x}_{t+1}) \otimes \phi(\mathbf{x}_t)], \quad (8)$$

$$\begin{aligned} \mathcal{C}_{3,x,1} &:= \mathbb{E}_{\mathbf{x}_{t+2}(\mathbf{x}_{t+1}=\mathbf{x})\mathbf{x}_t}[\phi(\mathbf{x}_{t+2}) \otimes \phi(\mathbf{x}_t)] \\ &= \mathbb{P}(\mathbf{x}_t = \mathbf{x}) \mathcal{C}_{3,1|2} \phi(\mathbf{x}), \end{aligned} \quad (9)$$

where  $\mathbf{x}_t$  denotes the continuous observation vector at time  $t$ ,  $\phi(\cdot)$  maps the real observation space to a Reproducing Kernel Hilbert Space (RKHS),  $\otimes$  denotes the tensor product, and  $\mathcal{C}_{3,1|2} := \mathcal{C}_{\mathbf{x}_{t+2}\mathbf{x}_t|\mathbf{x}_{t+1}}$  is a *conditional embedding operator* (Song et al., 2009). Using these embeddings, they derived an observable representation of the embedded HMM, which consists of the following parameters:

$$\beta_1 := U^\top \mu_1, \quad (10)$$

$$\beta_\infty := \mathcal{C}_{2,1}(U^\top \mathcal{C}_{2,1})^\dagger, \quad (11)$$

$$\mathcal{B}_x := (U^\top \mathcal{C}_{3,x,1})(U^\top \mathcal{C}_{2,1})^\dagger, \quad (12)$$

where  $U$  is the top- $k$  left singular vectors of  $\mathcal{C}_{2,1}$ . They then showed that the embedding of the predictive distribution  $\mathbb{P}(\mathbf{x}_t | \mathbf{x}_1, \dots, \mathbf{x}_{t-1})$  takes the form  $\mu_{\mathbf{x}_t | \mathbf{x}_1, \dots, \mathbf{x}_{t-1}} = \beta_\infty \mathcal{B}_{\mathbf{x}_1} \cdots \mathcal{B}_{\mathbf{x}_{t-1}} \beta_1$  and, as in the case of discrete observations, proposed estimates based on empirical averages  $\widehat{\mu}_1, \widehat{\mathcal{C}}_{2,1}, \widehat{\mathcal{C}}_{3,x,1}$ , and  $\widehat{U}$ , which is the top- $k$  left singular vectors of  $\widehat{\mathcal{C}}_{2,1}$ . Using the kernel trick and techniques from Kernel Principle Component Analysis (Schölkopf et al., 1998), they gave an estimation procedure that operates solely on finite-dimensional quantities. Moreover, to avoid the difficulty of partitioning the observation space required by estimation of  $\mathcal{B}_x$ , they proposed to estimate instead

$$\widehat{\mathcal{B}}_x := (U^\top \mathcal{C}_{3,1|2} \phi(\mathbf{x}))(U^\top \mathcal{C}_{2,1})^\dagger, \quad (13)$$

which is only a fixed multiplicative factor  $\mathbb{P}(\mathbf{x})$  away from  $\mathcal{B}_x$ , and have  $\mu_{\mathbf{x}_t | \mathbf{x}_1, \dots, \mathbf{x}_{t-1}}$  proportional to

$\beta_\infty \bar{\mathbf{B}}_{\mathbf{x}_1} \cdots \bar{\mathbf{B}}_{\mathbf{x}_{t-1}} \beta_1$ . Under some mild conditions, they established the consistency (with high probability) of their estimator for  $\mu_{\mathbf{x}_t | \mathbf{x}_1, \dots, \mathbf{x}_{t-1}}$  and gave a finite-sample bound on the estimation error.

In addition to estimation, Song et al. (2010) also discussed possible ways to carry out prediction. In particular, they showed that in the case of Gaussian RBF kernel,  $\hat{\mu}_{\mathbf{x}_t | \mathbf{x}_1, \dots, \mathbf{x}_{t-1}}$  takes the form of a nonparametric density estimator after proper normalization, and one may choose, from training data or a pool of samples, the observation with the highest predictive density as the prediction.

### 3. Spectral Learning of HMMs from Dynamic and Static Data

Suppose in addition to dynamic data, which can be time series of observations or triples of consecutive observations, we also have a set of *static data points*, which are drawn *independently* from the stationary distribution of the underlying HMM. We propose to improve the estimation of the observable representation of HMMs by solving regularized least square problems, which minimize a squared error term on the dynamic data *and* a regularization term based on the static data. As in existing work on spectral learning of HMMs, we assume that the dynamic data are observed after the HMM has fully mixed.

#### 3.1. Discrete Observations

Our method has two main steps. We first estimate  $P_{2,1}$ , and then  $\mathbf{b}_1, \mathbf{b}_\infty$ , and  $B_x$ 's. Let  $N$  denote the number of unique observation symbols. To make use of static data in estimating  $P_{2,1}$ , we note that the marginal of  $P_{2,1}$  is the stationary distribution of the discrete HMM. Moreover, from spectral learning methods we have the assumption of  $P_{2,1}$  being low-rank. We thus propose the following estimator  $\hat{P}_{2,1}$  defined as

$$\begin{aligned} \arg \min_P \quad & \frac{1}{2} \|W \odot (P - \hat{P}_{2,1})\|_F^2 + \tau \|P\|_* + \\ & \frac{u}{2} \left( \|\tilde{\mathbf{p}} - P\mathbf{1}\|_2^2 + \|\tilde{\mathbf{p}} - P^\top \mathbf{1}\|_2^2 \right), \quad (14) \\ \text{s.t.} \quad & \mathbf{1}^\top P \mathbf{1} = 1, P_{ij} \geq 0, \end{aligned}$$

where  $\tilde{\mathbf{p}}$  is the empirical observation distribution of *both the dynamic and the static data*,  $W$  is an indicator matrix such that  $W_{ij} = 1 \iff (\hat{P}_{2,1})_{ij} > 0$ ,  $\odot$  denotes the Hadamard product,  $\|\cdot\|_*$  denotes the matrix nuclear norm, a standard convex relaxation of matrix rank,  $\mathbf{1}$  is a vector of ones, and  $u, \tau > 0$  are regularization parameters. The objective in (14) minimizes the squared error from the dynamic-only estimate  $\hat{P}_{2,1}$  while penalizing the rank and the deviation

from the marginal  $\tilde{\mathbf{p}}$ . It is easy to see that (14) is a convex but non-smooth problem due to the matrix nuclear norm. Projected sub-gradient descent methods are a common way to solve such problems, but are known to suffer from slow convergence (Bertsekas, 1999). We solve (14) by a variant of the smoothing proximal gradient (SPG) method proposed by Chen et al. (2012), which achieves a provably faster convergence rate than projected sub-gradient methods but has a similar per-iteration time complexity. In Section 3.2 we use SPG to solve the continuous version of the estimation problem, which has a more general form, and hence describe more details there.

