

Simultaneous Learning of Nonlinear Manifold and Dynamical Models for High-dimensional Time Series

Rui Li, Tai-Peng Tian and Stan Sclaroff *

Computer Science Department, Boston University

{lir, tiantp, sclaroff}@cs.bu.edu

Abstract

The goal of this work is to learn a parsimonious and informative representation for high-dimensional time series. Conceptually, this comprises two distinct yet tightly coupled tasks: learning a low-dimensional manifold and modeling the dynamical process. These two tasks have a complementary relationship as the temporal constraints provide valuable neighborhood information for dimensionality reduction and conversely, the low-dimensional space allows dynamics to be learnt efficiently. Solving these two tasks simultaneously allows important information to be exchanged mutually. If nonlinear models are required to capture the rich complexity of time series, then the learning problem becomes harder as the nonlinearities in both tasks are coupled. The proposed solution approximates the nonlinear manifold and dynamics using piecewise linear models. The interactions among the linear models are captured in a graphical model. By exploiting the model structure, efficient inference and learning algorithms are obtained without oversimplifying the model of the underlying dynamical process. Evaluation of the proposed framework with competing approaches is conducted in three sets of experiments: dimensionality reduction and reconstruction using synthetic time series, video synthesis using a dynamic texture database, and human motion synthesis, classification and tracking on a benchmark data set. In all experiments, the proposed approach provides superior performance.

1. Introduction

High-dimensional time series encountered in computer vision tasks often have highly redundant representations. For instance, in video data, strong correlations exist among neighboring pixels in space-time. Similarly in human motion capture, 30-60 degrees of freedom are used to represent motion; however, movement at one joint is often coupled with motions at other joints. Such correlations significantly reduce the degrees of freedom in the time series.

Thus it can be argued that such time series can be economically represented by a dynamical process on a low-dimensional manifold. Recovering such representations depends on two distinct yet tightly coupled tasks: reducing the dimensionality and modeling the dynamical process.

We advocate for recovering the dynamical model parameters in concert with manifold learning. In isolation, recovering the dynamical model without dimensionality reduction is computationally inefficient. Conversely, dimensionality reduction without temporal information is “blind” as neighborhood information can only be approximated using Euclidean distance rather than using knowledge of temporal neighbors. Solving these two tasks simultaneously allows important information to be exchanged mutually. However, as nonlinear models are required to capture the rich complexity of these time series, the learning problem becomes harder as the nonlinearities in both tasks are now coupled.

To make learning tractable, we employ a divide and conquer approach: the nonlinear manifold is approximated by piecewise linear regions. Each local region is associated with its own linear dimensionality reducer and a linear dynamical model. Coordination among the local linear dimensionality reducers is needed to ensure consistent coordinates for the time series on the piecewise representation of the manifold, and to assure consistency among local linear dynamical models. Similarly, the linear dynamical models that approximate the nonlinear dynamical process on the manifold must be consistent with the observed high-dimensional time series. Such coordination and consistency constraints are enforced by estimating the parameters of the piecewise linear models together with the coordination parameters during manifold learning. Learning of the coordinated, piecewise representation is efficient, without oversimplifying the model of the underlying dynamical process.

Evaluation of our framework vs. competing approaches is conducted in experiments with three common data sets: dimensionality reduction and reconstruction for synthetic time series [13], synthesis of video textures [29], and human motion synthesis, classification and tracking on the benchmark of [22]. In all experiments, the proposed model pro-

¹This work was supported in part by NSF grants IIS 0308213, IIS 0329009, and CNS 0202067.

vides superior performance.

2. Related Work

NLDR techniques can be classified into embedding-based vs. mapping-based techniques. Embedding-based techniques [2, 10, 18, 23, 27] model the structure of the data that generates the manifold without providing mapping functions between the observation space and the latent space. Hence it is difficult to map new data into the latent space or from the latent space back to the observation space using embedding-based techniques. Regression methods have been used in [5, 24] to learn the mapping functions after the embedding.

Mapping-based techniques learn the nonlinear mapping functions either by modeling the nonlinear functions directly [3, 11, 21] or by using a combination of local linear models [4, 19, 26] during dimensionality reduction. These NLDR algorithms assume that the data are independently and identically distributed (i.i.d) even in applications where they are temporally correlated [9, 12, 28, 30]. Ignoring the temporal correlations causes inconsistencies in the learnt manifold as shown in [13, 14, 32].

To analyze time series, nonlinear dynamical models have been actively studied. Two main themes in nonlinear dynamical model learning are the use of a combination of linear models (e.g., [16]), and the use of nonlinear functions directly [8, 17]. The key problem with estimating dynamical model parameters in the observation space is that model parameter estimation does not scale well with the high dimensionality of the state space.

Some NLDR algorithms have been extended to incorporate temporal correlations during learning. Recent work [14, 32] combines a dynamical model with the standard Gaussian Process Latent Variable Model (GPLVM) [11] by augmenting the GPLVM cost function with terms from the kernel dynamics matrix. In [14, 32], the dynamical model parameters are considered as incidental and are marginalized out. Hence, this extension cannot be directly used for activity classification. Another concern with the approach of [14, 32] is the kernel sparsification problem; as there is no principled way to choose an active set for a dynamic sequence. Without sparsification, the full kernel matrix has to be inverted at each iteration of learning. Thus, it is difficult to apply the dynamics extension of GPVLM to large data sets. To avoid the discontinuity problem caused by the use of an active set, Snelson and Ghaharmani propose sparsification techniques that make use of psuedo-inputs [25]. There are still two open problems with [25]: how to choose the number of psuedo-inputs, and how to avoid overfitting. Furthermore, the success of applying such techniques to human tracking has yet to be demonstrated.

Lin, *et al.* [13] propose learn a piecewise linear model together with a global linear dynamical model in the latent

space. Their work extends [19] by using a global linear dynamical model in the latent space together with learning of the mapping functions between the latent space and the observation space. Their extension works well with large training data sets. However, the global linear dynamical model assumed in the latent space limits the types of dynamics that can be modeled.

