COUPLED EMBEDDING OF SEQUENTIAL PROCESSES USING GAUSSIAN PROCESS MODELS

BY KOOKSANG MOON

A dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Computer Science

Written under the direction of

Prof. Vladimir Pavlovic

and approved by

_____

_____

_____

_____

_____

New Brunswick, New Jersey

January, 2009

ABSTRACT OF THE DISSERTATION

Coupled Embedding Of Sequential Processes Using Gaussian Process Models

by Kooksang Moon

Dissertation Director: Prof. Vladimir Pavlovic

In this dissertation we consider the task of making predictions from high dimensional sequential data. Problems of this type arise in many practical scenarios, such as the estimation of 3D human figure motion from a sequence of images or the predictions of implied volatility trends from sequences of option market indicators in financial time-series analysis. However, direct predictions of this type are typically infeasible due to high dimensionality of both the input and the output data, as well as the existence of temporal dependencies. To address this task we present a novel approach to subspace modeling of dyadic high dimensional sequences which have a co-occurrence or regression relationship. Statistical reasoning suggests that predictions made through low dimensional subspaces may improve the performance of predictive models if such subspaces are properly selected. We show that selection of such optimal predictive subspaces can be made, and is largely analogous, to the task of designing a particular family of Gaussian processes (GP). As a consequence, many of the models we consider here can be seen as a generalization of the well-known GP regressors.

We first study the role of dynamics in subspace modeling of single sequence and propose a new family of marginal auto-regressive (MAR) models which can describe the space of all stable auto-regressive sequences. We utilize the MAR priors in a Gaussian process latent variable model (GPLVM) framework to represent the nonlinear dimensionality reduction process with

a dynamic constraint. To model the low dimensional embedding in the prediction tasks, we propose two alternative approaches: a generative model and direct predictive, discriminative model. For the generative modeling approach, we extend the framework of probabilistic latent semantic analysis (PLSA) models in a sequential setting. This dynamic PLSA approach results in a new generative model which learns a pair of mapping functions between the subspace and the two data sequences with a dynamic prior. For the discriminative modeling approach, we address the problem of learning optimal regressors that maximally reduce the dimension of the input while preserving the information necessary to predict the target values based on the sufficient dimensionality reduction concept. Instead of the iterative solutions of previous approaches, we show how a globally optimal solution in closed form can be obtained by formulating a related problem in a setting reminiscent of the GP regression. In the set of experiments on various vision and financial time-series prediction problems, the proposed two models achieve significant gains in accuracy of prediction as well as interpretability, compared to other dimension reduction and regression schemes.

Acknowledgements

I would like to thank my advisor Vladimir Pavlovic who supported and guided me from my third semester at Rutgers until now. His insight into machine learning greatly influenced my way of doing research. His enthusiasm and critical thinking on research topics has had a great impact on me. I owe him lots of gratitude for having me in the SEQAM lab and for his valuable comments and suggestions during numerous research meetings. I would also like to thank Dr. Hyuncheol Hwang for advising me on the financial data problem. My deepest gratitude also goes to the committee members: Dr. Dimitri Metaxas, Dr. Ahmed Elgammal, and Dr. Goce Trajcevski.

I am also grateful to Dr. Greg Slabaugh for his valuable comments and suggestions.

Special thanks go to my good colleagues, Rui Hwang and Pavel Kuksa in the SEQAM lab for their enjoyable research discussions with me.

Dedication

To my wife, Jeehyun, my lifelong companion: Thank you for all your patience and sacrifice during my Ph.D. study. Without you, this dissertation couldn't be finished.

To my son, Alexander: Thank you for your bright smile that cheers me a lot whenever I am tired.

To my father and mother: Thank you for your never-ending support and praying for me.

To my father-in-law and mother-in-law: Thank you for your great trust in me.

To my cousin, Caroline: Thank you for giving me a hand whenever I needed it.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

The objective of this thesis is to propose a general framework that utilizes the dimensionality reduction or subspace embedding to model the matching between sequences. We are in particular interested in the prediction tasks with two high dimensional sequences. Our intuition is that in these tasks, predictions made through low dimensional subspaces are able to improve the prediction accuracies if such subspaces are properly selected.

In many machine learning problems, we often deal with high dimension data sets, and this high dimensionality can be a significant obstacle to problem solving. Theoretically, the curse of dimensionality implies that the number of data points needed to model the structure of a high dimensional data set increases exponentially with the number of dimensions in the data space. However, in practice we found that the intrinsic representation of the data lies in a much smaller dimensional space, which enables us to do well with much smaller data sets. For example, in human motion modeling, the human body pose can be represented as a 62 dimensional vector (translation and joint angles) measured by the motion capture system. Despite the high dimensionality of body configuration space, it is well known that various human activities lie intrinsically on low dimensional manifold when considering the body kinematics.

Dimensionality reduction / subspace embedding methods such as Principal Components Analysis (PCA) play an important role in many data modeling tasks by selecting and inferring those features that lead to an intrinsic representation of the data. General purposes of dimensionality in machine learning includes the prediction performance improvements by filtering out redundant features and the improvements of learning efficiency by exploiting the models with fewer parameters and better generalization. As such, they have attracted significant attention in a number of machine learning areas, such as computer vision, where they have been

used to represent intrinsic spaces of shape, appearance, and motion. However, it is common that subspace projection methods applied in different contexts do not leverage the inherent properties of those contexts. For instance, the dynamic nature of sequential data or the intrinsic data structure of input in supervised learning is often ignored in the subspace learning process.

As for modeling the matching between two high dimensional sequences, learning the direct mapping between them results in complex models with poor generalization properties. Therefore, many previous approaches in computer vision and machine learning utilized the dimensionality reduction. However, most of them learn the two mappings between the two observations and the embedding subspace independently and in result the correlation between the two observations is weakened.

1.2 Single Sequence Modeling and Dimensionality Reduction

We first investigate the utility of the dimensionality reduction in a single sequence modeling procedure such as a human motion modeling. Modeling the dynamics of human figure motion is essential to many applications such as realistic motion synthesis in animation and human activity classification. Because the human pose is typically represented by more than 30 parameters (*e.g.* 59 joint angles in the marker-based motion capture system), modeling human motion is a complex task; dependent upon a sequence of high dimensional data. Suppose $\mathbf{y}_t$ is a M-dimensional vector consisting of joint angles at time t. Modeling human motion can be formulated as learning a dynamic system:

$$\mathbf{y}_t = h(\mathbf{y}_0, \mathbf{y}_1, ..., \mathbf{y}_{t-1}) + \mathbf{u}_t$$

where $\mathbf{u}_t$ is a (Gaussian) noise process.

A common approach to modeling linear motion dynamics would be to assume a T-th order linear auto-regressive (AR) model:

$$\mathbf{y}_t = \sum_{i=1}^{T} \mathbf{A}_i \mathbf{y}_{t-i} + \mathbf{u}_t \tag{1.1}$$

where $\mathbf{A}_i$ is the auto-regression coefficient matrix. For instance, second order AR models are sufficient for modeling of periodic motion and higher order models lead to more complex motion dynamics. However, as the order of the model increases the number of parameters

grows as $M^2 \cdot T + M^2$ (transition and covariance parameters). Learning this set of parameters may require large training sets and can be prone to overfitting.

Armed with the intuition that correlation between the limbs such as arms and legs always exists for a certain motion, many researchers have exploited the dynamics in the lower dimensional projected space rather than learning the dynamics in the high-dimensional pose space for human motion modeling. By inducing a hidden state $\mathbf{x}_t$ of dimension N ($M \gg N$) satisfying the first-order Markovian condition, modeling human motion is cast in the framework of dynamic Bayesian networks (DBNs) depicted in Figure 1.1:

$$\begin{aligned} \mathbf{x}_t &= f(\mathbf{x}_{t-1}) + \mathbf{w}_t \\ \mathbf{y}_t &= g(\mathbf{x}_t) + \mathbf{v}_t \end{aligned}$$

where $f(\cdot)$ is a transition function, $g(\cdot)$ represents any dimensional reduction operation, and $\mathbf{w}_t$ and $\mathbf{v}_t$ are (Gaussian) noise processes.

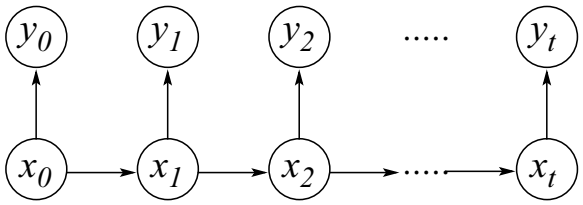


Figure 1.1: A graphical model for human motion modeling with subspace modeling.

The above DBN formalism implies that predicting the future observation $\mathbf{y}_{t+1}$ based on the past observation data $\mathcal{Y}_0^t = \{\mathbf{y}_0, \ldots, \mathbf{y}_t\}$ can be stated as the following inference problem:

$$P(\mathbf{y}_{t+1}|\mathcal{Y}_0^t) = \frac{P(\mathcal{Y}_0^{t+1})}{P(\mathcal{Y}_0^t)} = \frac{\sum_{\mathbf{x}_{t+1}} \cdots \sum_{\mathbf{x}_0} P(\mathbf{x}_0) \prod_{i=0}^{t} P(\mathbf{x}_{i+1}|\mathbf{x}_i) \prod_{i=0}^{t+1} P(\mathbf{y}_i|\mathbf{x}_i)}{\sum_{\mathbf{x}_t} \cdots \sum_{\mathbf{x}_0} P(\mathbf{x}_0) \prod_{i=0}^{t-1} P(\mathbf{x}_{i+1}|\mathbf{x}_i) \prod_{i=0}^{t} P(\mathbf{y}_i|\mathbf{x}_i)}.$$

This suggests that the dynamics of the observation (pose) sequence $\mathbf{Y}$ possesses a more complicated form. Namely, the pose $\mathbf{y}_t$ at time t becomes dependent on all previous poses $\mathbf{y}_{t-1}, \mathbf{y}_{t-2}, \ldots$ effectively resulting in an infinite order AR model. However, such a model can use a smaller set of parameters than the AR model of Equation (1.1) in the pose space. Assuming a first order linear dynamic system (LDS) $\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{w}$ and the linear dimensionality reduction process $\mathbf{y}_t = \mathbf{G}\mathbf{x}_t + \mathbf{v}$ where $\mathbf{F}$ is the transition matrix and $\mathbf{G}$ is the inverse of the dimensionality reduction matrix, the number of parameters to be learned is

$N^2 + N^2 + N \cdot M + M^2 = 2N^2 + M \cdot (N + M)$ (N^2 in F, NM in G and $N^2 + M^2$ in the two noise covariance matrices for $\mathbf{w}$ and $\mathbf{v}$). When $N \ll M$ the number of parameters of the LDS representation becomes significantly smaller than that of the "equivalent" AR model. That is, by learning both the dynamics in the embedded space and the subspace embedding model, we can effectively estimate $\mathbf{y}_t$ given all $\mathcal{Y}_0^{t-1}$ at any time t using a small set of parameters.

To illustrate the benefit of using the dynamics in the embedded space for human motion modeling, we take 12 walking sequences of one subject from CMU Graphics Lab Motion Capture Database [1] where the pose is represented by 59 joint angles. The poses are projected into a 3D subspace. Assume that the dynamics in the pose space and in the embedded space are modeled using the second order linear dynamics. We perform leave-one-out cross-validation for these 12 sequences - 11 sequences are selected as a training set and the one remaining sequence is reserved for a testing set. Let M_{pose} be the AR model in the pose space learned from this training set and M_{embed} be the LDS model in the latent space. Figure 1.2 shows the summary statistics of the two negative log-likelihoods of $P(Y_n|M_{pose})$ and $P(Y_n|M_{embed})$, where Y_n is a sequence reserved for testing.

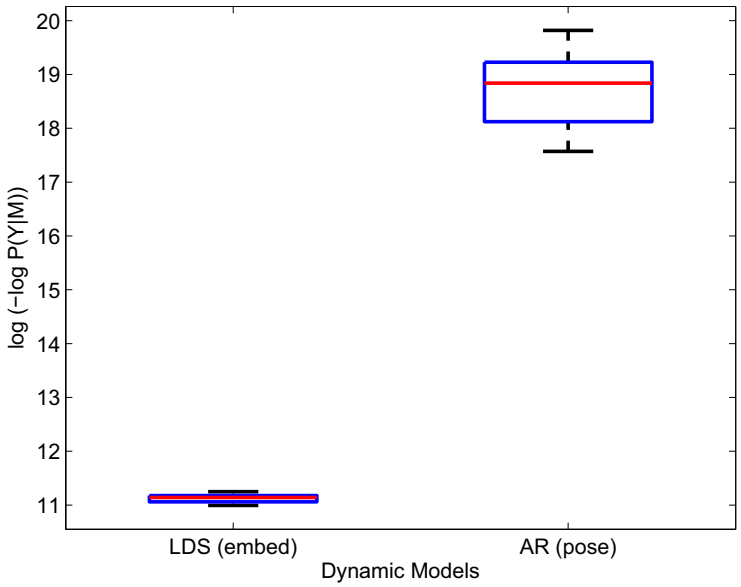


Figure 1.2: Comparison of generalization abilities of AR ("pose") and LDS ("embed") models. Shown are the medians, upper and lower quartiles (boxes) of the negative log likelihoods (in log space) under the two models. The whiskers depict the total range of the values. Note that lower values suggest better generalization properties (fit to test data) of a model.

The experiment indicates that with the same training data, the learned dynamics in the embedded space models the unseen sequences better than the dynamic model in the pose space. The large variance of $P(Y_n|M_{pose})$ for different training sets also indicates the overfitting problem that is generally observed in a statistical model that has too many parameters.

As shown in Figure 1.1, there are two processes in modeling human motion using a subspace embedding. One is learning the embedding model $P(\mathbf{y}_t|\mathbf{x}_t)$ and the other is learning the dynamic model $P(\mathbf{x}_{t+1}|\mathbf{x}_t)$. The problem of the previous approaches using the dimensionality reduction in human motion modeling is that these two precesses are decoupled into two separate stages in learning. However, coupling the two learning processes results in a better embedded space that preserves the dynamic nature of original data. For example, if the prediction by the dynamics suggests that the next state will be near a certain point we can learn a projection that retains the temporal information better than a naive projection, which disregards this prior knowledge. Our proposed framework formulates this coupling of the two learning processes in a probabilistic manner.

1.3 Nonlinear Dimensionality Reduction Using Gaussian Process

As briefly mentioned in Section 1.2, the subspace embedding process can be cast into the inverse problem of data generation problem. Let $g(\cdot)$ be a data generation process. Then the general formulation of data generation can be modeled as

$$\mathbf{y} = g(\mathbf{x}) + \mathbf{v} \tag{1.2}$$

where $\mathbf{x} \in \mathbb{R}^p$ can be any intrinsic low dimensional vector, $\mathbf{y} \in \mathbb{R}^d$ is any observation vector and $\mathbf{v}_t$ is the random noise vector. The dimensionality relationship should be $d > p$. Based on this formulation, the subspace embedding process can be represented as the inverse function, $g(\cdot)^{-1}$. And the task of dimensionality reduction becomes to infer the function, g or g^{-1}, explicitly or implicitly.

Depending on the selection of g, the approaches of dimensionality reduction can be categorized into two: linear and nonlinear methods. In linear methods, the original observation data is projected into a linear subspace. Principal Component Analysis (PCA) is the most well-known approach in this category. Nonlinear dimensional reduction methods are all other approaches

including the methods based on geometrical relationship of data points, extended nonlinear kernel PCA, and probabilistic nonlinear PCA using Gaussian Process. Our choice in the thesis is the probabilistic nonlinear dimensionality reduction using Gaussian Process. This approach is called Gaussian Process Latent Variable Model (GPLVM) and provides a nice probabilistic framework for dimensionality reduction modeling.

1.3.1 Gaussian Process

Here, we briefly review the concept of Gaussian process in the context of dimensionality reduction, based on [2,3]. Suppose that we are given a training dataset $D = \{(\mathbf{x}_i, \mathbf{y}_i)\}, i = 1, \ldots, N$ for dimensionality reduction modeling. To learn the data generation function g which can define the new data point from an arbitrary point (*e.g.* testing data point) in the latent space X, one needs to make assumptions about the characteristics of g. Depending on these assumptions, there have been two common approaches in learning the function g. The first approach restricts the class of functions in some parametric form and the second one considers the probability distribution over function space. When the first approach is selected in the learning, one has the obvious problem of the richness in class selection at the beginning. That is, the given data may not fit well into the selected class of function. And even when the function is modeled well by a certain class, there is a chance of overfitting which causes poor predictions for testing data. The second approach appears to have a similar problem because one should compute the probability distribution on infinite set of possible functions. However, Gaussian process makes it possible to place a prior over the entire function space. The Gaussian process is the generalization of a Gaussian distribution to a function space. As a Gaussian distribution is defined on all possible scalar values with its mean and covariance matrix, a Gaussian process is specified over infinite function space by a mean and a covariance function.

For simplicity, assume the conditional independency of individual dimension in $\mathbf{y}$, and then consider the function $f(\cdot)$ that fits into only a certain dimension of $\mathbf{y}$. Then the problem of learning this function f becomes a training problem of the regression model, $y = f(\mathbf{x}) + \varepsilon$ in which the covariate is a vector $\mathbf{x}$ and the target is a scalar value y with additive noise ε. Let $\mathbf{f} = \{f_i\}_{i=1}^{N} \in \Re^{N \times 1}$ be the vector of function values instantiated from a function $f(\cdot)$. If we assume a Gaussian prior on these values with zero mean and covariance matrix $\mathbf{K}$, then we

have

$$\begin{aligned} p(\mathbf{f}) &= \mathcal{N}(\mathbf{0}, \mathbf{K}) \\ &= (2\pi)^{-\frac{N}{2}} |\mathbf{K}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\mathbf{f}'\mathbf{K}^{-1}\mathbf{f}\right) \end{aligned} \tag{1.3}$$

Note that the covariance is built using the covariate $\mathbf{x}_i$. Now we want to utilize this knowledge about the function distribution in predicting the targets from a number of new input points $\mathbf{X}_*$. Assuming additive i.i.d. Gaussian noise $\varepsilon \sim \mathcal{N}(0, \sigma_n^2)$ we can easily combine a Gaussian process prior with a noise model to estimate a posterior over function. That is, when $\mathbf{f}_*$ is a vector of function values corresponding to $\mathbf{X}_*$, the conditional predictive distribution for Gaussian process regression also becomes Gaussian, $p(\mathbf{f}_*|\mathbf{X}, \mathbf{y}, \mathbf{X}_*) \sim \mathcal{N}(\bar{\mathbf{f}}_*, \Sigma)$, where

$$\bar{\mathbf{f}}_* = \mathbf{K}_{*,\mathbf{f}}(\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma_n^2\mathbf{I})^{-1}\mathbf{y} \tag{1.4}$$

$$\Sigma = \mathbf{K}_{*,*} - \mathbf{K}'_{\mathbf{f},*}(\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma_n^2\mathbf{I})^{-1}\mathbf{K}_{\mathbf{f},*}. \tag{1.5}$$

Then, one can also compute the marginal likelihood $p(\mathbf{y}|\mathbf{X})$ over the function value $\mathbf{f}$ by observing that $\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{K} + \sigma^2\mathbf{I})$,

$$\begin{aligned} p(\mathbf{y}|\mathbf{X}) &= \int p(\mathbf{y}|\mathbf{f}, \mathbf{X})p(\mathbf{f}|\mathbf{X})d\mathbf{f} \\ &= (2\pi)^{-\frac{N}{2}} |\mathbf{K} + \sigma^2\mathbf{I}|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\mathbf{y}'(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{y}\right\} \end{aligned} \tag{1.6}$$

The GP models has been applied to various machine learning problems because of their

1.3.2 Gaussian Process Latent Variable Model

Gaussian Process Latent Variable Model (GPLVM) is induced from probabilistic PCA as a dual representation of it [4]. Probabilistic PCA is a probabilistic extension of PCA and models a linear mapping between the p-dimensional latent space, $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ and the *centered* data set, $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]$ in D-dimensional space,

$$\mathbf{y}_n = \mathbf{W}\mathbf{x}_n + \eta_n \tag{1.7}$$

where η_n is a vector of noise term which is taken to be Gaussian distributed: $p(\eta) \sim \mathcal{N}(\mathbf{0}, \beta^{-1}\mathbf{I})$. By assuming $\mathbf{y}_n$ is i.i.d. and marginalizing the conditional probability given the latent space

($p(\mathbf{y}_n|\mathbf{x}_n, \mathbf{W}, \beta) = \mathcal{N}(\mathbf{y}_n|\mathbf{W}\mathbf{x}_n, \beta^{-1}\mathbf{I})$), one can found the solution for $\mathbf{W}$ by maximizing the likelihood,

$$p(\mathbf{Y}|\mathbf{W}, \beta) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{y}_n|\mathbf{0}, \mathbf{W}\mathbf{W}' + \beta^{-1}\mathbf{I}). \tag{1.8}$$

Instead of marginalizing the latent variables, one can marginalize the mapping $\mathbf{W}$. This marginalization results

$$\begin{aligned} p(\mathbf{Y}|\mathbf{X}, \beta) &= \int \prod_{n=1}^{N} p(\mathbf{y}_n|\mathbf{x}_n, \mathbf{W}, \beta) p(\mathbf{W}) d\mathbf{W} \\ &= (2\pi)^{-\frac{DN}{2}} |\mathbf{K}|^{-\frac{D}{2}} \exp\left\{ -\frac{1}{2} tr(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}') \right\} \end{aligned} \tag{1.9}$$

where $\mathbf{K} = \mathbf{X}\mathbf{X}' + \beta^{-1}\mathbf{I}$.

The GPLVM estimates the joint density of the data points ($\mathbf{Y}$) and their latent space representations ($\mathbf{X}$). The MAP estimates of $\mathbf{X}$ are used to represent a learned subspace.

1.4 Dyadic Sequences Modeling and Dimensionality Reduction

Modeling the matching between the two sequences is an important task in various signal and image processing problems such as object tracking, object pose estimation, image and signal denoising, and illumination direction estimation. The goal of modeling in these tasks is to make the accurate predictions given new inputs. The simplest approach to this problem is to learn the direct mapping between them. However, when the two sequences are high dimensional vectors, the direct mapping may result in a complex model with poor generalization. Therefore, many researchers in the machine learning community exploited the dimensionality reduction to learn the better models. The statistical reasoning about this approach is that given the proper subspace embedding, we can learn the simpler model with better generalization and make better predictions through it.

Our main interest in the thesis is how to learn the proper subspace embedding from a pair of sequences $\mathbf{X}$ and $\mathbf{Y}$ for the prediction tasks. We propose two ways to model the subspace embedding based on the relationship between them: generative way and discriminative way [5]. When we model the subspace in the generative way, we assume the co-occurrence of two sequences. In this approach, we are more interested in modeling the joint probability $P(\mathbf{X}, \mathbf{Y})$ given the subspace $\mathbf{Z}$. In general, the model in the generative approach describes the

casual dependencies and when the model assumption is correct the learning is easier than the discriminative approach with better generalization. In contrast, when we model the subspace embedding $\mathbf{Z}$ in the discriminative way, we focus on the regression between the input sequence $\mathbf{X}$ and the output sequence $\mathbf{Y}$. Therefore, the learning objective is to model the conditional likelihood $P(\mathbf{Y}|\mathbf{X})$ to optimize the prediction accuracy. The general advantage of the discriminative approach is that when the model assumption is incorrect, the learned model can lead to better prediction than the generative learning. Figure 1.3 depicts the graphical models of these two approaches.