To set  $\tau$  in the right scale, we use the following fact about matrix norms:

$$\|P_{2,1}\|_*/N \leq (r/N) \sqrt{\|P_{2,1}\|_\infty \|P_{2,1}\|_1}, \quad (15)$$

where  $r$  is the rank of  $P_{2,1}$ , and  $\|\cdot\|_\infty$  and  $\|\cdot\|_1$  denote matrix  $\infty$ -norm and 1-norm, respectively. Assuming stationarity, we have  $\|P_{2,1}\|_\infty = \|P_{2,1}\|_1 = \max_i \mathbf{p}_i$ , where  $\mathbf{p}$  is the stationary distribution of observations. Therefore,  $P_{2,1}$ 's average singular value is  $O((\max_i \mathbf{p}_i)/N)$ . As shown by Cai et al. (2010),  $\tau$  has an effect of soft-thresholding singular values of  $P_{2,1}$ , so we let  $\tau = \lambda \max_i \tilde{\mathbf{p}}_i / N$  and tune  $\lambda$  instead.

We then compute the SVD of  $\tilde{P}_{2,1}$ , denoting its top- $k$  left singular vectors as an  $N$ -by- $k$  matrix  $\tilde{U}$ , and obtain estimates of  $\mathbf{b}_1$  and  $\mathbf{b}_\infty$  in the same ways as (1) and (2) using  $\tilde{P}_{2,1}, \tilde{U}$ , and  $\tilde{\mathbf{p}}$ . To derive our estimator of  $B_x$ , we first note that the original estimator based on (3) is the solution to the following problem:

$$\hat{B}_x := \arg \min_B \|\hat{P}_{3,x,1} - \hat{U} B \hat{U}^\top \hat{P}_{2,1}\|_F^2, \quad (16)$$

showing that  $\hat{B}_x$  is a low-dimensional representation of  $\hat{P}_{3,x,1}$ . As in (14), we aim to regularize the least-square problem (16) with static data. Instead of constructing a regularization term directly from static data, we use our new estimator  $\tilde{P}_{2,1}$  based on the fact that  $(\mathbf{1}^\top P_{3,x,1})_j = (P_{2,1})_{xj}$  and  $(P_{3,x,1} \mathbf{1})_i = (P_{2,1})_{ix}$ , i.e., the marginals of  $\{P_{3,x,1}\}$  are equal to  $P_{2,1}$ . We thus propose the following estimator  $\{\tilde{B}_x\}$  defined as

$$\begin{aligned} \arg \min_{\{B_x\}} \sum_x \frac{1}{2} \|W_x \odot (\tilde{U} B_x \tilde{V}^\top - \tilde{P}_{3,x,1})\|_F^2 + \\ \frac{u}{2} \sum_{x,i} \left( (\tilde{P}_{2,1})_{ix} - (\tilde{U} B_x \tilde{V}^\top \mathbf{1})_i \right)^2 + \\ \frac{u}{2} \sum_{x,i} \left( (\tilde{P}_{2,1})_{xi} - (\mathbf{1}^\top \tilde{U} B_x \tilde{V}^\top)_i \right)^2, \quad (17) \\ \text{s.t.} \quad (\tilde{U} B_x \tilde{V}^\top)_{ij} \geq 0, \sum_x \mathbf{1}^\top \tilde{U} B_x \tilde{V}^\top \mathbf{1} = 1, \end{aligned}$$

where  $W_x$  is an indicator matrix such that  $(W_x)_{ij} > 0 \iff (\hat{P}_{3,x,1})_{ij} > 0$  and  $\tilde{V} := \tilde{U}^\top \tilde{P}_{2,1}$ . Note that we not only add regularization terms but also constrain the fitted matrices  $\{\tilde{U}B_x\tilde{V}^\top\}$  to lie on a simplex<sup>1</sup>, aiming to reduce negative values in the predictive distribution (6) during inference.

Eq. (17) is a quadratic program of  $k^2N$  variables under one linear equality constraint and  $N^3$  linear inequality constraints. When  $N$  is on the order of a few hundreds and  $k$  is a few tens, a reformulation that takes advantage of the block-diagonal structure in the Hessian of (17) can be solved quite efficiently with state-of-the-art optimization software, such as MOSEK ([www.mosek.com](http://www.mosek.com)). For larger problems, one possible solution is the Alternating Direction Method of Multipliers (Boyd et al., 2011), which handles constraints by minimizing the original objective augmented with a iteratively-refined constraint violation term. Our experiments in Section 4.1 have  $N = 100$ , so we solve (17) with MOSEK.

