Discriminative Learning of Dynamical Systems for Motion Tracking

Minyoung Kim and Vladimir Pavlovic
Department of Computer Science
Rutgers University, NJ 08854 USA
{mikim,vladimir}@cs.rutgers.edu

Abstract

We introduce novel discriminative learning algorithms for dynamical systems. Models such as Conditional Random Fields or Maximum Entropy Markov Models outperform the generative Hidden Markov Models in sequence tagging problems in discrete domains. However, continuous state domains introduce a set of constraints that can prevent direct application of these traditional models. Instead, we suggest to learn generative dynamic models with discriminative cost functionals. For Linear Dynamical Systems, the proposed methods provide significantly lower prediction error than the standard maximum likelihood estimator, often comparable to nonlinear models. As a result, the models with lower representational capacity but computationally more tractable than nonlinear models can be used for accurate and efficient state estimation. We evaluate the generalization performance of our methods on the 3D human pose tracking problem from monocular videos. The experiments indicate that the discriminative learning can lead to improved accuracy of pose estimation with no increase in computational cost of tracking.

1. Introduction

We consider the problem of tracking or state estimation of time-series motion sequences. The problem can be formulated as estimating a continuous multivariate state sequence, $\mathbf{x} = \mathbf{x}_1 \cdots \mathbf{x}_T$, from the measurement sequence, $\mathbf{y} = \mathbf{y}_1 \cdots \mathbf{y}_T$, where $\mathbf{x}_t \in \mathbb{R}^d$ and $\mathbf{y}_t \in \mathbb{R}^k$. Its applications in computer vision include 3D tracking of the human motion and pose estimation for moving objects from sequences of monocular or multi-camera images.

Learning of dynamic models for tracking is often accomplished by optimizing the likelihood of the measurement sequence, $P(\mathbf{y})$. Increased availability of high-precision motion capture tools and data opens a new possibility for learning models that directly optimize a tracker's prediction accuracy, $P(\mathbf{x}|\mathbf{y})$. However, the study of *discriminative* learning methods for tracking has only recently emerged in the computer vision community.

A problem resembling the state estimation in tracking, when $\mathbf{x}_t$ is a *discrete label* instead of continuous multivariate, is known as sequence tagging or segmentation. The most popular generative model in this realm is the Hidden Markov Model (HMM). Traditional Maximum Likelihood (ML) learning of generative models such as HMMs is not directly compatible with the ultimate goal of label prediction (namely, $\mathbf{x}$ given $\mathbf{y}$), as it optimizes the fit of the models to data jointly, $\mathbf{x}$ and $\mathbf{y}$. Recently, discriminative models such as Conditional Random Fields (CRFs) and Maximum Entropy Markov Models (MEMMs) were introduced to address the label prediction problem directly, resulting in superior performance to the generative models [9, 10].

Despite a broad success of discriminative models in the discrete state domain, the use of discriminative dynamic models for continuous multivariate state estimation is not widespread. One reason for this is that a natural reparameterization-based transformation of generative dynamic systems to conditional models may violate density integrability constraints and can often produce unstable dynamic systems. For example, an extension of Linear Dynamical System (LDS) to CRF imposes irregular constraints on the CRF parameters to ensure finiteness of the log-partition function, making convex or general gradient-based optimization complex and prone to numerical failure.

As an alternative to CRF-based models in continuous state sequence domains we propose to learn generative dynamic models discriminatively. This approach has been well studied in classification settings: Learning generative models such as Tree-Augmented Naive Bayes (TAN) or HMMs discriminatively via maximizing conditional likelihoods yields better prediction performance than the traditional maximum likelihood estimator [4, 6, 8, 12, 15]. Our main contribution in this paper is to extend this approach to dynamic models and the motion tracking problem. Namely, we learn dynamic models that directly optimize the accuracy of pose predictions rather than jointly increasing the likelihood of the object's visual appearance and pose.

We introduce two discriminative learning algorithms for

1-4244-1180-7/07/$25.00 ©2007 IEEE

generative probabilistic dynamical systems, $P(\mathbf{x}, \mathbf{y})$. One is to maximize the conditional log-likelihood of the entire state sequence $\mathbf{x}$, that is, $arg\max \log P(\mathbf{x}|\mathbf{y})$, while the other is for the individual state slices $\mathbf{x}_t$, namely, $arg\max(1/T)\sum_{t=1}^{T} \log P(\mathbf{x}_t|\mathbf{y})$. These objectives are not convex in general, however, the gradient-based optimization yields superior prediction performance to that of the standard ML algorithm. In addition, we devise computationally efficient methods for gradient evaluation as a part of the proposed framework.

For several human motions, we compare the prediction performance of the competing models including nonlinear and latent variable dynamic models. The proposed discriminative learning algorithms on LDS can provide significantly lower prediction error than the standard maximum likelihood estimator, often comparable to estimates of computationally more expensive and parameter sensitive nonlinear or latent variable models. Thus the discriminative LDS offers a highly desired combination of high estimation accuracy and low computational complexity.

The paper is organized as follows: In the next section we briefly review LDS. In Sec. 3, it is discussed why discriminative models can be problematic in the continuous multivariate state domain. Then in Sec. 4, the proposed discriminative learning algorithms for LDS are described, followed by how they can be extended to nonlinear models. After reviewing related prior work in Sec. 5, the evaluation on the motion tracking data appears in Sec. 6.

2. Linear Dynamical Systems

LDS assumes transition and emission densities to be linear Gaussian, conforming to the graphical representation in Fig. 1(a). The conditional densities of LDS are defined as:

$$\mathbf{x}_1 \sim \mathcal{N}(\mathbf{x}_1; \mathbf{m}_0, \mathbf{V}_0), \;\; \mathbf{x}_t|\mathbf{x}_{t-1} \sim \mathcal{N}(\mathbf{x}_t; \mathbf{A}\mathbf{x}_{t-1}, \mathbf{\Gamma}),$$
$$\mathbf{y}_t|\mathbf{x}_t \sim \mathcal{N}(\mathbf{y}_t; \mathbf{C}\mathbf{x}_t, \mathbf{\Sigma}). \quad (1)$$

The LDS parameter set is $\mathbf{\Theta}_{lds} = \{\mathbf{m}_0, \mathbf{V}_0, \mathbf{A}, \mathbf{\Gamma}, \mathbf{C}, \mathbf{\Sigma}\}$. The joint log-likelihood, $LL = \log P(\mathbf{x}, \mathbf{y})$[1] is (up to a constant):

$$LL = -\tfrac{1}{2}\Big[(\mathbf{x}_1 - \mathbf{m}_0)'\mathbf{V}_0^{-1}(\mathbf{x}_1 - \mathbf{m}_0) + \log|\mathbf{V}_0| + \sum_{t=2}^{T}(\mathbf{x}_t - \mathbf{A}\mathbf{x}_{t-1})'\mathbf{\Gamma}^{-1}(\mathbf{x}_t - \mathbf{A}\mathbf{x}_{t-1}) + \log|\mathbf{\Gamma}|^{T-1} + \sum_{t=1}^{T}(\mathbf{y}_t - \mathbf{C}\mathbf{x}_t)'\mathbf{\Sigma}^{-1}(\mathbf{y}_t - \mathbf{C}\mathbf{x}_t) + \log|\mathbf{\Sigma}|^{T}\Big], \quad (2)$$

where $\mathbf{M}'$ indicates the transpose of the matrix $\mathbf{M}$.

