Conditional State Space Models for Discriminative Motion Estimation

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

Abstract

We consider the problem of predicting a sequence of real-valued multivariate states from a given measurement sequence. Its typical application in computer vision is the task of motion estimation. State Space Models are widely used generative probabilistic models for the problem. Instead of jointly modeling states and measurements, we propose a novel discriminative undirected graphical model which conditions the states on the measurements while exploiting the sequential structure of the problem. The major benefits of this approach are: (1) It focuses on the ultimate prediction task while avoiding probably unnecessary effort in modeling the measurement density, (2) It relaxes generative models' assumption that the measurements are independent given the states, and (3) The proposed inference algorithm takes linear time in the measurement dimension as opposed to the cubic time for Kalman filtering, which allows us to incorporate large numbers of measurement features. We show that the parameter learning can be cast as an instance of convex optimization. We also provide efficient convex optimization methods based on theorems from linear algebra. The performance of the proposed model is evaluated on both synthetic data and the human body pose estimation from silhouette videos.

1. Introduction

We consider the regression problem on sequences which can be formulated as estimating a real-valued 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 motion estimation 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 the supervised learning techniques to be exploited. Among those, developing or learning models that directly optimize a tracker's prediction accuracy, $P(\mathbf{X}|\mathbf{Y})$, would be a most desirable approach. However, the study of *discriminative* framework in motion estimation has only recently emerged in the computer vision and related communities [10, 17, 20, 21].

A problem resembling the state estimation, when $\mathbf{x}_t$ is a *discrete label* instead of continuous multivariate, is known as the structured classification. The most popular generative model in this realm is the Hidden Markov Model (HMM). Recently, discriminative models such as Conditional Random Field (CRF) and Maximum Entropy Markov Model (MEMM) have been introduced to address the label prediction problem directly, resulting in performance superior to that of generative models [11, 13].

Despite a broad success of discriminative models in the discrete state domain, the use of discriminative dynamic models for real-valued multivariate state estimation is not widespread. One reason is that posing the density integrability constraints on the parameters of the discriminative models is not always straightforward. Although [10] implicitly resolved this issue by suggesting to learn generative models with discriminative objectives, the learning in general becomes a non-convex optimization problem.

In this paper, we propose a novel conditional undirected graphical model in the continuous state sequence domain. It relaxes generative models' assumption on the conditional independence of the measurements given the states. To address the integrability issue, we first show that the feasible parameter space can be a convex constrained set. This enables us to formulate the parameter learning as an instance of convex optimization. In particular, we introduce a novel membership algorithm for the feasible convex set using theorems from linear algebra. Based on this membership algorithm, we provide efficient convex optimization methods for the parameter learning.

In addition, we propose a new inference algorithm for our conditional model which takes linear time in the measurement dimension as opposed to the cubic time for Kalman filtering/smoothing for generative models. This potentially allows us to incorporate very large number of measurement features. Furthermore, we suggest a novel discriminative feature selection algorithm which enjoys a unique merit of the regression-type undirected model like our model. Finally, we evaluate the performance of the proposed model on both synthetic data and the human body pose estimation from silhouette videos.

2. Background: State Space Model (SSM)

SSM, often called the Linear Dynamical System (LDS), is a generative sequence model with a graphical representation shown in Fig. 1(a). Its joint distribution can be factorized into two parts; the Auto-Regressive (AR) model and the product of Gaussians, namely, $P(\mathbf{X},\mathbf{Y}) = P_{AR}(\mathbf{X}) \cdot P_{Gauss}(\mathbf{Y}|\mathbf{X})$, where

$$\begin{aligned} P_{AR}(\mathbf{X}) &= P(\mathbf{x}_1) \cdot \prod_{t=2}^{T} P(\mathbf{x}_t|\mathbf{x}_{t-1}) \\ &= \mathcal{N}(\mathbf{x}_1; m_0, V_0) \cdot \prod_{t=2}^{T} \mathcal{N}(\mathbf{x}_t; A\mathbf{x}_{t-1}, \Gamma), \quad (1) \end{aligned}$$

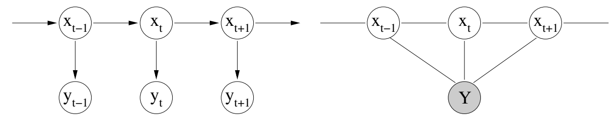
$$P_{Gauss}(\mathbf{Y}|\mathbf{X}) = \prod_{t=1}^{T} P(\mathbf{y}_t|\mathbf{x}_t) = \prod_{t=1}^{T} \mathcal{N}(\mathbf{y}_t; C\mathbf{x}_t, \Omega). \quad (2)$$

Here $\Theta = \{m_0, V_0, A, \Gamma, C, \Omega\}$ is the SSM parameter set.

For many problems that arise with only the measurement data, we learn SSM by optimizing the observation likelihood, $\log P(\mathbf{Y}|\Theta)$, usually by the EM algorithm [6]. In contrast, we assume a *supervised* setting throughout the paper. That is, the train data, $D = \{(\mathbf{X}^i, \mathbf{Y}^i)\}_{i=1}^{n}$, is comprised of both states and measurements. We denote the length of the i-th sequence by T_i. With the knowledge of the states, we have several options in choosing the learning objective. Among them, the standard generative learning (ML) maximizes the data joint likelihood, $\log P(\mathbf{X},\mathbf{Y}|\Theta)$, which gives a closed form solution.

The ML learning fits the model to data jointly on $\mathbf{X}$ and $\mathbf{Y}$, however, its objective is not necessarily the optimal choice for the state prediction. Motivated by this, the discriminative learning approaches have been introduced recently in [10]. Their two learning algorithms, namely, the conditional likelihood maximization (CML) and the slice-wise conditional likelihood maximization (SCML) optimize $\log P(\mathbf{X}|\mathbf{Y},\Theta)$ and $(1/T)\sum_t \log P(\mathbf{x}_t|\mathbf{Y},\Theta)$, respectively. These methods are shown to yield better prediction performance than ML in many situations including the problem of 3D human body pose estimation from silhouette images.