### 3.2. Continuous Observations

Our method for continuous observations builds on the Hilbert space embedding approach by Song et al. (2010), and consists of two main steps: estimating the feature covariance  $\mathcal{C}_{2,1}$  and then the observable representation  $\beta_1, \beta_\infty$ , and  $\mathcal{B}_x$ . Let the feature mappings of the dynamic data be organized into three matrices  $\Phi_1, \Phi_2$ , and  $\Phi_3$  such that their  $i$ -th columns  $\Phi_1^i, \Phi_2^i$ , and  $\Phi_3^i$  are consecutive and going forward in time. By the definition of the feature covariance (8), we have  $\mathcal{C}_{2,1} := \int \phi(\mathbf{x}) \otimes \phi(\mathbf{y}) p_{X_{t+1}X_t}(\mathbf{x}, \mathbf{y}) dx dy$ . If we have a set of feature points grouped column-wise as a feature matrix  $\Phi$ , and know exactly which pairs of points are consecutive in time via a (normalized) temporal adjacency matrix  $T_{2,1}$ , we may then compute the quantity  $\Phi T_{2,1} \Phi^\top$  as an unbiased estimator of  $\mathcal{C}_{2,1}$ . It is easy to see that  $\hat{\mathcal{C}}_{2,1} := \frac{1}{n} \Phi_2 \Phi_1^\top$  is one special case of such an estimator. To incorporate static data into our estimation procedure, we denote its feature matrix by  $\mathcal{Z}$  and consider another special case:

$$\tilde{\mathcal{C}}_{2,1} := \mathcal{Z}_2 P \mathcal{Z}_1^\top, \quad (18)$$

where  $\mathcal{Z}_1 := [\Phi_1 \ \mathcal{Z}]$  and  $\mathcal{Z}_2 := [\Phi_2 \ \mathcal{Z}]$ . It then suffices to estimate  $P$  subject to  $\mathbf{1}^\top P \mathbf{1} = 1$  and  $P_{ij} \geq 0$ .

Similar to Section 3.1, our estimation objective consists of three terms: the squared error between  $\hat{\mathcal{C}}_{2,1}$  and  $\tilde{\mathcal{C}}_{2,1}$ , penalization on  $\tilde{\mathcal{C}}_{2,1}$ 's rank, and deviation of

$\tilde{\mathcal{C}}_{2,1}$ 's marginal from the mean of the stationary distribution. The last term is based on the fact that, under the assumption of stationarity,  $\mathcal{C}_{2,1} \mathbf{f} = \mathbb{E}[\phi(X)]$  holds for some constant function  $\mathbf{f}$  in  $\mathcal{G}$  such that  $\mathbf{f}(\mathbf{x}) = \mathbf{f}^\top \phi(\mathbf{x}) = 1 \ \forall \mathbf{x}$ . Formally, our estimator  $\tilde{P}$  is the solution to the following convex program:

$$\begin{aligned} \min_P \quad & \frac{1}{2} \|\mathcal{Z}_2 P \mathcal{Z}_1^\top - \hat{\mathcal{C}}_{2,1}\|_{\mathcal{G} \otimes \mathcal{G}}^2 + \tau \|\mathcal{Z}_2 P \mathcal{Z}_1^\top\|_* + \\ & \frac{u}{2} \left( \left\| \mathcal{Z}_2 P \mathbf{1} - \frac{\mathcal{S} \mathbf{1}}{m_S} \right\|_{\mathcal{G}}^2 + \left\| \mathcal{Z}_1 P^\top \mathbf{1} - \frac{\mathcal{S} \mathbf{1}}{m_S} \right\|_{\mathcal{G}}^2 \right) \quad (19) \\ \text{s.t.} \quad & \mathbf{1}^\top P \mathbf{1} = 1, P_{ij} \geq 0, \end{aligned}$$

where we introduce  $\mathcal{S}$  and  $m_S$  to denote the feature matrix and the size of the entire set of static data and let  $\mathcal{Z}$  denote a sub-sample of it, mainly to limit the number of variables when the static dataset is very large. As shown in Appendix A, using the kernel trick and properties of the matrix trace and nuclear norm, we re-write the objective function in (19) as follows (dropping constants):

$$\begin{aligned} & \frac{1}{2} \text{Tr}(P^\top M_2 P M_1) - \text{Tr}(P^\top F) + \tau \|L_2^\top P L_1\|_* + \quad (20) \\ & \frac{u}{2} \mathbf{1}^\top (P^\top M_2 P + P M_1 P^\top) \mathbf{1} - u \mathbf{1}^\top (P^\top \boldsymbol{\mu}_2 + P \boldsymbol{\mu}_1), \end{aligned}$$

where  $\text{Tr}(\cdot)$  is the matrix trace,  $M_i := \mathcal{Z}_i^\top \mathcal{Z}_i$ ,  $\boldsymbol{\mu}_i := \frac{\mathcal{Z}_i^\top \mathcal{S} \mathbf{1}}{m_S}$ ,  $F := \mathcal{Z}_2^\top \hat{\mathcal{C}}_{2,1} \mathcal{Z}_1$ , and  $L_i$  is a finite matrix such that  $M_i = L_i L_i^\top$ . To set  $\tau$  in a proper scale, we use an inequality similar to (15) to upper-bound the average singular value of  $L_2^\top P L_1$ , and then replace the unknown  $P$  by the uniform distribution to have  $\tau := (\lambda/m^3) (\|L_2^\top \mathbf{1} \mathbf{1}^\top L_1\|_\infty \|L_2^\top \mathbf{1} \mathbf{1}^\top L_1\|_1)^{1/2}$ , where  $m$  is the size of  $P$  and  $\lambda > 0$  takes values in some reasonable range.

As mentioned in Section 3.1, we solve (19) with a variant of the smoothing proximal gradient (SPG) method outlined in Algorithm 1, which minimizes  $f_\mu(P)$ , a smooth approximation of (20) that approximates the non-smooth regularization  $\tau \|L_2^\top P L_1\|_*$  by

$$g_\mu(P) := \max_{\|Y\|_2 \leq 1} \tau \text{Tr}(Y^\top L_2^\top P L_1) - \frac{\mu}{2} \|Y\|_F^2, \quad (21)$$

where  $\mu \geq 0$  is a smoothing parameter,  $\|\cdot\|_2$  and  $\|\cdot\|_F$  denote the matrix spectral and Frobenius norms, respectively. Nesterov (2005) shows that (21) is continuously differentiable in  $P$  and  $\nabla g_\mu(P) = \tau L_2 Y^* L_1^\top$ , where  $Y^*$  is the optimal solution to (21) obtained by projecting  $(\tau/\mu) L_2^\top P L_1$  to the unit spectral-norm ball, i.e., truncating its singular values at 1. The update (22) for  $P^{(t+1)}$  requires projection onto a simplex, for which several efficient algorithms exist, such as the sorting-based method proposed by Duchi et al. (2008).