The task of inference is to compute the filtered state densities, $P(\mathbf{x}_t|\mathbf{y}_1, \ldots, \mathbf{y}_t)$, and the smoothed densities, $P(\mathbf{x}_t|\mathbf{y})$. The linear Gaussian assumption on LDS implies Gaussian posteriors that can be evaluated in linear time using the well-known Kalman filtering or RTS smoothing methods. We denote the means and the covariances of these posterior densities by: $\hat{\mathbf{m}}_t \triangleq E[\mathbf{x}_t|\mathbf{y}_1 \ldots \mathbf{y}_t]$, $\hat{\mathbf{V}}_t \triangleq V(\mathbf{x}_t|\mathbf{y}_1 \ldots \mathbf{y}_t)$, $\mathbf{m}_t \triangleq E[\mathbf{x}_t|\mathbf{y}]$, $\mathbf{V}_t \triangleq V(\mathbf{x}_t|\mathbf{y})$, and $\mathbf{\Sigma}_{t,t-1} \triangleq Cov(\mathbf{x}_t, \mathbf{x}_{t-1}|\mathbf{y})$.

To learn LDS, one needs to find $\mathbf{\Theta}_{lds}$ that optimizes a desired objective function. In the supervised setting that we assume throughout the paper, for the given train data $D = \{(\mathbf{x}^i, \mathbf{y}^i)\}_{i=1}^{n}$, the generative learning maximizes the joint log-likelihood, $\sum_{i=1}^{n} LL(\mathbf{x}^i, \mathbf{y}^i)$, which has a closed form by solving the equation that sets the gradient of (2) equal to 0. For instance, using the gradient w.r.t. $\mathbf{C}$ shown in (3), we have $\mathbf{C}^* = \big[\sum_{i=1}^{n}\sum_{t=1}^{T_i}\mathbf{y}_t^i\,\mathbf{x}_t^{i\prime}\big]\cdot\big[\sum_{i=1}^{n}\sum_{t=1}^{T_i}\mathbf{x}_t^i\,\mathbf{x}_t^{i\prime}\big]^{-1}$, where T_i is the length of the i-th sequence.

$$\frac{\partial LL}{\partial \mathbf{C}} = \mathbf{\Sigma}^{-1}\cdot\sum_{t=1}^{T}(\mathbf{y}_t\mathbf{x}_t' - \mathbf{C}\mathbf{x}_t\mathbf{x}_t') \quad (3)$$

The ML learning of the generative model is intended to fit the model to data jointly on $\mathbf{x}$ and $\mathbf{y}$. However, in tracking we are often more interested in finding a model that yields a high accuracy of predicting $\mathbf{x}$ from $\mathbf{y}$, an objective *not* achieved by ML learning in general. It is therefore tempting to employ discriminative models which explicitly focus on the desired goal. In the discrete state domain, CRFs and MEMMs are such models shown to outperform the generative models like HMMs. Unfortunately, as discussed in the next section, developing CRF- or MEMM-like discriminative models in the continuous multivariate state domain can be a challenge.

3. Discriminative Dynamic Models

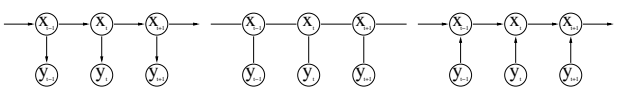
Analogous to extending HMMs to CRFs and MEMMs, we will extend LDS to conditional models that have the same representational capacity as LDS. This, for instance, reduces to exploiting 2nd-order moments (e.g., $\mathbf{x}_t\mathbf{x}_{t-1}'$) as local features for CRF, and a linear Gaussian local conditional density $P(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t)$ for MEMM.

3.1. Conditional Random Fields

CRF models the conditional probability of $\mathbf{x}$ given $\mathbf{y}$. Since $P(\mathbf{x}|\mathbf{y}) \propto P(\mathbf{x}, \mathbf{y})$, the log-conditional $\log P(\mathbf{x}|\mathbf{y})$ has the same form as (2) except that those terms that are not involved in $\mathbf{x}$ (e.g., $\mathbf{y}_t'\mathbf{\Sigma}^{-1}\mathbf{y}_t$) can be removed as they will be marginalized out into the log-partition function. We reparameterize $\mathbf{\Theta}_{lds}$ to CRF parameters so that the latter become linear coefficients for the CRF features. Specifically, the new CRF parameter set $\mathbf{\Theta}_{crf} = \{\mathbf{\Lambda}_b, \mathbf{\Lambda}_A, \mathbf{\Lambda}_C, \mathbf{\Lambda}_1, \mathbf{\Lambda}, \mathbf{\Lambda}_T\}$ satisfies:

$$\mathbf{\Lambda}_b \triangleq \mathbf{V}_0^{-1}\mathbf{m}_0, \;\; \mathbf{\Lambda}_A \triangleq \mathbf{\Gamma}^{-1}\mathbf{A}, \;\; \mathbf{\Lambda}_C \triangleq \mathbf{\Sigma}^{-1}\mathbf{C},$$
$$\mathbf{\Lambda}_1 \triangleq -\tfrac{1}{2}(\mathbf{V}_0^{-1} + \mathbf{A}'\mathbf{\Gamma}^{-1}\mathbf{A} + \mathbf{C}'\mathbf{\Sigma}^{-1}\mathbf{C}),$$
$$\mathbf{\Lambda} \triangleq -\tfrac{1}{2}(\mathbf{\Gamma}^{-1} + \mathbf{A}'\mathbf{\Gamma}^{-1}\mathbf{A} + \mathbf{C}'\mathbf{\Sigma}^{-1}\mathbf{C}),$$
$$\mathbf{\Lambda}_T \triangleq -\tfrac{1}{2}(\mathbf{\Gamma}^{-1} + \mathbf{C}'\mathbf{\Sigma}^{-1}\mathbf{C}). \quad (4)$$

[1] For brevity, we will often drop the dependency on $\mathbf{\Theta}$ in the notation.



(a) HMM or LDS (b) CRF (c) MEMM

Figure 1. Graphical Models: HMM (or LDS), CRF, and MEMM.

Then the LDS-counterpart CRF model can be written as:

$$P(\mathbf{x}|\mathbf{y}, \mathbf{\Theta}_{crf}) = \frac{\exp\big(\mathbf{\Phi}(\mathbf{x}, \mathbf{y}; \mathbf{\Theta}_{crf})\big)}{Z(\mathbf{y}; \mathbf{\Theta}_{crf})}, \qquad (5)$$

where the score function, $\mathbf{\Phi}(\mathbf{x}, \mathbf{y}; \mathbf{\Theta}_{crf}) = \mathbf{\Lambda}_b' \mathbf{x}_1 + \mathbf{x}_1' \mathbf{\Lambda}_1 \mathbf{x}_1 + \sum_{t=2}^{T-1} \mathbf{x}_t' \mathbf{\Lambda} \mathbf{x}_t + \mathbf{x}_T' \mathbf{\Lambda}_T \mathbf{x}_T + \sum_{t=2}^{T} \mathbf{x}_t' \mathbf{\Lambda}_A \mathbf{x}_{t-1} + \sum_{t=1}^{T} \mathbf{y}_t' \mathbf{\Lambda}_C \mathbf{x}_t$, and the normalizing partition function, $Z(\mathbf{y}; \mathbf{\Theta}_{crf}) = \int_{\mathbf{x}} \exp(\mathbf{\Phi}(\mathbf{x}, \mathbf{y}; \mathbf{\Theta}_{crf}))$.