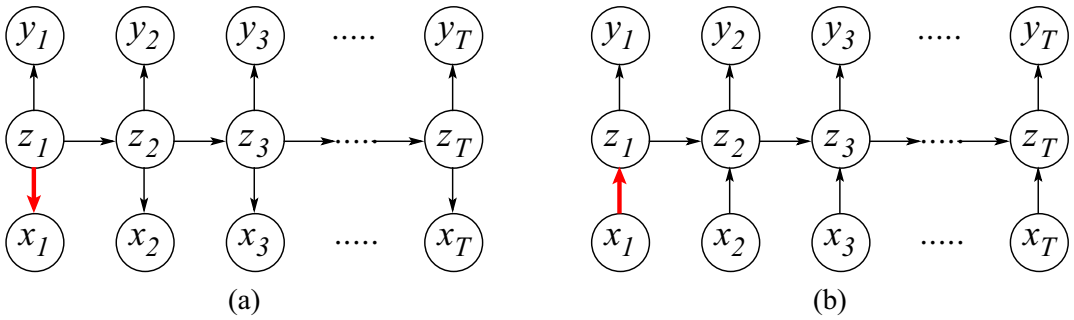


Figure 1.3: Graphical model for our approaches: (a) generative way (b) discriminative way.

1.5 Contribution

The main contributions of the thesis are:

- Nonlinear Dynamic System using a Marginal Autoregression Model (MAR): we present a new approach to subspace embedding of sequential data that explicitly accounts for their dynamic nature. We first model the space of sequences using a novel Marginal Auto-Regressive (MAR) formalism. A MAR model describes the space of sequences generated from all possible AR models. In the limit case, MAR describes all stable AR models. As such, the MAR model is weakly-parametric and can be used as a prior for an arbitrary sequence, without requiring the typical AR parameters such as the state transition matrix to be known. The embedding model is then defined using a probabilistic Gaussian Process Latent Variable (GPLVM) framework [9] with MAR as its prior. A GPLVM framework is particularly well suited for this task because of its probabilistic generative interpretation. The new hybrid GPLVM and MAR framework results in a

general model of the space of all nonlinear dynamic systems (NDS). It therefore has the potential to model nonlinear embeddings of a large family of sequences in theoretically sound manner. We empirically prove the advantage of our approach by applying the NDS model to modeling and tracking of the 3D human figure motion from a sequence of monocular images.

- Dynamic Probabilistic Latent Semantic Analysis (DPLSA): We propose a generative statistical approach to modeling sequential dyadic data that utilizes probabilistic latent semantic (PLSA) models. PLSA model has been successfully used to model the co-occurrence of dyadic data on problems such as image annotation where image features are mapped to word categories via latent variable semantics. We apply the PLSA approach to human motion tracking by extending it to a sequential setting where the latent variables describe intrinsic motion semantics linking human figure appearance to 3D pose estimates. This dynamic PLSA (DPLSA) approach is in contrast to many current methods that directly learn the often high-dimensional image-to-pose mappings and utilize subspace projections as a constraint on the pose space alone. As a consequence, such mappings may often exhibit increased computational complexity and insufficient generalization performance. We demonstrate the utility of the proposed model on a synthetic dataset and the task of 3D human motion tracking in monocular image sequences with arbitrary camera views. Our experiments show that the dynamic PLSA approach can produce accurate pose estimates at a fraction of the computational cost of alternative subspace tracking methods.

- Gaussian Process Manifold Kernel Dimensionality Reduction (GPMKDR): We addresses the problem of learning a low dimensional manifold that preserves information relevant for a general nonlinear regression. Instead of iterative solutions proposed in approaches to *sufficient dimension reduction* and its generalizations to kernel settings, such as the manifold kernel dimension reduction (mKDR), we show how a globally optimal solution in closed form can be obtained by formulating a related problem in a setting reminiscent of Gaussian Process (GP) regression. We then propose a generalization of the solution to arbitrary input points which is not usually mentioned in the previous literature. In a set of

experiments on various real world problems we show that the proposed GPMKDR can achieve significant gains in accuracy of prediction as well as interpretability, compared to other dimension reduction and regression schemes.

Chapter 2

Related Work

2.1 Subspace Embedding in Human Motion Modeling

Manifold learning approaches to motion modeling have attracted significant interest in the last several years. Brand [6] proposed nonlinear manifold learning that maps sequences of the input to paths of the learned manifold. Rosales and Sclaroff [7] proposed the Specialized Mapping Architecture (SMA) that utilizes forward mapping for the pose estimation task. Agarwal and Triggs [8] directly learned a mapping from image measurement to 3D pose using Relevance Vector Machine (RVM).

However, with high-dimensional data, it is often advantageous to consider a subspace e.g. the joint angles space that contains a compact representation of the actual figure motion. Principal Component Analysis (PCA) [9] is the most well-known linear dimensionality reduction technique. Although PCA has been applied to human tracking and other vision applications [10–12], it is insufficient to handle the non-linear behavior inherent to human motion. Non-linear manifold embedding of the training data in low dimensional spaces using isometric feature mapping (Isomap), Local linear (LLE) and spectral embedding [13–16], have shown success in recent approaches [17, 18]. While these techniques provide point-based embeddings implicitly modeling the nonlinear manifold through exemplars, they lack a fully probabilistic interpretation of the embedding process.

The GPLVM, a Gaussian Processes [19] model, produces a continuous mapping between the latent space and the high-dimensional data in a probabilistic manner [20]. Grochow *et al*. [21] use a Scaled GPLVM (SGPLVM) to model inverse kinematics for interactive computer animation. Tian *et al*. [22] use a GPLVM to estimate the 2D upper body pose from 2D silhouette features. However these approaches utilize simple temporal constraints in pose space that often introduce “curse of dimensionality” to nonlinear tracking methods such as particle filters.

Moreover, such methods fail to explicitly consider motion dynamics during the embedding process. Our work addresses both of these issues through the use of a novel marginal NDS model. Wang *et al*. [23] introduced Gaussian Process Dynamical Models (GPDM) that utilize dynamic priors for embedding. Our work extends the idea to tracking and investigates the impact of dynamics in the embedded space on tracking in real sequences.

2.2 Shared Subspace with Dyadic Data

Dyadic data refers to a domain with two sets of objects in which data is measured on pairs of units. One of the popular approaches for learning from this kind of data is the latent semantic analysis (LSA) that was devised for document indexing. Deerwester *et al*. [24] considered the term-document association data and used singular-value decomposition to decompose document matrix into a set of orthogonal matrices. LSA has been applied to a wide range of problems such as information retrieval and natural language processing [25, 26].

Probabilistic Latent Semantic Analysis (PLSA) [27] is a generalization of LSA to probabilistic settings. The main purpose of LSA and PLSA is to reveal semantic relations between the data entities by mapping the high dimensional data such text documents to a lower dimensional representation called latent semantic space. Some exemplary application areas of PLSA in computer vision include image annotation [28] and image category recognition [29, 30].

Human motion tracking is another application which model the matching between dyadic sequences. Recently, a GPLVM that produces a continuous mapping between the latent space and the high dimensional data in a probabilistic manner [20] was used for human motion tracking. Tian *et al*. [22] use a GPLVM to estimate the 2D upper body pose from 2D silhouette features. Urtasun *et al*. [31] exploit the SGPLVM for 3D people tracking. The GPDM [23] utilizing the dynamic priors for embedding is effectively used for 3D human motion tracking [32]. In [33], a marginal AR prior for GPLVM embedding is proposed and utilized for 3D human pose estimation from synthetic and real image sequences. Lawrence and Moore [34] propose the extension of GPLVM using a hierarchical model in which the conditional independency between human body parts is exploited with low dimensional non-linear manifolds. However,

these approaches utilize only the pose in latent space estimation and as a consequence, the optimized latent space cannot guarantee the proper dependency between the poses and the image observations in a regression setting.

Shon *et al*. [35] propose a shared latent structure model that utilizes the latent space that links corresponding pairs of observations from the multiple different spaces, and apply their model to image synthesis and robotic imitation of human actions. Although their model also utilizes GPLVM as the embedding model, their applications are limited to non-sequential cases and the linkage between two observations is explicit (*e.g*. image-image or pose-pose). The shared latent structure model using GPLVM is employed for pose estimation in [36]. This work focuses on the semi-supervised regression learning and makes use of unlabeled data (only pose or image) to regularize the regression model. In contrast, our work, using a statistical foundation of PLSA, focuses on the computational advantages of the shared latent space. In addition, it explicitly considers the latent dynamics and the multi-view setting ignored in [36].

2.3 Subspace Embedding with Regression

The problem of dimensionality reduction has been studied in many contexts including visualization of high dimensional data, noise reduction, and discovery of intrinsic data structure. Yan *et al*. [37] present a general framework called graph embedding that offers a unified view of linear and nonlinear dimensionality reduction methods. The original GPLVM produces a continuous manifold guided by one source of data (*e.g*. targets) in a probabilistic manner and can be extended to a shared latent variable model [35, 36] that deals with problems whose ultimate solution would best be represented by building a regressor between two domains (*e.g*. covariate and target). However, this extension does not explicitly postulate such a regressor and rather considers a *generative* model where both the covariate X and the target Y have a common but latent cause Z.

Li [38] first suggested to approach SDR as an inverse regression problem: if the distribution $P(Y|X)$ concentrates on a subspace of the input X space, then the inverse regression $E(X|Y)$ should lie in the same subspace. A technique known as the sliced inverse regression (SIR) was proposed, based on the idea that the sample mean of X is computed within each

slice of Y and PCA is used to aggregate these means into an estimate of effective subspace in regression. Since then many approaches such as Principal Hessian directions (PHd) [39], sliced average variance estimation (SAVE) [40], and contour regression [41] have been developed from the same methodological foundation. However, these methods, from an inverse regression perspective, have to impose the restrictive assumptions on the probability of X such as the elliptical symmetry of the marginal distribution. In addition, PHd and contour regression are applicable only to a one-dimensional response and the maximum dimension of a subspace of SIR is $p - 1$ when the output Y takes its value in a finite set of p elements.

Kernel Dimension Reduction (KDR) was recently proposed as another methodology for SDR [42, 43] in which no assumption regarding the marginal distribution of X is made. KDR treats the problem of dimensionality reduction as the one of finding a low-dimensional effective subspace for X and provides the contrast function for estimation of this space using reproducing kernel Hilbert spaces (RKHS). Alternatively, Sajama *et al*. [44] proposed a supervised dimensionality method using mixture models for a classification problem in which the subspace retaining the maximum possible mutual information between feature vectors and class labels is selected. However, it is limited only to classification and restricted to a Gaussian distribution. Yang *et al*. [45] proposed a way of modifying basic nonlinear dimensionality reduction methods (*e.g*. LLE) by taking into consideration prior information that exactly maps certain data points. The approach does not consider SDR and the side information for embedding is the prior knowledge of a correct embedding instead of the responses in regression.

Chapter 3

Marginal Nonlinear Dynamic System

Before we present our two approaches to the subspace embedding of dyadic sequences, We develop a framework incorporating dynamics into the process of learning low-dimensional representations of sequences. The chapter is organized as follows. We first define the family of MAR models and study some properties of the space of sequences modeled by MAR. Next, we show that MAR and GPLVM result in a model of the space of all NDS sequences and discuss its properties. The utility of the new framework is examined through a set of experiments with synthetic and real data. In particular, we apply the new framework to modeling and tracking of 3D human figure motion from a sequence of monocular images.

3.1 Marginal Auto-Regressive Model

In this section, a novel marginal dynamic model describing the space of all stable auto-regressive sequences is proposed to model the dynamics of an unknown subspace.

3.1.1 Definition

Consider a sequence $\mathbf{X}$ of length T of N-dimensional real-valued vectors $\mathbf{x}_t = [\mathbf{x}_{t,0}\mathbf{x}_{t,1}...\mathbf{x}_{t,N-1}] \in \Re^{1\times N}$. Suppose sequence $\mathbf{X}$ is generated by the first order AR model $AR(\mathbf{A})$:

$$\mathbf{x}_t = \mathbf{x}_{t-1}\mathbf{A} + \mathbf{w}_t,\ t = 0, ..., T-1 \tag{3.1}$$

where $\mathbf{A}$ is a specific $N \times N$ state transition matrix and $\mathbf{w}_t$ is a white iid Gaussian noise with precision, α: $\mathbf{w}_t \sim \mathcal{N}(0, \alpha^{-1}\mathbf{I})$. Assume that, without loss of generality, the initial condition $\mathbf{x}_{-1}$ has normal multivariate distribution with zero mean and unit precision: $\mathbf{x}_{-1} \sim \mathcal{N}(0, \mathbf{I})$.

We adopt a convenient representation of sequence $\mathbf{X}$ as a $T\times N$ matrix $\mathbf{X} = [\mathbf{x}'_0\mathbf{x}'_1...\mathbf{x}'_{T-1}]'$ whose rows are the vector samples from the sequence. Using this notation Equation (3.1) can

be written as

$$\mathbf{X} = \mathbf{X}_{\Delta}\mathbf{A} + \mathbf{W}$$

where $\mathbf{W} = [\mathbf{w}_0'\mathbf{w}_1'...\mathbf{w}_{T-1}']'$ and $\mathbf{X}_{\Delta}$ is a *shifted/delayed* version of $\mathbf{X}$, $\mathbf{X}_{\Delta} = [\mathbf{x}_{-1}'\mathbf{x}_0'...\mathbf{x}_{T-2}']'$. Given the state transition matrix $\mathbf{A}$ and the initial condition, the AR sequence samples have the joint density function

$$P(\mathbf{X}|\mathbf{A}, \mathbf{x}_{-1}) = (2\pi)^{-\frac{NT}{2}} \exp\left\{-\frac{1}{2} tr\left\{(\mathbf{X} - \mathbf{X}_{\Delta}\mathbf{A})(\mathbf{X} - \mathbf{X}_{\Delta}\mathbf{A})'\right\}\right\}. \tag{3.2}$$

The density in Equation (3.2) describes the distribution of samples in a T-long sequence for a particular instance of the state transition matrix $\mathbf{A}$. However, we are interested in the distribution of all AR sequences, regardless of the value of $\mathbf{A}$. In other words, we are interested in the marginal distribution of AR sequences, over all possible parameters $\mathbf{A}$.

Assume that all elements a_{ij} of $\mathbf{A}$ are iid Gaussian with zero mean and unit precision, $a_{ij} \sim \mathcal{N}(0, 1)$. Under this assumption, it can be shown [46] that the *marginal* distribution of the AR model becomes

$$\begin{aligned} P(\mathbf{X}|\mathbf{x}_{-1}, \alpha) &= \int_{\mathbf{A}} P(\mathbf{X}|\mathbf{A}, \mathbf{x}_{-1}) P(\mathbf{A}|\alpha) d\mathbf{A} \\ &= (2\pi)^{-\frac{NT}{2}} |\mathbf{K}_{xx}(\mathbf{X}, \mathbf{X})|^{-\frac{N}{2}} exp\left\{-\frac{1}{2} tr\{\mathbf{K}_{xx}(\mathbf{X}, \mathbf{X})^{-1}\mathbf{X}\mathbf{X}'\}\right\} \end{aligned} \tag{3.3}$$

where

$$\mathbf{K}_{xx}(\mathbf{X}, \mathbf{X}) = \mathbf{X}_{\Delta}\mathbf{X}_{\Delta}' + \alpha^{-1}\mathbf{I}. \tag{3.4}$$

We call this density the *Marginal AR* or MAR density. α is the hyperparameter of this class of models, $MAR(\alpha)$. Intuitively, Equation (3.3) favors those samples in $\mathcal{X}$ that do not change significantly from t to $t + 1$ and $t - 1$. The graphical representation of the MAR model is depicted in Figure 3.1. Different treatments of the nodes are represented by different shades.

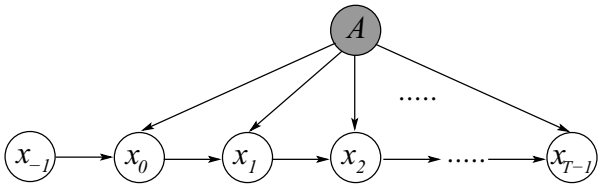


Figure 3.1: Graphical representation of MAR model. White shaded nodes are optimized while the grey shaded node is marginalized.

The MAR density models the distribution of all (AR) sequences of length T in the space $\mathcal{X} = \Re^{T \times N}$. Note that while the error process of an AR model has a Gaussian distribution, the MAR density is not Gaussian. We illustrate this in Figure 3.2. The figure shows joint pdf values for four different densities: MAR, periodic MAR (see Section 3.1.2), AR(2), and a circular Gaussian, in the space of length-two scalar-valued sequences $[\mathbf{x}_0 \mathbf{x}_1]'$. In all four cases we assume zero-mean, unit precision Gaussian distribution of the initial condition. All models have the mode at $(0, 0)$. The distribution of the AR model is multivariate Gaussian with the principal variance direction determined by the state transition matrix $\mathbf{A}$. However, the MAR models define non-Gaussian distributions with no circular symmetry and with directional bias. This property of MAR densities is important when viewed in the context of sequence subspace embeddings, which we discuss in Section 3.2.

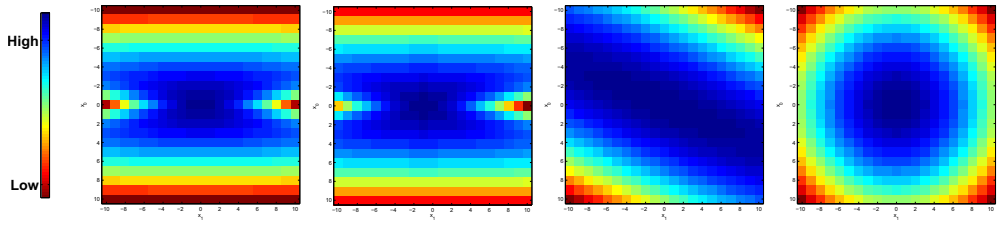


Figure 3.2: Distribution of length-two sequences of 1D samples under MAR, periodic MAR, AR, and independent Gaussian models.

3.1.2 Higher-Order Dynamics

The above definition of MAR models can be easily extended to families of arbitrary D-th order AR sequences. In that case the state transition matrix $\mathbf{A}$ is replaced by an $ND \times N$ matrix $\mathbf{A} = [\mathbf{A}_1' \mathbf{A}_2' ... \mathbf{A}_D']'$ and $\mathbf{X}_\Delta$ by $[\mathbf{X}_\Delta \mathbf{X}_{1\Delta} ... \mathbf{X}_{D\Delta}]$. Hence, a $MAR(\alpha, D)$ model describes a general space of all D-th order AR sequences. Using this formulation one can also model specific classes of dynamic models. For instance, a class of all periodic models can be formed by setting $\mathbf{A} = [\mathbf{A}_1' \ -\mathbf{I}]'$, where $\mathbf{I}$ is an identity matrix.

3.1.3 Nonlinear Dynamics

In Equation (3.1) and Equation (3.3) we assumed linear families of dynamic systems. One can generalize this approach to nonlinear dynamics of the form $\mathbf{x}_t = g(\mathbf{x}_{t-1}|\zeta)\mathbf{A}$, where $g(\cdot|\zeta)$ is a

nonlinear mapping to an L-dimensional subspace and $\mathbf{A}$ is a $L \times N$ linear mapping. In that case $\mathbf{K}_{xx}$ becomes a nonlinear kernel using justification similar to e.g. [20]. While nonlinear kernels often have potential benefits, such as robustness, they also preclude closed-form solutions of linear models. In our preliminary experiments we have not observed significant differences between MAR and nonlinear MAR.

3.1.4 Justification of MAR Models

The choice of the prior distribution of the AR model's state transition matrix leads to the MAR density in Equation (3.3). One may wonder, however, if the choice of iid $\mathcal{N}(0,1)$ results in a physically meaningful space of sequences. We suggest that, indeed, such choice may be justified.

Namely, Girko's circular law [47] states that if $\frac{1}{N}\mathbf{A}$ is a random $N \times N$ matrix with $\mathcal{N}(0,1)$ iid entries, then in the limit case of large $N(> 20)$ all real and complex eigenvalues of $\mathbf{A}$ are *uniformly distributed on the unit disk*. For small N, the distribution shows a concentration along the real line. Consequently, the resulting space of sequences described by the MAR model is that of *all stable AR systems*.

3.2 Nonlinear Dynamic System Models

In this section we develop a Nonlinear Dynamic System view of the sequence subspace reconstruction problem that relies on the MAR representation of the previous section. In particular, we use the MAR model to describe the structure of the subspace of sequences to which the extrinsic representation will be mapped using the GPLVM framework of [20].

3.2.1 Definition

Let $\mathbf{Y}$ be an extrinsic or measurement sequence of duration T of M-dimensional samples. Define $\mathbf{Y}$ as the $T \times M$ matrix representation of this sequence, similar to the definition in Section 3.1.1, $\mathbf{Y} = [\mathbf{y}_0'\mathbf{y}_1'...\mathbf{y}_{T-1}']'$. We assume that $\mathbf{Y}$ is a result of the process $\mathbf{X}$ in a lower-dimensional MAR subspace $\mathcal{X}$, defined by a nonlinear generative or forward mapping

$$\mathbf{Y} = f(\mathbf{X}|\theta)\mathbf{C} + \mathbf{V}.$$

$f(\cdot)$ is a nonlinear $\Re^N \rightarrow \Re^L$ mapping, $\mathbf{C}$ is a linear $L \times M$ mapping, and $\mathbf{V}$ is a Gaussian noise with zero-mean and precision β.

To recover the intrinsic sequence $\mathbf{X}$ in the embedded space from sequence $\mathbf{Y}$ it is convenient not to focus, at first, on the recovery of the specific mapping $\mathbf{C}$. Hence, we consider the family of mappings where $\mathbf{C}$ is a stochastic matrix whose elements are iid $c_{ij} \sim \mathcal{N}(0,1)$. Marginalizing over all possible mappings $\mathbf{C}$ yields a marginal Gaussian Process [19] mapping:

$$\begin{aligned} P(\mathbf{Y}|\mathbf{X},\beta,\theta) &= \int_{\mathbf{C}} P(\mathbf{Y}|\mathbf{X},\mathbf{C},\theta)P(\mathbf{C}|\beta)d\mathbf{C} \\ &= (2\pi)^{-\frac{MT}{2}}|\mathbf{K}_{yx}(\mathbf{X},\mathbf{X})|^{-\frac{M}{2}} exp\left\{-\frac{1}{2}tr\{\mathbf{K}_{yx}(\mathbf{X},\mathbf{X})^{-1}\mathbf{Y}\mathbf{Y}'\}\right\} \end{aligned}$$

where

$$\mathbf{K}_{yx}(\mathbf{X},\mathbf{X}) = f(\mathbf{X}|\theta)f(\mathbf{X}|\theta)' + \beta^{-1}\mathbf{I}.$$

Notice that in this formulation the $\mathbf{X} \rightarrow \mathbf{Y}$ mapping depends on the inner product $\langle f(\mathbf{X}), f(\mathbf{X})\rangle$. The knowledge on the actual mapping f is not necessary; a mapping is uniquely defined by specifying a positive-definite kernel $\mathbf{K}_{yx}(\mathbf{X},\mathbf{X}|\theta)$ with entries $\mathbf{K}_{yx}(i,j) = k(\mathbf{x}_i,\mathbf{x}_j)$ parameterized by the hyperparameter θ. A variety of linear and non-linear kernels (RBF, square exponential, various robust kernels) can be used as $\mathbf{K}_{yx}$. Hence, our likelihood model is a non-linear Gaussian process model, as suggested by [20]. Figure 3.3 shows the graphical model of NDS.