<sup>1</sup>These constraints may be infeasible if  $k$  is too small, but in our experiments we did not encounter this issue. When this happens one may choose the smallest  $k$  that makes the constraints feasible, and then solve (17).

**Algorithm 1** Smoothing Proximal Gradient for (19)

 Initialize  $Y^{(0)} = P^{(0)}$  to some feasible point.

 Set  $t := 0, \theta^{(0)} := 1, \eta := 10$ , and  $\gamma^{(0)} := 1$ .

**repeat**

 Find the smallest  $\kappa \in \{0, 1, \dots\}$  that satisfies

$$f_\mu(P^{(t+1)}) - f_\mu(Y^{(t)}) \leq \frac{\gamma^{(t+1)}}{2} \|P^{(t+1)} - Y^{(t)}\|_F^2 \\ + \text{Tr}((P^{(t+1)} - Y^{(t)})^\top \nabla f_\mu(Y^{(t)}))$$

 where  $\gamma^{(t+1)} := \eta^\kappa \gamma^{(t)}$  and

$$P^{(t+1)} := \arg \min_P \|Y^{(t)} - \nabla f_\mu(Y^{(t)})/\gamma^{(t+1)} - P\|_F^2 \\ \text{s.t. } P_{ij} \geq 0, \mathbf{1}^\top P \mathbf{1} = 1. \quad (22)$$

$$\theta^{(t+1)} := (1 + \sqrt{1 + 4(\theta^{(t)})^2})/2.$$

$$Y^{(t+1)} := P^{(t+1)} + \frac{\theta^{(t)} - 1}{\theta^{(t+1)}} (P^{(t+1)} - P^{(t)}).$$

 $t := t + 1$ .

**until** convergence or  $t = T_{\max}$ , an iteration limit.

The convergence theory of Chen et al. (2012) suggests setting<sup>2</sup>  $\mu = \epsilon/m$ ,  $m$  being the column dimension of  $\mathcal{Z}_2$ , so that the objective values (20) converge in  $O(1/\epsilon^2)$  iterations to at most  $\epsilon$  plus the minimum.

We then compute the top  $k$  left singular vectors of  $\tilde{\mathcal{C}}_{2,1}$  in a similar way to Kernel Principle Component Analysis (Schölkopf et al., 1998), starting with the fact that any left singular vector of  $\tilde{\mathcal{C}}_{2,1} = \mathcal{Z}_2 \tilde{P} \mathcal{Z}_1^\top$  can be expressed as  $\mathcal{Z}_2 \alpha$  for some  $\alpha \in \mathbb{R}^m$ , and any left singular vector of  $\tilde{\mathcal{C}}_{2,1}$  is an Eigenvector of  $\tilde{\mathcal{C}}_{2,1} \tilde{\mathcal{C}}_{2,1}^\top$  and vice versa. Thus we have

$$\mathcal{Z}_2 \tilde{P} M_1 \tilde{P}^\top M_2 \alpha = \mathcal{Z}_2 \tilde{P} \mathcal{Z}_1^\top \mathcal{Z}_1 \tilde{P}^\top \mathcal{Z}_2^\top (\mathcal{Z}_2 \alpha) = \omega \mathcal{Z}_2 \alpha \\ \iff M_2 \tilde{P} M_1 \tilde{P}^\top M_2 \alpha = \omega M_2 \alpha, \quad (23)$$

which is a generalized Eigensystem. Let  $\Omega$  denote the diagonal matrix formed by the top  $k$  generalized Eigenvalues of (23), and  $A$  denote the column concatenation of the corresponding generalized Eigenvectors. It is then clear that  $D := (A^\top M_2 A)^{-1/2}$  is diagonal, and we obtain a concise form of  $\tilde{\mathcal{C}}_{2,1}$ 's top  $k$  left singular vectors  $\tilde{U} = \mathcal{Z}_2 A D$ . We also have the following useful identity:

$$M_2 \tilde{P} M_1 \tilde{P}^\top M_2 A = M_2 A \Omega. \quad (24)$$

Next we describe our estimators for the observable rep-

<sup>2</sup>For solving (14) we set  $\mu = \epsilon/N$ .

resentation. First we have

$$\tilde{\beta}_1 := \tilde{U}^\top \mathcal{S} \mathbf{1} / m_S = D A^\top \mu_2, \quad (25)$$

$$\tilde{\beta}_\infty := \tilde{\mathcal{C}}_{2,1} (\tilde{U}^\top \tilde{\mathcal{C}}_{2,1})^\dagger = \mathcal{Z}_2 \tilde{P} M_1 \tilde{P}^\top M_2 A D \Omega^{-1} \quad (26)$$

by using the identity  $(\tilde{U}^\top \tilde{\mathcal{C}}_{2,1})^\dagger = \mathcal{Z}_1 \tilde{P}^\top M_2 A D \Omega^{-1}$  established from properties of pseudo inverse, (24), and the definition of  $D$ . To derive our estimator for  $\tilde{\beta}_x$  defined in (13), we start from the conditional covariance operator defined by Song et al. (2009)

$$\mathcal{C}_{3,1|2} := \mathcal{C}_{3,1,2} \mathcal{C}_{2,2}^{-1} \phi(\mathbf{x}), \quad \text{where}$$

$$\mathcal{C}_{3,1,2} := \mathbb{E}_{X_{t+2} X_t X_{t+1}} [\phi(X_{t+2}) \otimes \phi(X_t) \otimes \phi(X_{t+1})],$$

$$\mathcal{C}_{2,2} := \mathbb{E}_{X_{t+1}} [\phi(X_{t+1}) \otimes \phi(X_{t+1})].$$