The inference algorithm for SSM is well known as the Kalman filtering/smoothing. The algorithm computes the posterior means and covariances given the measurement $\mathbf{Y}$. Further details on learning and inference for SSM can be found in [2, 10].



(a) SSM (b) Conditional SSM

Figure 1. Graphical Models: SSM and Conditional SSM.

3. Conditional State Space Model (CSSM)

The proposed CSSM has an undirected graphical representation as shown in Fig. 1(b). Note that the model is conditioned on the entire measurement $\mathbf{Y}$ (as shaded), which relaxes SSM's assumption on the conditional independence of the measurements given the states. For discrete $\mathbf{X}$, it resembles CRF of [11], a well known discriminative model for the structured classification. In this sense, CSSM can be seen as an extension of CRF to the real-valued multivariate state domain, as is the SSM for HMM.

The conditional distribution of CSSM follows a Gibbs form, $P(\mathbf{X}|\mathbf{Y},\mathbf{w}) \propto e^{s(\mathbf{X},\mathbf{Y};\mathbf{w})}$, where $s : \mathbb{R}^{dT} \times \mathbb{R}^{kT} \to \mathbb{R}$ is a score function (or a negative energy) parameterized by $\mathbf{w}$. In particular, letting $\mathbf{w} = \{S \in \mathbb{R}^{d\times d}, Q \in \mathbb{R}^{d\times d}, E \in \mathbb{R}^{d\times h}\}$, we define the score function as:

$$\begin{aligned} s(\mathbf{X},\mathbf{Y};\mathbf{w}) &= -\frac{1}{2}\sum_{t=1}^{T} \mathbf{x}_t' S \mathbf{x}_t - \sum_{t=2}^{T} \mathbf{x}_t' Q \mathbf{x}_{t-1} \\ &\quad + \sum_{t=1}^{T} \mathbf{x}_t' E \cdot \phi(\mathbf{Y};t), \end{aligned} \quad (3)$$

where M' indicates the transpose of M, and $\phi(\mathbf{Y};t)$ is a h-dim measurement feature vector specialized at t-th position[1]. By some algebra, we have:

$$P(\mathbf{X}|\mathbf{Y},\{S,Q,E\}) \propto e^{-\frac{1}{2}\mathbf{X}'\mathbf{U}(S,Q)\mathbf{X} + \mathbf{b}(\mathbf{Y};E)'\mathbf{X}}, \quad (4)$$

where $\mathbf{U}(S,Q)$ is a $(T \times T)$ block tri-diagonal matrix, and $\mathbf{b}(\mathbf{Y};E)$ is a $(T \times 1)$ block vector as follows:

$$\mathbf{U} = \begin{bmatrix} S & Q' & 0 & \\ Q & S & Q' & \ddots \\ 0 & Q & S & \ddots \\ & \ddots & \ddots & \ddots \end{bmatrix}, \mathbf{b} = \begin{bmatrix} E \cdot \phi(\mathbf{Y};1) \\ \vdots \\ E \cdot \phi(\mathbf{Y};T) \end{bmatrix}. \quad (5)$$

Note that defining the score function as in Eq. 3 corresponds to have linear Gaussian dynamics features,

[1] One can also define it to be $\phi(\mathbf{Y})$ that depends on the entire $\mathbf{Y}$. In practice, however, one often uses a p-gram feature for some $p > 0$, possibly with some nonlinear mapping Ψ applied to it, namely, $\phi(\mathbf{Y};t) = \Psi(\mathbf{y}_{t-\lfloor \frac{p}{2} \rfloor}, \ldots, \mathbf{y}_{t+\lfloor \frac{p-1}{2} \rfloor})$. We denote dim($\phi(\mathbf{Y};t)$) by h.

$\varphi(\mathbf{x}_t, \mathbf{x}_{t-1}) := \{\mathbf{x}_t\mathbf{x}_t', \mathbf{x}_t\mathbf{x}_{t-1}'\}$, and (non)-linear emission features, $\varphi(\mathbf{x}_t, \mathbf{Y}) := \mathbf{x}_t \cdot \phi(\mathbf{Y}; t)'$. Even though one can define arbitrary triangle features, $\varphi(\mathbf{x}_t, \mathbf{x}_{t-1}, \mathbf{Y})$, in order to fully respect the graph structure in Fig. 1(b), there are obvious advantages of our choice of the features.

First, the score is a quadratic function in terms of $\mathbf{X}$, which implies that $P(\mathbf{X}|\mathbf{Y})$ in Eq. 4 is a Gaussian provided that the second-order coefficient matrix $\mathbf{U}$ is positive (semi)-definite. This enables us to enjoy many nice properties of Gaussian. Most of all, we guarantee that the distribution is *proper*, meaning that it is integrable (*i.e.*, the partition function, $\int_{\mathbf{X}} e^{s(\mathbf{X},\mathbf{Y})}$, is finite). Moreover, the decoding of the states, namely, $arg\max_{\mathbf{X}} s(\mathbf{X}, \mathbf{Y})$, is unique to the global optimum, and can be done efficiently (in $O(T)$) by the inference algorithm introduced in Sec. 3.3.

Second, by defining the dynamics features solely based on $\mathbf{X}$, we have $\mathbf{U}$ independent on $\mathbf{Y}$, which makes it possible to pose $\mathbf{U} \succeq 0$ regardless of $\mathbf{Y}$. In other words, the feasible parameter space, $\{(S, Q, E)|\mathbf{U}(S, Q) \succeq 0\}$, is not affected by $\mathbf{Y}$, allowing that a trained model can be used for unseen test measurement without worrying about the integrability. How the constraint $\mathbf{U}(S, Q) \succeq 0$ can be posed efficiently (in terms of S and Q) will be discussed in Sec. 3.2.