3. Formulation

Let $\mathcal{X}_T = \{\mathbf{x}_0, \dots, \mathbf{x}_{T-1}\}$ be the high-dimensional time series and $\mathcal{G}_T = \{\mathbf{g}_0, \dots, \mathbf{g}_{T-1}\}$ be the corresponding low-dimensional time series. We use \mathbb{R}^D to denote the high-dimensional observation space and \mathbb{R}^d to represent the low-dimensional latent space, hence $D \gg d$. We define $f_{dyn} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ to be the nonlinear dynamical function that drives the low-dimensional time series; assuming a first-order Markov process, we have:

$$\mathbf{g}_t = f_{dyn}(\mathbf{g}_{t-1}) + \mathbf{n}_{\mathbf{g},t}, \quad (1)$$

where $\mathbf{n}_{\mathbf{g},t}$ is a zero-mean, white Gaussian noise process. To map \mathbf{g}_t to observation \mathbf{x}_t , we define the nonlinear mapping function $f_{\mathbf{g} \rightarrow \mathbf{x}} : \mathbb{R}^d \rightarrow \mathbb{R}^D$ to be:

$$\mathbf{x}_t = f_{\mathbf{g} \rightarrow \mathbf{x}}(\mathbf{g}_t) + \mathbf{n}_{\mathbf{x},t}, \quad (2)$$

where $\mathbf{n}_{\mathbf{x},t}$ is also a zero-mean, white Gaussian noise process. Therefore, our problem can be formulated as a general dynamical system with nonlinear dynamical function f_{dyn} defined on the low-dimensional space and the nonlinear measurement function $f_{\mathbf{g} \rightarrow \mathbf{x}}$ that maps the latent coordinate \mathbf{g}_t to \mathbf{x}_t in the observation space.

We propose to approximate f_{dyn} and $f_{\mathbf{g} \rightarrow \mathbf{x}}$ using piecewise linear functions. The interactions among the linear functions are formulated in a graphical model as shown in Fig 1(b). Simultaneous learning of f_{dyn} (dynamical process) and $f_{\mathbf{g} \rightarrow \mathbf{x}}$ (and hence $f_{\mathbf{x} \rightarrow \mathbf{g}}$ if such mapping is bi-directional) is formulated as the model parameter estimation problem in this graphical model.

3.1. Mapping Functions $f_{\mathbf{g} \rightarrow \mathbf{x}}$ and $f_{\mathbf{x} \rightarrow \mathbf{g}}$

Mapping functions $f_{\mathbf{g} \rightarrow \mathbf{x}}$ and $f_{\mathbf{x} \rightarrow \mathbf{g}}$ allow us to associate high-dimensional observations with their corresponding low-dimensional representations and vice versa.

Mixtures of factor analyzers (MFA) [7] achieve the nonlinear approximation of $f_{\mathbf{g} \rightarrow \mathbf{x}}$ by using multiple linear factor analyzers (FA) [20]. Unfortunately, this type of mixture model does not describe a single, coherent low-dimensional coordinate system for the data since there is no constraint for the local coordinates of each component to agree. In our formulation, we prefer a global coordination scheme to produce a manifold so that as one traverses a connected path on the manifold, the internal coordinates change smoothly

and continuously even when the path crosses the domains of many different local models.

The graphical model of our globally coordinated MFA is shown in Fig. 1(a) where we use s to index the factor analyzers in the mixture. As in [19], the global coordination is achieved by maximizing the likelihood of data with an additional variational penalty term to encourage the internal coordinates of the FAs to agree. However, our model makes the training more efficient; it circumvents the need to solve the specific alignment transformations between local coordinates of individual FAs to the corresponding global coordinates. This is achieved by assuming a deterministic relationship between the local coordinates and their corresponding global coordinates. Removing local representations from the graphical model leads to a closed form solution for the optimal model parameters given the variational parameters as being observed in [31].

The additional variational penalty term is enforced by introducing a family of unimodal distributions of factorized form: $Q(\mathbf{g}, s|\mathbf{x}_n) = Q(\mathbf{g}|\mathbf{x}_n)Q(s|\mathbf{x}_n)$, where $Q(\mathbf{g}|\mathbf{x}_n) \sim \mathcal{N}(\mathbf{g}_n, \Sigma_n)$ and $Q(s|\mathbf{x}_n) = q_{n,s}$ is a scalar, and by encouraging $P(\mathbf{g}, s|\mathbf{x}_n)$ to be close to some member $Q(\mathbf{g}, s|\mathbf{x}_n)$ of this family; this implies $P(\mathbf{g}|\mathbf{x}_n, s_1) \approx P(\mathbf{g}|\mathbf{x}_n, s_2)$ for the same \mathbf{x}_n . Note that the factorized form of $Q(\mathbf{g}, s|\mathbf{x}_n)$ implies that \mathbf{g} is independent of the mixture component s given data point \mathbf{x}_n . Furthermore, $Q(\mathbf{g}|\mathbf{x}_n)$ is unimodal. These are exactly the constraints we want to impose on $P(\mathbf{g}|\mathbf{x}_n, s)$ to enforce global coordination. Now the objective function is a lower-bound on data log-likelihood using variational distribution $Q(\mathbf{g}, s|\mathbf{x}_n)$:

$$\Phi = \sum_{n,s} \int Q(\mathbf{g}, s|\mathbf{x}_n) \log \frac{P(\mathbf{x}_n, \mathbf{g}, s)}{Q(\mathbf{g}, s|\mathbf{x}_n)} d\mathbf{g}. \quad (3)$$

We estimate the MFA parameters together with the variational regularizing parameters $\{\mathbf{g}_n, \Sigma_n, q_{n,s}\}$ by iteratively optimizing Φ via coordinate ascent in learning.

Hence the mapping functions $f_{\mathbf{g} \rightarrow \mathbf{x}}$ and $f_{\mathbf{x} \rightarrow \mathbf{g}}$ are described by the following probabilistic relations between \mathbf{x}_n and \mathbf{g} :

$$P(\mathbf{g}|\mathbf{x}_n) = \sum_s P(\mathbf{g}|\mathbf{x}_n, s)P(s|\mathbf{x}), \quad (4)$$

$$P(\mathbf{x}_n|\mathbf{g}) = \sum_s P(\mathbf{x}_n|\mathbf{g}, s)P(s|\mathbf{g}). \quad (5)$$

We extend the globally coordinated MFA to incorporate dynamics in the following section so that f_{dyn} can be learnt together with the mapping functions.