Using a similar idea to (18), we encode the empirical distribution of triples of consecutive observations by a third-order tensor  $Q$  and have the following estimator

$$\tilde{\mathcal{C}}_{3,1|2} := \left( \sum_{i,j,l} Q_{ijl} \mathcal{Z}_3^i \otimes \mathcal{Z}_1^j \otimes \mathcal{Z}_2^l \right) \left( \frac{1}{m} \mathcal{Z}_2 \mathcal{Z}_2^\top + \nu I \right)^{-1},$$

where  $\mathcal{Z}_3 := [\Phi_3 \mathcal{Z}]$ ,  $\nu > 0$  is a regularization parameter, and superscripts denote column indices. We then define our estimator for  $\tilde{\beta}_x$  as

$$\tilde{\beta}_x := (\tilde{U}^\top (\tilde{\mathcal{C}}_{3,1|2} \phi(\mathbf{x}))) (\tilde{U}^\top \tilde{\mathcal{C}}_{2,1})^\dagger \quad (27)$$

$$= m \sum_l B_l \left( (M_2 + \nu m I)^{-1} \mathcal{Z}_2^\top \phi(\mathbf{x}) \right)_l, \quad (28)$$

where  $B_l \in \mathbb{R}^{k \times k}$  is a linear transformation of  $Q_{..l} \in \mathbb{R}^{m \times m}$ , the  $l$ th slice of  $Q$  along the third dimension:

$$B_l := \tilde{U}^\top \mathcal{Z}_3 Q_{..l} \mathcal{Z}_1^\top (\tilde{U}^\top \tilde{\mathcal{C}}_{2,1})^\dagger. \quad (29)$$

Note that in the usual setting of learning from dynamic data, the third-order tensor  $Q$  is diagonal and  $B_l$  becomes a rank-one matrix, so (28) reduces to the estimator proposed by Song et al. (2010).

The definitions above naturally lead to an estimation procedure that first estimates  $Q$  and then applies (29) to estimate  $B_l$ . However, such a procedure involves  $m^3$  variables when the quantities of interest consist of only  $km^2$  variables. We thus propose to estimate  $B_l$ 's directly. Viewing (29) as the solution to

$$\arg \min_{B_l} \|Q_{..l} - \tilde{U} B_l \tilde{V}^\top\|_F^2, \quad \text{where}$$

$$\tilde{U} := (\tilde{U}^\top \mathcal{Z}_3)^\dagger = (D A^\top \mathcal{Z}_2^\top \mathcal{Z}_3)^\dagger = (D A^\top M_{23})^\dagger,$$

$$\tilde{V}^\top := (\mathcal{Z}_1^\top (\tilde{U}^\top \tilde{\mathcal{C}}_{2,1})^\dagger)^\dagger = (M_1 \tilde{P}^\top M_2 A D \Omega^{-1})^\dagger,$$

we propose to estimate  $B_l$ 's by the following:

$$\arg \min_{\{B_l\}} \frac{1}{2} \|\tilde{\mathcal{C}}_{3,1,2}(\{B_l\}) - \hat{\mathcal{C}}_{3,1,2}\|_{\mathcal{G} \otimes \mathcal{G} \otimes \mathcal{G}}^2 + \quad (30)$$

$$\frac{u}{2} \left( \|\tilde{\mathcal{C}}_{3,.,2}(\{B_l\}) - \tilde{\mathcal{C}}_{2,1}\|_{\mathcal{G} \otimes \mathcal{G}}^2 + \|\tilde{\mathcal{C}}_{.,1,2}(\{B_l\})^\top - \tilde{\mathcal{C}}_{2,1}\|_{\mathcal{G} \otimes \mathcal{G}}^2 \right)$$

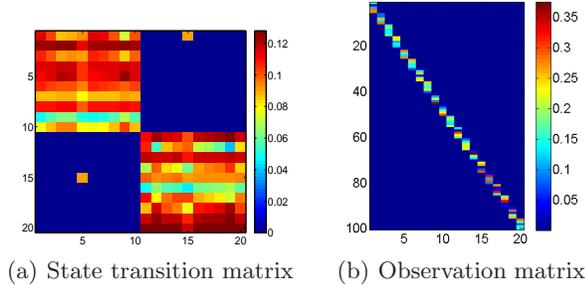


Figure 1. Discrete HMM model parameters

in which

$$\tilde{\mathcal{C}}_{3,1,2}(\{B_l\}) := \sum_{i,j,l} (\tilde{U} B_l \tilde{V}^\top)_{ij} \mathbf{z}_3^i \otimes \mathbf{z}_1^j \otimes \mathbf{z}_2^l, \quad (31)$$

$$\tilde{\mathcal{C}}_{3,\cdot,2}(\{B_l\}) := \sum_{i,j,l} (\tilde{U} B_l \tilde{V}^\top)_{ij} \mathbf{z}_3^i \otimes \mathbf{f}^\top \mathbf{z}_1^j \otimes \mathbf{z}_2^l, \quad (32)$$

$$\tilde{\mathcal{C}}_{\cdot,1,2}(\{B_l\}) := \sum_{i,j,l} (\tilde{U} B_l \tilde{V}^\top)_{ij} \mathbf{f}^\top \mathbf{z}_3^i \otimes \mathbf{z}_1^j \otimes \mathbf{z}_2^l. \quad (33)$$

Again, our estimation objective consists of a squared error term on the observed tri-variance and two regularization terms on the deviation of the marginals  $\tilde{\mathcal{C}}_{3,\cdot,2}$  and  $\tilde{\mathcal{C}}_{\cdot,1,2}$  from our estimated co-variance  $\tilde{\mathcal{C}}_{2,1}$ . As shown in Appendix B, we use kernel tricks to rewrite the objective function (30) in terms of finite-dimensional quantities. Moreover, by re-defining the notation  $B$  to be a  $k^2$ -by- $m$  matrix whose  $l$ -th column denotes the column concatenation of the  $k$ -by- $k$  matrix  $B_l$ , we obtain the following succinct form of (30) (dropping constants):

$$\min_B \frac{1}{2} \text{Tr}(B^\top C B M_2) - \text{Tr}(J^\top B) \quad (34)$$

with an analytical solution  $C^{-1} J M_2^{-1}$ , where  $C$  and  $J$  are defined<sup>3</sup> in Appendix B.

## 4. Experiments

We compare our proposed methods with the original spectral algorithms (Section 2) that only use dynamic data. In the case of discrete observations we conduct a simulation study, and we apply the algorithms for continuous observations to an activity monitoring dataset.