In fact, our choice of the features is reasonable in the following sense: When ignoring the emission features $\varphi(\mathbf{x}_t, \mathbf{Y})$, our model is roughly[2] equivalent to the AR model. On the other hand, by dropping the dynamics features $\varphi(\mathbf{x}_t, \mathbf{x}_{t-1})$, and further assuming that $\phi(\mathbf{Y}; t) = [\ldots, k(\mathbf{y}_t, \mathbf{y}_j), \ldots]'$ for some kernel function $k(\cdot, \cdot)$, our model reduces to a multivariate kernel regression model with i.i.d. pairs $\{(\mathbf{x}_t, \mathbf{y}_t)\}_{t=1}^T$ across t.

3.1. Feasible Parameter Space

We first specify the feasible parameter space in which the parameter learning will take place. The feasible parameters are those that make the Gaussian in Eq. 4 have a positive definite precision matrix $\mathbf{U}_T(S, Q)$. Here we use the subscript T to indicate a $(T \times T)$ block matrix for the length-T sequence. As we *do not know a priori the test sequence length*, it is reasonable to define the feasible parameter space as $\mathcal{W} = \{(S, Q, E)|\mathbf{U}_T(S, Q) \succeq 0, \forall T > 0\}$. Since we have an inclusion relation that $\mathbf{U}_{T+1} \succeq 0$ implies $\mathbf{U}_T \succeq 0$ [3], it is easy to see that $\mathcal{W} = \{(S, Q, E)|\mathbf{U}_\infty(S, Q) \succeq 0\}$. By abusing notation, we use $\mathbf{U}$ to indicate $\mathbf{U}_T$ for either a fixed T or arbitrary large T, which is clear in the context.

3.2. Parameter Learning

We consider learning the parameters $\mathbf{w} = (S, Q, E)$ of the model for a given train data $\{(\mathbf{X}^i, \mathbf{Y}^i)\}_{i=1}^n$. One may expect to exploit the Gaussian properties in learning as $P(\mathbf{X}|\mathbf{Y}, \mathbf{w}) = \mathcal{N}(\mathbf{X}; \mathbf{U}(S, Q)^{-1} \cdot \mathbf{b}(\mathbf{Y}; E), \mathbf{U}(S, Q)^{-1})$. However, this is precluded by the following reasons: (1) the dimensions of data (or the sizes of $\mathbf{U}$) are not the same due to potentially unequal length sequences, (2) inverting $\mathbf{U}$ is usually infeasible, and (3) constraining the structure of $\mathbf{U}$ (Eq. 5) is difficult.

Instead, we optimize the log-likelihood [4] w.r.t. $\mathbf{w} \in \mathcal{W}$:

$$\min_{\mathbf{w}} -\sum_{i=1}^{n} \log P(\mathbf{X}^i|\mathbf{Y}^i, \mathbf{w}), \ \text{ s.t. } \ \mathbf{w} \in \mathcal{W}. \tag{6}$$

Note that the negative log-likelihood is convex in $\mathbf{w}$ as it can be decomposed into the *linear* negative score function and the *convex* log-partition function as follows:

$$-\log P(\mathbf{X}|\mathbf{Y}, \mathbf{w}) = -s(\mathbf{X}, \mathbf{Y}; \mathbf{w}) + \log Z(\mathbf{Y}; \mathbf{w}), \tag{7}$$

where the partition function $Z(\mathbf{Y}; \mathbf{w}) = \int_{\mathbf{X}} e^{s(\mathbf{X},\mathbf{Y};\mathbf{w})}$. Furthermore, it is easy to see that the feasible constraint set $\mathcal{W}$ is a convex (cone) in terms of $\mathbf{w}$. To see this, for $\mathbf{w}_1 = (S_1, Q_1, E_1) \in \mathcal{W}$ and $\mathbf{w}_2 = (S_2, Q_2, E_2) \in \mathcal{W}$, note that $(\nu \cdot \mathbf{w}_1 + \eta \cdot \mathbf{w}_2) \in \mathcal{W}$ for any $\nu, \eta \geq 0$. Thus Eq. 6 is an instance of convex optimization: convex cost + convex conic constraint.

The major concern in this optimization is how to pose the constraint $\mathbf{U} \succeq 0$ efficiently. The straightforward approach is to consider $\mathbf{U}$ as an arbitrary positive definite matrix with a set of linear constraints to define the structure of $\mathbf{U}$ in Eq. 5. More specifically,

$$\begin{gathered}
\mathbf{U}_{I,I+1} = \mathbf{U}_{I-1,I}, \ \ \mathbf{U}_{I+1,I} = \mathbf{U}_{I,I-1}, \ \text{ for } \ I \geq 2, \\
\mathbf{U}_{1,2} = \mathbf{U}_{2,1}', \ \ \mathbf{U}_{I,J} = 0_{d\times d}, \ \text{ for } \ |I - J| \geq 2, \\
\mathbf{U}_{I,I} = \mathbf{U}_{I-1,I-1}, \ \text{ for } \ I \geq 2, \ \text{ and } \ \mathbf{U} \succeq 0,
\end{gathered} \tag{8}$$

where $\mathbf{U}_{I,J}$ is the (I, J)-th block of $\mathbf{U}$. However, plugging it into the general convex solvers would be too slow. Moreover, the popular method like back-projection by eigenvalue truncation (*e.g.* [26]) for the positive definite constraint may be intractable due to the size of $\mathbf{U}$.

Rather, we propose a novel optimization method based on the gradient descent, where the gradient is taken w.r.t. $\mathbf{w} = (S, Q, E)$, instead of the whole block matrix $\mathbf{U}$. The gradient of Eq. 7 is denoted as $\nabla_{\mathbf{w}} = \{\partial S, \partial Q, \partial E\}$, where

$$\begin{aligned}
\partial S &= \frac{1}{2}\sum_{t=1}^{T}\Big(\mathbf{x}_t\mathbf{x}_t' - \mathrm{E}[\mathbf{x}_t\mathbf{x}_t'|\mathbf{Y}]\Big), \\
\partial Q &= \sum_{t=2}^{T}\Big(\mathbf{x}_t\mathbf{x}_{t-1}' - \mathrm{E}[\mathbf{x}_t\mathbf{x}_{t-1}'|\mathbf{Y}]\Big), \\
\partial E &= -\sum_{t=1}^{T}\Big(\mathbf{x}_t \cdot \phi(\mathbf{Y}; t)' - \mathrm{E}[\mathbf{x}_t|\mathbf{Y}] \cdot \phi(\mathbf{Y}; t)'\Big).
\end{aligned} \tag{9}$$

[2] There exists some discrepancy at the boundaries $t = 1$ and $t = T$.