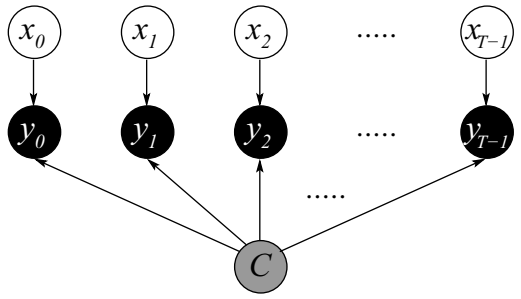


Figure 3.3: Graphical model of NDS. White shaded nodes are optimized while the grey shaded node is marginalized and the black shaded nodes are observed variables.

By joining the MAR model and the NDS model, we have constructed a Marginal Nonlinear Dynamic System (MNDS) model that describes the joint distribution of all measurement and all intrinsic sequences in a $\mathcal{Y} \times \mathcal{X}$ space:

$$P(\mathbf{X},\mathbf{Y}|\alpha,\beta,\theta) = P(\mathbf{X}|\alpha)P(\mathbf{Y}|\mathbf{X},\beta,\theta). \tag{3.5}$$

The MNDS model has a MAR prior $P(\mathbf{X}|\alpha)$, and a Gaussian process likelihood $P(\mathbf{Y}|\mathbf{X},\beta,\theta)$. Thus it places the intrinsic sequences $\mathbf{X}$ in the space of all AR sequences. Given an intrinsic sequence $\mathbf{X}$, the measurement sequence $\mathbf{Y}$ is zero-mean normally distributed with the variance determined by the nonlinear kernel $\mathbf{K}_{yx}$ and $\mathbf{X}$.

3.2.2 Inference

Given a sequence of measurements $\mathbf{Y}$ one would like to infer its subspace representation $\mathbf{X}$ in the MAR space, without needing to first determine a particular family of AR models $AR(\mathbf{A})$, nor the mapping $\mathbf{C}$. Equation (3.5) shows that this task can be, in principle, achieved using the Bayes rule $P(\mathbf{X}|\mathbf{Y},\alpha,\beta,\theta) \propto P(\mathbf{X}|\alpha)P(\mathbf{Y}|X,\beta,\theta)$.

However, this posterior is non-Gaussian because of the nonlinear mapping f and the MAR prior. One can instead attempt to estimate the mode $\mathbf{X}^*$

$$\mathbf{X}^* = \arg\max_{\mathbf{X}} \{\log P(\mathbf{X}|\alpha) + \log P(\mathbf{Y}|\mathbf{X},\beta,\theta)\}$$

using nonlinear optimization such as the Scaled Conjugate Gradient in [20].

To effectively use a gradient-based approach, one needs to obtain expressions for gradients of the log-likelihood and the log-MAR prior. Note that the expressions for MAR gradients are more complex than those of e.g. GP due to a linear dependency between $\mathbf{X}$ and $\mathbf{X}_\Delta$ (see Appendix A).

3.2.3 Learning

The MNDS space of sequences is parameterized using a set of hyperparameters (α,β,θ) and the choice of the nonlinear kernel $\mathbf{K}_{yx}$. Given a set of sequences $\{\mathbf{Y}^{(i)}\}, i = 1,..,S$ the learning task can be formulated as a ML/MAP estimation problem

$$(\alpha^*,\beta^*,\theta^*)|_{\mathbf{K}_{yx}} = \arg\max_{\alpha,\beta,\theta} \prod_{i=1}^{S} P(\mathbf{Y}^{(i)}|\alpha,\beta,\theta).$$

One can use a generalized EM algorithm to obtained the ML parameter estimates recursively from two fixed-point equations:

E-step:

$X^{(i)*} = \arg\max_X P(Y, X^{(i)}|\alpha^*, \beta^*, \theta^*)$

M-step:

$(\alpha^*, \beta^*, \theta^*) = \arg\max_{(\beta,\alpha,\theta)} \prod_{i=1}^{K} P(Y^{(i)}, X^{(i)*}|\alpha, \beta, \theta)$

3.2.4 Learning of Explicit NDS Model

Inference and learning of MNDS models results in the embedding of the measurement sequence $\mathbf{Y}$ into the space of all NDS/AR models. Given $\mathbf{Y}$, the embedded sequences $\mathbf{X}$ estimated in Section 3.2.3 and MNDS parameters α, β, θ, the explicit AR model can be easily reconstructed using the ML estimation of sequence $\mathbf{X}$, e.g.:

$$\mathbf{A}^* = (\mathbf{X}'_\Delta \mathbf{X}_\Delta)^{-1} \mathbf{X}'_\Delta \mathbf{X}.$$

Because the embedding was defined as a GP, the likelihood function $P(\mathbf{y}_t|\mathbf{x}_t, \beta, \theta)$ follows a well-known result from GP theory: $\mathbf{y}_t|\mathbf{x}_t \sim \mathcal{N}(\mu, \sigma^2\mathbf{I})$ where

$$\mu = \mathbf{Y}'\mathbf{K}_{yx}(\mathbf{X}, \mathbf{X})^{-1}\mathbf{K}_{yx}(\mathbf{X}, \mathbf{x}_t) \tag{3.6}$$

$$\sigma^2 = \mathbf{K}_{yx}(\mathbf{x}_t, \mathbf{x}_t) - \mathbf{K}_{yx}(\mathbf{X}, \mathbf{x}_t)'\mathbf{K}_{yx}(\mathbf{X}, \mathbf{X})^{-1}\mathbf{K}_{yx}(\mathbf{X}, \mathbf{x}_t). \tag{3.7}$$

The two components fully define the explicit NDS.

In summary, a complete sequence modeling algorithm consists of the following set of steps.

Input : Measurement sequence $\mathbf{Y}$ and kernel family $\mathbf{K}_{yx}$

Output: $NDS(\mathbf{A}, \beta, \theta)$

1) Learn subspace embedding $MNDS(\alpha, \beta, \theta)$ model of training sequences $\mathbf{Y}$ as described in Section 3.2.3.

2) Learn explicit subspace and projection model $NDS(\mathbf{A}, \beta, \theta)$ of $\mathbf{Y}$ as described in Section 3.2.4.

Algorithm 1: NDS learning.

3.2.5 Inference in Explicit NDS Model

The choice of the nonlinear kernel $\mathbf{K}_{yx}$ results in a nonlinear dynamic system model of training sequences $\mathbf{Y}$. The learned model can then be used to infer subspace projections of a new

sequence from the same family. Because of the nonlinearity of the embedding, one cannot apply the linear forward-backward or Kalman filtering/smoothing inference. Rather, it is necessary to use nonlinear inference methods such as (I)EKF or particle filtering/smoothing.

It is interesting to note that one can often use a relatively simple sequential nonlinear optimization in place of the above two inference methods:

$$\mathbf{x}_t^* = \arg\max_{\mathbf{x}_t} P(\mathbf{y}_t|\mathbf{x}_t, \beta^*, \theta^*) P(\mathbf{x}_t|\mathbf{x}_{t-1}^*, \mathbf{A}^*).$$

Such sequential optimization yields local modes of the true posterior $P(\mathbf{X}|\mathbf{Y})$. While one would expect such approximation to be valid in situations with few ambiguities in the measurement space and models learned from representative training data, our experiments show the method to be robust across a set of situations. However, dynamics seem to play a crucial role in the inference process.

3.2.6 Example

We illustrate the concept of MNDS on a simple synthetic example. Consider the AR model $AR(2)$ from Section 3.1. Sequence $\mathbf{X}$, generated by the model, is projected to the space $\mathcal{Y} = \Re^{2\times 3}$ using a linear conditional Gaussian model $\mathcal{N}(\mathbf{XC}, \mathbf{I})$. Figure 3.4 shows negative likelihood over the space $\mathcal{X}$ of the MNDS, a marginal model (GP) with independent Gaussian priors, a GP with the exact $AR(2)$ prior, and a full LDS with exact parameters. All likelihoods are computed for the fixed $\mathbf{Y}$. Note that the GP with Gaussian prior assumes no temporal structure in the data. This example shows that, as expected, the maximum likelihood subspace

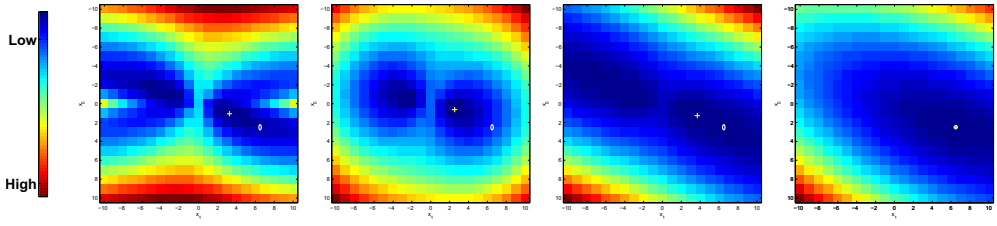


Figure 3.4: Negative log-likelihood of length-two sequences of 1D samples under MNDS, GP with independent Gaussian priors, GP with exact AR prior and LDS with the true process parameters. "o" mark represents the optimal estimate $\mathbf{X}^*$ inferred from the true LDS model. "+" shows optimal estimates derived using the three marginal models.

estimates of the MNDS model fall closer to the "true" LDS estimates than those of the non-sequential model. This property holds in general. Figure 3.5 shows the distribution of optimal negative log likelihood scores, computed at corresponding $\mathbf{X}^*$, of the four models over a 10000 sample of $\mathbf{Y}$ sequences generated from the true LDS model. Again, one notices that MNDS

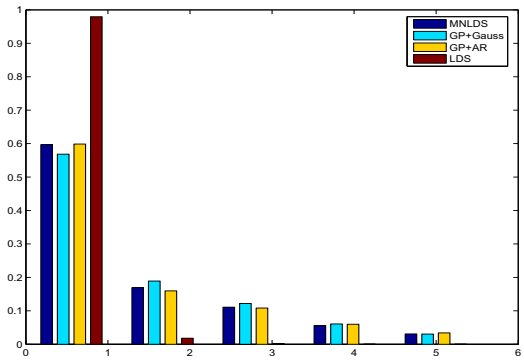


Figure 3.5: Normalized histogram of optimal negative log-likelihood scores for MNDS, a GP model with a Gaussian prior, a GP model with exact AR prior and LDS with the true parameters.

has a lower mean and mode than the non-sequential model, GP+Gauss, indicating MNDS's better fit to the data. This suggests that MNDS may result in better subspace embeddings than the traditional GP model with independent Gaussian priors.

3.3 Human Motion Modeling using MNDS

When the dimension of image feature vector $\mathbf{z}_t$ is much smaller than the dimension of pose vector $\mathbf{y}_t$ (e.g. 10-dimensional vector of alt Moments vs. 59-dimensional joint angle vector of motion capture data), estimating the pose given the feature becomes the problem of predicting a higher dimensional projection in the model $P(\mathbf{Z}|\mathbf{Y}, \theta_{zy})$. It is an undetermined problem. In this case, we can utilize the practical approximation by modeling $P(\mathbf{Y}|\mathbf{Z})$ rather than $P(\mathbf{Z}|\mathbf{Y})$ - It yielded better results and still allowed a fully GP-based framework. That is to say, the mapping into the 3D pose space from the feature space is given by a Gaussian process model $P(\mathbf{Y}|\mathbf{Z}, \theta_{yz})$ with a parametric kernel $\mathbf{K}_{yz}(\mathbf{z}_t, \mathbf{z}_t|\theta_{yz})$.

As a result, the joint *conditional* model of the pose sequence $\mathbf{Y}$ and intrinsic motion $\mathbf{X}$, given the sequence of image features $\mathbf{Z}$ is approximated by

$$P(\mathbf{X}, \mathbf{Y}|\mathbf{Z}, \mathbf{A}, \beta, \theta_{yz}, \theta_{yx}) \approx P(\mathbf{Y}|\mathbf{Z}, \theta_{yz})P(\mathbf{X}|\mathbf{A})P(\mathbf{Y}|\mathbf{X}, \beta, \theta_{yx}).$$

3.3.1 Learning

In the training phase, both the image features $\mathbf{Z}$ and the corresponding poses $\mathbf{Y}$ are known. Hence, the learning of GP and NDS models becomes decoupled and can be accomplished using the NDS learning formalism presented in the previous section and a standard GP learning approach [19].

Input : Image sequence $\mathbf{Z}$ and joint angle sequence $\mathbf{Y}$

Output: Human motion model.

1) Learn Gaussian Process model $P(\mathbf{Y}|\mathbf{Z}, \theta_{yz})$ using *e.g.* [19].

2) Learn NDS model $P(\mathbf{X}, \mathbf{Y}|\mathbf{A}, \beta, \theta_{yx})$ as described in Section 3.2.

Algorithm 2: Human motion model learning.

3.3.2 Inference and Tracking

Once the models are learned they can be used for tracking of the human figure in video. Because both NDS and GP are nonlinear mappings, estimating current pose $\mathbf{y}_t$ given a previous pose and intrinsic motion space estimates $P(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}|\mathbf{Z}_{0..t})$ will involve nonlinear optimization or linearizion, as suggested in Section 3.2.5. In particular, optimal point estimates $\mathbf{x}_t^*$ and $\mathbf{y}_t^*$ are the result of the following nonlinear optimization problem:

$$(\mathbf{x}_t^*, \mathbf{y}_t^*) = \arg\max_{\mathbf{x}_t, \mathbf{y}_t} P(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{A}) P(\mathbf{y}_t|\mathbf{x}_t, \beta, \theta_{yx}) P(\mathbf{y}_t|\mathbf{z}_t, \theta_{yz}). \tag{3.8}$$

The point estimation approach is particularly well suited for a particle-based tracker. Unlike some traditional approaches that only consider the pose space representation, tracking in the low dimensional intrinsic space has the potential to avoid problems associated with sampling in high-dimensional spaces.

A sketch of the human motion tracking algorithm using a particle filter with N_P particles and weights $(w^{(i)}, i = 1, ..., N_P)$ is shown below. We apply this algorithm to a set of tracking problems described in Section 3.4.2.

Input : Image $\mathbf{z}_t$, Human motion model (GP+NDS) and prior point estimates

$$(w_{t-1}^{(i)}, \mathbf{x}_{t-1}^{(i)}, \mathbf{y}_{t-1}^{(i)})|\mathbf{Z}_{0..t-1}, i = 1, ..., N_P.$$

Output: Current pose/intrinsic state estimates

$$(w_t^{(i)}, \mathbf{x}_t^{(i)}, \mathbf{y}_t^{(i)})|\mathbf{Z}_{0..t}, i = 1, ..., N_P$$

1) Draw the initial estimates $\mathbf{x}_t^{(i)} \sim p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}, \mathbf{A})$.

2) Compute the initial poses $\mathbf{y}_t^{(i)}$ from the initial $\mathbf{x}_t^{(i)}$ and NDS model.

3) Find optimal estimates $(\mathbf{x}_t^{(i)}, \mathbf{y}_t^{(i)})$ using nonlinear optimization in Equation (3.8). 4) Find point weights

$w_t^{(i)} \sim P(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}, \mathbf{A})P(\mathbf{y}_t^{(i)}|\mathbf{x}_t^{(i)}, \beta, \theta_{yx})P(\mathbf{y}_t^{(i)}|\mathbf{z}_t, \theta_{yz})$.

Algorithm 3: Particel filter in human motion tracking.

3.4 Experiments

3.4.1 Synthetic Data

In our first experiment we examine the utility of MAR priors in a subspace selection problem. A second order AR model is used to generate sequences in a $\Re^{T\times 2}$ space; the sequences are then mapped to a higher dimensional nonlinear measurement space. An example of the measurement sequence, a periodic curve on the Swiss-roll surface, is depicted in Figure 3.6.

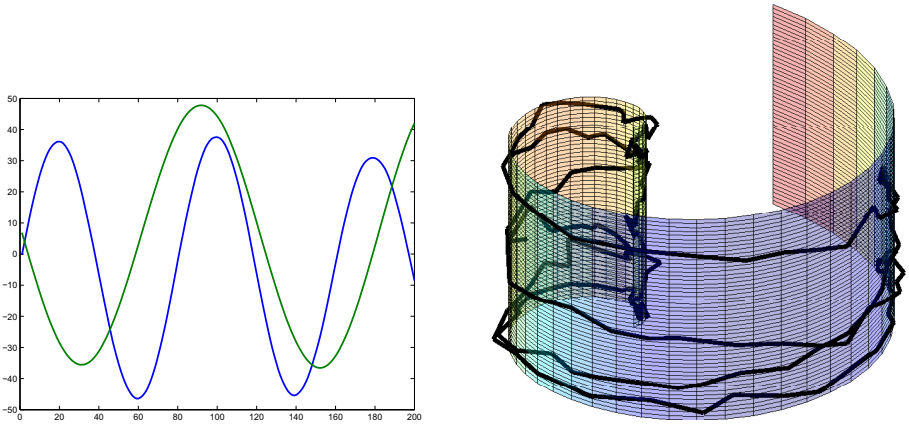


Figure 3.6: A periodic sequence in the intrinsic subspace and the measured sequence on the Swiss-roll surface.

We apply two different methods to recover the intrinsic sequence subspace: MNDS with an

RBF kernel and a GPLVM with the same kernel and independent Gaussian priors. Estimated embedded sequences are shown in Figure 3.7. The intrinsic motion sequence inferred by the

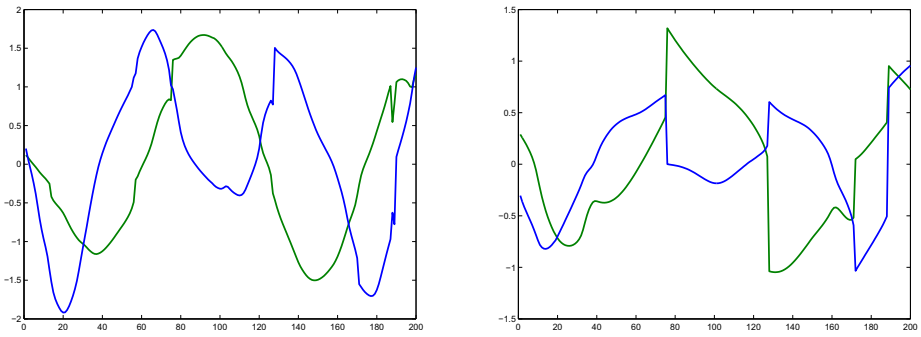


Figure 3.7: Recovered embedded sequences. Left: MNDS. Right: GPLVM with iid Gaussian priors.

MNDS model more closely resembles the "true" sequence in Figure 3.6. Note that one dimension (blue/dark) is reflected about the horizontal axis, because the embeddings are unique up to an arbitrary rotation. These results confirm that proper dynamic priors may have crucial role in learning of embedded sequence subspaces. We study the role of dynamics in tracking in the following section.

3.4.2 Human Motion Data

We conducted experiments using a database of motion capture data for a 59 d.o.f. body model from the CMU Graphics Lab Motion Capture Database [1]. Figure 3.8 shows the latent space resulting from the original GPLVM and our MNDS model. Note that there are breaks in the intrinsic sequence of the original GPLVM. On the other hand, the trajectory in the embedded space of MNDS model is smoother, without sudden breaks. Note that the precision for the points corresponding to the training poses is also higher in our MNDS model.

For the experiments on human motion tracking, we utilize synthetic images as our training data similar to [8,22]. Our database consists of seven walking sequences of around 2000 frames total. The data was generated using software (3D human model and Maya binaries) generously provided by the authors of [48, 49]. We train our GP and NDS models with one sequence of 250 frames and test on the remaining sequences. In our experiments, we exclude 15 joint angles that exhibit small movement during walking (e.g. clavicle and figures joint) and use

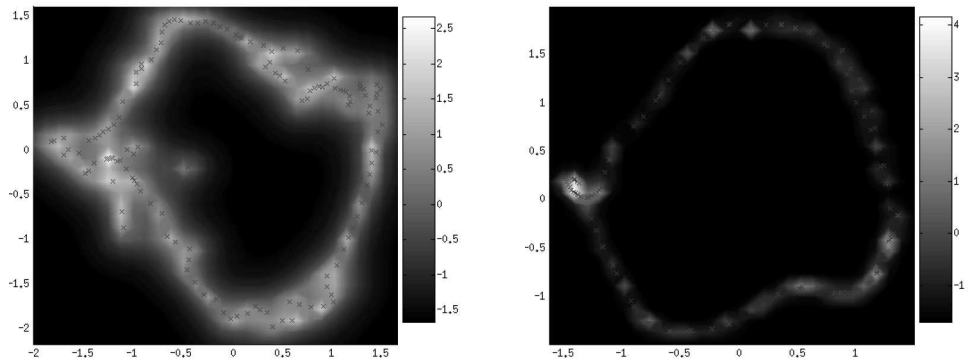


Figure 3.8: Latent space with the grayscale map of log precision. Left: pure GPLVM. Right: MNDS.

the remaining 44 joints. Our choice of image features are the silhouette-based Alt moments used in [7, 22]. The scale and translational invariance of Alt moments makes them suitable to a motion modeling task with little or no image-plane rotation.

In the model learning phase we utilize the approach proposed in Section 3.2. Once the model is learned, we apply the two tracking/inference approaches in Section 3.3 to infer motion states and poses from sequences of silhouette images. The pose estimation results with the two different models show little difference. The big difference between two models is the speed, which we discuss in the following Section 4.4.2.

Figure 3.9 depicts a sequence of estimated poses. The initial estimates for gradient search are determined by the nearest neighborhood matching in the Alt moments space alone. To evaluate our NDS model, we estimate the same input sequence with the original GPLVM tracking in [22]. Although the silhouette features are informative for human pose estimation, they are also prone to ambiguities such as the left/right side changes. Without proper dynamics modeling, the original GPLVM fails to estimate the correct poses because of this ambiguity.

The accuracy of our tracking method is evaluated using the mean RMS error between the true and the estimated joint angles [8], $D(\mathbf{y}, \mathbf{y}') = \frac{1}{44}\sum_{i=1}^{44} |(\mathbf{y}_i - \mathbf{y}'_i) mod \pm 180^o|$. The first column of Figure 3.10 displays the mean RMS errors over the 44 joint angles, estimated using three different models. The testing sequence consists of 320 frames. The mean error for the NDS model is in the range $3^o \sim 6^o$. The inversion of right and left legs causes significant errors in the original GPLVM model. Introduction of simple dynamics in the pose space similar to [31] was not sufficient to rectify the "static" GPLVM problem. The second column of Figure 3.10

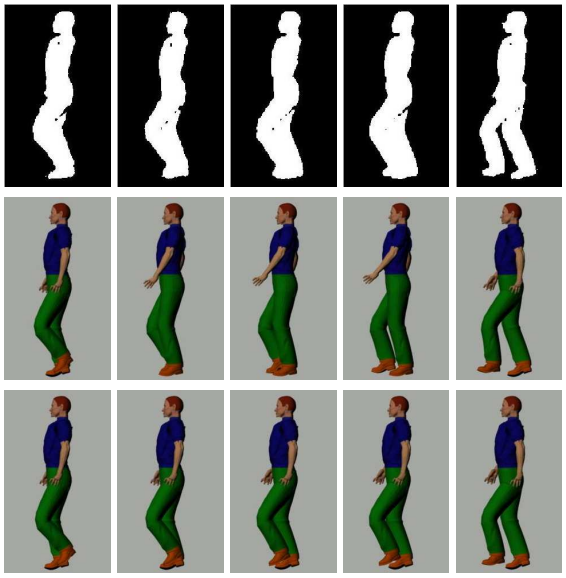


Figure 3.9: Tracking results. First row: input image silhouettes. Remaining rows show reconstructed poses. Second row: GPLVM model. Third row: NDS model.

shows examples of trajectories in the embedded space corresponding to the pose estimates with the three different models. The points inferred from our NDS model follow the path defined by the MAR model, making them temporally consistent. The other two methods produced less-than-smooth embeddings.