### 4.1. Simulation

We create a discrete HMM with 20 states and 100 observation symbols. The state transition probability matrix is of rank nearly 7. The heatmaps of the

<sup>3</sup>When the kernel is positive definite, it is easy to verify that both  $C$  and  $M_2$  are invertible.

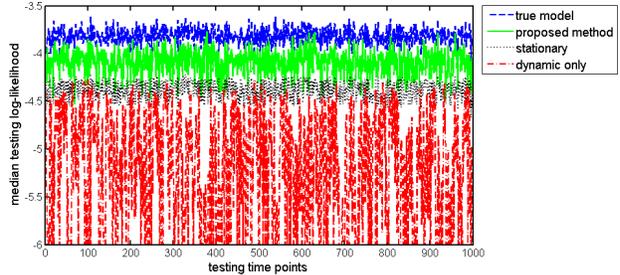


Figure 2. Median testing log-likelihood. The y-axis lower limit is set to -6 for better visualization; the red dashed line actually takes values as small as -17.

state transition probability and the state-conditioned observation probability matrices are in Figures 1(a) and 1(b). From this HMM we generate 50 datasets, each containing a training sequence of length 1000 initialized from the stationary distribution as the dynamic data, a set of 10000 observations independently drawn from the stationary distribution as the static data, and a testing sequence of length 1000, also initialized from the stationary distribution. We set the dimension  $k = 7$ , and for the proposed estimate set  $u = 100$  and  $\lambda = 15$ . We then perform filtering and prediction along the testing sequence. To give bounds on the prediction performance, we also give prediction results by the true observable representation and the stationary distribution.

Figure 2 shows the median testing log-likelihood over 50 experiments at each testing time point. The proposed estimator outperforms the dynamic-only estimator at most time points. For each pair among the four predictions, we performed paired t-tests of their testing likelihoods at all time points, and counted the number of time points at which one prediction outperforms the other statistically significantly. The results are in Table 1. The proposed estimator predicts better than the dynamic-only estimator at all time points and the stationary distribution at many time points, but these two other methods never predict significantly better than the proposed method. It is surprising that the dynamic-only estimator performs even worse than the stationary distribution. As pointed out by Siddiqi et al. (2010), the filtering and prediction steps (6) do not guarantee non-negativity of the probability estimates, especially when, as in the current experiment, there is few dynamic data. Indeed, we observe quite a few negative values in the dynamic-only estimates and replace them with  $10^{-12}$ . This is an indication of unreliable estimates leading to poor prediction. On the contrary, the proposed estimates almost always take non-negative values.

Table 1. Paired t tests results. Each cell shows the number of testing time points at which the row method outperforms the column method statistically significantly. The total number of testing time points is 999.

	true	proposed	dynamic	stationary
true		827	999	975
proposed	0		999	470
dynamic	0	0		0
stationary	0	0	999	

## 4.2. IMU Measurements of Human Activities

The PAMAP2 physical activity monitoring dataset (Reiss & Stricker, 2012) contains recordings of 18 different physical activities performed by 9 subjects wearing 3 inertial measurement units (IMUs) and a heart-rate monitor. Each subject follows a protocol to perform a sequence of activities with breaks in between. For our experiment we use data collected from subject 101 while walking and running. We focus our experiment on recordings from the three IMUs, and for each IMU only use the 3D-acceleration data ( $\text{ms}^{-2}$ ) with scale  $\pm 16\text{g}$ , as recommended by the authors, and the 3D-gyroscope data ( $\text{rad/s}$ ), resulting in an observation space of  $6 \times 3 = 18$  dimensions. Subject 101 performs walking and running for approximately 3.5 minutes each, and we discard the first and the last 10 seconds of data to remove transitioning between activities. To make the experiment more interesting, we break the IMU recordings into short segments of 10 seconds each and interleave the walking segments with the running ones to generate a sequence of alternating activities. The IMUs operate at a sampling frequency of 100Hz, so each segment has 1000 data points and the entire sequence has 39265 data points. We normalize each of the 18 dimensions to be zero-mean and standard deviation 1. Figure 3 shows one of the dimensions from the first 2000 data points, revealing significant differences between walking and running.

We take the last 4256 data points as the testing sequence, and generate 10 training datasets as follows. We randomly sample  $n$  triples of consecutive observations from the first 35000 data points as the dynamic data, and another non-overlapping set of  $m + m_S$  single observations as the static data, in which  $m$  points are used to form  $\mathcal{Z}$  and the rest  $m_S$  points constitute  $\mathcal{S}$  in the proposed algorithm. The values of  $n$ ,  $m$ , and  $m_S$  are:  $n \in \{25, 50, 100, 200\}$ ,  $m \in \{500, 1000\}$ , and  $m_S = 4000$ . We use the Gaussian RBF kernel  $\kappa(\mathbf{x}, \mathbf{x}') := \exp(-\|\mathbf{x} - \mathbf{x}'\|^2/\sigma^2)$ , and set  $\sigma^2$  to be half of the median squared pairwise distances of the dynamic data. The dimension  $k$ , i.e., the number of top left singular vectors, is set to 20 for  $n = 25$  and 50 for the rest. The proposed algorithm has three reg-

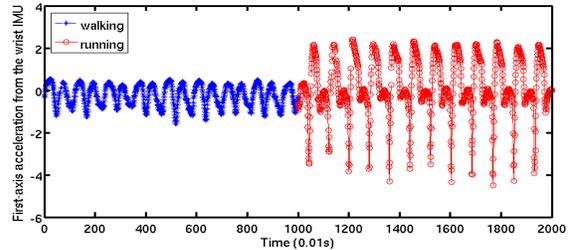


Figure 3. First-axis acceleration from the wrist IMU

ularization parameters:  $u_P$  and  $\lambda$  in (19) and  $u_B$  in (34). We determine these parameters by minimizing 5-fold<sup>4</sup> cross validation error on the dynamic data over a cube of values  $(\log_2 u_P, \log_2 \lambda, \log_2 u_B) \in \{-8, -6, \dots, 6\} \times \{-9, -7, \dots, 1\} \times \{-5, -3, \dots, 9\}$ .