[3] It is because $\mathbf{U}_T$ is a principal sub-matrix of $\mathbf{U}_{T+1}$.

[4] Perhaps one may add a penalty term $\lambda\|\mathbf{w}\|^2$ for some scale hyperparameter λ, however, we skip it in the derivation for simplicity. Note that adding the term does not affect the convexity of our cost function.

The posterior means that appear in $\nabla_{\mathbf{w}}$ can be easily computed by the inference algorithm in Sec. 3.3. Following the general gradient descent approach, the next iterate is determined as $\mathbf{w}_{\eta}^{\text{new}} = \mathbf{w} + \eta \cdot \mathcal{G}(\nabla_{\mathbf{w}})$, where $\eta > 0$ is a chosen step-size, and $\mathcal{G}(\nabla_{\mathbf{w}})$ is an operator that determines the descent direction. For instance, $\mathcal{G}(\nabla_{\mathbf{w}}) = -\nabla_{\mathbf{w}}$ for the steepest descent while $\mathcal{G}(\nabla_{\mathbf{w}}) = -H_{\mathbf{w}}^{-1} \cdot \nabla_{\mathbf{w}}$ for the Newton method, where $H_{\mathbf{w}}$ is the Hessian at $\mathbf{w}$.

The main idea is to adjust the step size η so that $\mathbf{w}_{\eta}^{\text{new}}$ belongs to $\mathcal{W}$. For some trial η, if $\mathbf{w}_{\eta}^{\text{new}} \notin \mathcal{W}$, then we reduce η (*e.g.*, $\eta \leftarrow \frac{1}{2}\eta$). The reason we do not enlarge η is that due to the convexity of $\mathcal{W}$, $\mathbf{w}_{\nu}^{\text{new}} \notin \mathcal{W}$ for any $\nu \geq \eta$, otherwise $\mathbf{w}_{\eta}^{\text{new}}$ that lies on the line segment ending at $\mathbf{w}$ and $\mathbf{w}_{\nu}^{\text{new}}$ should have resided in $\mathcal{W}$. If $\mathbf{w}_{\eta}^{\text{new}} \in \mathcal{W}$, we do a line-search: find $\eta* \in [0, \eta]$ that minimizes the cost. Note that any step size in $[0, \eta]$ results in a feasible model due to convex $\mathcal{W}$.

This approach requires an efficient membership algorithm that determines whether $\mathbf{U}(S, Q) \succeq 0$ or not, for arbitrary S and Q. Fortunately, from linear algebra there are some known theorems that study $\mathbf{U}$ which is also known as a special type of block Toeplitz matrix. The following theorem is from [4, 5] where its proof is therein.

Theorem 1 $\mathbf{U}(S, Q) \succeq 0$ *if and only if* $Z_j \succeq BB'$ *for all* $j \geq 1$, *where* $B = S^{-1/2}QS^{-1/2}$, *and* $\{Z_j\}$ *is recursively defined as:* $Z_0 = I$, $Z_j = I - B'Z_{j-1}^{-1}B,\ j \geq 1$. *(I is the* $(d \times d)$ *identity matrix.)*

Notice that checking the condition $Z_j \succeq BB'$ for $(d \times d)$ matrices is quite easy. The following theorem provides a sufficient condition and a necessary condition which are both non-iterative (See [4] for the proofs).

Theorem 2 *(Sufficiency)* $\mathbf{U}(S, Q) \succeq 0$ *if* $\|B\|_2 \leq 1/2$. *(Necessity)* $\rho(B) \leq 1/2$ *if* $\mathbf{U}(S, Q) \succeq 0$.

Here, $\|B\|_2$ is the 2-norm (or the largest singular value of B) and $\rho(B)$ is the spectral radius of B (or the largest eigenvalue magnitude).

Our iterative membership algorithm (See Alg. 1) is based on Thm. 1, while the conditions in Thm. 2 are also used to quickly filter out the members or the non-members for $\mathcal{W}$. In our experiments, the maximum number of iterations chosen as $J = 1,000$ works well without feasibility violation. The overall learning algorithm that uses the membership algorithm with a bisection step-size adjustment is described in Alg 2.

Algorithm 1 Membership to $\mathcal{W}$

Input: S and Q.
Output: TRUE if $\mathbf{U}(S, Q) \succeq 0$, FALSE otherwise.
if $S \not\succeq 0$ **then**
  return FALSE.
end if
$B = S^{-1/2}QS^{-1/2}$.
if $\rho(B) > 1/2$ **then**
  return FALSE.
end if
if $\|B\|_2 \leq 1/2$ **then**
  return TRUE.
end if
$Z_0 = I$. Choose a sufficiently large number J.
for $j = 1, \ldots, J$ **do**
  $Z_j = I - B'Z_{j-1}^{-1}B$.
  if $Z_j \not\succeq BB'$ **then**
    return FALSE.
  end if
end for
return TRUE.