3.2. Incorporate Dynamics

Given the model depicted in Fig. 1(a), we can extend it to incorporate temporal information in the form of the graphical model shown in Fig. 1(b). Now the observations

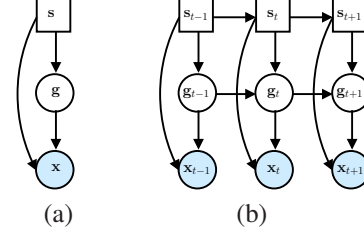


Figure 1. (a). The modified model for the globally coordinated of mixture of factor analyzers. (b). Our proposed latent dynamical model. Square nodes are hidden discrete states while the circle nodes are hidden nodes. The shaded nodes are observations.

$\{\mathbf{x}_t\}$ form temporal sequence generated from the collaboration of the discrete Markov process $\{s_t\}$ and continuous Markov process $\{\mathbf{g}_t\}$.

The model in Fig. 1(b) is a generalization of the switching linear dynamical system (SLDS) [16] by switching among multiple linear dynamical models defined on the low-dimensional globally coordinated latent space. Compared to the dynamic globally coordinated model (DGCM) proposed in [13], the switching linear dynamical models in the latent space in our approach can capture a richer set of dynamics. Simultaneous learning of the nonlinear dynamics and nonlinear manifold is achieved by modeling the interactions among the linear models that define both the dynamics and the mappings from \mathbf{g}_t to observation \mathbf{x}_t .

The discrete state variables are vectors, $s_t \in \{\mathbf{e}_0, \dots, \mathbf{e}_{S-1}\}$, where \mathbf{e}_i is the indicator vector of dimension S with the i -th entry equal to 1. We use $s_{t,i}$ to indicate that $s_t = \mathbf{e}_i$. Let π_0 be the initial state distribution and $\mathbf{\Pi}$ be the state transition matrix, where $\mathbf{\Pi}(i, j) = P(s_{t+1} = \mathbf{e}_i | s_t = \mathbf{e}_j)$; therefore, $s'_{t+1,i} \mathbf{\Pi}_{s_{t,j}}$ represents the state transition probability of $P(s_{t+1} = \mathbf{e}_i | s_t = \mathbf{e}_j)$. Let $\mathcal{S}_T = \{s_0, \dots, s_{T-1}\}$, $\mathcal{G}_T = \{g_0, \dots, g_{T-1}\}$ and $\mathcal{X}_T = \{x_0, x_1, \dots, x_{T-1}\}$. The joint distribution for the graphical model shown in Fig. 1(b) is defined as:

$$P(\mathcal{S}_T, \mathcal{G}_T, \mathcal{X}_T) = P(s_0) \prod_{t=1}^{T-1} P(s_t | s_{t-1}) \\ \times P(g_0 | s_0) \prod_{t=1}^{T-1} P(g_t | g_{t-1}, s_t) \prod_{t=0}^{T-1} P(x_t | g_t, s_t). \quad (6)$$

The dynamical system is defined on the globally coordinated space with the observation being tied to the individual factor analyzers. The following set of state-space equations describe the dynamical system:

$$\begin{aligned} \mathbf{g}_t &= \mathbf{F}(s_t) \mathbf{g}_{t-1} + \mathbf{n}_{\mathbf{g},t}(s_t), \quad t > 0, \\ \mathbf{g}_0 &= \mathbf{n}_0(s_0), \quad t = 0, \\ \mathbf{x}_t &= \mathbf{\Lambda}(s_t)(\mathbf{g}_t - \boldsymbol{\kappa}(s_t)) + \boldsymbol{\mu}(s_t) + \mathbf{n}_{\mathbf{x},t}(s_t), \quad \forall t. \end{aligned} \quad (7)$$

$\Lambda(\mathbf{s}_t)$, $\boldsymbol{\mu}(\mathbf{s}_t)$ and $\boldsymbol{\kappa}(\mathbf{s}_t)$ are globally coordinated MFA parameters that parameterize the mapping $f_{\mathbf{g} \rightarrow \mathbf{x}}$ and $\mathbf{F}(\mathbf{s}_t)$ represents the piecewise linear approximation of f_{dyn} . The corresponding noise processes are assumed to be independently distributed Gaussians, where $\mathbf{n}_{\mathbf{g},t}(\mathbf{s}_t) \sim \mathcal{N}(\mathbf{0}, \Sigma(\mathbf{s}_t))$ for $t > 0$, $\mathbf{n}_0(\mathbf{s}_0) \sim \mathcal{N}(\mathbf{g}_0(\mathbf{s}_0), \Sigma_0(\mathbf{s}_0))$ for $t = 0$ and $\mathbf{n}_{\mathbf{x},t}(\mathbf{s}_t) \sim \mathcal{N}(\mathbf{0}, \Psi(\mathbf{s}_t)), \forall t$.

Let $\Theta = \{\{\mathbf{F}_s, \Sigma_s, \boldsymbol{\mu}_s, \Psi_s, \Lambda_s, \boldsymbol{\kappa}_s\}, \Sigma_0, \boldsymbol{\pi}_0, \Pi\}$ be the set of model parameters. We need to solve the learning problem $\Theta^* = \arg \max_{\Theta} \log P(\mathcal{X}_T | \Theta)$, and the inference problem $P(\mathcal{S}_T, \mathcal{G}_T | \mathcal{X}_T, \Theta)$, *i.e.*, computing the joint distribution of the hidden state sequence \mathcal{S}_T and \mathcal{G}_T given the observation sequence \mathcal{X}_T and model parameters Θ .