The (conditional) log-likelihood, $\log P(\mathbf{x}|\mathbf{y}, \mathbf{\Theta}_{crf})$, is concave in $\mathbf{\Theta}_{crf}$ because $\mathbf{\Phi}(\mathbf{x}, \mathbf{y}; \mathbf{\Theta}_{crf})$ is linear and $\log Z(\mathbf{y}; \mathbf{\Theta}_{crf})$ is convex. However, the reparameterization produces unexpected constraints on the CRF parameter space. In fact, the set of constraints revealed during the inference phase is defined in a recursive manner (See Appendix I for details), where it is difficult to pose such constraints in the optimization. This, in turn, makes the seemingly convex optimization infeasible.

3.2. Maximum Entropy Markov Models

MEMM has a graphical structure depicted in Fig. 1(c). Despite the well-known *label bias* problem, its simple learning procedure that does not require forward/backward recursion is very attractive. Given a complete data $\{(\mathbf{x}, \mathbf{y})\}$, the likelihood function can be factored into terms related with individual slices $(\mathbf{x}_{t-1}, \mathbf{y}_t, \mathbf{x}_t)$ and subsequently treated as a set of independent slice instances. Learning MEMM is equivalent to training a *static* classifier or regression function $P(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t)$ for the *iid* data with the output $\{\mathbf{x}_t\}$ and the input $\{(\mathbf{x}_{t-1}, \mathbf{y}_t)\}$.

MEMM with the linear Gaussian conditional, namely,

$$\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t \sim \mathcal{N}(\mathbf{x}_t; \mathbf{A}_x \mathbf{x}_{t-1} + \mathbf{A}_y \mathbf{y}_t + \mathbf{e}, \mathbf{W}), \qquad (6)$$

can be seen as a counterpart of LDS. The prediction is done by the recursion, $P(\mathbf{x}_t|\mathbf{y}) = \int_{\mathbf{x}_{t-1}} P(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t) \cdot P(\mathbf{x}_{t-1}|\mathbf{y})$. Note that in MEMMs the smoothed posterior $P(\mathbf{x}_t|\mathbf{y})$ equals the filtered posterior $P(\mathbf{x}_t|\mathbf{y}_1, ..., \mathbf{y}_t)$, effectively removing the influence of future samples on current state estimates. The mean estimate $\tilde{\mathbf{m}}_t = E[\mathbf{x}_t|\mathbf{y}]$ is:

$$\tilde{\mathbf{m}}_t = \mathbf{A}_x \tilde{\mathbf{m}}_{t-1} + \mathbf{A}_y \mathbf{y}_t + \mathbf{e}. \qquad (7)$$

(7) points to another deficiency of linear MEMMs. The next state estimate is linearly related with the previous state mean, where the coefficient $\mathbf{A}_x$ is determined by the multivariate linear regression learning with data treated slicewise independently. If the learned $\mathbf{A}_x$ is unstable[2], the state estimates become unbounded. As a result, the state estimation error can be significantly amplified in this MEMM setting.

This behavior may be reduced when non-linear or non-Gaussian noise models are used. In [19], for instance, a complex nonlinear regression function (Bayesian Mixture of Experts) was applied to the 3D human body pose estimation problem. However, the failure of simple linear MEMM points to prevalent role of local functions over the MEMM's overall discriminative model structure. In other words, the success of MEMM may be strongly dependent on the performance of the employed *static* regression functions.

4. Discriminative Learning of LDS

Our analysis of traditional conditional dynamic models points to possible modes of failure when such models are applied to continuous state domains. To address these deficiencies we suggest to learn the generative LDS model with discriminative cost functionals. As the discriminative learning of TAN or HMM has shown to outperform generative learning in classification settings, the same approach can be brought to benefit the task of motion tracking in continuous domains. We propose two discriminative objectives to solve the problem of discriminative learning of LDS. The optimal parameter estimation is accomplished by an efficient gradient search on the two objectives. We also show how the discriminative learning task can be extended to a general family of nonlinear dynamic models.

4.1. Conditional Likelihood Maximization (CML)

The goal of CML learning is to find LDS parameters that maximize the conditional likelihood of $\mathbf{x}$ given $\mathbf{y}$, an objective directly related to our goal of accurate state prediction. The conditional log-likelihood objective for the data $(\mathbf{x}, \mathbf{y})$ is defined as:

$$CLL = \log P(\mathbf{x}|\mathbf{y}) = \log P(\mathbf{x}, \mathbf{y}) - \log P(\mathbf{y}). \qquad (8)$$

CLL objective is, in general, non-convex in the model parameter space. However, the objective can be locally optimized using a general gradient search. The gradient of CLL with respect to $\mathbf{\Theta}_{lds}$ is:

$$\frac{\partial CLL}{\partial \mathbf{\Theta}_{lds}} = \frac{\partial \log P(\mathbf{x}, \mathbf{y})}{\partial \mathbf{\Theta}_{lds}} - \frac{\partial \log P(\mathbf{y})}{\partial \mathbf{\Theta}_{lds}}. \qquad (9)$$

The first term, the gradient of the *complete* log-likelihood (i.e., the Fisher score) is easy to obtain (e.g., (3)). The second term, the gradient of the *observation* log-likelihood, is

[2] Eigenvalues of matrix A have absolute magnitudes exceeding 1.

the expected Fisher score w.r.t. the posterior, namely,

$$\frac{\partial \log P(\mathbf{y})}{\partial \mathbf{\Theta}_{lds}} = \int_{\mathbf{x}} P(\mathbf{x}|\mathbf{y}) \frac{\partial \log P(\mathbf{x},\mathbf{y})}{\partial \mathbf{\Theta}_{lds}} = E_{P(\mathbf{x}|\mathbf{y})}\left[\frac{\partial \log P(\mathbf{x},\mathbf{y})}{\partial \mathbf{\Theta}_{lds}}\right]. \quad (10)$$

Hence, CLL gradient is the difference between the Fisher score on the data $(\mathbf{x},\mathbf{y})$ and the expected Fisher score by the model given $\mathbf{y}$ only. Because the Fisher score (e.g., (3)) is a sum of 2nd-order moments (i.e., those related with $\mathbf{x}_t\mathbf{x}_t'$, $\mathbf{x}_t\mathbf{x}_{t-1}'$, or $\mathbf{x}_t$), the expected Fisher score can be easily computed once we have the posterior $P(\mathbf{x}|\mathbf{y})$. For example, $\frac{\partial \log P(\mathbf{y})}{\partial \mathbf{C}} = E_{P(\mathbf{x}|\mathbf{y})}\left[\frac{\partial LL}{\partial \mathbf{C}}\right] = \mathbf{\Sigma}^{-1} \cdot \sum_{t=1}^{T}\left[\mathbf{y}_t\mathbf{m}_t' - \mathbf{C}(\mathbf{m}_t\mathbf{m}_t' + \mathbf{V}_t)\right]$, where $\mathbf{m}_t$ and $\mathbf{V}_t$ are smoothed means and variances, respectively.