Algorithm 2 CSSM Learning via Step-Size Adjustment

(a) Initialize $\mathbf{w}$.
(b) Compute $\nabla_{\mathbf{w}}$.
(c) $\eta \leftarrow 1$.
(d) $\mathbf{w}_{\eta} = (S_{\eta}, Q_{\eta}, E_{\eta}) \leftarrow \mathbf{w} + \eta \cdot \mathcal{G}(\nabla_{\mathbf{w}})$.
if Membership(S_{η}, Q_{η}) == TRUE **then**
  Do line-search to find η^*.
  $\mathbf{w} \leftarrow \mathbf{w} + \eta^* \cdot \mathcal{G}(\nabla_{\mathbf{w}})$. goto (b).
else
  $\eta \leftarrow \eta/2$. goto (d).
end if

3.3. Inference and Decoding

The inference is the task to compute the posterior distribution, $P(\mathbf{X}|\mathbf{Y})$, for the given $\mathbf{Y}$, while the decoding is to find its mode, $\mathbf{X}^* = arg\max_{\mathbf{X}} P(\mathbf{X}|\mathbf{Y})$. In our model, due to the conditional Gaussianity (Eq. 4) and the chain-structure (Fig. 1(b)), the Gaussian posteriors on the cliques, namely, $P(\mathbf{x}_t|\mathbf{Y})$ and $P(\mathbf{x}_t, \mathbf{x}_{t-1}|\mathbf{Y})$ completely represents the full posterior $P(\mathbf{X}|\mathbf{Y})$. Moreover, the decoding coincides with the posterior means, that is, $\mathbf{X}_t^* = \mathrm{E}[\mathbf{x}_t|\mathbf{Y}]$. The well-known Kalman filtering/smoothing is the algorithm to compute $P(\mathbf{x}_t|\mathbf{Y})$ and $P(\mathbf{x}_t, \mathbf{x}_{t-1}|\mathbf{Y})$. Here we derive an alternative approach based on the general message passing for the undirected graphical model.

For the given $\mathbf{Y}$, the potential function $M_t(\cdot)$ defined on the clique at time t is:

$$\begin{aligned} M_t(\mathbf{x}_t, \mathbf{x}_{t-1}) &= e^{-\frac{1}{2}\mathbf{x}_t' S \mathbf{x}_t - \mathbf{x}_t' Q \mathbf{x}_{t-1} + \mathbf{x}_t' E \phi(\mathbf{Y}; t)}, \quad t \geq 2, \\ M_1(\mathbf{x}_1) &= e^{-\frac{1}{2}\mathbf{x}_1' S \mathbf{x}_1 + \mathbf{x}_1' E \phi(\mathbf{Y}; 1)}. \end{aligned} \tag{10}$$

We recursively define the forward messages with the initial condition, $\alpha_1(\mathbf{x}_1) = M_1(\mathbf{x}_1)$, and for $t = 2, \ldots, T$,

$$\alpha_t(\mathbf{x}_t) = \int_{\mathbf{x}_{t-1}} \alpha_{t-1}(\mathbf{x}_{t-1}) \cdot M_t(\mathbf{x}_t, \mathbf{x}_{t-1}). \tag{11}$$

Since $\alpha_t(\mathbf{x}_t)$ is an unnormalized Gaussian, it can be represented by a triplet, $(r_t, P_t, q_t) \in (\mathbb{R}, \mathbb{R}^{d\times d}, \mathbb{R}^d)$, which implies $\alpha_t(\mathbf{x}_t) = r_t \exp(-\frac{1}{2}\mathbf{x}_t' P_t \mathbf{x}_t + q_t'\mathbf{x}_t)$. Following the recursion in Eq. 11, we can compute (r_t, P_t, q_t) recursively as follows:

$$
\begin{aligned}
t = 1;\quad & r_1 = 1,\ \ P_1 = S_1,\ \ q_1 = E\cdot\phi(\mathbf{Y};1),\\
t \geq 2;\quad & r_t = r_{t-1}\cdot \left|2\pi P_{t-1}^{-1}\right|^{1/2}\cdot e^{\frac{1}{2}q_{t-1}' P_{t-1}^{-1} q_{t-1}},\\
& P_t = S - Q P_{t-1}^{-1} Q',\\
& q_t = E\cdot\phi(\mathbf{Y};t) - Q P_{t-1}^{-1} q_{t-1}.
\end{aligned}
\tag{12}
$$

It is important to notice that Eq. 12 makes sense only if the integrability condition in Eq. 11, that is, $P_t \succeq 0$ for all t, is satisfied. we can guarantee that for the feasible model that resides in $\mathcal{W}$, this condition always holds. The partition function is easily obtained from the forward messages, in particular, $Z(\mathbf{Y}) = \int_{\mathbf{x}_T} \alpha_T(\mathbf{x}_T)$[5].

Similarly, we can define the backward messages with the initial, $\beta_T(\mathbf{x}_T) = 1$, and for $t < T$,

$$
\beta_t(\mathbf{x}_t) = \int_{\mathbf{x}_{t+1}} \beta_{t+1}(\mathbf{x}_{t+1})\cdot M_{t+1}(\mathbf{x}_{t+1}, \mathbf{x}_t). \tag{13}
$$

The triplet, $(h_t, F_t, g_t) \in (\mathbb{R}, \mathbb{R}^{d\times d}, \mathbb{R}^d)$, for representing $\beta_t(\mathbf{x}_t) = h_t \exp(-\frac{1}{2}\mathbf{x}_t' F_t \mathbf{x}_t + g_t'\mathbf{x}_t)$, can be computed as:

$$
\begin{aligned}
t = T;\quad & h_T = 1,\ \ F_T = 0_{d\times d},\ \ g_T = 0_{d\times 1},\\
t < T;\quad & h_t = h_{t+1}\cdot \left|2\pi \tilde{F}_{t+1}^{-1}\right|^{1/2}\cdot e^{\frac{1}{2}\tilde{g}_{t+1}' \tilde{F}_{t+1}^{-1}\tilde{g}_{t+1}},\\
& F_t = -Q'\tilde{F}_{t+1}^{-1} Q,\ \ g_t = -Q'\tilde{F}_{t+1}^{-1}\tilde{g}_{t+1},
\end{aligned}
\tag{14}
$$

where $\tilde{F}_t := F_t + S$ and $\tilde{g}_t := g_t + E\cdot\phi(\mathbf{Y};t)$. Another integrability condition in Eq. 13, $\tilde{F}_t \succeq 0$ for all t, also holds for the feasible parameters.