4. Learning Algorithm

We take a variational approach to learn the model parameters and optimize the lower bound of the log likelihood by applying Jensen's inequality, $\log P(\mathcal{X}_T | \Theta) \geq \Phi$, where

$$\begin{aligned} \Phi &= \sum_{\mathcal{S}_T} \int Q(\mathcal{G}_T, \mathcal{S}_T | \mathcal{X}_T, \Theta) \log \left(\frac{P(\mathcal{S}_T, \mathcal{G}_T, \mathcal{X}_T | \Theta)}{Q(\mathcal{G}_T, \mathcal{S}_T | \mathcal{X}_T, \Theta)} \right) d\mathcal{G}_T \\ &= \sum_{\mathcal{S}_T} \left[\int Q(\mathcal{G}_T, \mathcal{S}_T | \mathcal{X}_T, \Theta) \log P(\mathcal{X}_T, \mathcal{G}_T, \mathcal{S}_T | \Theta) d\mathcal{G}_T \right. \\ &\quad \left. - \int Q(\mathcal{G}_T, \mathcal{S}_T | \mathcal{X}_T, \Theta) \log Q(\mathcal{G}_T, \mathcal{S}_T | \mathcal{X}_T, \Theta) d\mathcal{G}_T \right]. \quad (8) \end{aligned}$$

$Q(\mathcal{G}_T, \mathcal{S}_T | \mathcal{X}_T, \Theta)$ is an approximation of $P(\mathcal{G}_T, \mathcal{S}_T | \mathcal{X}_T)$. Hence, the first term of Eq. 8 approximates of the expected log-likelihood of the standard EM algorithm. The second term can be regarded as a regularization term given that it models the entropy of the approximate variational distribution. An outline of the learning algorithm is given in Alg. 1.

Algorithm 1 EM-like Learning Algorithm

1: **E-step:** Variational inference to obtain the approximate posterior distribution:

$$P(\mathcal{G}_T, \mathcal{S}_T | \mathcal{X}_T, \Theta^i) \approx Q(\mathcal{G}_T, \mathcal{S}_T | \mathcal{X}_T, \Theta^i). \quad (9)$$

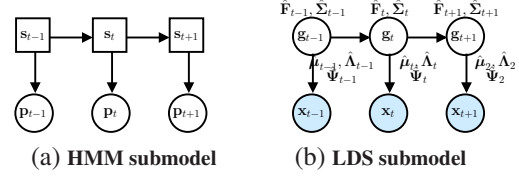
2: **M-step:** Maximize Φ with respect to Θ :

$$\Theta^{i+1} = \arg \max_{\Theta} \Phi(\Theta^i). \quad (10)$$

5. Inference Algorithm

The exact inference is intractable [6] for the graphical model defined Fig. 1(b). We propose a variational inference algorithm and use $Q(\mathcal{G}_T, \mathcal{S}_T | \mathcal{X}_T, \Theta)$ to approximate $P(\mathcal{G}_T, \mathcal{S}_T | \mathcal{X}_T, \Theta)$. We factorize $Q(\mathcal{G}_T, \mathcal{S}_T | \mathcal{X}_T, \Theta)$ into two components:

$$Q(\mathcal{G}_T, \mathcal{S}_T | \mathcal{X}_T, \Theta) = Q(\mathcal{S}_T | \mathcal{X}_T, \Theta) Q(\mathcal{G}_T | \mathcal{X}_T, \Theta). \quad (11)$$



The factorized form of $Q(\mathcal{G}_T, \mathcal{S}_T | \mathcal{X}_T, \Theta)$ implies we can approximate the original model (Fig. 1(b)) with two decoupled models: one is a hidden Markov model (HMM) defined on \mathcal{S}_T with a set of variational parameters (Fig. 5(a)) $\eta_S = \{p_0, \dots, p_{T-1}\}$, where p_0, \dots, p_{T-1} are the output probabilities; and the other is a linear dynamic system (LDS) defined on \mathcal{G}_T with a set of variational parameters (Fig. 5(b)) where $\eta_G = \{\hat{\mathbf{g}}_0, \hat{\Sigma}_0, \hat{\Sigma}_1, \dots, \hat{\Sigma}_{T-1}, \hat{\mathbf{F}}_1, \dots, \hat{\mathbf{F}}_{T-1}, \hat{\Lambda}_0, \dots, \hat{\Lambda}_{T-1}, \hat{\boldsymbol{\mu}}_0, \dots, \hat{\boldsymbol{\mu}}_{T-1}, \hat{\Psi}_0, \dots, \hat{\Psi}_{T-1}\}$.

The expectation of joint log likelihood $\mathcal{L} = \log P(\mathcal{S}_T, \mathcal{G}_T, \mathcal{X}_T)$ with respect to $Q(\mathcal{G}_T | \mathcal{X}_T, \Theta)$ has the form of the joint log-likelihood function of a HMM (Fig. 5(a)), and similarly the expectation of \mathcal{L} with respect to $Q(\mathcal{S}_T | \mathcal{X}_T, \Theta)$ has the form of the joint log-likelihood function of a time-varying LDS (Fig. 5(b)). Hence we can derive the alternating updates for η_S and η_G . Given the HMM sufficient statistics $\langle \mathbf{s}_t \rangle$, we can obtain the time-varying LDS parameters η_G and vice versa. We only summarize the inference algorithm in Alg. 2 as the detailed update equations can be derived by following the formulas provided in [15].

Algorithm 2 Variational Inference Algorithm

- 1: error = inf;
 - 2: Initialize $\langle \mathbf{s}_t \rangle$;
 - 3: **while** error > maxError **do**
 - 4: Compute η_G ;
 - 5: Run LDS smoother to compute sufficient statistics $\langle \mathbf{g}_t \rangle$, $\langle \mathbf{g}_t \mathbf{g}_t^T \rangle$ and $\langle \mathbf{g}_t \mathbf{g}_{t-1}^T \rangle$;
 - 6: Compute η_S ;
 - 7: Run HMM inference to compute sufficient statistics $\langle \mathbf{s}_t \rangle$;
 - 8: Update approximation error based on KL divergence.
 - 9: **end while**
-

6. Experiments

Comparative studies with competing approaches [13, 16] are carried out on three sets of experiments to demonstrate the advantages of our approach.