We applied the algorithm to tracking of various real monocular image sequences. The data used in these experiments was the sideview sequence in CMU mobo database made publicly available under the HumanID project [50]. Figure 3.11 shows one example of our tracking result. This testing sequence consists of 340 frames. Because a slight mismatch in motion dynamics between the training and the test sequences, reconstructed poses are not geometrically perfect. However the overall result sequence depicts a plausible walking motion that agrees with the observed images.

It is also interesting to note that in a number of tracking experiments, it was sufficient to carry a very small number of particles (~ 1) in the point-based tracker of Algorithm 3. In most cases all particles clustered in a small portion of the motion subspace $\mathcal{X}$, even in ambiguous situations induced by silhouette-based features. This indicates that the presence of dynamics had an important role in disambiguating statically similar poses.

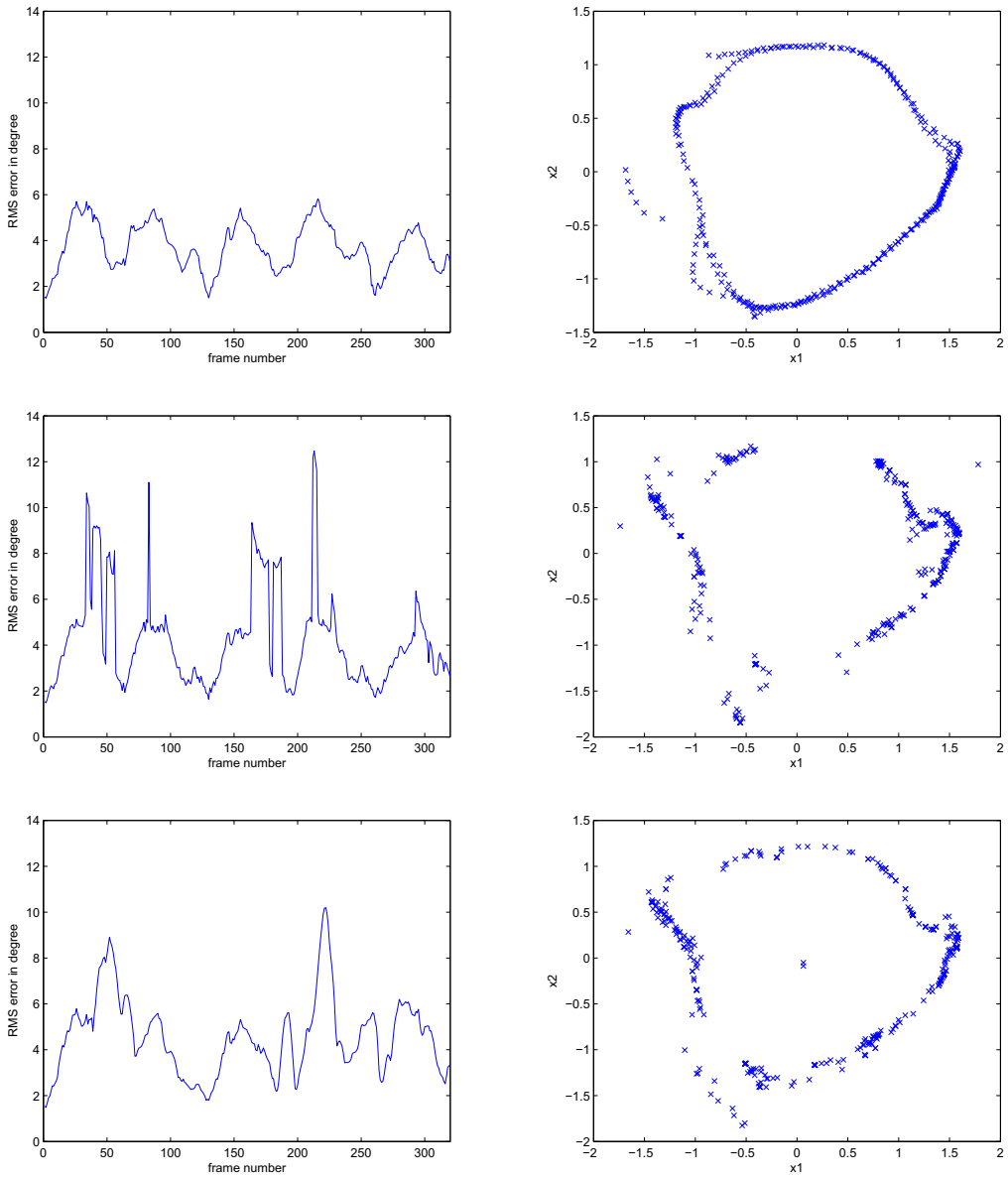


Figure 3.10: Mean angular pose RMS errors and 2D latent space trajectories. First row: tracking using our NDS model. Second row: original GPLVM tracking. Third row: tracking using simple dynamics in the pose space.

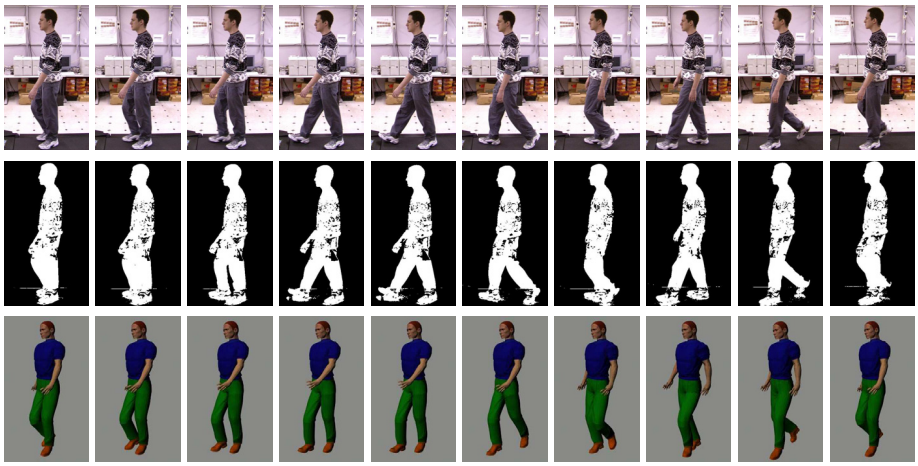


Figure 3.11: Tracking results. First row: input real walking images. Second row: image silhouettes. Third row: images of the reconstructed 3D pose.

3.5 Summary and Contribution

We proposed a novel method for embedding of sequences into subspaces of dynamic models. In particular, we propose a family of marginal AR (MAR) subspaces that describe all stable AR models. We show that a generative nonlinear dynamic system (NDS) can then be learned from a hybrid of Gaussian (latent) process models and MAR priors, a marginal NDS (MNDS). As a consequence, learning of NDS models and state estimation/tracking can be formulated in this new context. Several synthetic examples demonstrate the potential utility of the NDS framework and display its advantages over traditional static methods in dynamic domains. We also test the proposed approach on the problem of the 3D human figure tracking in sequences of monocular images. Our results indicate that dynamically constructed embeddings using NDS can resolve tracking ambiguities that may plague static as well as less principled dynamic approaches.

Chapter 4

Dynamic Probabilistic Latent Semantic Analysis

In this chapter, we present our generative way to model the subspace embedding of dyadic sequences. In particular, we focus on the human motion tracking task where we utilize the latent space to model the matching between the input image features, x and the poses, y. Although we suggested one possible way to model the human motion tracking using our MNDS model in Section 3.3, the model is not appropriate for the real video tracking with a few restrictions such as a high computational cost. Therefore, we propose the novel DPLSA model that utilizes the marginal dynamic prior to learn the latent space of dyadic sequential data. We then propose the new framework for human motion modeling based on the DPLSA model and suggest learning and inference methods in this specific modeling context. The framework can be directly extended for multiple viewpoints by using the mixture model in the space of the latent variables and the image features. The utility of the the new framework is examined thorough a set of experiments of tracking 3D human figure motion from synthetic and real image sequences.

4.1 Motivation

Estimating 3D body pose from 2D monocular images is a fundamental problem for many applications ranging from surveillance to advanced human-machine interfaces. However, the shape variation of 2D images caused by changes in pose, camera setting, and viewpoints makes this a challenging problem. Computational approaches to pose estimation in these settings are often characterized by complex algorithms and a tradeoff between the estimation accuracy and computational efficiency. In this chapter we propose low-dimensional embedding method for 3D pose estimation that exhibits both high accuracy, tractable estimation, and invariance to viewing direction.

3D human pose estimation from monocular 2D images can be formulated as the task of

matching an image of the tracked subject to the most likely 3D pose. To learn such a mapping one needs to deal with a dyadic set of high dimensional objects - the poses, y and the image features, x. Because of the high dimensionality of the two spaces, learning a direct mapping $x \rightarrow y$ often results in complex models with poor generalization properties. One way to solve this problem is to map the two high dimensional vectors to a lower dimensional subspace z: $z \rightarrow x$ and $z \rightarrow y$ [17,51]. However, in these approaches, the correlation between the pose and the image feature is weakened by learning the two mappings independently and the temporal relationship is ignored during the embedding procedure.

4.2 Dynamic PLSA with GPLVM

The starting point of our framework design is the symmetric parameterization of Probabilistic Latent Semantic Analysis [27]. In this setting the co-occurrence data $x \in X$ and $y \in Y$ are associated via an unobserved latent variable $z \in Z$:

$$P(x, y) = \sum_{z \in Z} P(z)P(x|z)P(y|z). \tag{4.1}$$

With a conditional independence assumption, the joint probability over data can be easily computed by marginalizing over the latent variable. We extend the idea to the case in which the two sets of objects, X and Y are sequences and the latent variable z_t is only associated with the dyadic pair (x_t, y_t) at time t. And we solve the dual problem by marginalizing the parameters in the conditional probability models instead of marginaling of Z.

Consider the sequence of length T of M-dimensional vectors, $Y = [y_1 y_2 ... y_T]$, where y_t is a human pose (*e.g.* joint angles) at time t. The corresponding sequence $X = [x_1 x_2 ... x_T]$ represents the sequence of N-dimensional image features observed for the given poses. The key idea of our Dynamic Probabilistic Latent Semantic Analysis (DPLSA) model is that the correlation between the pose Y and the image feature X can be modeled using a latent-variable model where two mappings between the latent variable Z and X and between Z and Y are defined using a Gaussian Process latent variable model of [20]. In other words, Z can be regarded as the intrinsic subspace that X and Y jointly share. The graphical representation of DPLSA for human motion modeling is depicted in Figure 4.1.

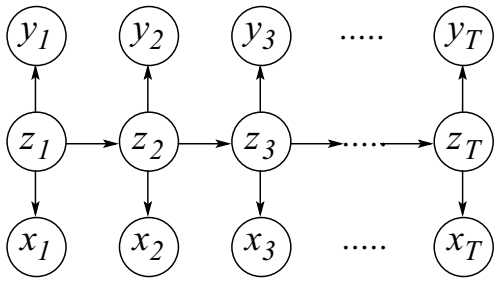


Figure 4.1: Graphical model of DPLSA.

We assume that sequence $Z \in \Re^{D\times T}$ of length T is generated by possibly nonlinear dynamics modeled as a known mapping ϕ parameterized by parameter γ_x [23, 32] such as

$$z_t = A_1\phi_{t-1}(z_{t-1}|\gamma_{z,t-1}) + A_2\phi_{t-2}(z_{t-2}|\gamma_{z,t-2}) + \ldots + w_t. \tag{4.2}$$

Then the first order nonlinear dynamics are characterized by the kernel matrix

$$K_{zz} = \phi(Z_\Delta|\gamma_z)\phi(Z_\Delta|\gamma_z)^T + \alpha^{-1}I. \tag{4.3}$$

The model can further be generalized to higher order dynamics.

The mapping from Z to Y is a generative model defined using a GPLVM [20]. We assume that the relationship between the latent variable and the pose is nonlinear with additive noise, v_t a zero-mean Gaussian noise with covariance $\beta_y^{-1}I$:

$$y_t = Cf(z_t|\gamma_y) + v_t. \tag{4.4}$$

C represents a linear mapping matrix and $f(\cdot)$ is a nonlinear mapping function with a hyperparameter γ_y. By choosing the simple prior of a unit covariance , zero mean Gaussian distribution on the element c_{ij} in C and z_t, marginalization of C results in a mapping:

$$P(Y|Z,\beta_y) \sim |K_{yz}|^{-M/2}\exp\left\{-\frac{1}{2}tr\{K_{yz}^{-1}YY^T\}\right\} \tag{4.5}$$

where

$$K_{yz}(Z,Z) = f(Z|\gamma_y)f(Z|\gamma_y)^T + \beta_y^{-1}I. \tag{4.6}$$

Similarly, the mapping from the latent variable Z into the image feature X can be defined by

$$x_t = Dg(z_t|\gamma_x) + u_t. \tag{4.7}$$

The marginal distribution of this mapping becomes

$$P(X|Z,\beta_x) \sim |K_{xz}|^{-N/2} \exp\left\{-\frac{1}{2} tr\{K_{xz}^{-1} X X^T\}\right\} \tag{4.8}$$

where

$$K_{xz}(Z,Z) = g(Z|\gamma_x) g(Z|\gamma_x)^T + \beta_x^{-1} I. \tag{4.9}$$

Notice that the kernel functions and parameters are different in the two mappings from the common latent variable sequence Z to X and to Y.

The joint distribution of all co-occurrence data and all intrinsic sequence in a $\mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$ space is finally modeled as

$$P(X,Y,Z|\theta_x,\theta_y,\theta_z) = P(Z|\theta_z) P(X|Z,\theta_x) P(Y|Z,\theta_y) \tag{4.10}$$

where $\theta_x \equiv \{\beta_x, \gamma_x\}$ and $\theta_y \equiv \{\beta_y, \gamma_y\}$ represent the sets of hyperparameters in the two mapping functions from Z to X and from Z to Y. θ_z represents a set of hyperparameters in the dynamic model (*e.g.* α for a linear model and α, γ_z for a nonlinear model).

4.2.1 Human Motion Modeling Using Dynamic PLSA

In human motion modeling, one's goal is to recover two important aspects of human motion from image features: (1) 3D posture of the human figure in each image and (2) an intrinsic representation of the motion. Given a sequence of image features X, the joint *conditional* model of the pose sequence Y and the corresponding embedded sequence Z can be expressed as

$$P(Y,Z|X,\theta_z,\theta_y,\theta_x) \propto P(Z|\theta_z) P(Y|Z,\theta_y) P(X|Z,\theta_x). \tag{4.11}$$

Notice that the two mapping processes $P(X|Z)$ and $P(Y|Z)$ have different noise models which can account for different factors (*e.g.* motion capture noise for the pose and camera noise for the image) that influence one but not the other process.

4.2.2 Learning

The human motion model is parameterized by a set of hyperparameters θ_x, θ_y and θ_z, and the choice of kernel functions, K_{yz} and K_{xz}. Given both the sequence of poses and the corresponding image features, the learning task is to infer the subspace sequence Z in the marginal

dynamics space and the hyperparameters. Using the Bayes rule and (Equation 4.11) the joint likelihood is in the form

$$P(X, Y, Z, \theta_x, \theta_y, \theta_z) = P(Z|\theta_z)P(Y|Z, \theta_y)P(X|Z, \theta_x)P(\theta_x)P(\theta_y)P(\theta_z). \quad (4.12)$$

To mitigate the overfitting problem, we utilize priors over the hyperparameters [23,32,35] such as $P(\theta_z) \propto \alpha^{-1}$ (or $\alpha^{-1}\gamma_z^{-1}$), $P(\theta_x) \propto \beta_x^{-1}\gamma_x^{-1}$ and $P(\theta_y) \propto \beta_y^{-1}\gamma_y^{-1}$.

The task of estimating the mode Z^* and the hyperparameters, $\{\theta_x^*, \theta_y^*, \theta_z^*\}$ can then be formulated as the ML/MAP estimation problem

$$\{Z^*, \theta_x^*, \theta_y^*, \theta_z^*\} = \arg \max_{Z, \theta_x, \theta_y, \theta_z} \{\log P(Z|\theta_z) + \log P(Y|Z, \theta_y) + \log P(X|Z, \theta_x)\} \quad (4.13)$$

which can be achieved using a generalized gradient optimization such as CG, SCG or BFG. The task's nonconvex objective can give rise to point-based estimates of the posterior $P(Z|X, Y)$ that can be obtained by starting the optimization process from different initial points.

4.2.3 Inference and Tracking

Having learned the DPLSA on training data X and Y, the motion model can be used effectively in inference and tracking. Because we have two conditionally independent GPs, estimating current pose (distribution) y_t and estimating current point z_t in the embedded space can be decoupled. Given image features x_t in frame t, the optimal point estimate z_t^* is the result of the following nonlinear optimization

$$z_t^* = \arg \max_{z_t} P(z_t|z_{t-1}, \theta_z)P(x_t|z_t, \theta_x). \quad (4.14)$$

Due to the GP nature of the dependencies, the second term assumes conditional Gaussian form, however its dependency on z_t is nonlinear [20] even with linear motion models in z. As a result, the tracking posterior $P(z_t|x_t, x_{t-1}, \ldots)$ may become highly multimodal. We utilize a particle-based tracker for our final pose estimation during tracking. However, because the search space is the low dimensional embedding space, only a small number of particles (< 20, empirical result) is sufficient for tracking allowing us to effectively avoid the computational problems associated with sampling in high dimensional spaces.

A sketch of this procedure using a particle filter based on the sequential importance sampling algorithm with N_P particles and weights $(w^{(i)}, i = 1, ..., N_P)$ is shown below.

Input : Image x_t, Human motion model *e.g.* (Equation 4.10) and prior point estimates $(w_{t-1}^{(i)}, z_{t-1}^{(i)}, y_{t-1}^{(i)})|X_{0..t-1}, i = 1, ..., N_P$.

Output: Current intrinsic state estimates $(w_t^{(i)}, z_t^{(i)})|X_{0..t}, i = 1, ..., N_P$

1) Draw the initial estimates $z_t^{(i)} \sim p(z_t|z_{t-1}^{(i)}, \theta_x)$.

2) Find optimal estimates $z_t^{(i)}$ using nonlinear optimization in (Equation 4.14).

3) Find point weights $w_t^{(i)} \sim P(z_t^{(i)}|z_{t-1}^{(i)}, \theta_z)P(x_t^{(i)}|z_t^{(i)}, \theta_z)$.

Algorithm 4: Particle filter in human motion tracking.

Finally, because the mapping from Z to Y is a GP function, we can easily compute the distribution of poses y_t for each particle $z_t^{(i)}$ by using the well known result from GP theory: $P(y_t|z_t^{(i)}) \sim \mathcal{N}(\mu^{(i)}, {\sigma^{(i)}}^2 I)$.

$$\mu^{(i)} = \mu_Y + Y^T K_{yz}(Z, Z)^{-1} K_{yz}(Z, z_t^{(i)}) \tag{4.15}$$

$${\sigma^{(i)}}^2 = K_{yz}(z_t^{(i)}, z_t^{(i)}) - K_{yz}(Z, z_t^{(i)})^T K_{yz}(Z, Z)^{-1} K_{yz}(Z, z_t^{(i)}) \tag{4.16}$$

where μ_Y is the mean of training set. The distribution of poses at time t is thus approximated by a Gaussian mixture model. The mode of this distribution can be selected as the final pose estimate.

4.3 Mixture Models for Unknown View

The image feature for a specific pose can vary according to a camera viewpoint and orientation of the person with respect to the imaging plane. In a dynamic PLSA framework, the view point factor R can be easily combined into the generative model $P(X|Z)$ that represents the image formation process.

$$P(X, Y, Z, R|\theta_x) = P(Z|\theta_z)P(Y|Z)P(X|Z, R)P(R). \tag{4.17}$$

While the continuous representation of R is possible, learning such a representation from a finite set of view samples may be infeasible in practice. As an alternative, we use a quantized

set of view points and suggest a mixture model,

$$P(X|Z,\beta_x,\gamma_x) = \sum_{r=1}^{S} P(X|Z,R=r,\beta_x^r,\gamma_x^r)P(R=r) \tag{4.18}$$

where S denotes the number of views. Note that all the kernel parameters (β_x^r, γ_x^r) can be potentially different for different r.

4.3.1 Learning

Collecting enough training data for a large set of view points can be a tedious task. Instead, by using realistic synthetic data generated by 3D rendering software which allows us to simulate a realistic humanoid model and render textured images from a desired point of view, one can build a large training set for multi-view human motion model. In this setting one can simultaneously use all views to jointly estimate a complete set of DPLSA parameters as well as the latent space Z. Given the pairs of the pose and the corresponding image features with viewpoint, learning the complete mixture models reduces to joint optimization of

$$P(Y,Z,X_1,X_2,...,X_S,R_1,...,R_S) = P(Z)P(Y|Z)\prod_s P(X_s|Z,R_s=s)P(R_s=s). \tag{4.19}$$

where S is the number of quantized views. The optimization of Z and model parameters is a straightforward generalization of the method described in Section 4.2.2.

4.3.2 Inference and Tracking

The presence of an unknown viewing direction during tracking necessitates its estimation in addition to that of the latent state z_t. This joint estimation of z_t and R can be accomplished by directly extending the particle tracking method of Section 4.2.3. This approach is reminiscent of [17] in that it maintains the multiple view-based manifolds representing the various mappings caused from different view points.

4.4 Experiments

4.4.1 Synthetic Data

In our first experiment we demonstrate the advantage of our DPLSA framework on the subspace selection problem. We also compare the predictive ability of DPLSA when estimating the sequence Y from the observation X.

We generate a set of synthetic sequences using the following model: intrinsic motion Z is generated with two periodic functions in $\Re^{T\times 2}$ space. The sequences are then mapped to a higher dimensional space of Y (in $\Re^{T\times 7}$) through a mapping which is a linear combination of nonlinear features ${z_1}^2, z_1, z_2, {z_2}^2$. X (in $\Re^{T\times 3}$) is finally generated by mapping Y into a non-linear lower observation space in a similar manner. Examples of the three sequences are depicted in Figure 4.2. This model is reminiscent of the generative process that may reasonably

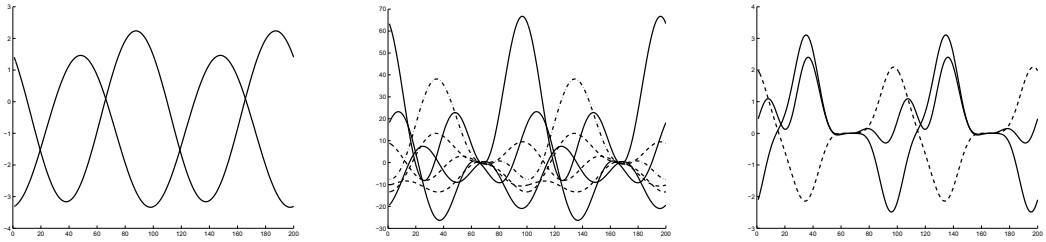


Figure 4.2: A example of synthetic sequences. Left: Z in the intrinsic subspace. Middle: Y generated from Z Right: X generated from Y

model the mapping from intrinsic human motion to image appearance/features.

We apply three different motion modeling approaches to model the relationship between the intrinsic motion Z, the 3D "pose" space Y and the "image feature" space X. The first method (Model 1) is the manifold mapping of [17] which learns the embedding space using LLE or Isomap based on the observation X and optimizes the mapping between Z and Y using Generalized RBF interpolation. The second approach (Model 2) is the human motion modeling using Marginal Nonlinear Dynamic System (MNDS) in Section 3.3, a model that attempts to closely approximate the data generation process. Model 3 is our proposed DPLSA approach described in Section 4.2.1. During the learning process, the initial embedding is estimated using probabilistic PCA. Initial kernel hyperparameter values were typically set to 1, except for the dynamic models where the variances were initially assigned values of 0.01.