Observing that $P(\mathbf{x}_t|\mathbf{Y}) = \frac{1}{Z(\mathbf{Y})}\alpha_t(\mathbf{x}_t)\cdot\beta_t(\mathbf{x}_t)$ and $P(\mathbf{x}_t, \mathbf{x}_{t-1}|\mathbf{Y}) = \frac{1}{Z(\mathbf{Y})}\alpha_{t-1}(\mathbf{x}_{t-1})\cdot M_t(\mathbf{x}_t, \mathbf{x}_{t-1})\cdot\beta_t(\mathbf{x}_t)$, we have the following Gaussian posterior:

$$
P(\mathbf{x}_t, \mathbf{x}_{t-1}|\mathbf{Y}) = \mathcal{N}\Big(\begin{bmatrix}\mu_t\\ \mu_{t-1}\end{bmatrix}, \begin{bmatrix}\Sigma_t & \Sigma_{t,t-1}\\ \Sigma_{t,t-1}' & \Sigma_{t-1}\end{bmatrix}\Big), \tag{15}
$$

where $\mu_t = \mathrm{E}[\mathbf{x}_t|\mathbf{Y}] = (P_t + F_t)^{-1}\cdot(q_t + g_t)$, $\Sigma_t = \mathrm{V}(\mathbf{x}_t|\mathbf{Y}) = (P_t + F_t)^{-1}$, and $\Sigma_{t,t-1} = \mathrm{Cov}(\mathbf{x}_t, \mathbf{x}_{t-1}|\mathbf{Y}) = -\tilde{F}_t^{-1} Q(P_{t-1} - Q'\tilde{F}_t^{-1}Q)^{-1}$. Notice that for a feasible model, all the posterior quantities are proper (*i.e.*, covariances are positive definite) since they are marginals of the full Gaussian (Eq. 4).

It is interesting to consider the proposed inference algorithm in light of the standard Kalman smoothing for SSM. In time complexity, the proposed method takes $O(T\cdot(d^3 + dh))$ time, while the Kalman smoothing requires $O(T\cdot(d^3 + h^3 + dh))$ [6], where we assume that the corresponding SSM employs $\phi(\mathbf{Y};t)$ as its t-th observation. The linear time in the measurement dimension enables CSSM to incorporate a large number of measurement features. In terms of numerical issues, we empirically observe that the proposed inference algorithm in conjunction with the step size adjustment (Sec. 3.2) is numerically more stable than the Kalman smoothing for SSM, especially when d is large.

4. Variants of CSSM

In this section we discuss two extensions to CSSM: the feature selection algorithm and inclusion of the nonlinear dynamics features.

4.1. Discriminative Feature Selection

A typical benefit of regression-type undirected conditional models is their eligibility for feature selection, especially when the data contains a combination of relevant and a large number irrelevant (noisy) features. Given a pool of candidate features denoted as $\mathcal{F}$, our discriminative feature selection algorithm selects a sparse subset that contains the most salient and discriminative features.

The algorithm is essentially a greedy forward selection (boosting): at each stage, a new feature is added to the current feature set. We follow [12]'s functional gradient derivation to compute the gain for each candidate feature $f \in \mathcal{F}$. For the current feature set $\phi(\mathbf{Y})$, we have the following criterion (gain) for f:

$$
L(f;\phi(\mathbf{Y})) = \Big\| \sum_t f(\mathbf{Y};t)\cdot\big(\mathbf{x}_t - \mathrm{E}_{\phi(\mathbf{Y})}[\mathbf{x}_t|\mathbf{Y}]\big)\Big\|_1, \tag{16}
$$

where the expectation is taken with respect to the learned model with the current feature set $\phi(\mathbf{Y})$.

The gain can be interpreted as a discrepancy between the empirical features and the model expected features. However, Eq. 16 is slightly different from the derivation in [12] in that we take $\|\cdot\|_1$ to make the multivariate quantity to scalar. Notice that if the state is discrete as in [12], the quantity in $\|\cdot\|_1$ is scalar. Once the gains are computed, the new feature to be added is $f^* = \max_{f\in\mathcal{F}} L(f;\phi(\mathbf{Y}))$.

This also makes intuitive sense because the feature that is the most positively correlated with the current error is the one to be selected. However, according to Eq. 16, simply scaling up f results in a better feature. It is thus crucial to normalize the features before computing the gains, namely, $f(\mathbf{Y};t) = f(\mathbf{Y};t)/|\sum_t f(\mathbf{Y};t)|$. After adding the feature f^* to our model, we exclude it from $\mathcal{F}$, followed by learning parameters of the new model.

[5]This must be the same to the Gaussian normalizing constant from Eq. 4, namely, $(2\pi)^{dT/2}\cdot|\mathbf{U}|^{-1/2}\cdot\exp(\frac{1}{2}\mathbf{b}'\mathbf{U}^{-1}\mathbf{b})$.

[6]Recently, it has been shown that the cubic in h can be reduced to quadratic, using the least-square method. See [25] for details.

4.2. Nonlinear Dynamics

Although CSSM is derived for linear dynamics features, it can be easily extended to incorporate nonlinear dynamics features. Motivated by [19], we suggest to use an invertible nonlinear mapping $\Psi(\cdot)$ to obtain a new embedded states $\zeta_t = \Psi(\mathbf{x}_t)$. For instance, if the mapping is related to a polynomial kernel with odd degree, it is invertible and the inversion can be found in a closed form. In the embedded space we may assume that ζ conforms to linear dynamics despite nonlinear dynamics in the original space. In the linear ζ space, all the previous derivations do not change. It is only required to transform the estimated ζ back to the original space: $\mathbf{x}_t = \Psi^{-1}(\zeta_t)$.

5. Related Work

While discriminative modeling of discrete-state dynamics such as CRFs has received significant attention recently, similar models in the continuous state space has been rarely explored. Recent works that deal with discriminative dynamics models in continuous domain do not consider the integrability (or feasibility) conditions on the parameter space discussed here. Hence such approaches had to resort to sampling or discretization to circumvent the integrability issue. For instance, [17] introduced a novel conditional model with switching latent variables, however, the inference is done by MCMC which would be computationally intensive for high dimensions. [21] tried to discretize the continuous state space into grids, which may require a huge number of poses to be known in order to have a good approximation.