The well-known EM algorithm for LDS in the unsupervised setting (i.e., training data $= \{\mathbf{y}^i\}$) takes advantage of it in the M-step. In this case, $\sum_i \log P(\mathbf{y}^i)$ is the objective whose gradient is derived as in (10). Setting it to 0 gives no analytical solution, however, the EM follows an iterative update scheme: (1) (E-step) for the current iterate $\mathbf{\Theta}_{lds}$, compute $P(\mathbf{x}|\mathbf{y}^i, \mathbf{\Theta}_{lds})$, and (2) (M-step) solve $E\left[\sum_i \frac{\partial \log P(\mathbf{x},\mathbf{y}^i)}{\partial \mathbf{\Theta}_{lds}}\right] = 0$ to $\mathbf{\Theta}_{lds}$ as the next iterate, where the latter expectation is w.r.t. the posterior obtained from the E-step. This, under fairly general conditions, guarantees monotonic improvement of the objective for each update.

However, the EM algorithm cannot be directly applied to CLL optimization since the Jensen's inequality for lower bound does not hold. Instead, we use a gradient ascent optimization such as the conjugate gradient search or the efficient BFGS optimization that has shown to yield best results in traditional CRF learning [17].

4.2. Slicewise Conditional Likelihood Maximization

The goal of CML is to find a model that minimizes the *joint* estimation error for the entire state sequence, $\mathbf{x}$. In most motion tracking problems, however, it is more natural to consider the prediction error at each *time slice* independently. In the discrete state domain, this notion is directly related to minimization of the Hamming distance between the target and the inferred states. In the continuous domain, we consider the Slicewise Conditional Likelihood Maximization (SCML) as the following objective:

$$SCLL = \frac{1}{T}\sum_{t=1}^{T} \log P(\mathbf{x}_t|\mathbf{y}). \quad (11)$$

$SCLL$ has been introduced as an alternative objective for CRF in the discrete domain sequence tagging problem [7]. Note that evaluating the objective itself requires a forward/backward or Kalman filtering/smoothing. SCML learning is subsequently based on the gradient optimization.

We extend the approach of [7] to LDS. For clarity, we distinguish the states in *train data* from *random variables* by denoting the former as $\overline{\mathbf{x}}$ while the latter as $\mathbf{x}$. It is easy to see that $SCLL$ gradient can be written as:

$$\frac{\partial SCLL}{\partial \mathbf{\Theta}_{lds}} = \frac{1}{T}\sum_{t=1}^{T} \frac{\partial \log P(\overline{\mathbf{x}}_t,\mathbf{y})}{\partial \mathbf{\Theta}_{lds}} - \frac{\partial \log P(\mathbf{y})}{\partial \mathbf{\Theta}_{lds}}. \quad (12)$$

Since the second term is dealt in (10), we will focus on the first term of (12). It can be shown that the first term excluding $(1/T)$ is equivalent to:

$$\sum_{t=1}^{T}\int_{\mathbf{x}\setminus\mathbf{x}_t} P(\mathbf{x}\setminus\mathbf{x}_t|\overline{\mathbf{x}}_t,\mathbf{y}) \cdot \frac{\partial \log P(\mathbf{x},\mathbf{y})}{\partial \mathbf{\Theta}_{lds}}\Big|_{\mathbf{x}_t=\overline{\mathbf{x}}_t}, \quad (13)$$

where $\mathbf{x}\setminus\mathbf{x}_t$ means *set-minus*, excluding $\mathbf{x}_t$ from $\mathbf{x}$. Recalling that the Fisher score, $\frac{\log P(\mathbf{x},\mathbf{y})}{\partial \mathbf{\Theta}_{lds}}$, is a sum of 2nd-order moment terms, let $\mathbf{f}(\mathbf{x}_j,\mathbf{x}_{j-1})$ be one of them. This enables us to evaluate $E[\mathbf{f}(\mathbf{x}_j,\mathbf{x}_{j-1})]$ individually (w.r.t. the unnormalized density $\sum_{t=1}^{T} P(\mathbf{x}\setminus\mathbf{x}_t|\overline{\mathbf{x}}_t,\mathbf{y})$), while later on all the expectations of terms that compose the Fisher score have to be summed to obtain the quantity in (13).

For each $j = 2,\ldots,T$, the expectation, $E[\mathbf{f}(\mathbf{x}_j,\mathbf{x}_{j-1})]$ w.r.t. $\sum_{t=1}^{T} P(\mathbf{x}\setminus\mathbf{x}_t|\overline{\mathbf{x}}_t,\mathbf{y})$, can be written as:

$$\begin{aligned} &E_{P(\mathbf{x}_j|\overline{\mathbf{x}}_{j-1},\mathbf{y})}\left[\mathbf{f}(\mathbf{x}_j,\overline{\mathbf{x}}_{j-1})\right] + E_{P(\mathbf{x}_{j-1}|\overline{\mathbf{x}}_j,\mathbf{y})}\left[\mathbf{f}(\overline{\mathbf{x}}_j,\mathbf{x}_{j-1})\right] \\ &\quad + \textstyle\sum_{t=1}^{j-2} E_{P(\mathbf{x}_j,\mathbf{x}_{j-1}|\overline{\mathbf{x}}_t,\mathbf{y})}\left[\mathbf{f}(\mathbf{x}_j,\mathbf{x}_{j-1})\right] \\ &\quad + \textstyle\sum_{t=j+1}^{T} E_{P(\mathbf{x}_j,\mathbf{x}_{j-1}|\overline{\mathbf{x}}_t,\mathbf{y})}\left[\mathbf{f}(\mathbf{x}_j,\mathbf{x}_{j-1})\right]. \end{aligned} \quad (14)$$

The first two terms in (14) are the expectations w.r.t. the posteriors given the neighbor (previous and next, respectively) state. These posteriors are both Gaussians, namely,

$$\begin{aligned} P(\mathbf{x}_{t+1}|\mathbf{x}_t,\mathbf{y}) &= \mathcal{N}(\mathbf{x}_{t+1};\mathbf{F}_{t+1}\mathbf{x}_t + \mathbf{b}_{t+1}, \mathbf{R}_{t+1}), \\ P(\mathbf{x}_t|\mathbf{x}_{t+1},\mathbf{y}) &= \mathcal{N}(\mathbf{x}_t;\mathbf{G}_t\mathbf{x}_{t+1} + \mathbf{c}_t, \mathbf{S}_t), \end{aligned} \quad (15)$$

where $\mathbf{F}_{t+1} = \mathbf{\Sigma}_{t+1,t}\mathbf{V}_t^{-1}$, $\mathbf{G}_t = \mathbf{\Sigma}_{t+1,t}'\mathbf{V}_{t+1}^{-1}$, $\mathbf{b}_{t+1} = \mathbf{m}_{t+1} - \mathbf{F}_{t+1}\mathbf{m}_t$, $\mathbf{c}_t = \mathbf{m}_t - \mathbf{G}_t\mathbf{m}_{t+1}$, $\mathbf{R}_{t+1} = \mathbf{V}_{t+1} - \mathbf{F}_{t+1}\mathbf{\Sigma}_{t+1,t}'$, and $\mathbf{S}_t = \mathbf{V}_t - \mathbf{G}_t\mathbf{\Sigma}_{t+1,t}$. (Recall that $\mathbf{\Sigma}_{t,t-1} \triangleq Cov(\mathbf{x}_t,\mathbf{x}_{t-1}|\mathbf{y})$.)

The last two terms in (14) are the expectations w.r.t. $P(\mathbf{x}_j,\mathbf{x}_{j-1}|\overline{\mathbf{x}}_t,\mathbf{y})$, the posteriors given the past $(t < j-1)$ and the future $(t > j)$ state, respectively. Computing these posteriors requires another forward (for $t < j-1$) and backward (for $t > j$) recursion on j, which together with the Kalman filter forms the two-pass forward/backward algorithm for SCML learning. We refer the reader to Appendix II for details on the derivation of the second-pass recursion[3].