After learning the model parameters, we perform filtering and prediction along the testing sequence. As mentioned in Section 2, the Hilbert space embedding of the predictive distribution takes the form of a non-parametric density estimator thanks to the Gaussian RBF kernel, and we predict the next observation by selecting from  $\mathcal{S}$ , the  $m_S$  static data points, the one with the highest predictive density. For each predicted observation we compute the squared error against the true observation, and for each predicted sequence we take the median and the mean of the squared prediction errors as sequence-wise performance indicators. Figure 4(a) gives the boxplot of the 10 median prediction errors, showing that the proposed method of incorporating static data improves on the prediction performance more significantly when the dynamic data size  $n$  is small. Figure 4(b) gives the boxplot of the 10 means, demonstrating a similar trend of improvement except when  $n = 50$ . Looking more into that result, we find that it is the running part of the testing sequence the proposed method fails to predict better, possibly due to the more extreme values and changes in its IMU readings, as shown in Figure 3.

## 5. Conclusions

We propose spectral learning algorithms for HMMs that incorporate static data as regularization. Experiments on synthetic and real human activities data demonstrate a clear advantage of using static data when dynamic data is limited. Theoretical guarantees for our methods are still unclear and worth further investigation. Also interesting is applying the proposed methods to dynamic modeling tasks where dynamic data is much more difficult to obtain than static data.

<sup>4</sup>We only split the dynamic data but not the static data.

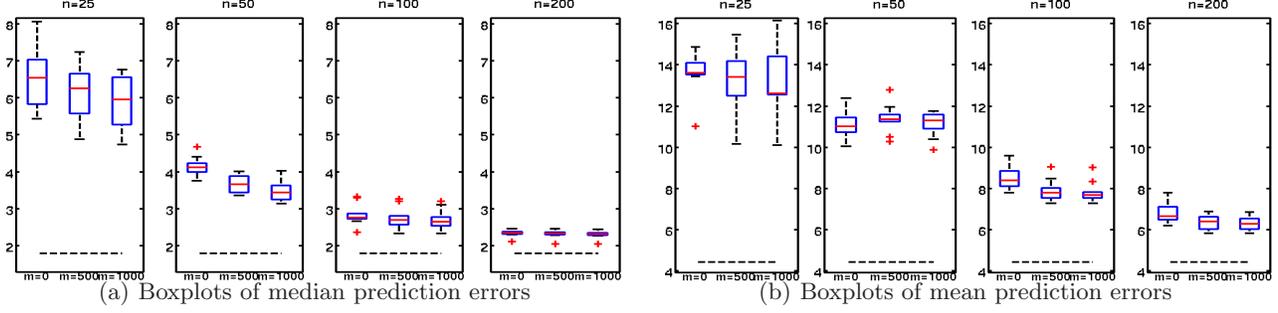


Figure 4. Prediction performance on the IMU data. The black-dashed line is obtained by using  $n = 5000$  dynamic data points, serving as a performance limit.

## A. Derivation of (20)

Using properties of the matrix trace and the kernel trick, we immediately have

$$\begin{aligned} \frac{1}{2} \|\mathcal{Z}_2 P \mathcal{Z}_1^\top - \hat{\mathcal{C}}_{2,1}\|_{\mathcal{G} \otimes \mathcal{G}}^2 &\propto \frac{1}{2} \text{Tr}(P^\top M_2 P M_1) - \text{Tr}(P^\top F), \\ \frac{u}{2} \left( \|\mathcal{Z}_2 P \mathbf{1} - \frac{\mathbf{S} \mathbf{1}}{m_S}\|_{\mathcal{G}}^2 + \|\mathcal{Z}_1 P^\top \mathbf{1} - \frac{\mathbf{S} \mathbf{1}}{m_S}\|_{\mathcal{G}}^2 \right) \\ &\propto \frac{u}{2} \mathbf{1}^\top (P^\top M_2 P + P M_1 P^\top) \mathbf{1} - u \mathbf{1}^\top (P^\top \boldsymbol{\mu}_2 + P \boldsymbol{\mu}_1). \end{aligned}$$

Let  $\lambda_i(\cdot)$  denotes the  $i$ -th Eigenvalue of a matrix. We then rewrite the nuclear norm term:

$$\begin{aligned} \tau \|\mathcal{Z}_2 P \mathcal{Z}_1^\top\|_* &= \tau \sum_i \sqrt{\lambda_i(\mathcal{Z}_2 P L_1^\top L_1 P^\top \mathcal{Z}_2^\top)} \\ &= \tau \sum_i \sqrt{\lambda_i(L_1^\top P^\top L_2 L_2^\top P L_1)} = \tau \|L_2^\top P L_1\|_*, \end{aligned}$$

## B. Derivation of (34)

We begin by defining some notations:

$$\begin{aligned} H &:= \tilde{U}^\top M_3 \tilde{U}, \quad R := \tilde{V}^\top M_1 \tilde{V}, \quad \mathbf{u} := \tilde{U}^\top \mathbf{1}, \quad \mathbf{v} := \tilde{V}^\top \mathbf{1}, \\ F_1 &:= \Phi_1^\top \mathcal{Z}_1 \tilde{V}, \quad F_2 := \frac{\Phi_2^\top \mathcal{Z}_2}{n}, \quad F_3 := \Phi_3^\top \mathcal{Z}_3 \tilde{U}. \end{aligned}$$

Let  $\text{vec}(X)$  be the vector resulting from column concatenation of a matrix  $X$ ,  $\text{diag}(\mathbf{x})$  be the diagonal matrix with the vector  $\mathbf{x}$  being its main diagonal. Superscripts denote column indices. Using properties of the matrix trace and the kernel trick, we re-write the three terms in (34) as follows. For the first term we have