In robotics community, [1] empirically studied several objectives for learning of dynamical systems. In contrast to [1]'s ad-hoc optimization method, [10] provided efficient gradient-based optimization algorithms for discriminative objectives to dynamical systems. Compared to CSSM, their objectives are optimized w.r.t. the parameters of the generative model, which results in non-convex optimization. [20] successfully extended MEMM of [13] with Bayesian mixtures of experts for 3D pose estimation. However, their ability to successfully infer states from observations mostly depends on the modeling capacity of the regression functions, not on the choice of the discriminative dynamic model objective.

Recent work on the human motion tracking encompasses a variety of approaches, among them dynamic model based methods ([8, 14, 15]), nonlinear manifold embedding ([3, 16, 18]), and Gaussian process based latent variable models ([24]). In our approach, we consider a conditional dynamic models, formulate a convex program, and show that it can be used for accurate and computationally efficient dynamic pose estimation.

Model	CSSM	ML	CML	SCML
L2-Error	1.54 ± 0.21	1.79 ± 0.26	1.59 ± 0.22	1.30 ± 0.12
Log-Perp.	4.26 ± 0.35	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 conditional state space model in a set of experiments that include synthetic data as well as human motion capture data. For the baseline comparison, we include SSM, learned via not only the standard maximum likelihood estimator (ML), but also the discriminative learning algorithms, CML and SCML, proposed in [10]. In notation, for instance, we denote CML-learned SSM by SSM-CML or simply CML interchangeably. Unless stated otherwise, the gradient-based learning algorithms (our CSSM, SSM-CML, and SSM-SCML) start from SSM-ML as an initial iterate. For CSSM, we can also easily find the CSSM parameters that correspond to the SSM-ML parameters.

6.1. Synthetic Data

We devise a dynamic model from which the sequences are sampled. The model has second-order dynamics and emission, specifically, $\mathbf{x}_t = \frac{1}{2}\mathbf{A}_1\mathbf{x}_{t-1} + \frac{1}{2}\mathbf{A}_2\mathbf{x}_{t-2} + \mathbf{v}_t$, and $\mathbf{y}_t = \frac{1}{2}\mathbf{C}_1\mathbf{x}_t + \frac{1}{2}\mathbf{C}_2\mathbf{x}_{t-1} + \mathbf{w}_t$, where $\mathbf{v}_t$ and $\mathbf{w}_t$ are Gaussian white noises. Note that this model must be structurally more complex than SSM.

We sampled 10 sequences of lengths $\sim\ 150$ with $\dim(\mathbf{x}_t) = 3$ and $\dim(\mathbf{y}_t) = 2$. Table 1 shows the performance of the competing models for the leave-one-out validation. Here we use two error measures: (1) L2-Error is the norm-2 difference averaged over time slices, namely, $(1/T)\sum_{t=1}^{T} ||\overline{\mathbf{x}}_t - \mathbf{m}_t||_2$, where $\overline{\mathbf{x}}$ is the ground truth, and $\mathbf{m}$ is the model estimated state sequence. (2) Log-Perplexity, defined as $-(1/T)\sum_{t=1}^{T} \log P(\overline{\mathbf{x}}_t|\mathbf{Y},\Theta)$, measures the variance of the estimate which is not usually captured by L2-Error measure. Note that the smaller numbers are better for both measures.

Although SSM-CML and CSSM optimize the same objective, the SSM-CML learning is in general non-convex with many local optima. When the learning starts from SSM-ML, both models exhibit very similar test errors as shown in Table 1. To see the difference, we learn both models with the same initial parameters that are randomly chosen. CSSM recorded test error 1.83 which is not very close to the one at the global optimum, however, this is much better than 2.30, the test error of SSM-CML. This implies that CSSM is less sensitive to the initial iterate.

As our result in Table 1 demonstrates, SCML is empirically more robust than CML and CSSM, especially with sub-optimal model structures [9, 10]. However, its learning time is quadratic in sequence length, which may prevent it

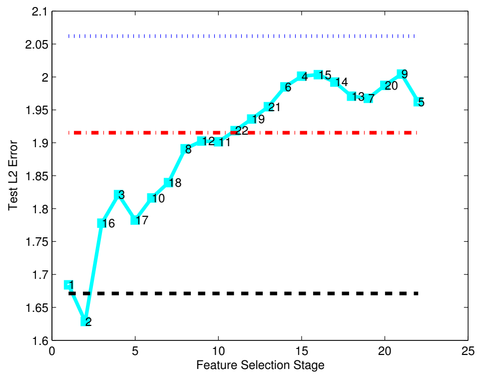


Figure 2. Feature selection. The solid (cyan) line indicates test L2 errors of CSSM learned with the features selected by our algorithm. It starts from empty feature, adds a feature at each stage, up to full features (22-nd stage). Each number indicates the feature index (dim), where the original relevant features corresponds to 1 and 2. The test errors of SSM learned with full 22 features are depicted as: ML by dotted (blue), CML by dotted-dashed (red), and SCML by dashed (black).

from being applied to large scale data.

Feature Selection

We next apply the feature selection algorithm discussed in 4.1. In addition to the original $(k = 2)$-dim measurement features, we add extra 20 irrelevant noisy features that are generated randomly. First, SSM is learned via ML, CML, and SCML with this 22-dim feature set, where the test L2 errors are 2.06, 1.92, and 1.67, respectively. On the other hand, CSSM feature selection algorithm (starting from empty feature) successfully finds the two original (relevant) features for its first two stages. Fig. 2 shows the features found by our feature selection algorithm and the corresponding test errors.

6.2. Human Motion Data

We evaluate the performance of the proposed method on the task of 3D body pose estimation from human motion data. The CMU motion capture dataset[7] provides the ground-truth body poses (3D joint angles), which makes it possible to compare competing methods quantitatively. Among the original 59 angles at 31 articulation points, we selectively use $(d = 39)$-dim by excluding less significant joint angles around fingers and toes as well as those that rarely vary over time.