We use DGCM to denote Lin *et al.*'s approach [13] and SLDS to denote the model proposed in [16]. We obtained the SLDS code from the authors of [16] and we implemented the DGCM proposed in [13]. All three approaches are implemented in un-optimized Matlab. As EM or coordinate ascent algorithms are used in all three approaches, proper initialization is necessary. To initialize the model in our approach and DGCM, the dimensionality of the latent space is chosen experimentally. To avoid over-fitting,

we adopt a variational Bayesian approach [1] to choose the number of mixture components automatically for our approach and DGCM. We follow the technique proposed in [16] to initialize SLDS. First order linear dynamical systems (LDS) are used in all three approaches.

The results reported are based on our implementations. Experiments are conducted on a PC with Intel dual-core 3.46GHz CPU with 4GB memory.

6.1. Synthetic Data

	DGCM	Our Approach
Number of Training Data	1500	1500
Dimensionality of Training Data \mathbf{x}	3	3
Dimensionality of Latent Coordinate \mathbf{g}	2	2
Number of States	10	10
Training Time	~ 3 min	~ 5 min

Table 1. Experimental setup for experiments with synthetic data (Sec 6.1). Number of states in DGCM refer to the number of factor analyzers in the mixture. In our approach, each state comprises a factor analyzer and its corresponding dynamical model.

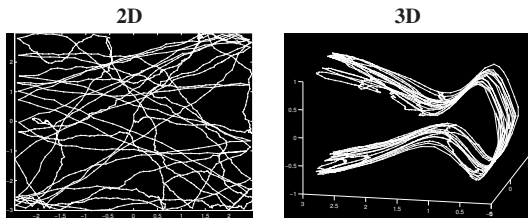


Figure 2. Visualization of the ground truth synthetic data set.

A synthetic data set is used in this experiment to quantify the information loss of dimensionality reduction and reconstruction. The data set is similar to the one used by [13]. 1500 2D data points are generated by a 2D random walk bounced off the boundaries in a patch $[-2.5, 2.5] \times [-3, 3]$. The bouncing at the boundaries introduces nonlinear motion. The 2D data are then lifted to 3D by a mapping function $f(x, y) = (x, |y|, \sin(\pi y)(y^2 + 1)^{-2} + 0.3y)$. Fig. 2 provides a visualization of the ground truth data set. The 1500 3D ground truth points are used as training data. We compare our approach with DGCM and Table 1 shows the setup of the experiment.

	2D		3D		3D	
	$f_{\mathbf{x} \rightarrow \mathbf{g}}$	σ	$f_{\mathbf{g} \rightarrow \mathbf{x}}$	σ	$f_{\mathbf{g} \rightarrow \mathbf{x}}(f_{\mathbf{x} \rightarrow \mathbf{g}})$	σ
DGCM	0.2958	0.2234	1.1993	0.7387	1.2347	0.7491
Our Approach	0.1102	0.0913	0.4854	0.2291	0.6507	0.6192

Table 2. Comparison of dimensionality reduction ($f_{\mathbf{x} \rightarrow \mathbf{g}}$) and reconstruction ($f_{\mathbf{g} \rightarrow \mathbf{x}}$). MSE stands for mean squared error and σ stands for standard deviation of MSE.

To quantitatively evaluate the mapping $f_{\mathbf{x} \rightarrow \mathbf{g}}$, we compute the mean squared error (MSE) between ground truth 2D data and inferred 2D data. Similarly, to evaluate $f_{\mathbf{g} \rightarrow \mathbf{x}}$,

MSE is computed between ground truth 3D data and by applying $f_{\mathbf{g} \rightarrow \mathbf{x}}$ on the ground truth 2D data to reconstruct the 3D sequence. Finally, to evaluate the bidirectional mapping, $f_{\mathbf{g} \rightarrow \mathbf{x}}(f_{\mathbf{x} \rightarrow \mathbf{g}})$, MSE is computed on 3D data by first applying $f_{\mathbf{x} \rightarrow \mathbf{g}}$ on the ground truth 3D data and then applying $f_{\mathbf{g} \rightarrow \mathbf{x}}$ to reconstruct 3D data from the inferred 2D data points. These error statistics are reported in Table 2. The mapping functions learnt by our approach are more accurate in terms of smaller MSE and standard deviation σ . In all cases, our approach cuts the MSE by more than half.

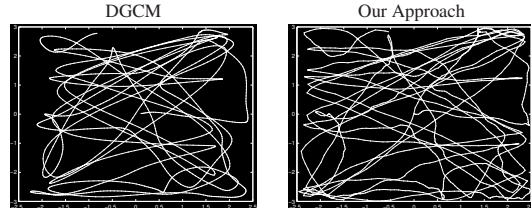


Figure 3. Visualization of the 2D trajectories obtained by applying $f_{\mathbf{x} \rightarrow \mathbf{g}}$ learnt by DGCM and our approach on the ground truth 3D training data shown in Fig. 2. The visual result obtained from our implementation of DGCM is consistent with the result reported in [12].

In the visualization (Fig. 3) of inferred 2D trajectories, $f_{\mathbf{x} \rightarrow \mathbf{g}}$ learnt by our approach produces a 2D trajectory that is closer to the ground truth 2D trajectory (Fig. 2). The mapping function learnt by DGCM produces an overly smoothed trajectory. This is because of switching of the linear dynamical models used in our approach is able to capture the sudden bouncing motion occurred at the patch boundaries more accurately. This leads to the overall improvement in terms of smaller MSE and σ over DGCM.

6.2. Dynamic Texture

	DGCM	Our Approach
Length of the Flag Sequence	250	250
Length of the Wave Sequence	350	350
Dimensionality of \mathbf{x}	104256(= 288 × 362)	104256
Dimensionality of \mathbf{g}	20	20
Number of States	3	3
Training Time	~ 5 min	~ 8 min

Table 3. Experimental setup of experiment (Sec. 6.2) with dynamic texture.