[3] Apart from [7]'s discrete-domain two-pass algorithm which takes $O(T)$ time, our algorithm requires $O(T^2)$ time since $O(T)$ Gaussians generated from each j-th step of the second-pass forward/backward need to be stored (See Appendix II). [7]'s linear time is due to the ease of representing a sum of probability mass functions as a single function compactly. The quadratic time could be a problem in large sequence lengths, however, segmentation to shorter sequences will be helpful.

4.3. Extension to Nonlinear Dynamical Systems

CML and SCML learning can be similarly applied to the nonlinear dynamical systems (NDS). In NDS, the posterior can be evaluated via Extended Kalman filtering/smoothing based on the approximated linear model (e.g., [3]) or using various particle filter methods, depending on the dimensionality of the state space. Since the Fisher score for NDS is no more a sum of 2nd-order moments, rather a complex nonlinear function, the evaluation of the expectation $E[\mathbf{f}(\mathbf{x}_t, \mathbf{x}_{t-1})]$ becomes difficult. However, following [3] we can approximate any nonlinear functions by RBF networks, namely,

$$\begin{aligned} \mathbf{x}_t|\mathbf{x}_{t-1} &\sim \mathcal{N}(\mathbf{x}_t; \mathbf{A}_\mathbf{k}\mathbf{k}(\mathbf{x}_{t-1}) + \mathbf{A}\mathbf{x}_{t-1}, \mathbf{\Gamma}), \\ \mathbf{y}_t|\mathbf{x}_t &\sim \mathcal{N}(\mathbf{y}_t; \mathbf{C}_\mathbf{k}\mathbf{k}(\mathbf{x}_t) + \mathbf{C}\mathbf{x}_t, \mathbf{\Sigma}), \end{aligned} \tag{16}$$

where $\mathbf{k}(\mathbf{x}_t) \triangleq [k(\mathbf{x}_t, \mathbf{u}_1), \ldots, k(\mathbf{x}_t, \mathbf{u}_L)]'$ is a vector of RBF kernels evaluated on the known centers $\{\mathbf{u}_l\}_{l=1}^L$. For $k(\mathbf{x}_t, \mathbf{u}_l) = e^{-\frac{1}{2}(\mathbf{x}_t - \mathbf{u}_l)'\mathbf{S}_l^{-1}(\mathbf{x}_t - \mathbf{u}_l)}$, where $\mathbf{S}_l$ is the kernel covariance, the nonlinear part in the Fisher score takes a specific form such as $\mathbf{k}(\mathbf{x}_t)\mathbf{k}(\mathbf{x}_t)'$, $\mathbf{k}(\mathbf{x}_t)\mathbf{k}(\mathbf{x}_{t-1})'$, or $\mathbf{x}_t\mathbf{k}(\mathbf{x}_t)'$, and has a closed-form expectation w.r.t. a Gaussian (approximated) posterior. As a result, gradient terms necessary for CML/SCML optimization in RBF nonlinear dynamic models also possess closed-form expressions.

In the evaluation, we have verified that for LDS, the discriminative algorithms improve the generative learning significantly. For NDS, however, the improvement is not as significant as the linear case. In other words, the choice of learning objective for nonlinear models appears less critical. However, the generalization performance of the nonlinear models can be very sensitive to the choice of the kernel centers and the kernel hyperparameters. In Sec. 6, we demonstrate that discriminatively learned linear models can be comparable to even well-tuned nonlinear models.

5. Prior Work

While discriminative learning of discrete-state dynamic models such as HMMs, CRFs and MEMMs has received significant attention recently, learning of similar models in the continuous space has been rarely explored. In robotics community, [1] empirically studied several objectives for learning of continuous-state dynamical systems. In contrast to [1]'s ad-hoc optimization method, our work is the first to provide efficient gradient optimization algorithms for discriminative objectives, by extending the method of [7] to dynamical systems in continuous domains.

The recent work on the human motion tracking problem can be roughly categorized into: dynamic model based ([5, 13, 14]), nonlinear manifold embedding ([2, 16, 18, 23]), and Gaussian process based latent variable models ([11, 21, 22]) to name a few. In our approach, we consider a generative family of models and show that it can be used for accurate and computationally efficient pose estimation, if coupled with a proper learning objective.

Related with the discriminative paradigm, [19] successfully employed a MEMM-like model with Bayesian mixtures of experts for 3D pose estimation. In general, MEMMs are sensitive to label-bias [9]. Their ability to successfully infer states from observations mostly depends on the modeling capacity of the regression functions and not on the choice of discriminative dynamic model objective. Unlike MEMMs, the discriminatively learned generative dynamic models could also be used for motion synthesis.

Model	ML	CML	SCML
L2 Error	1.79 ± 0.26	1.59 ± 0.22	1.30 ± 0.12
Log-Perplexity	4.76 ± 0.40	4.49 ± 0.34	3.80 ± 0.25

Table 1. Test errors and log-perplexities for synthetic data.

6. Evaluation

We evaluate our discriminative dynamical system modeling approach in a set of experiments that include synthetic data as well as the CMU motion capture dataset[4]. The proposed models are denoted as CML and SCML, the LDS models learned via the methods in Sec. 4.1 and Sec. 4.2, respectively. ML is the standard maximum likelihood estimator for LDS. We also include comparison with nonlinear and latent-variable dynamic models, as described in Sec. 6.2.

6.1. Synthetic Data

We synthesize data from a devised model which is structurally more complex than LDS. The model has 2nd-order dynamics and emission: $\mathbf{x}_t = \mathbf{A}_1\mathbf{x}_{t-1} + \mathbf{A}_2\mathbf{x}_{t-2} + \mathbf{v}_t$, and $\mathbf{y}_t = \mathbf{C}_1\mathbf{x}_t + \mathbf{C}_2\mathbf{x}_{t-1} + \mathbf{w}_t$, where $\mathbf{v}_t$ and $\mathbf{w}_t$ are Gaussian white noises. This experiment demonstrates how the learning algorithms behave for the incorrect model structure, emphasizing the fact that it is usually difficult to figure out the correct model structure in many applications.

The evaluation is done by leave-one-out validation for 10 sampled sequences of lengths about 150, where $\dim(\mathbf{x}_t) = 3$ and $\dim(\mathbf{y}_t) = 2$. The test performances of competing learning methods are depicted in Table 1. The L2 error is the average norm-2 difference, $(1/T)\sum_{t=1}^T ||\overline{\mathbf{x}}_t - \mathbf{m}_t||_2$, where $\overline{\mathbf{x}}$ is the ground truth, and $\mathbf{m}$ is the estimated state sequence. The log-perplexity is defined as $-(1/T)\sum_{t=1}^T \log P(\overline{\mathbf{x}}_t|\mathbf{y}, \mathbf{\Theta})$, which captures the variance of the estimate. The smaller number is better for both measures. We also visualized the estimated sequences in Fig. 2.

The result shows that the prediction performance is improved by the proposed methods, while the significance is stronger for SCML than CML. It also implies that discrim-

[4] http://mocap.cs.cmu.edu/.