We evaluate predictive accuracy of the models in inferring Y from X. We generate 25 sequences using the procedure described above. We generate the testing sequence X by adding white noise to the training sequence and infer Y from this X. Table 4.1 shows individual mean square error (MSE) rates of predicting all 7 dimensions in Y. All values are normalized with respect to the total variance in the true Y. The results demonstrate that the DPLSA model outperforms both the LLE-based model as well as the MNDS. We attribute the somewhat surprising result when compared to Model 2 to the sensitivity of this model to estimates of the initial parameters of $X \rightarrow Y$ mapping. This problem can be mitigated by careful manual selection of the initial parameters, a typically burdensome task. However, another crucial advantage of our DPLSA model over Model 2 is the computational cost in inferring Y. For instance, the mean number of iterations of scaled CG optimization is 72.09 for Model 3 and 431.82 for Model 2. This advantage will be further exemplified in the next set of experiments.

Table 4.1: MSE rates of predicting Y from X.

Model	$\bar{e}_{y_1}$	$\bar{e}_{y_2}$	$\bar{e}_{y_3}$	$\bar{e}_{y_4}$	$\bar{e}_{y_5}$	$\bar{e}_{y_6}$	$\bar{e}_{y_7}$	$\sum \bar{e}_{y_i}$
LLE+GRBF	0.06	0.14	0.08	0.06	0.27	0.16	0.04	0.81
MNDS	0.02	0.06	0.06	0.06	0.13	0.10	0.04	0.47
DPLSA	0.03	0.06	0.03	0.03	0.10	0.08	0.02	**0.34**

4.4.2 Synthetic Human Motion Data

Single view point

In a controlled study, we conducted experiments using a database of motion capture data for a 59 d.o.f. body model from the CMU Graphics Lab Motion Capture Database [1] and synthetic video sequences. We used five walking sequences from three different subjects and four running sequences from two different subjects. The models were trained and tested on different subjects to emphasize the robustness of the approach to changes in the motion style. Initial model values, prior to learning updates, were set in the manner described in Section 4.4.1. We exclude six joint angles that exhibit very small variances but are very noisy (*e.g.* clavicle and finger). The human figure images are rendered using Maya using the software generously provided by the authors of [48, 49]. Following this, we extract the silhouette images to compute the 10-dimensional Alt moment image features as in [22]. Also the 3D latent space is employed for

all motion tracking experiments. Figure 4.3 depicts the log precision of $P(Y|Z)$ and $P(X|Z)$ on the 2D projection of the latent space learned from one cycle of walking sequence. Note that the precisions around the embedded Z is different in the two spaces even though Z is commonly shared by both GPLVM models. To evaluate our DPLSA model in human motion

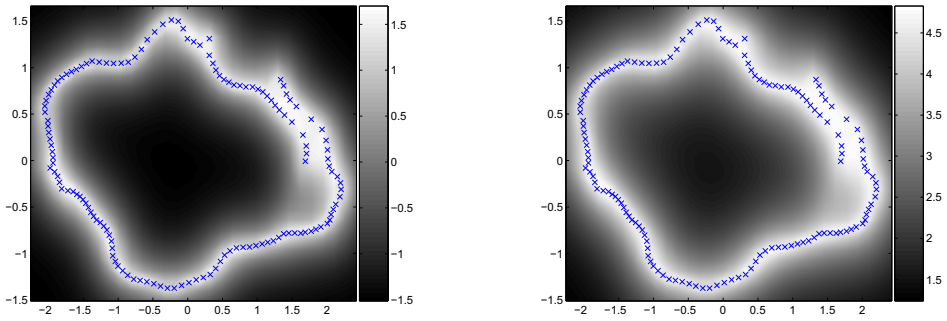


Figure 4.3: Latent spaces with the grayscale map of log precision. Left: $P(Y|Z)$. Right: $P(X|Z)$.

tracking, we again compare it to the MNDS model in [33] that utilizes the direct mapping between the poses and the image features. The models are learned from one specific motion sequence and tested on different sequences. Figure 4.4 shows the mean error in the 3D joint position estimation and the number of iterations in SCG per frame during tracking. We use an error metric similar to the one in [52]. The error between estimated pose $\widehat{Y}$ and the ground truth pose Y from motion capture is $E(Y, \widehat{Y}) = \sum_{j=1}^{J} \parallel \widehat{y}_j - y_j \parallel /J$ where y_j is the 3D location of a specific joint and J is the number of joints considered in the error metric. We choose 9 joints which have a wide motion range (*e.g.* throx and right & left wrist, humerus, femur and tibia). The height of human figure in this virtual space is 28 and the error unit can be computed relatively (*e.g.* when the height of man is $175cm$, the error unit is $175/28 \approx 6.25cm$). When the model was learned from one walking sequence and tested on four other sequences, the average error was 1.91 for MNDS tracking and 1.85 for DPLSA tracking. Results of full pose estimation for one running sequence are depicted in Figure 4.5. The models achieve similarly a good accuracy in pose estimation. A distinct difference between the two models, however, is exhibited in the computational complexity of the learning and inference stages as shown in Figure 4.4. The DPLSA model, on average, requires 1/5th of the iterations to achieve the same level of accuracy as the competing model. This can be explained by the presence of complex

direct interaction between high dimensional features and pose states in the competing model. In DPLSA such interactions are summarized via the low dimensional subspace. As a result, the DPLSA representation is potentially more suitable for real-time tracking.

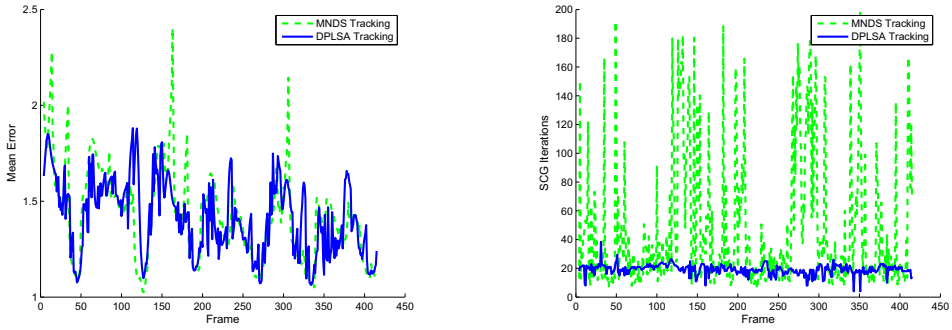


Figure 4.4: Tracking performance comparison. Left: pose estimation accuracy. Right: mean number of iterations of SCG.

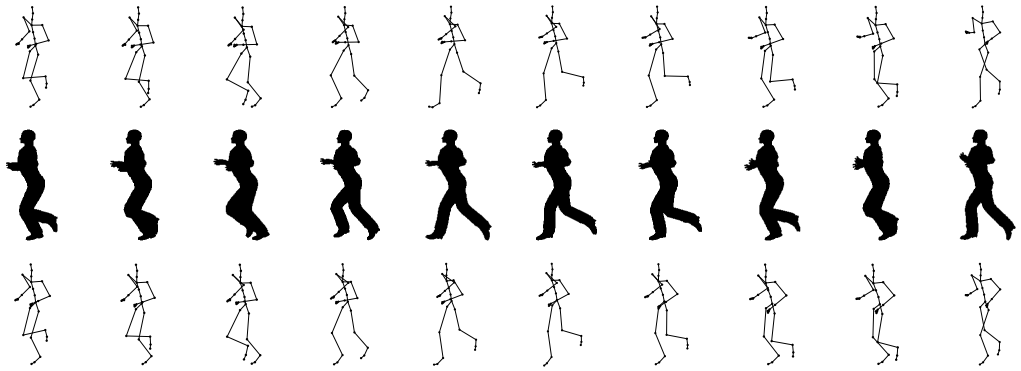


Figure 4.5: Input silhouettes and 3D reconstructions from a known viewpoint of $\frac{\pi}{2}$. First row: true poses. Second rows: silhouette images. Third row: estimated poses.

Comparison between MNDS and DPLSA

The distinctive difference between the human motion model using MNDS in Section 3.3 and DPLSA is the complexity in the learning and inference stages. The complexity in computing the objective function in the GPLVM is proportional to the dimension of a observation (pose) space. For the approximation of MNDS, a 44 dimensional pose is the observation for the two GP models, $P(\mathbf{Y}|\mathbf{Z})$ and $P(\mathbf{Y}|\mathbf{X})$. However, for DPLSA the pose is the observation for only one GP model $P(\mathbf{Y}|\mathbf{Z})$ and the observation of the other GP model $P(\mathbf{X}|\mathbf{Z})$ (Alt Moments) has only 10 dimensions. This makes learning of DPLSA less complex than learning of MNDS. In

addition, in inference only the latent variable (e.g. 3-dimension) is optimized in DPLSA while the optimization in MNDS deals with both the latent variable and the pose (3-dimensions + 44-dimension in our experiments). As a result, DPLSA requires significantly fewer iterations of the nonlinear optimization search, leading to a potentially more suitable algorithm for real-time tracking.

Multiple view points

We used a one person sequence to learn the mixture models of the 8 different views (view angles $= \frac{\pi}{4}i, i = 1, 2, \ldots, 8$ in clockwise direction, 0 for frontal view) and human motion model. We made testing sequences by picking different motion capture sequences and rendering the images from eight viewpoints. Figure 4.6 shows one example of 3D tracking results from two different viewpoints. In the experiment, the viewpoint of input images is unknown and inferred frame by frame during tracking with pose estimation. Although the pose estimation for some ambiguous silhouettes is erroneous, our system can track the pose until the end of sequence with the proper viewpoint estimation. Furthermore, the 3D reconstructions are matched well to the true poses. Notice that the last two rows depict poses viewed from $3\pi/4$, *i.e.* the subject walking in the direction of the top left corner.

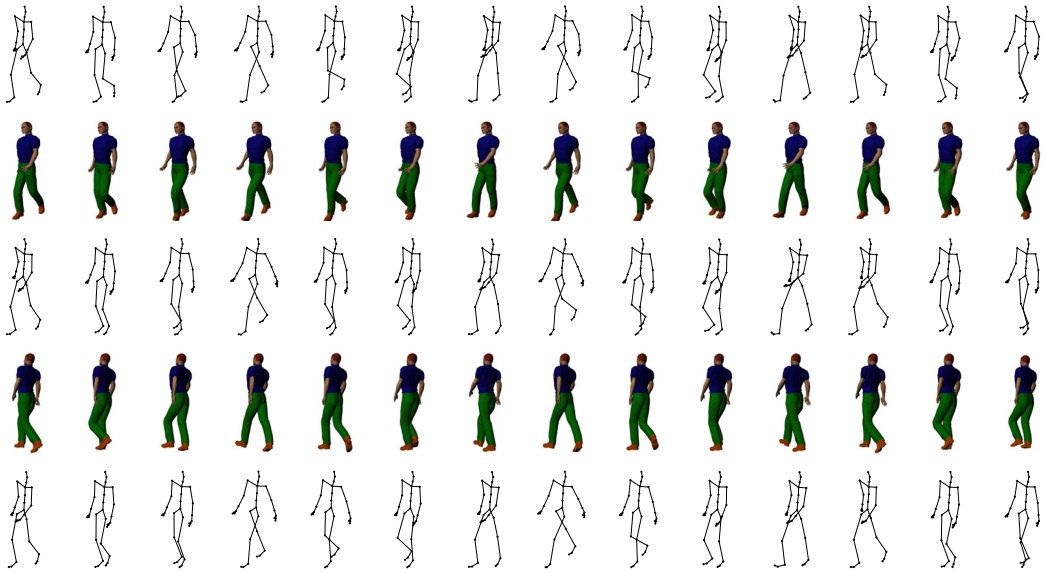


Figure 4.6: Input images with unknown view point and 3D reconstructions using DPLSA tracking. First row: true pose. Second and third rows: $\frac{\pi}{4}$ view angle. Fourth and fifth rows: $\frac{3\pi}{4}$ view angle.

4.4.3 Real Video Sequence

We applied our method to tracking of real monocular image sequences with a fixed viewpoint. We used the sideview sequences from CMU Mobo database [50]. The DPLSA model was trained on walking sequences from the Mocap data and tested on the motion sequences from the Mobo set. Figure 4.7 shows two example sequences of our tracking result. The lengths of the testing sequence are 300 and 340 frames. Although the frame rates and the walking style are different and there exists noise in the silhouette images, the reconstructed pose sequence depicts a plausible walking motion that agrees with the observed images.

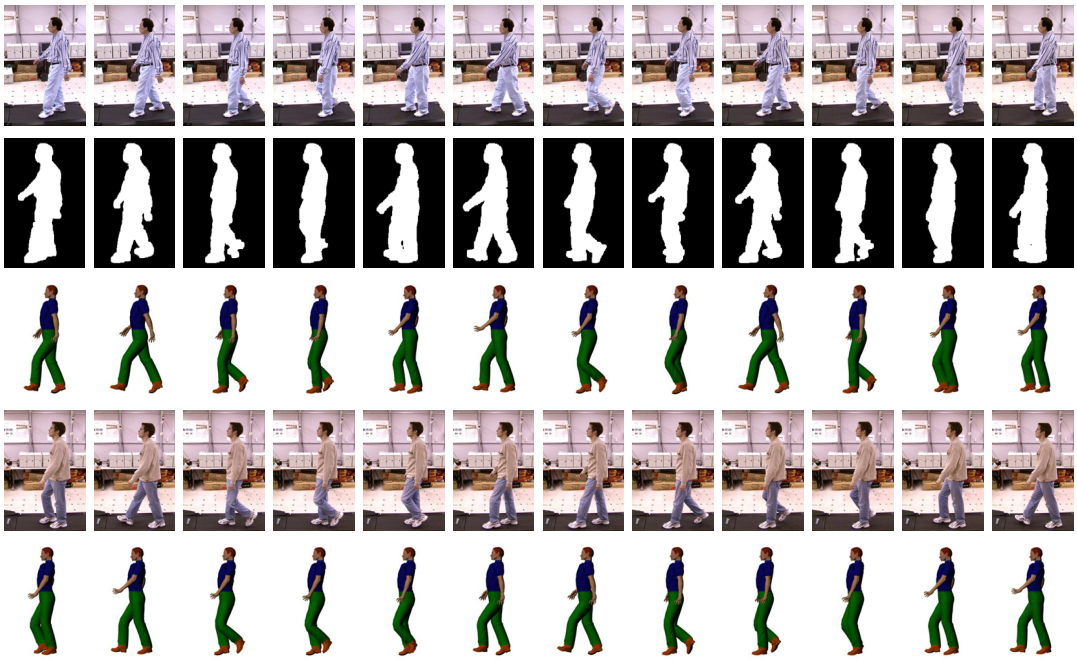


Figure 4.7: Tracking results. First row: input real walking images of subject 22. Second row: image silhouettes. Third row: images of the reconstructed 3D poses. Fourth row: input real walking images of subject 15. Fifth row: images of the reconstructed 3D poses.

4.5 Summary and Contribution

We have reformulated the shared latent space approach for learning models from sequential dyadic data. The reformulated generative statistical model is called the Dynamic Probabilistic Latent Semantic Analysis (DPLSA), which extends the successful PLSA formalism to a continuous state estimation problem of mapping sequences of human figure appearances in images

to estimates of the 3D figure pose. Our experimental results indicate that the DPLSA formalism can result in highly accurate trackers that exhibit a fractional computational cost of the traditional subspace tracking methods. Moreover, the method is easily amenable to extensions to unknown or multi-view camera tracking tasks.

Chapter 5

Gaussian Process Manifold Kernel Dimensionality Reduction

In this chapter we consider the discriminative modeling of the subspace embedding which preserves information relevant for a general nonlinear regression. This chapter is organized as follows. We first suggest an approximate approach to our full discriminative model and then introduce KDR and mKDR models. And we describe the reformulation of mKDR and show that the solution of mKDR becomes an eigen-decomposition task. We also relate this solution to the Gaussian Process regression models. Next, we propose a way to extend the model so that it is defined everywhere in the covariate space rather than only on the training data points. On a few examples, we illustrate the benefits of our approach, contrasted with the original mKDR optimization problem. The utility of the new method is further examined through a set of experiments with real data.

5.1 Approximation

Our ultimate goal of modeling subspace embedding in the discriminative way is making predictions in the full sequence level by utilizing the dynamic constraints as described in Figure 1.3(b). However, in this full discriminative model, it is not easy to model the dynamics in the unknown embedding space explicitly due to the dependency between x and z. That is, we should model the conditional probability $p(z_t|z_{t-1}, x_{t-1})$ to represent the dynamics in the embedding space, which may be difficult to learn. Therefore, we assume the i.i.d. condition on each instance and treat slices as depicted in Figure 5.1.

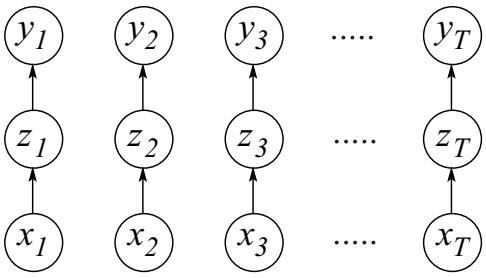


Figure 5.1: Graphical model of our approximation to the full discriminative model.

5.2 Motivation

The goal of dimensionality reduction in statistical learning problems is mainly *feature selection* in which we seek linear or nonlinear combinations of the original set of variables from the data. The setting for the learning mechanism for this purpose can be divided into two: unsupervised and supervised learning. In unsupervised learning, only a set of random vectors X is observed and we aim to learn the mapping from these observations to the low dimensional manifold as in Section 3.2. On the other hand, in supervised learning, desired responses or label Y corresponding to the input observation X is also available. The task of subspace embedding for regression is to find a low dimensional subspace embedding Z of the input X for regressing the output Y in the supervised learning framework. The dimensionality reduction for regressor can be beneficial for efficient regressor design with a reduced input dimension by filtering out noise in the original input X or discovering the essential information (*e.g.* Z) for predicting the output Y.

The goal of information-preserving manifold regression is to find a manifold that separates, in terms of probabilistic dependency, the target $Y \in \mathbb{R}^q$ from the covariate $X \in \mathbb{R}^p$. The basic idea of Sufficient Dimensionality Regression (SDR) is to reduce the dimension of X without losing information on the regression model, $P(Y|X)$. Specifically, the aim is to discover a low-dimensional projection satisfying the following conditional independence

$$Y \perp\!\!\!\perp X|\Phi_s X, \tag{5.1}$$

where Φ_s is the orthogonal projection of $\mathbb{R}^p$ onto the dimension-reduction subspace (DRS), $\mathcal{S}$. There exist several $\mathcal{S}$ for most regressions and one instead considers the subspace obtained by intersecting all the DRS's. If this subspace satisfies (Equation 5.1), it is called the central

subspace (central DRS) [53].

The previous approaches in the supervised setting have mainly focused on reduction to a linear manifold to avoid complexity [38, 42, 54]. Due to this limitation, the formulations in many approaches are based on the strong assumption on the linear manifold representation of covariate data. However, it is obvious that in many practical situations, this assumption can be too restrictive and render the approach unable to fully utilize the role of supervised data in manifold learning. To overcome this limitation, Nilsson et al. [55] recently proposed a method called manifold Kernel Dimension Reduction (mKDR) that finds nonlinear central spaces, which resort to a nonconvex gradient optimization in finding the space. We propose that instead of this iterative solution, there exists a closed-form solution to a related problem that results in a central subspace. Moreover, we show how this process relates to and extends the well-known Gaussian Process regression [56].

5.3 KDR and Manifold KDR

The idea of manifold KDR (mKDR) is to construct the dimension-reduction subspace in a regression which incorporates the intrinsic manifold structure of covariates. We briefly review this idea following [42, 43, 55].

5.3.1 KDR

The core idea of KDR is to characterize conditional independence in terms of cross-covariance operators on reproducing kernel Hilbert spaces (RKHS). Because the conditional independence assertion in (Equation 5.1) is equivalent to finding a low-dimensional projection Φ_s which makes $(I - \Phi_s)X$ and Y conditionally independent given $\Phi_s X$, the dimension reduction problem can be formulated as an optimization problem expressed in terms of covariance operators.

When $(\mathcal{H}_\mathcal{X}, k_\mathcal{X})$ and $(\mathcal{H}_\mathcal{Y}, k_\mathcal{Y})$ are RKHS's of functions on $\mathcal{X}$ and $\mathcal{Y}$, respectively, with the kernels $k_\mathcal{X}$ and $k_\mathcal{Y}$, the cross-covariance operator of (X, Y) is defined for all $f \in \mathcal{H}_\mathcal{X}$ and $g \in \mathcal{H}_\mathcal{Y}$ as follows:

$$\langle g, \Sigma_{YX} f \rangle_{\mathcal{H}_\mathcal{Y}} = E_{XY}\left[(f(X) - E_X[f(X)])(g(Y) - E_Y[g(Y)])\right] \tag{5.2}$$

The conditional covariance operator $\Sigma_{YY|X}$ is defined using covariance operators:

$$\Sigma_{YY|X} = \Sigma_{YY} - \Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XY}. \tag{5.3}$$

When $Z = F^{\mathrm{T}}X \in \mathcal{S}$ where F is a projection matrix such that $F^{\mathrm{T}}F = I$, it is proved in [43] that $\mathcal{S}$ is the central subspace if and only if $\Sigma_{YY|X} = \Sigma_{YY|Z}$. And for empirical samples, F, characterizing the central subspace, is the matrix that minimizes $Tr[\widehat{\Sigma}_{YY|Z}]$ where $\widehat{\Sigma}_{YY|Z}$ is the empirical version of the conditional covariance operator (Equation 5.2).

Let $\{x_i, y_i\}_{i=1}^N$ be a set of N data samples drawn from the joint distribution $P(X, Y)$ of targets Y and covariates X and $\{z_i = F^{\mathrm{T}}x_i\}$. Furthermore, let K_{yy} and K_{zz} denote the Gram matrices computed over $\{y_i\}$ and $\{z_i\}$. [43] shows that finding the central space is equivalent to solving the following optimization problem:

$$\begin{aligned} min_F \quad & Tr\left[K_{yy}^c \left(K_{zz}^c + N\epsilon I\right)^{-1}\right] \\ \text{s.t.} \quad & F^{\mathrm{T}}F = I \end{aligned} \tag{5.4}$$

where ϵ is a regularization coefficient and K_{yy}^c and K_{zz}^c are the centralized versions of K_{yy} and K_{zz}[1].

5.3.2 Manifold KDR

The mKDR approaches the central subspace estimation problem by combining the manifold preserving topological and geometrical properties of the data space into the KDR framework. In [55] the method of normalized Laplacian eigenmaps is first utilized for the unsupervised manifold learning. Given N data points $\{x_i \in \mathbb{R}^p\}_{i=1}^N$ and the weighted graph matrix W linking the data points, the normalized graph Laplacian matrix, $\mathcal{L}$ is defined as

$$\mathcal{L} = D^{-1/2}(D - W)D^{-1/2}. \tag{5.5}$$

where D is the diagonal matrix of the row sums of W, $D = W\mathbf{1}$. Let $\{v_m \in \mathbb{R}^N\}_{m=0}^{N-1}$ be the eigenvectors of L, ordered to their eigenvalues with v_0 having the smallest eigenvalue ($= 0$). Then the projection of the data to the lower dimensional manifold of dimension M is given by $[u_1, u_2, \ldots, u_N] = [v_1, v_2, \ldots, v_M]^{\mathrm{T}}$.