$$\begin{aligned} &\|\tilde{\mathcal{C}}_{3,1,2}(\{B_l\}) - \hat{\mathcal{C}}_{3,1,2}\|_{\mathcal{G} \otimes \mathcal{G} \otimes \mathcal{G}}^2 \\ &\propto \sum_d \text{Tr} \left( \sum_{l,l'} (\mathcal{Z}_2^l)_d (\mathcal{Z}_2^{l'})_d \tilde{V} B_l^\top \tilde{U}^\top M_3 \tilde{U} B_{l'} \tilde{V}^\top M_1 \right) - \\ &\quad 2 \sum_d \text{Tr} \left( \sum_l \tilde{V} B_l^\top \tilde{U}^\top (\mathcal{Z}_2^l)_d \mathcal{Z}_3^\top \Phi_3 \frac{\text{diag}((\Phi_2)_d)}{n} \Phi_1^\top \mathcal{Z}_1 \right) \\ &= \text{Tr} \left( \sum_{ll'} (M_2)_{ll'} B_l^\top H B_{l'} R - 2 \sum_l B_l^\top F_3^\top \text{diag}(F_2^l) F_1 \right), \end{aligned}$$

and then for the second term

$$\begin{aligned} &\|\tilde{\mathcal{C}}_{3,2}(\{B_l\}) - \hat{\mathcal{C}}_{2,1}\|_{\mathcal{G} \otimes \mathcal{G}}^2 \propto \\ &\text{Tr}([B_1 \mathbf{v} \cdots B_m \mathbf{v}]^\top H [B_1 \mathbf{v} \cdots B_m \mathbf{v}] M_2) - \\ &2 \text{Tr}([B_1 \mathbf{v} \cdots B_m \mathbf{v}]^\top \tilde{U}^\top M_{32} \tilde{P} M_{12}) = \\ &\text{Tr} \left( \sum_{il} (M_2)_{il} B_i^\top H B_l \mathbf{v} \mathbf{v}^\top - 2 \sum_i B_i^\top \tilde{U}^\top M_{32} \tilde{P} M_{12}^i \mathbf{v}^\top \right), \end{aligned}$$

and finally for the third term

$$\begin{aligned} &\|\tilde{\mathcal{C}}_{3,1,2}(\{B_l\})^\top - \hat{\mathcal{C}}_{2,1}\|_{\mathcal{G} \otimes \mathcal{G}}^2 \propto \\ &\text{Tr}([B_1^\top \mathbf{u} \cdots B_m^\top \mathbf{u}] M_2 [B_1^\top \mathbf{u} \cdots B_m^\top \mathbf{u}]^\top R) - \\ &2 \text{Tr}([B_1^\top \mathbf{u} \cdots B_m^\top \mathbf{u}] M_2 \tilde{P} M_1 \tilde{V}) = \\ &\text{Tr} \left( \sum_{ij} (M_2)_{ij} B_i^\top \mathbf{u} \mathbf{u}^\top B_j R - 2 \sum_i B_i^\top \mathbf{u} (M_2^i)^\top \tilde{P} M_1 \tilde{V} \right). \end{aligned}$$

To further simplify these expressions, we re-define the notation  $B$  to be a  $k^2$ -by- $m$  matrix whose  $l$ -th column  $B^l$  denotes column concatenation of the  $k$ -by- $k$  matrix  $B_l$  in the above expressions. With the new notation and the identity:

$$\text{vec}(XYZ) = (Z^\top \circ X) \text{vec}(Y) \quad (35)$$

where  $\circ$  denotes the Kronecker product, we obtain the succinct form (34) in which

$$\begin{aligned} C &:= R \circ H + u((\mathbf{v} \mathbf{v}^\top) \circ H + R \circ (\mathbf{u} \mathbf{u}^\top)), \\ J &:= (F_1 \circ F_3)^\top [\text{vec}(\text{diag}(F_2^1)) \cdots \text{vec}(\text{diag}(F_2^m))] \\ &\quad + u \left( (\mathbf{v} \circ (\tilde{U}^\top M_{32} \tilde{P})) M_{12} + ((\tilde{V}^\top M_1 \tilde{P}^\top) \circ \mathbf{u}) M_2 \right). \end{aligned}$$

## References

- Balle, Borja and Mohri, Mehryar. Spectral learning of general weighted automata via constrained matrix completion. In *Advances in Neural Information Processing Systems 25*, pp. 2168–2176, 2012.
- Balle, Borja, Quattoni, Ariadna, and Carreras, Xavier. Local loss optimization in operator models: A new insight into spectral learning. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pp. 1879–1886, 2012.
- Bertsekas, Dimitri P. *Nonlinear Programming*. Athena Scientific, Belmont, MA 02178-9998, second edition, 1999.
- Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.
- Cai, J.F., Candès, E.J., and Shen, Z. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.
- Chen, X., Lin, Q., Kim, S., Carbonell, J.G., and Xing, E.P. Smoothing proximal gradient method for general structured sparse regression. *The Annals of Applied Statistics*, 6(2):719–752, 2012.
- Duchi, John, Shalev-Shwartz, Shai, Singer, Yoram, and Chandra, Tushar. Efficient projections onto the  $\ell_1$ -ball for learning in high dimensions. In *Proceedings of the 25th International Conference on Machine Learning*, pp. 272–279, 2008.
- Hsu, Daniel, Kakade, Sham M., and Zhang, Tong. A spectral algorithm for learning hidden Markov models. In *Proceedings of the Twenty-Second Annual Conference on Learning Theory*, 2009.
- Huang, Tzu-Kuo and Schneider, Jeff. Learning autoregressive models from sequence and non-sequence data. In *Advances in Neural Information Processing Systems 24*, pp. 1548–1556. 2011.
- Nesterov, Y. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005.
- Rabiner, Lawrence R. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, 1989.
- Reiss, A. and Stricker, D. Introducing a new benchmarked dataset for activity monitoring. In *Wearable Computers (ISWC), 2012 16th International Symposium on*, pp. 108–109. IEEE, 2012.
- Schölkopf, B., Smola, A., and Müller, K.R. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.
- Siddiqi, Sajid M., Boots, Byron, and Gordon, Geoffrey J. Reduced-rank hidden Markov models. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, 2010.
- Song, Le, Huang, Jonathan, Smola, Alex, and Fukumizu, Kenji. Hilbert space embeddings of conditional distributions with applications to dynamical systems. In *Proceedings of the 26th International Conference on Machine Learning*, 2009.
- Song, Le, Boots, Byron, Siddiqi, Sajid, Gordon, Geoffrey, and Smola, Alex. Hilbert space embeddings of hidden Markov models. In *Proceedings of the 27th International Conference on Machine Learning*, 2010.