We can synthesize data on the manifold by using f_{dyn} to generate time series in the low-dimensional latent space. We can then use $f_{\mathbf{g} \rightarrow \mathbf{x}}$ to map the low-dimensional time series back to the high-dimensional observation space. In this experiment, videos from a dynamic texture database [29] are used for training DGCM and our method. We then synthesize textures from the trained models. Table 3 shows the setup of the experiment.

Similar to Sec 6.1, we quantify the information loss by computing the MSE of normalized intensity values (range

from 0 to 1) between the training video frames and the reconstructed frames. The reconstructed frames are obtained by first applying $f_{x \rightarrow g}$ to the training video frames to get the coordinates in the latent space, and then applying $f_{g \rightarrow x}$ to the latent coordinates. The error statistics of $f_{x \rightarrow g}$ and $f_{g \rightarrow x}$ are not evaluated separately as there is no ground truth low-dimensional data. The error statistics are shown in Table 4. We can see that the images reconstructed from the manifold learnt by our approach are closer to the training images in terms of smaller MSE and standard deviation σ . Our approach reduces the MSE by 35% and the σ by 31% for the flag sequence, and 30% and 26% for the wave sequence.

	Flag Sequence		Wave Sequence	
	Mean Err.	σ	Mean Err.	σ
DGCM	0.0249	0.0378	0.0300	0.0316
Our approach	0.0161	0.0258	0.0210	0.0235

Table 4. Comparison of reconstruction error from training frames.

Sample synthesized frames from dynamic texture sequences are shown in Fig 4. The images synthesized by our approach are much crisper than those obtained by DGCM, especially when there is a sudden change of dynamics (it is more evident in the submitted video). Subtle details like the folds of the flag and foam on the wave are crisper in the images synthesized by our approach. This observation is consistent with the evaluation with the synthetic data, where our approach is able to handle sudden changes of motion through switching among multiple dynamical models.

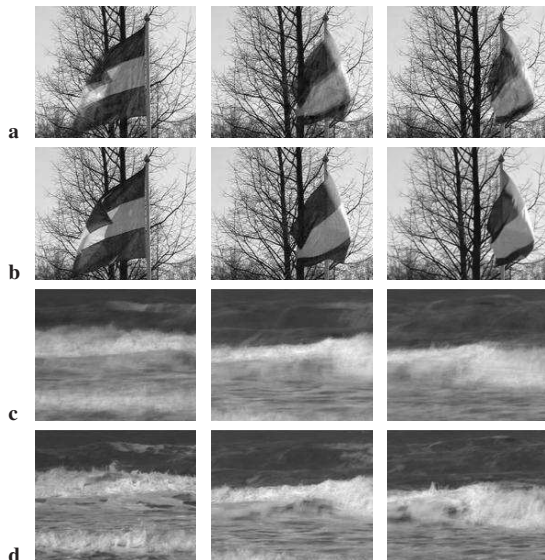


Figure 4. Comparison of texture synthesis results: **a.** DGCM, and **b.** our approach. **c.** and **d.** are the results from the wave sequence. The folds of the flag and the foam of the wave are crisper than those produced by DGCM.

	SLDS	DGCM	Our Approach
Length of the Mocap Sequence	1500	1500	1500
Dimensionality of \mathbf{x}	28	28	28
Dimensionality of \mathbf{g}	-	3	3
Number of States	17	12	12
Training Time	~ 45 min	~ 28 min	~ 33 min

Table 5. Experimental setup for human motion analysis (Sec. 6.3). In SLDS, number of states refer to the number of dynamical models used.

6.3. Human Motion Analysis

We test our approach on the tasks of human motion synthesis, classification and tracking to demonstrate the advantages of modeling dynamics on the low-dimensional manifold with multiple linear dynamical models. The *Boxing* sequences of S1 from the benchmark datasets [22] are used. In all three experiments, the motion capture sequence (containing multiple cycles of the boxing action) from Session 3 is used to train the model. Table 5 summarizes the experimental setup.

6.3.1 Human Motion Synthesis

We compare our approach with DGCM for the task of reconstructing human body configurations from the learnt low-dimensional manifolds; *i.e.*, we compute the average joint angle error between the training data and the reconstructed data by applying $f_{g \rightarrow x}(f_{x \rightarrow g})$ on the training data. Figure 5 shows the average reconstruction error of each joint location of the upper body limbs over all the training frames. The errors reported on the left and right upper-body joints are not exactly the same due to asymmetrical limb movements. We can see that DGCM tends to make more errors at the shoulders. This can cause large errors for the joints at the elbows when we convert from the joint angle representation to the actual 3D human body. The errors made by our approach at the shoulder joints are at least one standard deviation smaller than those made by DGCM. This is because our motion model is able to capture the nonlinear limb movements effectively in the latent space.

We apply the learnt f_{dyn} and $f_{g \rightarrow x}$ by DGCM and our approach to synthesize 100 frames. We also use SLDS [16] as a motion model to synthesize 100 frames. Sample synthesized frames are shown in Fig. 6. The undesirable synthesis results are shown in red border and they are produced by SLDS and DGCM. We can see that the propagated error at the shoulder joints introduces unnatural configurations of the lower arms. In comparison, our approach is able to produce more natural boxing actions when compared to [16, 13] thanks to the temporally consistent learning of the low-dimensional manifold and effective modeling of nonlinear dynamics using interacting linear models.