[1] The centralized kernel matrix of size $N \times N$ can be computed as $K^c = (I - \frac{1}{N}\mathbf{1}\mathbf{1}^{\mathrm{T}})K(I - \frac{1}{N}\mathbf{1}\mathbf{1}^{\mathrm{T}})$ where $\mathbf{1}$ is the vector with all elements equal to 1.

Given a M-dimensional nonlinear manifold U of covariates X, the central subspace is parameterized as a low dimensional linear transformation of this embedding, as in the KDR framework. This parameterization can be achieved by constructing the following explicit mapping:

$$K(\cdot, F^{\mathrm{T}}x_i) \approx \Phi^{\mathrm{T}}u_i \tag{5.6}$$

where K is a kernel function to map a point $F^{\mathrm{T}}x_i$ in the central subspace to the RKHS. With this smooth mapping Φ, we can approximate the Gram matrix on the RKHS as

$$\langle K(\cdot, F^{\mathrm{T}}x_i), K(\cdot, F^{\mathrm{T}}x_i)\rangle \approx u_i^{\mathrm{T}}\Phi\Phi^{\mathrm{T}}u_i. \tag{5.7}$$

The original optimization problem (Equation 5.4) is modified using the approximated Gram matrix:

$$\begin{aligned} min_{\Phi} \quad & Tr\left[K_{yy}^{c}\left(U^{\mathrm{T}}\Phi\Phi^{\mathrm{T}}U + N\epsilon I\right)^{-1}\right] \\ \text{s.t.} \quad & \Phi\Phi^{\mathrm{T}} \geq 0 \\ & Tr(\Phi\Phi^{\mathrm{T}}) = 1. \end{aligned} \tag{5.8}$$

Note that unit trace: $Tr(\Phi\Phi^{\mathrm{T}}) = 1$ is introduced as a convenience constraint that does not impact the centrality, but prevents unbounded Φ.

5.4 Reformulated Manifold KDR

Instead of solving the optimization problem (Equation 5.8) of manifold mKDR, we solve a related problem

$$\begin{aligned} min_{\Phi} \quad & J(\Phi) = Tr\left[K_{yy}^{c}K(\Phi)^{-1}\right] + M\log|K(\Phi)| \\ \text{s.t.} \quad & \Phi\Phi^{\mathrm{T}} = \Lambda \geq 0, \end{aligned} \tag{5.9}$$

where $K(\Phi) = U^{\mathrm{T}}\Phi\Phi^{\mathrm{T}}U + N\epsilon I$, M is the dimension of the manifold of X, Λ is a diagonal matrix and $|\cdot|$ denotes the determinant of the matrix. Our objective J is the objective function of original mKDR with a regularization term which plays the same role of $Tr(\Phi\Phi^{\mathrm{T}}) = 1$ in (Equation 5.8). The optimal solution, Φ^{*}, then satisfies the condition:

$$\nabla J(\Phi^{*}) = 0. \tag{5.10}$$

After taking the gradient with respect to Φ (and leaving out the superscript "*" for brevity), (Equation 5.10) leads to

$$UK(\Phi)^{-1}K_{yy}^{c}K(\Phi)^{-1}U^{\mathrm{T}}\Phi = MUK(\Phi)^{-1}U^{\mathrm{T}}\Phi. \tag{5.11}$$

Let $\Phi = ALB^{\mathrm{T}}$ be the SVD of Φ. Then, one can show (see Appendix B) that (Equation 5.11) can be written as

$$\frac{1}{M}UK_{yy}^{c}U^{\mathrm{T}}A = A(N\epsilon I + L^2) \tag{5.12}$$

Hence, an optimal solution for Φ can be found from the solution of the eigenvalue problem (Equation 5.12) as

$$\Phi^* = AL, \tag{5.13}$$

where A is the matrix of eigenvectors of $S = \frac{1}{M}UK_{yy}^{c}U^{\mathrm{T}}$ and the entries of L are $l_i = (\lambda_i - N\epsilon)^{1/2}$ where λ_i are the eigenvalues of S^2. Note that $\lambda_i - N\epsilon > 0$ guarantees positive definiteness of the constraint.

To find an optimal embedding of the training data one retains the columns of A that correspond to M largest values of the entries L. The embedded points in the central space are then

$$Z = LA^{\mathrm{T}}U. \tag{5.14}$$

The regressor $X \rightarrow Y$ can now be constructed by learning a regressor from the central space points Z to the targets Y. Learning such a regressor is typically easier than the direct $X \rightarrow Y$ regression, and is additionally less prone to adverse influence of noise or irrelevant features in the input. One reason for this is that Z can be viewed as those (filtered) features of the input that are most relevant for predicting the target.

5.4.1 Gaussian Process mKDR

The cost function we consider in (Equation 5.9) draws direct similarity to the Gaussian Process (GP) [56] and the Gaussian Process Latent Variable model (GPLVM) [57]. A GP objective typically assumes a linear kernel in the target y and a nonlinear kernel in its covariate. In our

[2] Φ^* is independent of B, an arbitrary rotation factor.

case, Z could be viewed as a (linear) covariate of the nonlinear GP $Z \to Y$. However, Z is additionally constrained to lie on the nonlinear subspace of X, $Z = \Phi^{\mathrm{T}} U$, $\Phi\Phi^{\mathrm{T}} = \Lambda$, unlike the typical Gaussian iid assumption of traditional GPs. Therefore, the problem of finding a central subspace-based manifold regression is equivalent to that of finding a Gaussian Process latent central subspace manifold. As we showed above, the stated problem has an optimal solution and reduces to eigensolution. This solution is reminiscent to the one of finding the latent covariate in a general GPLVM [57]. Thus we call our method Gaussian Process Manifold KDR (GPMKDR) in contrast to the original mKDR. However, note that our model is different from a shared latent model using GPLVM [35,36]. In contrast to our SDR approach, this shared latent variable extension considers the two generative mappings from the latent cause Z to X and Y, which relies on the joint iid Gaussian assumption of Z as well as the GP assumption in $P(Y|Z)$ and $P(X|Z)$. We specifically make no such assumption on $P(X|Z)$. This is the difference between direct discriminative models such as ours and the discriminative models induced by the generative ones. The algorithm for a general GPMKDR embedding is shown below.

Input : Covariate $X = \{x_i\}_{i=1}^N$ and response $Y = \{y_i\}_{i=1}^N$

Output: Linear mapping Φ^*

1) Compute the M-dimensional embedding U of Laplacian eigenmaps from X.

2) Compute the eigenvalues, $\{\lambda_i\}_{i=1}^M$ and the eigenvectors $\{e_i\}_{i=1}^M$ of the following matrix S

$$S = \frac{1}{M} U K_{yy}^c U^{\mathrm{T}}$$

where λ_i are sorted and e_i are ordered to their eigenvalues with e_1 having the largest eigenvalue λ_1.

3) Compute the diagonal matrix L where $l_{ii} = (\lambda_i - N\epsilon)^{1/2}, i = 1, \ldots, d$ and build the matrix $A = [e_1, \ldots, e_d]$ where d is the dimension of central subspace.

4) Compute $\Phi^* = AL$.

Algorithm 5: Gaussian Process mKDR Algorithm.

5.5 Extended Mapping for Arbitrary Covariates

Original derivation of mKDR [55] and our GPMKDR formulation in the previous section are based on embedding of a fixed set of training points. One way to generalize this to arbitrary points in the covariate space is via a functional mapping from $X \rightarrow U$.

To accomplish this, we consider Z to be a general mapping from the RKHS of X, $Z = \alpha K_{xx}$; each Z is a linear combination of the rows of Gram matrix K_{xx} of covariate X. Knowing an optimal Z we have

$$\alpha = Z K_{xx}^{-1} \ . \tag{5.15}$$

Consequently, any new test point x can be projected onto Z as

$$z(x) = \sum_{i=1}^{N} \alpha_i K_{xx}(x_i, x) \ . \tag{5.16}$$

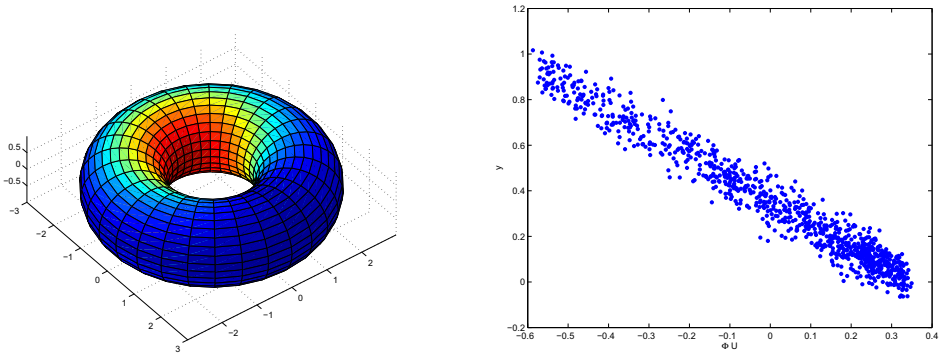
where α_i is the i-th column of the matrix α.

5.6 Experiments

We first demonstrate the effectiveness of our solution by carrying out a set of experiments suggested originally in [55]. We then consider three vision-related problems: illumination estimation, human pose estimation, and digit subspace visualization. Parameters of all methods presented in this section are selected for best performance. Unless noted otherwise, we employ Gaussian RBF kernels.

5.6.1 Comparison with mKDR

The first experiment focuses on analyzing data points that lie on a torus surface. The 3D coordinate of points on the torus is given by $x_1 = (2 + \cos\theta_r)\cos\theta_p$, $x_2 = (2 + \cos\theta_r)\sin\theta_p$, and $x_3 = \sin\theta_r$ where θ_r is the rotation angle and θ_p is the polar angle. This space is augmented to a 10D covariate space by adding 7-dimensional random noise vectors $\{x_i\}_{i=4}^{10}$, $x_i \sim \mathcal{N}(0, 0.1)$. The response is $y = \sigma[-1.7(\sqrt{(\theta_r - \pi)^2 + (\theta_p - \pi)^2} - 0.6\pi)]$ where $\sigma[\cdot]$ is the sigmoid function. The resulting torus whose surface is colored according to the target value y is shown in Figure 5.2(a). 950 data points are generated by randomly sampling θ_r and θ_p over $[0, 2\pi] \times [0, 2\pi]$. The target is further corrupted by noise. We compute the 1D central subspace



(a) (b)

Figure 5.2: 3D torus and central subspace of data randomly sampled on the torus.

from $M = 50$ bottom eigenvectors in the graph Laplacian. The resulting central subspace is shown in Figure 5.2(b).

Using the hyperparameter setting of [55], the iterative solution converges after 349 iterations. However, monotonic convergence to a global optima is not guaranteed. Figure 5.3(a) shows the objective function for the first 25 iterations oscillating around the limit value which is higher than the values attained in early iteration steps. This behavior can be remedied by reducing M, the dimension of the initial covariate embedding (*e.g.* LE), but can affect the quality of the induced central space. The original iterative solution is sensitive to the setting of parameters (*e.g.* tolerance) and the simple gradient descent method is typically inappropriate. It may be possible, but not always trivial, to find parameters that improve convergence properties of the algorithm or employ a more complex nonlinear search solution. Also note that the iterative algorithm requires inversion of an $N \times N$ matrix at each step, unlike our GPMKDR solution that follows from a single eigen-problem.

Figure 5.3(b) shows the Frobenius-distances between two Ωs ($\Omega = \Phi^{\mathsf{T}}\Phi$) computed from our closed-form solution and the iterative solution. The two matrices are first normalized using Forbenius norm and the distance is computed between these two normalized matrices at each iteration. It shows that our closed-form solution is the optimal solution of the iterative solution.

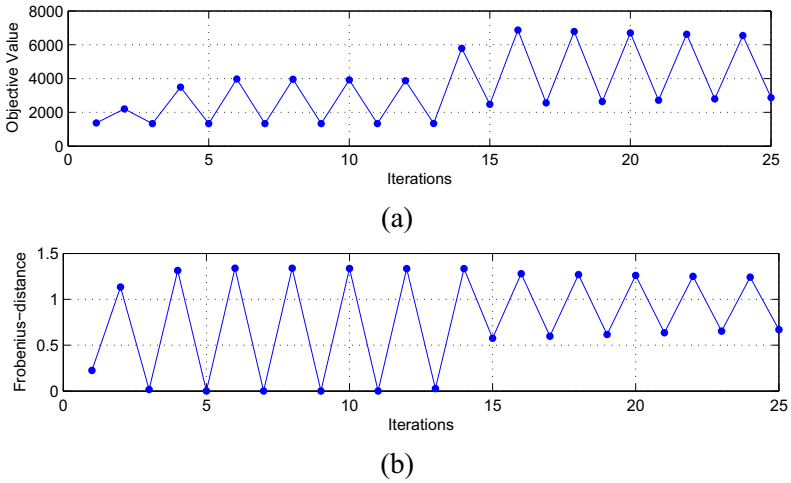


Figure 5.3: Comparison of two solutions. (a) Objective function values of the iterative solution during iterations, (b) Frobenius-distances between the closed-form solution and the iterative solutions.

We also test our method on the global temperature prediction task used in [55]. The responses of this regression problem are 3168 satellite measurements of temperatures in December 2004 extracted from the MSU (Microwave Sounding Units) channel of the TMT (Temperature Middle Troposphere) [58] while the covariates are the latitude and longitude. Figure 5.4(a) is the color coded world map displaying the trend of this channel. While there are only two covariates, their domain is not Euclidean. It is therefore nontrivial to learn a proper model for this regression problem.

In [55] it is shown that the relationship between the central space projection and the temperatures is largely linear. Figure 5.4(c) shows that our GPMKDR solution captures this relationship using $M = 100$ eigenvectors in LE. We next use a linear regression model and predict the temperatures from the projection. Figure 5.4(b) and Figure 5.4(e) display the predicted temperatures and the prediction error respectively. The prediction from the central subspace matches the temperature patterns well, even across local regions such as the Antarctic, with average error of 0.6750. The only exceptions are few areas of extreme climate (e.g. Himalayas).

Using the same $M = 100$, the original mKDR method displays oscillations and converges to a local minimum with an error of 3.0848. Figure 5.4(d) shows the scatter plot of the projection against the temperatures for this estimate of Φ. The target-central space relationship fails to be linear and results in large prediction errors as in Figure 5.4(f). Improved performance, as

demonstrated in [55], may be achieved by carefully adjusting the mKRD parameters, a step not necessary in our proposed solution.

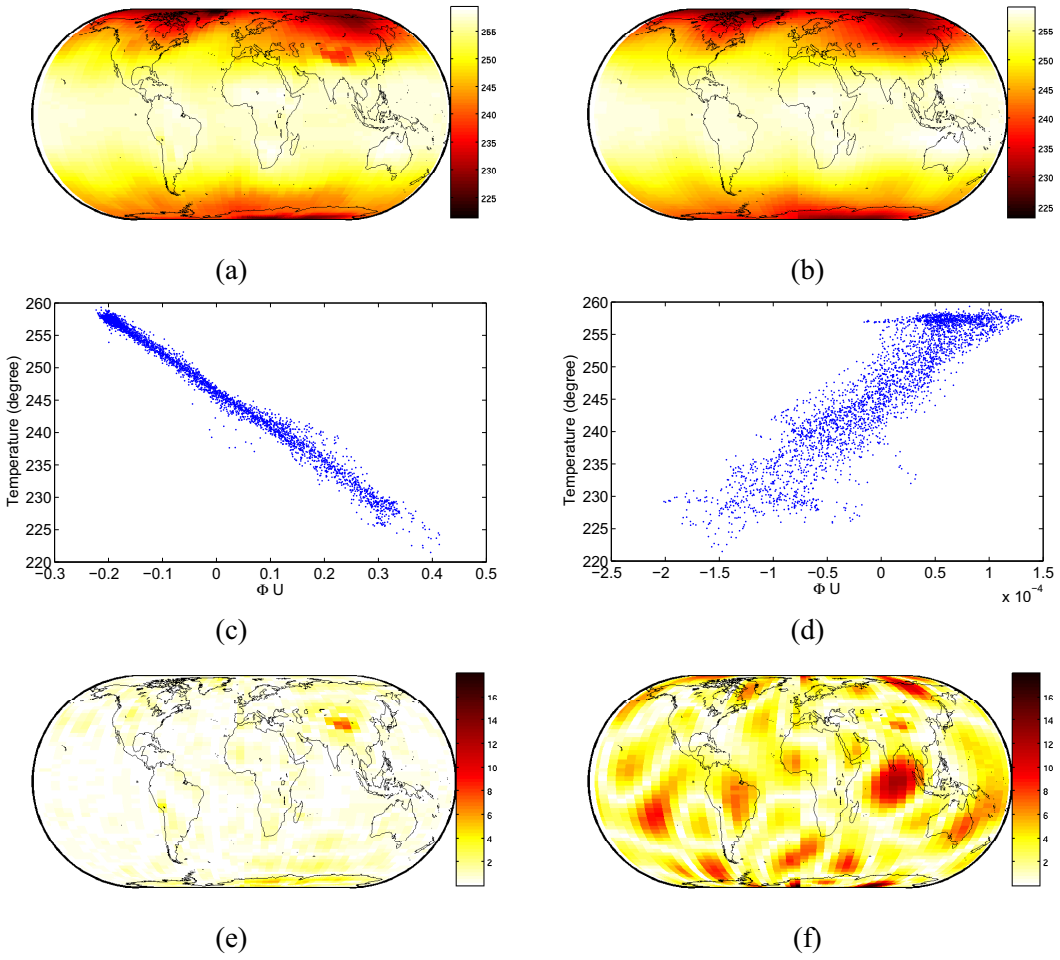


Figure 5.4: Comparison between solutions to global temperature regression analysis: (a) Map of the global temperature in Dec. 2004, (b) prediction with from closed-form solution, (c)(d) central subspaces, and (e)(f) prediction errors.

5.6.2 Illumination Estimation

We consider the task of estimating illumination direction from images. The illumination estimation becomes a regression problem where covariates are image pixel intensities and the response is the illuminant direction.

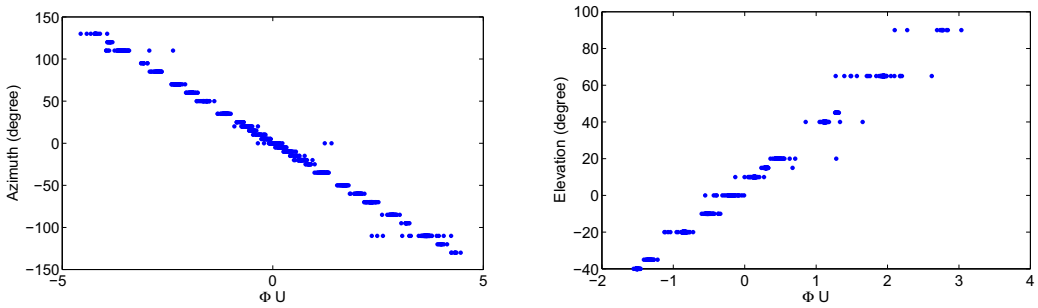
Our experiments are based on the extended Yale Face Database B with 2432 face images of 38 subjects under 64 illumination conditions [59, 60]. The illumination directions are defined by two angles with respect to the camera axis: azimuth and elevation. We resized the images

(a) (b)

Figure 5.5: Sample images from extended Yale Face Database B: (a) various azimuth angles and (b) various elevation angles.

to 96×84 pixels leading to 8064-dimensional covariates. Figure 5.5 shows several exemplar images of faces illuminated from different directions.

Our dataset consists of 20 randomly selected subjects. We remove images with very low contrast and randomly select 925 images for training and 300 images for testing. We then applied the GPMKDR algorithm to this data set.

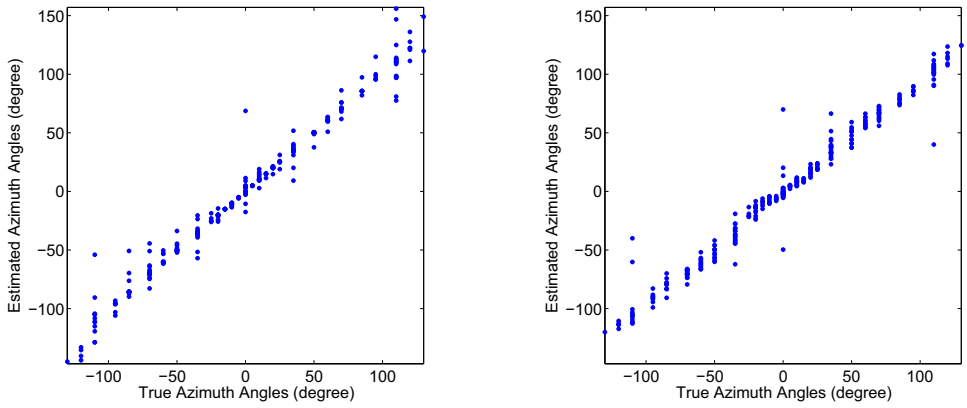


(a) (b)

Figure 5.6: First and second dimension of central subspace for Yale face database B; (a) Scatter plot of first dimension against azimuth angle; (b) Scatter plot of second dimension against elevation angle.

As seen in Figure 5.6 (a)(b), the first and second dimension of the central subspace have a largely linear relationship with the azimuth and elevation. As a result, we can build the linear regression model from the 2D central subspace to the illumination direction represented by the two angles. Using this linear regression model, we estimate the direction of illumination from input images in the training set. We compare the performance of GPMKDR to a Nadaraya-Watson kernel (NWK) regression where the covariates are the images and the responses are two illumination angles[3]. Figure 5.7 and Figure 5.8 shows the scatter plot of two estimated angles. The average error of GPMKDR+Linear regression in predicting azimuth is $5.15^\circ \pm 7.82$, similar to that of NWK, $5.77^\circ \pm 8.50$. However, the GPMKDR+Linear performs significantly better in

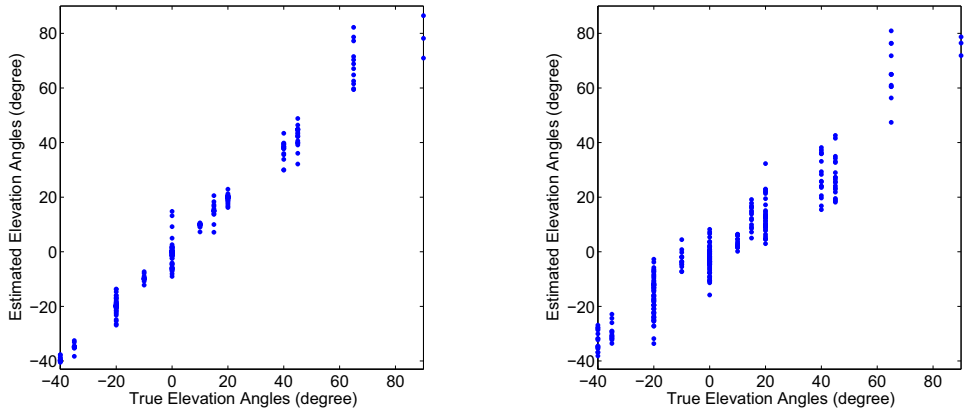
[3] Application of SIR to this and other similar domains is challenged due to the high dimensionality of covariates.



(a) (b)

Figure 5.7: Azimuth angle estimation results: (a) GPMKDR+Linear regression and (b) NWK regression.

estimating the elevation, shown in Figure 5.8. Its average error is $2.71° \pm 2.63$, whereas NWK regression results in $7.22° \pm 5.61$.



(a) (b)

Figure 5.8: Elevation angle estimation results: (a) GPMKDR+Linear regression and (b) NWK regression.