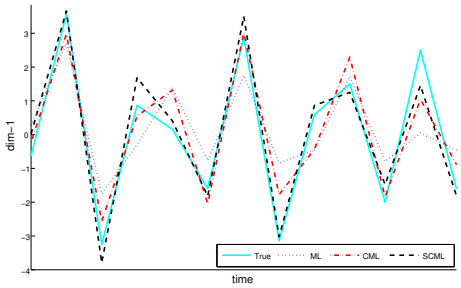


Figure 2. Visualization of estimated sequences for synthetic data. It shows the estimated states (for dim-1) at $t = 136 \sim 148$. The ground truth is depicted by solid (cyan) line, ML by dotted (blue), CML by dotted-dashed (red), and SCML by dashed (black).

inative learning can be useful for enhancing the restricted performance of the generatively trained models with (possibly) suboptimal structures.

6.2. Human Motion Data

We evaluate the performance of the proposed methods on the task of 3D pose estimation from real human motion data. The CMU motion capture dataset provides the ground-truth body poses (3D joint angles), which makes it possible to compare competing methods quantitatively. Among the original 59 joint angles, we used only 39-dim by excluding less significant fingers and toes as well as joint angles that rarely vary over time. Here we include three different motions: walking, picking-up a ball, and running. For each motion, 5 or 6 sequences from one subject are gathered to perform leave-one-out validation. Sequences are about 150-frame long, containing 1 or 2 motion cycles. The measurement is a 10-dim Alt-Moment feature vector extracted from the monocular silhouette image (e.g., [20]). The images are taken by a single camera at a fixed view.

Typically, we will demonstrate how comparable the performance of the proposed algorithms on LDS is to that of nonlinear models learned generatively. We briefly discuss two nonlinear models that are used in our evaluation.

The first model is NDS defined as (16). Since it is computationally demanding to use all poses $\mathbf{x}_t$ in the train data for RBF kernel centers $\mathbf{u}_l$, we instead adopt a sparse greedy kernel selection technique. It selects a pose from the pose pool (of all train poses) one at a time, according to a certain criterion (e.g., maximizing the data likelihood). Deciding the number of poses (or kernel centers) to be added is crucial for generalization performance. In our experiment, we use cross-validation among the several candidates (e.g., 5%, 10%, or 20% of the pool). The kernel covariance $\mathbf{S}_l$ for each center $\mathbf{u}_l$ is estimated in a way that the neighbor points of $\mathbf{u}_l$ have kernel values one half of its peak value [3]. This generates reasonably smooth kernels.

The second model is the latent variable nonlinear dynamic model, denoted as LVN. As it is broadly believed that the realizable poses lie in a low dimensional space, it is useful to introduce latent variables $\mathbf{z}_t$ embedded from the poses $\mathbf{x}_t$. One possible way to devise LVN is to place dynamics on $\mathbf{z}_t$, assuming $\mathbf{x}_t$ and $\mathbf{y}_t$ are generated nonlinearly (with RBF kernels) by $\mathbf{z}_t$. Learning LVN can be done by EM algorithm on the linear approximated model as introduced in [3]. Initial subspace mapping for LVN is determined by PCA dim-reduction on the train poses. Similarly to NDS, the number of kernels is determined by cross-validation. We use $\dim(\mathbf{z}_t) = 3$.

Motions	Err.	ML	CML	SCML	NDS	LVN
Walk	SJA	19.20	18.31	17.19	18.91	18.01
	FJA	22.57	22.73	20.78	20.84	19.05
	S3P	15.28	14.79	13.53	14.62	13.99
	F3P	20.02	20.28	17.07	16.59	14.96
Pick-up	SJA	35.03	33.15	30.56	33.50	32.23
	FJA	42.28	38.89	36.99	41.25	32.10
	S3P	22.60	21.27	19.33	21.14	20.49
	F3P	25.20	24.36	23.83	25.35	20.40
Run	SJA	23.35	22.11	19.39	21.26	19.08
	FJA	21.87	22.09	20.92	21.86	19.76
	S3P	21.52	19.85	16.96	18.41	16.97
	F3P	20.40	20.43	18.43	18.42	17.65

Table 2. Average test errors. The error types are abbreviated as 3 letters: The first indicates smoothed (S) or filtered (F), followed by 2 letters meaning that the error is measured in either the joint angle space (JA) or the 3D articulation point space (3P) (e.g., SJA = smoothed error in the joint angle space). The unit in the 3D point space can be deemed by the height of the human model ~ 25.

Table 2 shows the average test (norm-2) errors of competing methods. We recorded the smoothed $(\mathbf{x}_t|\mathbf{y})$ and the filtered $(\mathbf{x}_t|\mathbf{y}_1, \ldots, \mathbf{y}_t)$ estimation errors for both the (joint angle) pose space and the 3D articulation point space. The latter can be easily evaluated by mapping the estimated joint angles to the body skeleton model provided in the dataset. As shown, the proposed algorithms have significantly lower prediction errors than ML learning, while exhibiting comparable (or often superior) performance to the nonlinear models possibly with latent variables.

It should be noticed that the filtered estimation errors of the proposed methods are not as outstanding as the smoothed ones. This is probably due to their smoothing-based objectives. It is interesting, yet left as future work, to see the performance of the modified objectives based on filtering. When comparing two discriminative algorithms, SCML yields superior performance to CML consistently for all motions. This is expected from the $SCLL$ objective which is more closely related with the ultimate error measure. Note also that the inference (tracking) of CML or SCML is the standard Kalman filtering/smoothing, which is

much faster than the approaches based on particles or nonlinear optimization (e.g., [11, 19, 21, 22]).

In Fig. 3, selected frames of the estimated body skeletons are illustrated to compare SCML with the standard linear and nonlinear models.

7. Conclusion

We introduced novel discriminative learning algorithms for generative family of dynamical systems. The proposed approaches yield accurate and computationally efficient pose estimation for motion data. As a future work, we plan to extend our methods to deal with the settings where the motion capture data is assumed noisy (e.g., occlusions). In addition we will apply our approaches to piece-wise linear models such as switching LDS (e.g., [14]) which can handle problematic motions that may contain rapid changes in motion types.

Acknowledgements

This work was supported by the NSF IIS grant 0413105.

References

[1] P. Abbeel, A. Coates, M. Montemerlo, A. Y. Ng, and S. Thrun. Discriminative training of Kalman filters, 2005. In Proceedings of Robotics: Science and Systems.

[2] A. Elgammal and C.-S. Lee. Inferring 3D body pose from silhouettes using activity manifold learning, 2004. CVPR.

[3] Z. Ghahramani and S. Roweis. Learning nonlinear dynamical systems using an EM algorithm, 1999. NIPS.

[4] R. Greiner and W. Zhou. Structural extension to logistic regression: Discriminative parameter learning of belief net classifiers, 2002. In Proc. of annual meeting of the AAAI.

[5] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi. Robust online appearance models for visual tracking. *IEEE Trans. on PAMI*, 25(10):1296–1311, 2001.

[6] Y. Jing, V. Pavlovic, and J. M. Rehg. Efficient discriminative learning of Bayesian Network Classifier via boosted augmented Naive Bayes, 2005. ICML.

[7] S. Kakade, Y. Teh, and S. Roweis. An alternate objective function for Markovian fields, 2002. ICML.

[8] M. Kim and V. Pavlovic. Discriminative learning of mixture of Bayesian Network Classifiers for sequence classification, 2006. CVPR.

[9] J. Lafferty, A. McCallum, and F. Pereira. Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data, 2001. ICML.

[10] A. Mccallum, D. Freitag, and F. Pereira. Maximum Entropy Markov Models for information extraction and segmentation, 2000. ICML.