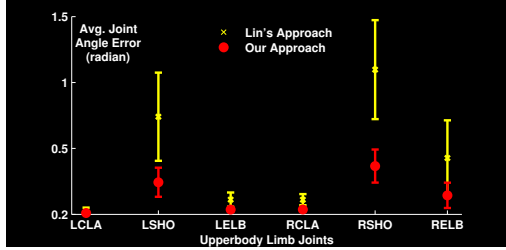


Figure 5. Comparison of reconstruction error. Our approach has smaller reconstruction error (both mean and standard deviation) than DGCM (Lin’s approach), especially at the joints higher on the hierarchy of the kinematic chain. The short form joint labels are: LCLA (left clavicle), LSHO (left shoulder), LELB (left elbow) and RCLA, RSHO and RELB refer to the corresponding joints on the right upper body limbs

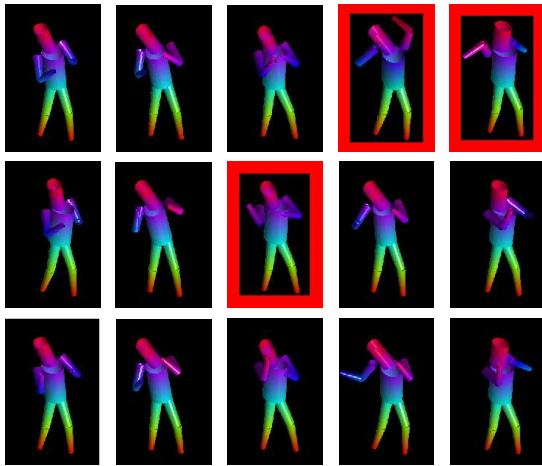


Figure 6. Comparison of synthesis results. The first row shows synthesized frames using SLDS, the second row DGCM and the last row our approach. Undesirable synthesized results are shown with red border.

6.3.2 Human Motion Classification

As we approximate f_{dyn} with multiple linear motion models, we can do motion classification when we associate each model with a class label. This experiment with the boxing sequence demonstrates such classification capability. The test sequence comprises 300 frames in this experiment. We compare our model with the SLDS model proposed in [16]. Note that in the SLDS model, the observation and hidden states of the continuous layer are of the same dimensionality (28), while the hidden states of the continuous layer in our model are of much lower dimension (3 in the current setup). In our approach, the 6 states for the forward punch are considered as one class and the 6 states for the upward punch are considered as another class. Similarly, for the 17 states used for the SLDS model, the 7 states being labeled as forward punch are considered as one class and the other 10 are considered the upward punch class. SLDS state labels

are set to maximize the classification accuracy.

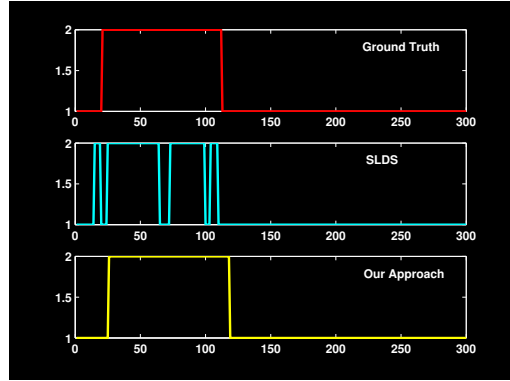


Figure 7. Comparison of classification results. The horizontal axis shows the frame indices, while the vertical axis show class labels with 1 refers to upward punch and 2 refers to forward punch. Our approach produces accurate classification results compared to SLDS, where there is abrupt change of class labels in SLDS.

As shown in Fig. 7, our approach achieves 95% classification accuracy. At the state transition, our approach tends to delay the transition a little bit more for about 5-10 frames. This can be explained by global coordination mechanism which counteracts the abrupt switching. As there is no such mechanism in the SLDS model and the high-dimensional states are less discriminative, SLDS tends to switch among different classes more frequently and hence has a lower classification accuracy of 90.3% for this data set.

6.3.3 Human Motion Tracking

	SLDS	DGCM	Our Approach
Mean marker error (mm)	569.90	380.02	187.50
σ (mm)	209.18	74.97	39.73
Processing time per frame (second)	~ 120	~ 32	~ 41

Table 6. Comparison of tracker errors and processing time per frame. Our approach takes slightly more time per frame compared to DGCM with an improved accuracy of 50% both in terms of mean and standard deviation of the marker error.

In this experiment, we use the learnt f_{dyn} and $f_{g \rightarrow x}$ to provide prior information for 3D human motion tracking. The tracker we use here is similar to [12, 24]. We test the Boxing sequences from Session 1 and 2 of S1 and evaluate the tracker accuracy from the online evaluation tool provided by [22]. The tracker errors reported in Table 6 are computed based on the criteria defined in [22]. From Table 6, we can see that the mean error for recovered virtual joint marker positions (see [22] for detail) is within 250 mm and less than half of the errors reported for SLDS and DGCM. Sample tracked frames are shown in Fig. 8. We can see that the tracker that uses the priors from our approach is

able to lock on to the limbs over time while the other two approaches fail. Our tracker is also able to generalize fairly well for motion with slight variation from the training data as the training sequence and testing sequences are captured at different times with the same test subject. The promising results show that the proposed model can be used effectively in tracking applications. However, its generalization performance needs further investigation.

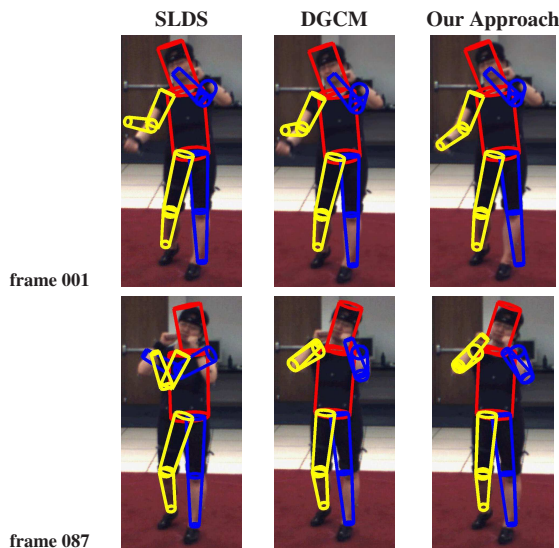


Figure 8. Sample tracked frames. Both SLDS and DGCM fail to lock on the right lower arm in frame 1. SLDS fails to track both arms in frame 87. More results can be seen in the submitted video.

7. Conclusions and Future Work

A general method is proposed for efficient simultaneous learning a nonlinear low-dimensional manifold and a nonlinear dynamical model for high-dimensional time series. Previous approaches have difficulty of handling large datasets [32] or modeling complex nonlinear dynamical behavior [13]. The main contribution is the proposed solution, which exploits the coordinated piecewise linear models to overcome these difficulties. Extensive experiments verify the efficiency and effectiveness of the proposed solution.