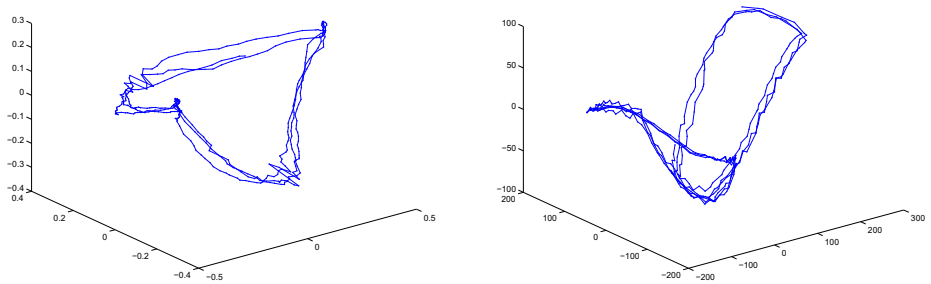
The Yale database is typically used for estimating the accuracy of face recognition under diverse illumination conditions. The ability to accurately estimate the illumination direction, as shown in Figure 5.7 and Figure 5.8, may be used in such a setting (*e.g.* , 3D face model-based approaches) to improve the recognition results without the need for explicit physical illumination models such as [60].

5.6.3 Human Motion Estimation

Our approach is also evaluated on the problem of human motion embedding and 3D figure pose estimation. In applications such as the human motion tracking, finding pose and motion embedding manifolds can be beneficial for improving the tracker's robustness or interpreting the motion properties. Typical embedding approaches, *c.f.* [17], consider manifolds of the pose or image space without regard to each other or the ultimate goal of estimating the pose from images. Nonlinear methods such as the Isomap or LLE are often employed for that purpose.

Our experiments use a database of motion capture data from CMU [1] and synthetic image sequences. The human figure images are rendered from the captured pose data using a 3D human model, and then binary silhouette images are extracted. In this regression problem, the covariates are the silhouette images of size 160×100 and the responses are the 3D pose joint angles represented by a 59 dimensional vector.

We first compare the quality of embedding produced by the central space to a 3D subspace of an image-based method, typically used in motion analysis. Figure 5.9 shows the manifolds computed by our GPMKDR and Isomap for one walking sequence (trial 03, subject 7). When the embedding is determined solely by the locality of covariates, one can observe crossing points induced by the intrinsic left/right side ambiguity of a silhouette image. Such points typically result in tracking failures unless additional information is used. GPMKDR, on the other hand, restructures the manifold with the pose information, removing the crossing points.



(a) (b)

Figure 5.9: Dimensionality Reductions for walking sequence, (a) GPMKDR and (b) Isomap.

We next apply the GPMKDR coupled with a Gaussian Process (GP) regression to model the mapping from the 3D central subspace to the pose space. Every third frame of 03 walking

sequence from subject 35 is used for training. We test the model on ten different sequences of the same subject. We compare the pose prediction performance of our GPMKDR+GP regressor to the GP regression model that maps directly from the silhouette images to the poses, without an intermediate subspace. The error is computed for 19 major joint angles with largest variance. The average test error is 0.8571^o for our GPMKDR+GP regression model, compared to 0.8991^o for the GP regression model. Despite the small average difference, certain poses remain predicted more accurately by our model, as illustrated in Figure 5.10. Moreover, the cost of learning a GP model with a 16000-dimensional input significantly exceeds that of a GP with a 3D input. Such high dimensionality may also lead to adverse numerical precision effects.

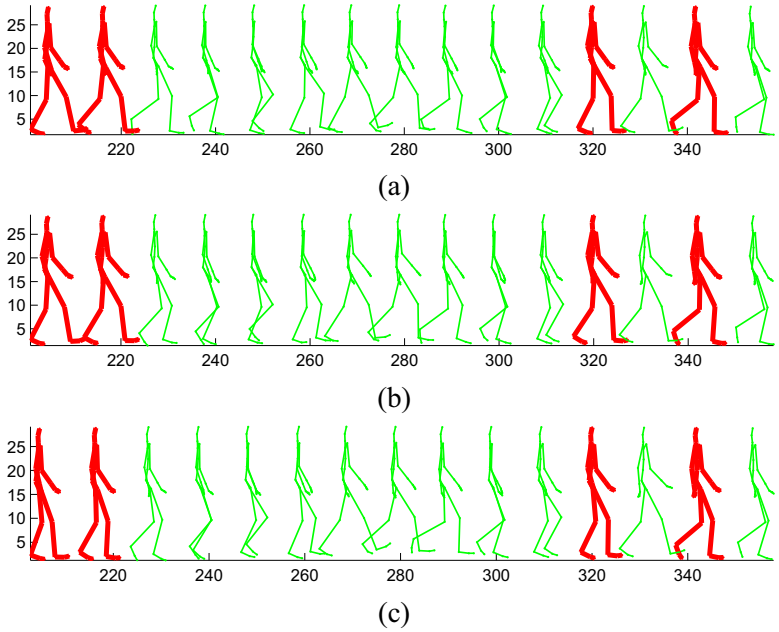


Figure 5.10: Comparison of two models. (a) True walking poses, (b) estimated poses using GPMKDR+GP regression model and (c) estimated pose using GP regression on image inputs.

5.6.4 Digit Visualization

We evaluate our GPMKDR method with three other dimension reduction methods on the problem of digit subspace visualization. The goal is to induce, from images of handwritten digits and possibly their labels, low dimensional subspaces that reflect a structure (*e.g.* digit identity) in this data. While predicting digit labels is not a regression task, central space methods can still be used in this setting for the purpose of subspace visualization.

In our experiments we contrasted GPMKDR to one supervised method, SIR, and two unsupervised methods, Laplacian Eigenmaps and Kernel PCA. All four models allow eigensolutions to the embedding problem. We report experiments on two handwritten digit databases: ORHD [61], MNIST [62] and and USPS [63]. Image data contains variations in appearance, style, and orientation and is used in the experiments without preprocessing (*e.g.* rotation correction).

For the first set, a 32×32 binary digit image is divided into nonoverlapping 4×4 blocks and the number of "1" pixels are counted in each block. Thus, the input covariate is represented lexicographically as a 16 dimensional vector. We randomly sampled 300 images for each digit. The 2D projections obtained by the four methods are illustrated in Figure 5.11.

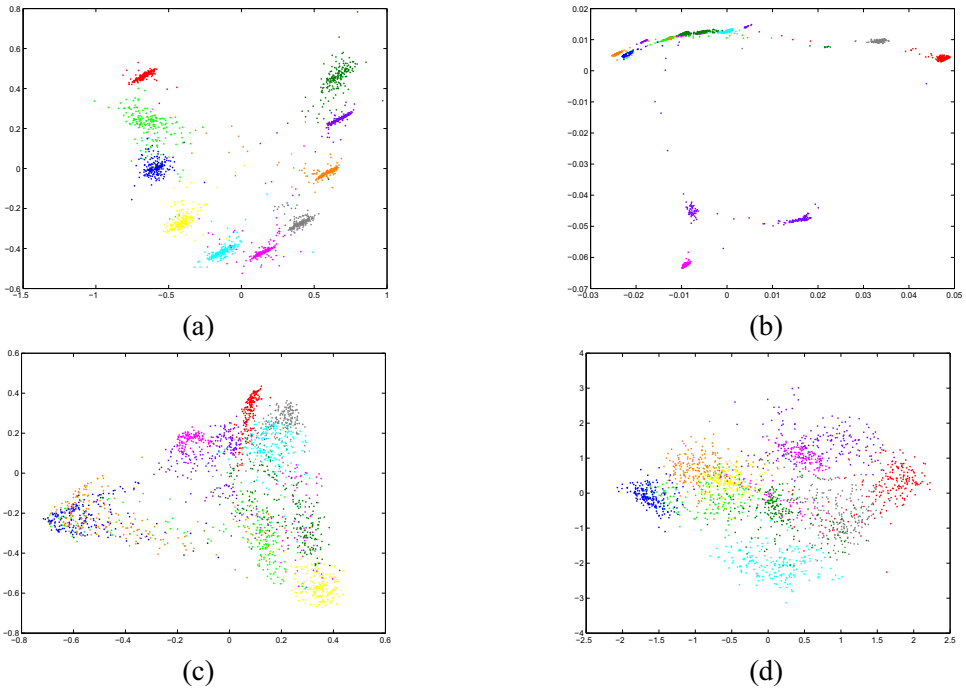


Figure 5.11: Embedding space for ORHD: (a) GPMKDR, (b) LE, (c) KPCA, and (d) SIR.

The MNIST database consists of 70,000 sample images of size 28×28 and the image has a 8-bit grayscale pixel value. The input vector of length 784 is projected to the low-dimensional spaces by four different methods. 400 random samples are selected for each digit. Figure 5.12 shows the projections to a 3D subspace of the MNIST dataset.

The USPS database contains 1,100 examples of 8-bit grayscale images for each digit. The size of images is 16×16. We randomly sample 400 images per digit. The projection results to

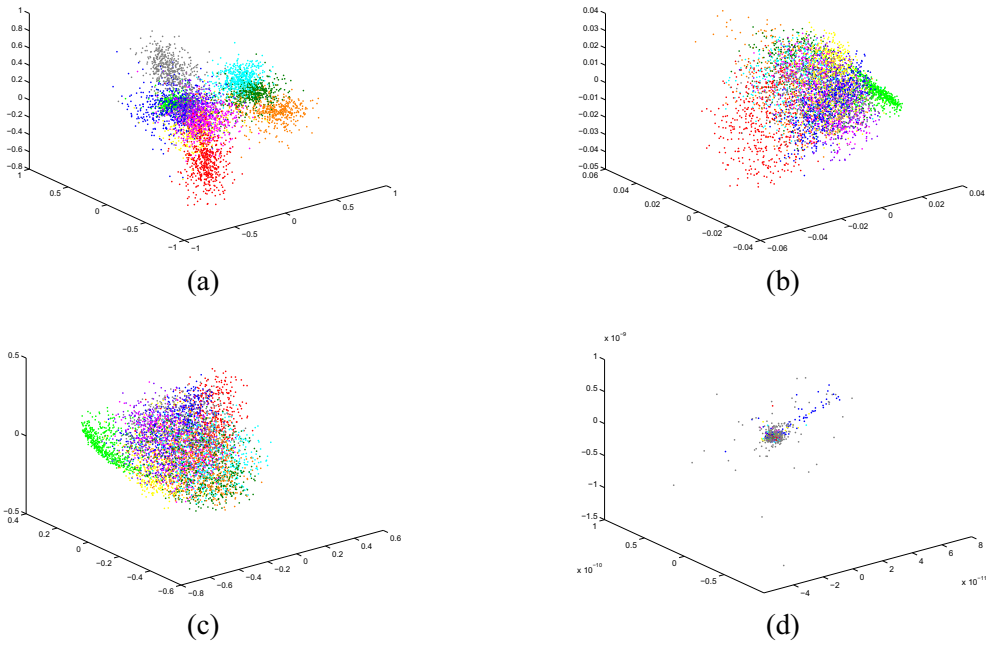
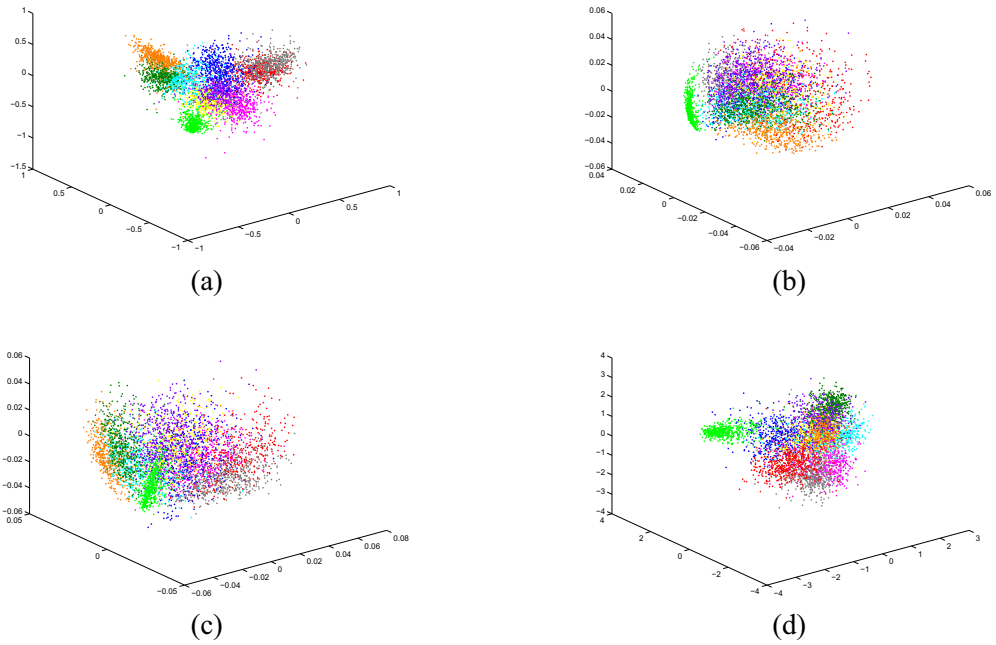


Figure 5.12: Embedding space for MNIST: (a) GPMKDR, (b) NPE, (c) KPCA, and (d) SIR.

the 3D embedding spaces are depicted in Figure 5.13. Observe that the classes are well separated in the central subspace of GPMKDR in all databases. The central subspace of GPMKDR is clearly distinguishable from the projections of LE even though the former is created from an LE manifold.

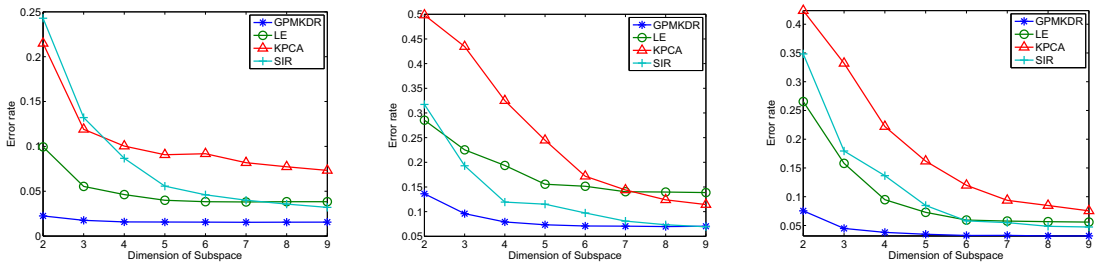
To quantify the quality of the low-dimensional embedding, we estimate the kNN classification error in the projected subspaces. We report results from five different random samplings of the data and report the average error rates. We also display energy compactness of the embeddings (normalized sum of the retained eigenvalues). Figure 5.14 and Figure 5.15 show the scores as a function of the subspace dimension. Because the maximum dimension for SIR is $\#\,\text{classes} - 1$, we investigate only 2 to 9 dimensions. GPMKDR-based central space display anticipated grouping of data points according to the digit labels. The advantage of GPMKDR over competing methods is especially significant for lower dimensional embeddings. Most of GPMKDR energy is concentrated in few eigenvalues that rapidly lead to good separation in the subspace, as measured by the error rates. SIR produces inferior structure. We have observed significant variation in its performance as a function of the number of slices, but none leads to performance better than GPMKDR. Unsupervised LE and KPCA recovered, as expected, less



(a) (b) (c) (d)

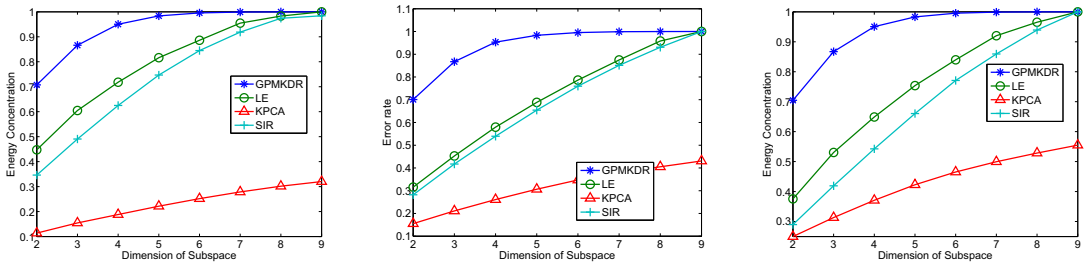
Figure 5.13: Embedding space for USPS: (a) GPMKDR, (b) LE, (c) KPCA, and (d) SIR.

structure than GPMKDR. KPCA showed consistently unsatisfactory performance when fewer than 9 dimension is used. Surprisingly, LE appeared to consistently outperform SIR on both sets even though 3D visualization results for SIR seem structurally more appealing. Poor SIR performance may be attributed to, among other factors, the underlying covariate distribution assumptions.



(a) (b) (c)

Figure 5.14: Error rate: (a) ORHD, (b) MNIST, and (c) USPS.



(a) (b) (c)

Figure 5.15: Energy concentration: (a) ORHD, (b) MNIST, and (c) USPS.

5.7 Summary and Contribution

We have proposed a novel dimension reduction approach called Gaussian Process Manifold Kernel Dimensional Reduction, induced by reformulating the manifold kernel dimensional reduction (mKDR) in the Gaussian Process (GP) framework. In this framework, a closed-form solution for mKDR is given by the maximum eigenvalue-eigenvector solution to a kernelized problem. We also suggest a way to generalize the approach to arbitrary points in the covariate space. The new algorithm has been applied to several synthetic and real-world datasets. The closed-form solution eliminates the need for parameter settings of iterative mKDR and can significantly reduce its complexity. Our preliminary results on vision applications such as the illumination and 3D human pose estimation indicate that our approach can result in a regressors with high accuracy and reduced computational requirements.

Chapter 6

Application in Financial Data

6.1 Preliminaries

In this chapter we apply our GPMKDR model to financial data, especially to solve the problem of implied volatility surface (IVS) prediction in the option market. An option in the stock market is a contract which conveys the right to buy or to sell a particular stock at a certain price (*i.e*. strike) at some time on or before a certain day (*i.e*. expiration). There are two kinds of options: call and put options. Buying a call option gives the buyer to right to buy a specific quantity of a stock at a certain strike price at some time or before expiration and buying a put option gives the right to sell. The theoretical value of an option can be evaluated according to several models (*e.g*. Black-Scholes model [64] and binomial options pricing model [65]) which utilize the quantitative techniques based on the concept of risk neutral pricing and stochastic calculus. Among the various factors affect the option price, volatility represents the measure of uncertainty or risk about the size of changes in an asset's value. A higher volatility means that the price of asset can change dramatically over a short time period in either direction while a lower volatility means that the price fluctuates at a steady pace. The *Implied volatilities* of a option is the volatility directly derived from the market price of the option based on an option pricing model. The level or behavior of implied volatility represents the state of the option market and is used as a market risk indicator [66].

From a specific pricing model, one can derive the IVS by applying the model to a set of options across different strikes and expirations. This volatility surface is utilized as a key financial variable for trading, hedging, and the risk management of various equity portfolios in the financial market. Because the IVS is a stochastic variable which indicates the current market move and risk [67], it is a key financial variable for market makers at option trading desks who keep monitoring and updating the volatility surface they trade on. Risk managers

also estimate the impact of large market movements by analyzing implied volatility shifts or other deformations of the IVS. Therefore, it is a very important task to model the IVS dynamics properly and predict the future volatility surface from the current market condition.

Next, we briefly introduce the mathematical concept of IVS in option pricing and the difficulties in IVS modeling. We then specify our problem in IVS prediction with high frequency tick data.

6.1.1 Implied Volatility Surface

The factors that determine the value of an option include the current stock price, the strike, time to expiration, interest rates, volatility, and cash dividends paid. For simplicity, assume a option for a non-dividend paying stock with the current stock price, S_t with expiration date T and strike price K. When the fixed interest rate is given, the price of option is a function which depends on the option pricing model, M:

$$P_M : (S_t, K, T, \sigma) \rightarrow P_M(S_t, K, T, \sigma) \in \mathbb{R} \geq 0 \tag{6.1}$$

where σ is the volatility, which is a statistical measure that shows how much the return of stock underlying the asset will fluctuate between now and the expiration. Let us now consider the option price is known from a market, which is denoted by $P_{market}(K, T)$. Then the implied volatility $\sigma_I(K, T)$ of the option is defined as the value of the volatility parameter which equates the market price with the price determined by the model M:

$$P_M(S_t, K, T, \sigma_I(K, T)) = P_{market}(K, T). \tag{6.2}$$

With a fixed stock price at the current time, we then have the unique implied volatility that represents the characteristics of option depending on the expiration T and the strike K. And the collection of these values results in a parametric surface which is called the *implied volatility surface*:

$$\sigma_I : (K, T) \rightarrow \sigma_I(K, T). \tag{6.3}$$

Figure 6.1(a) depicts the IVS graph. Note that the surface is built by using the IVS collected in 15 minutes and these real data points are marked using a small red dot (black in gray print). The way to build this IV surface will be fully described in Section 6.3 again.

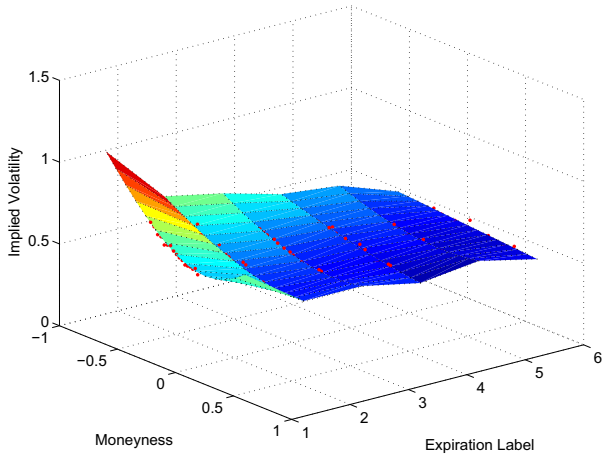
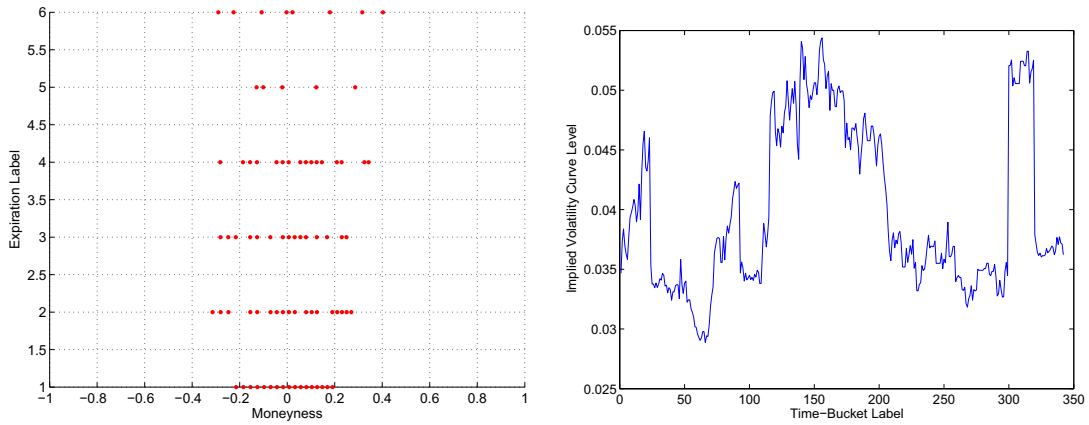


Figure 6.1: 3D implied volatility surface example (based on the option trade between 9:36AM and 9:41AM on Sep. 30, 2008).