[11] K. Moon and V. Pavlovic. Impact of dynamics on subspace embedding and tracking of sequences, 2006. CVPR.

[12] A. Y. Ng and M. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and Naive Bayes, 2002. NIPS.

[13] B. North, M. I. A. Blake, and J. Rittscher. Learning and classification of complex dynamics. *IEEE Trans. on PAMI*, 25(9):1016–1034, 2000.

[14] V. Pavlovic, J. M. Rehg, and J. MacCormick. Learning switching linear models of human motion, 2000. NIPS.

[15] F. Pernkopf and J. Bilmes. Discriminative versus generative parameter and structure learning of Bayesian Network Classifiers, 2005. ICML.

[16] A. Rahimi, T. Darrell, and B. Recht. Learning appearance manifolds from video, 2005. CVPR.

[17] F. Sha and F. Pereira. Shallow parsing with conditional random fields, 2003. In Proc. of Human Language Technology-NAACL.

[18] C. Sminchisescu and A. Jepson. Generative modeling for continuous non-linearly embedded visual inference, 2004. ICML.

[19] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Discriminative density propagation for 3D human motion estimation, 2005. CVPR.

[20] T.-P. Tian, R. Li, and S. Sclaroff. Articulated pose estimation in a learned smooth space of feasible solutions, 2005. In Proceedings of IEEE Workshop in CVPR.

[21] R. Urtasun, D. Fleet, A. Hertzmann, and P. Fua. Priors for people tracking from small training sets, 2005. ICCV.

[22] R. Urtasun, D. J. Fleet, and P. Fua. 3D people tracking with Gaussian process dynamical models, 2006. CVPR.

[23] Q. Wang, G. Xu, and H. Ai. Learning object intrinsic structure for robust visual tracking, 2003. CVPR.

Appendix I: CRF Inference and Constraints

We derive the forward recursion, and in turn introduce a set of constraints to be met for the density integrability. The backward recursion which we skip here due to the space limit also yields a similar type of constraints. In the chain-structure as shown in Fig. 1(b), the potential function M_t defined on the clique at time t is:

$$M_1(\mathbf{x}_1|\mathbf{y}) = e^{\mathbf{x}_1'\mathbf{\Lambda}_1\mathbf{x}_1+\mathbf{\Lambda}_b'\mathbf{x}_1+\mathbf{y}_1'\mathbf{\Lambda}_C\mathbf{x}_1},$$
$$M_t(\mathbf{x}_t,\mathbf{x}_{t-1}|\mathbf{y}) = e^{\mathbf{x}_t'\mathbf{\Lambda}\mathbf{x}_t+\mathbf{x}_t'\mathbf{\Lambda}_A\mathbf{x}_{t-1}+\mathbf{y}_t'\mathbf{\Lambda}_C\mathbf{x}_t}, \;\; t\geq 2,$$

where we replace $\mathbf{\Lambda}$ by $\mathbf{\Lambda}_T$ when $t=T$. With the initial condition, $\alpha_1(\mathbf{x}_1|\mathbf{y}) = M_1(\mathbf{x}_1|\mathbf{y})$, the forward message is defined recursively (for $t\geq 2$) as,

$$\alpha_t(\mathbf{x}_t|\mathbf{y}) = \textstyle\int_{\mathbf{x}_{t-1}} \alpha_{t-1}(\mathbf{x}_{t-1}|\mathbf{y})\cdot M_t(\mathbf{x}_t,\mathbf{x}_{t-1}|\mathbf{y}).$$

Since $\alpha_t(\mathbf{x}_t|\mathbf{y})$ is an unnormalized Gaussian, it can be represented by a triple $(r_t,\mathbf{P}_t,\mathbf{q}_t) \in (\mathbb{R},\mathbb{R}^{d\times d},\mathbb{R}^d)$, where $\alpha_t(\mathbf{x}_t|\mathbf{y}) = r_t\exp(\mathbf{x}_t'\mathbf{P}_t\mathbf{x}_t+\mathbf{q}_t'\mathbf{x}_t)$. For a (feasible) $\mathbf{\Theta}_{crf}$,

$$r_t = r_{t-1}\left|-\pi\mathbf{P}_{t-1}^{-1}\right|^{1/2}\exp(-\tfrac{1}{4}\mathbf{q}_{t-1}'\mathbf{P}_{t-1}^{-1}\mathbf{q}_{t-1}),$$
$$\mathbf{q}_t = \mathbf{\Lambda}_C'\mathbf{y}_t - \tfrac{1}{2}\mathbf{\Lambda}_A\mathbf{P}_{t-1}^{-1}\mathbf{q}_{t-1}, \quad \text{for} \;\; 2\leq t\leq T, \quad \text{and}$$
$$\mathbf{P}_t = \mathbf{\Lambda} - \tfrac{1}{4}\mathbf{\Lambda}_A\mathbf{P}_{t-1}^{-1}\mathbf{\Lambda}_A', \quad \text{for} \;\; 2\leq t\leq T-1,$$

with the boundary conditions: $r_1=1, \mathbf{P}_1=\mathbf{\Lambda}_1, \mathbf{q}_1 = \mathbf{\Lambda}_b+\mathbf{\Lambda}_C'\mathbf{y}_1$, and $\mathbf{P}_T = \mathbf{\Lambda}_T - \frac{1}{4}\mathbf{\Lambda}_A\mathbf{P}_{T-1}^{-1}\mathbf{\Lambda}_A'$. The above