We focus on the walking motion. We gather 5 sequences from one subject and perform leave-one-out validation. The sequence lengths are about 150. The measurement features are $(k = 10)$-dim Alt-Moments extracted from the silhouette image (e.g., [22]). The images are taken by a single camera at a fixed view.

Since the 3D joint angles and the moment features tend to conform to a rather complex nonlinear structure, it will be interesting to compare with standard nonlinear dynamic models that may contain some latent variables. At the end of the section, we briefly describe two popular nonlinear models that are also used in our performance comparison.

Table 2 shows the average test L2 errors of the competing methods. The proposed CSSM has significantly lower errors than SSM-ML and more importantly, SSM-CML. This indicates that the discriminative model can achieve significant improvement beyond the discriminative learning of generative models. CSSM also has superior performance to the nonlinear models. This implies that the discriminative model even with a simple linear structure can have potential to outperform complex nonlinear models in terms of generalization performance. Moreover, the proposed inference algorithm is much faster than the approaches based on particles or nonlinear optimization used for the nonlinear models (*e.g.*, [20, 23, 24]).

Nonlinear Dynamical System (NDS)

While NDS has the same graphical structure as SSM, it is modeled by nonlinear dynamics and emission functions. The nonlinear functions are often represented as a sum of RBF kernels and linear functions, namely, $\mathbf{x}_t = C_x k(\mathbf{x}_{t-1}) + A_x \mathbf{x}_{t-1} + \mathbf{v}_t$ and $\mathbf{y}_t = C_y k(\mathbf{x}_t) + A_y \mathbf{x}_t + \mathbf{w}_t$. Here $k(\mathbf{x}_t) := [k(\mathbf{x}_t, \mathbf{u}_1), \ldots, k(\mathbf{x}_t, \mathbf{u}_L)]'$ is a vector of RBF kernels evaluated on the predefined centers $\{\mathbf{u}_l\}$. In order to reduce the overhead maintaining the entire train poses, one often includes the most salient poses $\mathbf{u}_l$ to form a sparse active set.

Although NDS is very powerful in representation, its generalization performance is in general very sensitive to the choice of the active set as well as the kernel scale (Gaussian precision) hyperparameters. In the experiment, we adopt a greedy kernel selection technique which adds a pose from the train pool one at a time, according to a certain criterion (e.g., maximizing data likelihood). The size of the active set is chosen by cross validation among several candidates. The kernel scale parameters are estimated in a way that the neighbor points of each center $\mathbf{u}_l$ have kernel values one half of its peak value [7]. This generates a reasonably smooth kernel. We learn NDS generatively by maximizing the joint likelihood.

Latent Variable Nonlinear Model (LVN)

As it is broadly believed that the realizable poses live in a low dimensional latent space, it is useful to introduce latent variables $\mathbf{z}_t$ embedded point of the pose $\mathbf{x}_t$. A typical way

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

CSSM	ML	CML	SCML	NDS	LVN
16.85	19.20	18.31	17.19	18.91	18.01

Table 2. Average test L2 errors for CMU walking motion data.

to represent LVN is to place dynamics on $\mathbf{z}_t$, while assuming $\mathbf{x}_t$ and $\mathbf{y}_t$ are generated nonlinearly from $\mathbf{z}_t$. Learning LVN is done by the EM algorithm on the linear approximated model as introduced in [7]. Initial subspace mapping for LVN is determined by PCA dim-reduction on the train poses. Similar to NDS, the hyperparameters (e.g., the number of kernel centers) are determined by cross validation. In the experiment, we set $\dim(\mathbf{z}_t) = 3$.

7. Conclusion

We proposed a novel discriminative undirected graphical model for dynamics in real-valued multivariate state domain. We address the integrability issue by casting the learning problem as a convex optimization subject to a convex constrained feasible parameter set. The proposed inference algorithm takes linear time in the measurement dimension as opposed to the cubic time for Kalman filtering, which allows us to incorporate large numbers of measurement features. In the human body pose estimation problem, the proposed model achieved superior prediction performance to that of the complex nonlinear dynamic models. As a future work, we plan to extend our model to piece-wise linear models such as switching LDS (e.g., [15]) which can handle composite motions of different styles and types.

Acknowledgements

This work was supported in part by NSF grant IIS-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] Y. Bar-Shalom and X.-R. Li. *Estimation and tracking : principles, techniques, and software*. Boston : Artech House, 1993.

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

[4] J. C. Engwerda. On the existence of the positive definite solution of the matrix equation $X + A^T X^{-1} A = I$. *Linear Algebra and Its Application*, 194:91–108, 1993.

[5] J. C. Engwerda, A. C. M. Ran, and A. L. Rijkeboer. Necessary and sufficient conditions for the existence of a positive definite solution of the matrix equation $X + A^* X^{-1} A = Q$. *Linear Algebra and Its Application*, 186:255–275, 1993.

[6] Z. Ghahramani and G. E. Hinton. Parameter estimation for linear dynamical systems, 1996. University of Toronto Technical Report CRG-TR-96-2.

[7] Z. Ghahramani and S. Roweis. Learning nonlinear dynamical systems using an EM algorithm, 1999. In Advances in NIPS.

[8] 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.

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

[10] M. Kim and V. Pavlovic. Discriminative learning of dynamical systems for motion tracking, 2007. CVPR.

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

[12] J. Lafferty, X. Zhu, and Y. Liu. Kernel conditional random fields: Representation and clique selection, 2004. ICML.

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

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

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

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

[17] D. Ross, S. Osindero, and R. Zemel. Combining discriminative features to infer complex trajectories, 2006. ICML.

[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. Conditional visual tracking in kernel space, 2005. NIPS.

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

[21] L. Taycher, D. Demirdjian, T. Darrell, and G. Shakhnarovich. Conditional Random People: Tracking humans with CRFs and grid filters, 2006. CVPR.

[22] 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.

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

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

[25] R. van der Merwe and E. Wan. The square-root unscented Kalman filter for state and parameter-estimation, 2001. In Proceedings of ICASSP.

[26] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information, 2002. NIPS.