Currently the number of states is chosen independently of the dynamical models using a variational Bayesian approach. The dimensionality of the latent space is chosen empirically. We are investigating a full-fledged variational Bayesian formulation [1] for choosing the optimal model setup, *i.e.*, the number of components, dimensionality of the latent space and the order of the dynamical models. Another question with the proposed approach is its generalization performance and we are investigating methods to quantify how well the model generalizes for a given application.

References

- [1] M. Beal. *Variational Algorithms for Approximate Bayesian Inference*. PhD thesis, Gatsby Computational Neuroscience Unit, University College London, 2003. 5, 8
- [2] M. Belkin and P. Niyogi. Laplacian Eigenmaps and spectral techniques for embedding and clustering. In *Proc. of NIPS*, pages 585–591, 2001. 2
- [3] C. M. Bishop, M. Svensén, and C. K. I. Williams. GTM: the Generative Topographic Mapping. *Neural Computation*, 10(1):215–234, 1998. 2
- [4] M. Brand. Charting a manifold. In *Proc. of NIPS*, pages 961–968, 2002. 2
- [5] A. Elgammal and C.-S. Lee. Inferring 3D body pose from silhouettes using activity manifold learning. In *CVPR*, pages 681–688, 2004. 2
- [6] Z. Ghahramani. Learning Dynamic Bayesian Networks. In C. Giles and M. Gori, editors, *Lecture Notes in Artificial Intelligence, Adaptive Processing of Sequences and Data Structures*, pages 168–197. Berlin: Springer-Verlag, 1998. 4
- [7] Z. Ghahramani and G. E. Hinton. The EM algorithm for mixtures of factor analyzers. Technical Report CRG-TR-96-1, University of Toronto, 1996. 2
- [8] Z. Ghahramani and S. Roweis. Learning nonlinear dynamical systems using an EM algorithm. In *Proc. of NIPS*, volume 11, pages 431–437, 1998. 2
- [9] K. Grochow, S. L. Martin, A. Hertzmann, and Z. Popovic. Style-based inverse kinematics. In *ACM Computer Graphics (SIGGRAPH)*, pages 522–531, 2004. 2
- [10] O. Jenkins and M. Mataríć. A Spatio-temporal Extension to Isomap Nonlinear Dimensionality Reduction. In *Proc. ICML*, pages 56–73, 2004. 2
- [11] N. Lawrence. Gaussian Process Latent Variable Models for Visualisation of High Dimensional Data. In *Proc. of NIPS*, volume 16, 2004. 2
- [12] R. Li, M.-H. Yang, S. Sclaroff, and T.-P. Tian. Monocular Tracking of 3D Human Motion with a Coordinated mixture of factor analyzers. In *Proc. ECCV*, volume 2, pages 137–150, 2006. 2, 5, 7
- [13] R.-S. Lin, C.-B. Liu, M.-H. Yang, N. Ahuja, and S. Levinson. Learning Nonlinear Manifolds from Time Series. In *Proc. ECCV*, volume 3, pages 239–250, 2006. 1, 2, 3, 4, 5, 6, 8
- [14] K. Moon and V. Pavlovic. Impact of Dynamics on Subspace Embedding and Tracking of Sequences. In *CVPR*, pages 198–205, 2006. 2
- [15] S. M. Oh, A. Ranganathan, J. M. Rehg, and F. Dellaert. A variational inference method for switching linear dynamic systems. Technical Report GIT-GVU-05-16, Georgia Institute of Technology, 2005. 4
- [16] V. Pavlovic, J. M. Rehg, and J. MacCormick. Learning switching linear models of human motion. In *Proc. of NIPS*, volume 13, pages 626–632, 2000. 2, 3, 4, 5, 6, 7
- [17] L. Ralaivola and F. d’Alché-Buc. Dynamical modeling with kernels for nonlinear time series prediction. In *Proc. of NIPS*, volume 16, 2004. 2
- [18] S. Roweis and L. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, pages 2323–2326, 2000. 2
- [19] S. Roweis, L. Saul, and G. E. Hinton. Global Coordination of Local Linear Models. In *Proc. of NIPS*, volume 14, pages 889–896, 2001. 2, 3
- [20] D. Rubin and D. Thayer. EM algorithm for ML factor analysis. *Psychometrika*, 47(1), 1982. 2
- [21] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, 10(1):1299–1319, 1998. 2
- [22] L. Sigal and M. Black. Humaneva: Synchronized video and Motion Capture Dataset for Evaluation of Articulated Human Motion. Technical Report CS-06-08, Brown University, 2006. 1, 6, 7

- [23] V. Silva and J. Tenenbaum. Global versus Local Methods in Non-linear Dimensionality Reduction. In *Proc. of NIPS*, pages 705–712, 2003. 2
- [24] C. Sminchisescu and A. Jepson. Generative modelling for continuous non-linearly embedded visual inference. In *Proc. ICML*, pages 140–147, 2004. 2, 7
- [25] E. Snelson and Z. Ghahramani. Sparse gaussian processes using pseudo-inputs. In *Proc. of NIPS*, pages 1259–1226. 2
- [26] W.-Y. Teh and S. Roweis. Automatic alignment of local representations. In *Proc. of NIPS*, pages 841–848, 2002. 2
- [27] J. Tenenbaum, V. Silva, and J. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, pages 2319–2323, 2000. 2
- [28] T.-P. Tian, R. Li, and S. Sclaroff. Articulated pose estimation in a learned smooth space of feasible solutions. In *CVPR Learning Workshop*, 2005. 2
- [29] <http://www.cwi.nl/projects/dyntex/>. 1, 5
- [30] R. Urtasun, D. J. Fleet, A. Hertzmann, and P. Fua. Priors for people tracking from small training sets. In *Proc. ICCV*, pages 403–410, 2005. 2
- [31] J. Verbeek. Learning non-linear image manifolds by combining local linear models. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 39(10):1864–1875, 2006. 3
- [32] J. M. Wang, D. Fleet, and A. Hertzmann. Gaussian Process Dynamical Models. In *Proc. of NIPS*, volume 18, pages 1441–1448, 2005. 2, 8