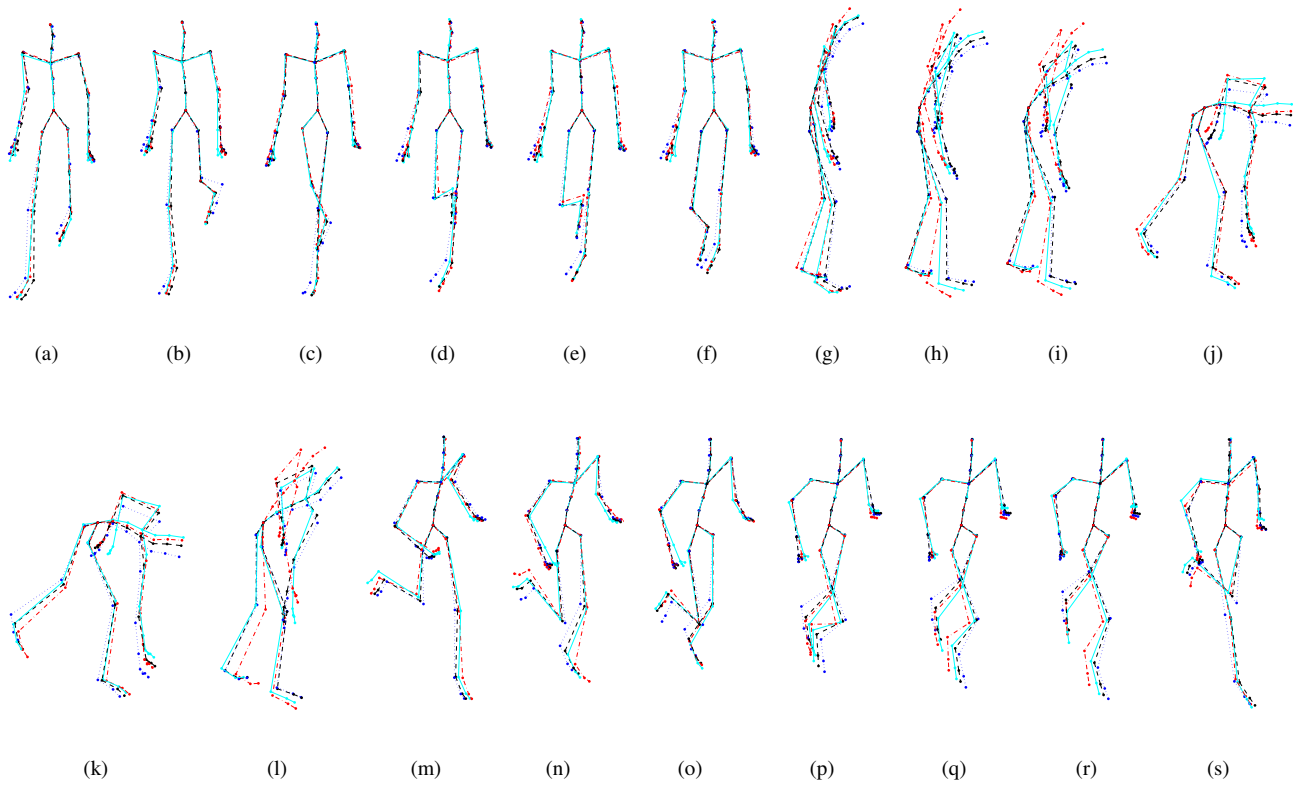


Figure 3. Skeleton snapshots for walking (a−f), picking-up a ball (g−l), and running (m−s): The ground-truth is depicted by solid (cyan) lines, ML by dotted (blue), SCML by dashed (black), and latent variable nonlinear model (LVN) by dotted-dashed (red).

derivation makes sense only if $\mathbf{P}_t$ is negative definite for every t. This guarantees proper forward messages α_t; in particular, the partition function, $Z(\mathbf{y}) = \int_{\mathbf{x}_T} \alpha_T(\mathbf{x}_T|\mathbf{y})$, is finite guaranteeing a proper (integrable) density.

Appendix II: Second-Pass Forward/Backward

Noting that $P(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{y}) \cdot P(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y})$ is a Gaussian on $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$, we denote its means as: $\boldsymbol{\mu}_t^1 \triangleq E[\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}]$ and $\boldsymbol{\mu}_t^2 \triangleq E[\mathbf{x}_{t+1}|\mathbf{x}_{t-1}, \mathbf{y}]$. We define the second-pass forward message as: $\tilde{\alpha}_j(\mathbf{x}_j, \mathbf{x}_{j-1}) = \sum_{t=1}^{j-2} P(\mathbf{x}_j, \mathbf{x}_{j-1}|\overline{\mathbf{x}}_t, \mathbf{y})$, for $j = 3, \ldots, T$. It turns out to be a sum of $(j-2)$ Gaussians in the following reason. Initially for $j = 3$, $\tilde{\alpha}_3(\mathbf{x}_3, \mathbf{x}_2) = P(\mathbf{x}_3, \mathbf{x}_2|\overline{\mathbf{x}}_1, \mathbf{y})$, or equivalently, $P(\mathbf{x}_3|\mathbf{x}_2, \mathbf{y}) \cdot P(\mathbf{x}_2|\overline{\mathbf{x}}_1, \mathbf{y})$ is *a* Gaussian. Suppose that $\tilde{\alpha}_{j-1}(\mathbf{x}_{j-1}, \mathbf{x}_{j-2})$ be a sum of $(j-3)$ Gaussians. In the forward recursion:

$$\tilde{\alpha}_j(\mathbf{x}_j, \mathbf{x}_{j-1}) = P(\mathbf{x}_j|\mathbf{x}_{j-1}, \mathbf{y}) \cdot \int_{\mathbf{x}_{j-2}} \tilde{\alpha}_{j-1}(\mathbf{x}_{j-1}, \mathbf{x}_{j-2}) + P(\mathbf{x}_j|\mathbf{x}_{j-1}, \mathbf{y}) \cdot P(\mathbf{x}_{j-1}|\overline{\mathbf{x}}_{j-2}, \mathbf{y}),$$

it is easy to see that the first term of RHS is a sum of $(j-3)$ Gaussians by the inductive assumption, while the second term is another Gaussian. In particular, it can be shown that the m-th Gaussian component of $\tilde{\alpha}_j(\mathbf{x}_j, \mathbf{x}_{j-1})$, has the mean denoted by $\begin{bmatrix} \tilde{\boldsymbol{\mu}}_j^1(m) \\ \tilde{\boldsymbol{\mu}}_j^2(m) \end{bmatrix}$ and the covariance by $\begin{bmatrix} \tilde{\boldsymbol{\Sigma}}_j^{11}(m) & \tilde{\boldsymbol{\Sigma}}_j^{12}(m) \\ \tilde{\boldsymbol{\Sigma}}_j^{21}(m) & \tilde{\boldsymbol{\Sigma}}_j^{22}(m) \end{bmatrix}$ satisfying the recursion:

$$\tilde{\boldsymbol{\mu}}_j^1(m) = \tilde{\boldsymbol{\mu}}_{j-1}^2(m), \;\; \tilde{\boldsymbol{\mu}}_j^2(m) = \mathbf{F}_j \tilde{\boldsymbol{\mu}}_{j-1}^2(m) + \mathbf{b}_j,$$
$$\tilde{\boldsymbol{\Sigma}}_j^{22}(m) = \mathbf{F}_j \tilde{\boldsymbol{\Sigma}}_{j-1}^{22}(m) \mathbf{F}_j' + \mathbf{R}_j, \;\; \tilde{\boldsymbol{\Sigma}}_j^{11}(m) = \tilde{\boldsymbol{\Sigma}}_{j-1}^{22}(m),$$
$$\tilde{\boldsymbol{\Sigma}}_j^{21}(m) = \tilde{\boldsymbol{\Sigma}}_j^{12}(m)' = \mathbf{F}_j \tilde{\boldsymbol{\Sigma}}_{j-1}^{22}(m),$$

for $m = 1, \ldots, j-3$, and for the last $(j-2)$-nd component,

$$\tilde{\boldsymbol{\mu}}_j^1(j-2) = \boldsymbol{\mu}_{j-1}^1, \;\; \tilde{\boldsymbol{\mu}}_j^2(j-2) = \boldsymbol{\mu}_{j-1}^2,$$
$$\tilde{\boldsymbol{\Sigma}}_j^{22}(j-2) = \mathbf{F}_j \mathbf{R}_{j-1} \mathbf{F}_j' + \mathbf{R}_j, \;\; \tilde{\boldsymbol{\Sigma}}_j^{11}(j-2) = \mathbf{R}_{j-1},$$
$$\tilde{\boldsymbol{\Sigma}}_j^{21}(j-2) = \tilde{\boldsymbol{\Sigma}}_j^{12}(j-2)' = \mathbf{F}_j \mathbf{R}_{j-1}.$$

In the same manner, the backward message, defined as $\tilde{\beta}_j(\mathbf{x}_j, \mathbf{x}_{j-1}) = \sum_{t=j+1}^{T} P(\mathbf{x}_j, \mathbf{x}_{j-1}|\overline{\mathbf{x}}_t, \mathbf{y})$, can be derived as a sum of $(T-j)$ Gaussians. By summing up the expectations with respect to these Gaussians, the last two terms in (14) can be computed, ultimately obtaining the $SCLL$ gradient in (12).