6.1.2 Difficulties in IVS Prediction

Modeling the IVS with high frequency tick data is a challenging task. First, as observed in Figure 6.2(a), the instances of implied volatilities can be very sparse and can be missing on the sub-regions over the moneyness axis. For example, the implied volatilities for the options belonging to the first closest expiration are observed only near-the-money (where the strike is close to the current stock price, around 0 in the graph). However, in order to use the surface as a variable in the application, we need the observations on the every desired grid. The popular methods to this problem are to utilize a non-parametric approximation such as Nadaraya-Watson estimator [68,69] or to model the IVS in a parametric form [70–72]. In our experiment, we utilize the parametric fitting for an individual curve that can represent the detail movements of IVS over the different expirations.

The movement of IVS is affected by various factors such as underlying stock price movements for bid/ask/trade, the trading volume ratio between the different segments over moneyness axis, and the volume change in stock trade. In addition, the IVS itself has its own view represented from the its price inputs (bid/ask/high/low/open/close price, volume-weighted-average-price(VWAP)). Therefore, it is not an easy task to consider all these factors in predicting the future IVS. As a result, none of the previous approaches attempt to take advantage of these factors.



(a) (b)

Figure 6.2: Implied Volatility Surface Analysis: (a) IVS as seen from top (b) volatility surface level evolution using the implied volatility curve of second closest expiration in the days between Sep. 29 and Oct. 3.

6.1.3 Previous Approaches

The motivation to the study of IVS as a foundation for a market-based approach is well described in [68]. Before this work, the main stream of implied volatility time series analysis focuses on curves (*e.g.* at the money implied volatility smiles), not surface. The smoothed volatility surface is obtained using a non-parametric Nadaraya-Watson estimator. Then they apply the Karhunen-Loeve decomposition (a generalization of PCA to higher dimensional random fields) on the surface generated from the daily variations of the logarithm of implied volatility. The dynamics of individual eigenvalue sequences are modeled separately and the analysis focuses on the correlation between the principal component and the time. Fengler *et al*. [69] extend the volatility curve model to the surface by utilizing common principal components analysis (CPC). They exploit a small number of factors common to several maturity groups to represent a group structure given by the option surface data. The various parametric modeling of surface is also studied for IVS. Goncalves *et al*. [70] fit the volatility surface into the following parametric model by ordinary least squares (OLS):

$$\ln \sigma_i = \beta_0 + \beta_1 M_i + \beta_1 M_i^2 + \beta_3 \tau_i + \beta_4 (M_i \times \tau_i) + \varepsilon_i \tag{6.4}$$

where M_i is the time-adjusted measure of moneyness, τ_i is the maturity and ε_i is the random error term for $i = 1, \ldots, N$ (N is the number of options available across the surface). A vector AR model has been also utilized to model the dynamics of the coefficient vector $\beta = \{\beta_j\}_{j=0}^{4}$.

However, this type of parametric fitting seems insufficient to represent the subtle changes in the surface. Fengler *et al*. [71] considered the sparsity of IVS data with respect to the time-to-maturity axis. For the previous studies using the nonparametric approximations (*e.g*. Nadaraya-Watson estimator), the bandwidth in the time-to-maturity dimension must be very large to cover the large gaps between the expirations. Therefore it causes an estimation bias with severe irregular observations across the moneyness. A semiparametric factor model (SFM) as an solution to this problem is also proposed and the dynamic structure of the IVS is represented by the movement of basis functions in a finite dimensional function space. When $Y_{i,j}$ is the log-implied volatility, $X_{i,j}$ is the the vector of moneyness and time-to-maturity, i is an index of time, and j is the index of the strikes. In this approach, the IVS is fitted to the following model:

$$Y_{i,j} \approx m_0(X_{i,j}) + \sum_{l=1}^{L} \beta_{i,l} m_l(X_{i,j}) \tag{6.5}$$

where m_i are smooth basis functions and β_i are weights depending on time i. After fitting, the dynamics of β_i is analyzed by applying the classical vector autoregressive model. Because of the iterative process in fitting, this method can be computational demanding and the prediction performance is also limited. To overcome this shortfall, Audrino *et al*. [72] propose the semi-parametric factor model utilizing an additive expansion of simple fitted regression trees estimated by boosting techniques. Beginning from a initial model, a tree-boosting algorithm sequentially minimizes the residuals of observed and estimated implied volatilities. A cross-validation strategy is utilized to find an optimal stopping value for the tree boosting. They propose to use the split variables in a tree as factors. However, instead of using these factors alone, the authors add the factors from Heston-Nandi-GARCH model, which are very high dimensional factors. In the experiments of measuring the prediction accuracy for out-of-sample data, the method is compared to other standard approaches such as sticky-moneyness model and PCA with ARMA-GARCH model.

All the previous works utilized the stock index (*e.g*. S&P500 and DAX) based on daily historical data. There are no works for the intraday prediction of the IVS of individual stock option and no previous study considers both stock and option as the input factors. However, our purpose is to predict the volatility surface from all available high-frequency tick data recorded for every trade and quote in both stock and option market. In particular, we do not limit the

option selection to the index option.

6.2 Problem Formulation

We formulate the problem of predicting future IVS from the current market data as a regression problem. In this regression problem, the input is the observed data representing the current market condition (*e.g*. underlying stock price change, bid/ask volume ratio, the current IVS from bid/ask prices for the options *etc*.) and the output is the future IVS at the certain time. Figure 6.2(b) shows that the level of volatility surface keeps changing according to the various market conditions and time. The observation that shifts in the level of implied volatility are highly correlated across strikes and maturities suggests that their joint dynamics is driven by a small number of factors. It also indicates the usability of intrinsically low dimensional features which are directly linked to the movements of the IVS.

6.3 Data

Although all the previous approaches to model the implied volatilities surface are based on daily data, our dataset is obtained from the intraday high frequency tick data. The implied volatility surface is computed from call and put option prices with different strikes and expirations on a few stock symbols. To build the dataset, we first collect the time-bucket data from the raw tick data file. The time-bucket data is a formatted collection of tick data which represents the tendency of orders and trades made in a certain time frame window (*e.g*. 5 or 15 minutes). For example, the time-bucket data includes the open/close/high/low prices, the total sizes (volume), and the volume weighted average prices (VWAP) of all traded or quoted stocks and options in a specified time segment. After collecting the time-bucket data, we can easily compute the implied volatilities from all available option prices. To construct the IVS, only out-of-the-money options are used: put options are used for moneyness $m < 0$, call options for moneyness $m > 0$. As for m, we use the log *moneyness* $m = log(K/F)$ (where $F = S_t \exp\{r(T-t)\}$ is the forward price of the stock at the expiration time T when the stock price is S_t at the current time T and the risk-free interest rate is r) instead of the strike, which describes the intrinsic value of an option with regarding to its current stock price. To get the smoothed volatility

surface on the fixed grid, we utilized the following quadratic fitting for the implied volatility curve for an individual expiration:

$$\sigma_I(m) = \gamma_1(1 + \gamma_2 m + \gamma_3 m^2) \tag{6.6}$$

where γ_1 is the level of IV curve and γ_2 and γ_3 represent the skewness and the kurtosis (or smileness) of the curve respectively. From this curve fitting, we estimate implied volatilities over the fixed grid in the moneyness and expiration space.

The variables used in the experiment are defined in Table 6.1. The target of the regression is the parameters $\{\gamma_1^i, \gamma_2^i, \gamma_3^i\}_{i=1}^n$ (n is the number of expirations) of IVS at time $t+1$.

Table 6.1: Variables included in the input.

Variable Name	dim.	Description
$stockQuotePriceRatio_t$	1	Fraction that expresses the ratio of ask VWAP to bid VWAP of underlying stock trading at time t
$stockQuoteVolumeRatio_t$	1	Fraction that expresses the ratio of ask volume to bid volume of underlying stock trading at time t
$stockPriceChange_t$	1	Fraction that expresses the ratio of stock trading price at time t to price at time $t-1$
$QuadBidParams_{t-1,t}$	$d \times n \times 2$	Parameters for IVS from the quoted option bid price at time $t-1$ and t (d is the number of parameters and n is the number of expirations in the option data)
$QuadAskParams_{t-1,t}$	$d \times n \times 2$	Parameters for IVS from the quoted option ask price at time $t-1$ and t
$volumeBidRatio_t$	$3 \times n$	Ratio of three volumes of quoted bid options for out-of-money put, in-the-money, and out-of-money call at time t
$volumeAskRatio_t$	$3 \times n$	Ratio of three volumes of quoted ask options for out-of-money put, in-the-money, and out-of-money call at time t

6.4 Results

To investigate the usefulness of our GPMKDR framework in this IVS prediction problem, we compare the predictive accuracy of various regression models. We apply four regression models: Linear, Nadaraya-Watson, GP, and our GPMKDR. The model parameters which should be manually set are tuned through cross-validation.

The accuracy of the prediction is measured using mean error after we normalize the scale of the output dimensions. In the real market, the key parameter which the most market participants is more interested in are the level (γ_1) and skewness (γ_2) of IVS, especially for short term options which are far more actively traded than long term options. If the level or skewness of the volatility surface are accurately predicted, it is quite straightforward to develop profitable trading strategies from them. However, as for the kurtosis, unless the prediction is highly accurate, it is relatively difficult to trade it. Therefore, it is practically reasonable for us to be more interested in these two parameters. We report the prediction accuracy on the level, γ_1, and the skewness, γ_2 of the implied volatility curves of two closest expirations. Table 6.2 shows the mean errors in predicting these outputs by four regression methods. The models are trained using the option data (240 time-bucket samples) of Google (symbol: GOOG) collected from September and October in 2008. Then we test the models on the option data of the same stock collected in 4 days (100 time-bucket samples) of November, 2008. Using the same settings, we test the methods on the options for another stock and sector Exchange-Traded Fund (ETF): Apple (symbol: AAPL) and Financial Select Sector (symbol: XLF). The mean errors in predicting four output parameters are shown respectively in Table 6.3 and Table 6.4. For all symbols, the accuracy of GPMKDR in predicting the level of the implied curve for the closest expiration (γ_1 of first expiration) is better than any other regressors. Note that the options on this curve are usually traded heavier (more liquid) than the other options. For the other output parameters, though our GPMKDR performs very well for the index option XLF, the prediction accuracy of GPMKDR of this stock option is lower than the other three regression methods. It is because some information in the original input related to these output parameters is lost while we select the limited number of dimensions in the latent space. We are able to improve the prediction accuracies for these outputs by learning the central subspace only from them.

Table 6.2: Prediction error mean and variance for GOOG.

Key Output	Linear	NW	GP	GPMKDR
γ_1 of 1st Exp.	0.528 ± 0.065	0.653 ± 0.110	0.709 ± 0.106	0.423 ± 0.060
γ_2 of 1st Exp.	0.434 ± 0.148	0.387 ± 0.125	0.260 ± 0.087	0.506 ± 0.094
γ_1 of 2nd Exp.	0.408 ± 0.049	0.335 ± 0.041	0.344 ± 0.032	0.214 ± 0.016
γ_2 of 2nd Exp.	0.404 ± 0.230	0.472 ± 0.336	0.490 ± 0.281	0.813 ± 0.244

Table 6.5 shows p-value from the statistical hypothesis tests between GPMKDR and the

Table 6.3: Prediction error mean and variance for AAPL.

Key Output	Linear	NW	GP	GPMKDR
γ_1 of 1st Exp.	0.971 ± 0.400	0.820 ± 0.225	0.402 ± 0.067	0.369 ± 0.104
γ_2 of 1st Exp.	0.406 ± 0.102	0.391 ± 0.094	0.512 ± 0.124	0.566 ± 0.129
γ_1 of 2nd Exp.	0.309 ± 0.065	0.110 ± 0.013	0.300 ± 0.060	0.379 ± 0.157
γ_2 of 2nd Exp.	0.283 ± 0.064	0.354 ± 0.069	0.297 ± 0.054	0.314 ± 0.071

Table 6.4: Prediction error mean and variance for XLF.

Key Output	Linear	NW	GP	GPMKDR
γ_1 of 1st Exp.	2.132 ± 3.562	1.258 ± 0.562	1.385 ± 0.350	0.683 ± 0.128
γ_2 of 1st Exp.	1.808 ± 6.921	0.795 ± 0.479	0.540 ± 0.146	0.521 ± 0.126
γ_1 of 2nd Exp.	1.139 ± 0.727	0.879 ± 0.344	0.931 ± 0.258	0.373 ± 0.071
γ_2 of 2nd Exp.	3.400 ± 36.441	1.025 ± 0.867	0.751 ± 0.264	1.325 ± 0.961

other regressors by using *t-test* and *Willcox Signed Rank test*. For these tests we focus on the task of predicting the level (γ_1) of the implied volatility curve for the closest expiration. For GOOG dataset, GPMKDR shows a similar sampling distribution to the linear regressor and for AAPL dataset, GPMKDR and GP have a significant similarity. The statistical test results show that our GPMKDR has a unique advantage on the task of predicting the IVS of index options that fluctuate in a wide range.

Table 6.5: Statistical model comparison.

Symbol	T-test			Wilcoxon signed-rank test		
	Linear	NW	GP	Linear	NW	GP
GOOG	0.571	0.00331	5.83e-5	0.34	3.42e-10	2.32e-13
AAPL	8.44e-15	1.15e-13	0.41	1.1e-16	1.82e-16	0.0642
XLF	6.14e-10	1.32e-6	4.8e-13	1.06e-7	0.0011	2e-13

Chapter 7

Conclusion

In this dissertation we have studied extensions of nonlinear dimensionality reduction applied to various contexts. Our focus is on a coupled subspace embedding induced from a pair of sequences, and takes account of the intrinsic structures of both sequences and characterizes the relationship. By utilizing this subspace, we improve the prediction accuracy in the prediction tasks with high dimensional input data.

In Chapter 3 we introduced the Marginal Autoregression (MAR) model and Marginal Nonlinear Dynamic System (MNDS) to model the dimensionality reduction process of single sequence. In contrast to the other subspace embedding models, our MNDS is the dimension reduction process which exploits the dynamic nature of the data sequence by utilizing MAR as a dynamic prior. MAR is the dynamic model representing all stable AR models, which marginalizes out the model parameters using Gaussian process (GP) prior. We test the utility of MNDS framework on the problem of 3D human figure tracking in sequence of monocular silhouette images. The results show that a dynamically constrained subspace using MNDS can effectively resolve the ambiguities in silhouette images and result in more accurate pose estimates than using the general static embedding without a dynamic approach.

In Chapter 4 we proposed Dynamic Probabilistic Latent Semantic (DPLSA) models which represent the embedding process taking account to the co-occurrence of dyadic sequences in the generative way. The experimental results on 3D pose estimation indicate that the our DPLSA formalism can achieve high accuracies with factional computation cost of the traditional tracking methods utilizing the subspace embedding. Therefore, the proposed model has the potential to handle complex classes of tracking problems, such as challenging rapidly changing motions, when coupled with multiple model frameworks such as switching dynamic models. Future work can address these new directions as well as focus on continuing extensive evaluation of

DPLSA on additional motion datasets. As we noted in our experiments on 3D human figure tracking, the style problem should be also resolved for improved automatic human tracking.

In Chapter 5 we proposed the novel dimension reduction approach called Gaussian Process Manifold Kernel Dimension Reduction (GPMKDR) to use the dimensionality reduction to make predictions in the discriminative way. Our GPMKDR model is the reformulation of the previous manifold Kernel Dimension Reduction (mKDR) approach which discovers a subspace embedding that best preserves information relevant to a nonlinear regression. Instead of an iterative solution without a convergence guarantee, our model provides a globally optimal solution in a closed form which is given by the eigen-decomposition. This framework eliminates the need for parameter setting of an iterative process and reduces the computational cost for learning. The results on various real datasets indicates the our GPMKDR can achieve high accuracy in prediction of regression with small computational costs. Our future work focus on the full discriminative dynamic model exploiting the temporal information in our GPMKDR framework.

In Chapter 6 we apply our GPMKDR regression framework to real financial data. We formulated the problem of predicting the implied volatility surface (IVS) from the current market data inputs as a regression problem. The high dimensionality of input including all available market data and the existence of small hidden factors inducing the IVS movements indicates that our GPMKDR model can be an ideal fit to this problem. And the experimental results also show that our approach results in more accurate predictions when compared to other general regression methods.

In the appendices we have gathered additional details relevant for the computations performed in the previous chapters.

Appendix A

MAR Gradient

Log-likelihood of the MAR model is, using Equation (3.3) and leaving out the constant term,

$$L = \frac{N}{2} \log |K_{xx}| + \frac{1}{2} tr \left\{ K_{xx}^{-1} X X' \right\} \tag{A.1}$$

with $K_{xx} = K_{xx}(X, X)$ defined in Equation (3.4). The gradient of L with respect to X is

$$\frac{\partial L}{\partial X} = \frac{\partial X_\Delta}{\partial X} \frac{\partial L}{\partial K_{xx}} \frac{\partial K_{xx}}{\partial X_\Delta} + \left. \frac{\partial L}{\partial X} \right|_{X_\Delta}. \tag{A.2}$$

X_Δ can be written as a linear operator on X,

$$X_\Delta = \Delta \cdot X, \;\; \Delta = \begin{bmatrix} 0_{(T-1)\times 1} & I_{(T-1)\times(T-1)} \\ 0 & 0_{1\times(T-1)} \end{bmatrix}, \tag{A.3}$$

where 0 and I denote zero vectors and identity matrices of sizes specified in the subscripts. It is now easily follows that

$$\frac{\partial L}{\partial X} = \Delta' \left(N K_{xx}^{-1} - K_{xx}^{-1} X X' K_{xx}^{-1} \right) \Delta \cdot X + K_{xx}^{-1} X. \tag{A.4}$$

Appendix B

GPMKDR Derivation

To derive (Equation 5.12) we follow steps similar to those outlined in [57]. Given the objective J as (Equation 5.9), it can be shown that the gradient of the objective can be written as

$$\frac{1}{2}\nabla J(\Phi) = -UK(\Phi)^{-1}K_{yy}^{c}K(\Phi)^{-1}U^{\mathrm{T}}\Phi + MUK(\Phi)^{-1}U^{\mathrm{T}}\Phi.$$

Hence, the solution Φ has to satisfy

$$UK(\Phi)^{-1}K_{yy}^{c}K(\Phi)^{-1}U^{\mathrm{T}}\Phi = MUK(\Phi)^{-1}U^{\mathrm{T}}\Phi. \tag{B.1}$$

If ALB^{T} is the SVD decomposition of Φ, using the Woodbury matrix inversion lemma the term $K(\Phi)^{-1}$ becomes

$$K(\Phi)^{-1} = \frac{1}{N\epsilon}\left[I - U^{\mathrm{T}}AL\left(N\epsilon I + L^2\right)^{-1}LA^{\mathrm{T}}U\right], \tag{B.2}$$

where we used the fact that $UU^{\mathrm{T}} = I$ and $A^{\mathrm{T}}A = I$. We now substitute (B.2) into (5.11). Note that the following holds

$$\begin{aligned} K(\Phi)^{-1}U^{\mathrm{T}}\Phi &= \frac{1}{N\epsilon}\left[I - U^{\mathrm{T}}AL\left(N\epsilon I + L^2\right)^{-1}LA^{\mathrm{T}}U\right]U^{\mathrm{T}}AL \\ &= \frac{1}{N\epsilon}U^{\mathrm{T}}AL\left[I - \left(N\epsilon I + L^2\right)^{-1}L^2\right] \\ &= U^{\mathrm{T}}AL\left(N\epsilon I + L^2\right)^{-1}. \end{aligned} \tag{B.3}$$

Similarly,

$$\begin{aligned} UK(\Phi)^{-1} &= \frac{1}{N\epsilon}U\left[I - U^{\mathrm{T}}AL\left(N\epsilon I + L^2\right)^{-1}LA^{\mathrm{T}}U\right] \\ &= \frac{1}{N\epsilon}\left[I - AL\left(N\epsilon I + L^2\right)^{-1}LA^{\mathrm{T}}\right]U. \end{aligned} \tag{B.4}$$

Substituting (B.3) and (B.4) into (5.11) results in

$$\begin{aligned} &\left[I - AL\left(N\epsilon I + L^2\right)^{-1}LA^{\mathrm{T}}\right]UK_{yy}^{c}U^{\mathrm{T}}AL(N\epsilon I + L^2)^{-1} \\ &\quad = M\left[I - AL\left(N\epsilon I + L^2\right)^{-1}LA^{\mathrm{T}}\right]AL. \end{aligned} \tag{B.5}$$

Premultiplying[1] both sides by $\left[I - AL\left(N\epsilon I + L^2\right)^{-1} LA^{\mathrm{T}}\right]^{-1}$ and postmultiplying by $(N\epsilon I + L^2)L^{-1}$ finally yields

$$\frac{1}{M} U K_{yy}^{c} U^{\mathrm{T}} A = A(N\epsilon I + L^2). \tag{B.6}$$

[1]The $M \times M$ matrix is nonsingular as it can be written in the form $N\epsilon\left(N\epsilon I + AL^2A^{\mathrm{T}}\right)^{-1}$.

References

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

[2] D. J. C. MacKay. Introduction to gaussian processes. In *C. M. Bishop Edt, Neural Networks and Machine Learning*, volume 168 of *NATO ASI Series*, pages 133–165. Springer, 1998.

[3] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2005.

[4] N. D. Lawrence. Probabilistic non-linear principal component analysis with gaussian process latent variable models. *Journal of Machine Learning Research*, 6:1783–1816, 2005.

[5] A. Y. Ng and M. I. Jordan. On discriminative vs. generative classi ers: A comparison of logistic regression and naive bayes. In *NIPS*. 2001.

[6] Matthew Brand. Shadow puppetry. In *CVPR*, volume II, pages 1237–1244, 1999.

[7] R. Rosales and S. Sclaroff. Specialized mappings and the estimation of human body pose from a single image. In *Workshop on Human Motion*, pages 19–24, 2000.

[8] A. Agarwal and B. Triggs. 3d human pose from silhouettes by relevance vector regression. In *CVPR*, pages II 882–888, 2004.

[9] I. T. Jolliffe. *Principal Component Analysis*. Springer Verlag, 1986.

[10] H. Sidenbladh, M. J. Black, and D. J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *ECCV*, pages 702–718, 2000.

[11] D. Ormoneit, H. Sidenbladh, M. J. Black, and T. Hastie. Learning and tracking cyclic human. In *NIPS*, pages 894–900. 2001.

[12] R. Urtasun, D. J. Fleet, and P. Fua. Monocular 3d tracking of the golf swing. In *CVPR*, page 1199, 2005.

[13] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

[14] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, December 2000.

[15] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[16] Q. Wang, G. Xu, and H. Ai. Learning object intrinsic structure for robust visual tracking. In *CVPR (2)*, pages 227–233, 2003.

[17] A. Elgammal and C. Lee. Inferring 3d body pose from silhouettes using activity manifold learning. In *CVPR*, volume 2, pages 681–688, 2004.

[18] C. Sminchisescu and A. Jepson. Generative modeling for continuous non-linearly embedded visual inference. In *ICML*, New York, NY, USA, 2004. ACM Press.

[19] C. K. I. Williams and D. Barber. Bayesian classification with Gaussian processes. *Pattern Analysis and Machine Intelligence*, 20(12):1342–1351, 1998.

[20] N. D. Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. In *NIPS*. 2004.

[21] K. Grochow, S. L. Martin, A. Hertzmann, and Z. Popovic. Style-based inverse kinematics. *ACM Transactions on Graphics*, 23(3):522–531, 2004.

[22] T. Tian, R. Li, and S. Sclaroff. Articulated pose estimation in a learned smooth space of feasible solutions. In *CVPR*, 2005.

[23] J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models. In *NIPS*. 2005.

[24] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *JASIST*, 41(6):391–407, 1990.

[25] P. W. Peter and S. T. Dumais. Personalized information delivery: an analysis of information filteringmethods. *Communications of the ACM*, 35:51–60, December 1992.

[26] J. R. Bellegarda. Exploiting both local and global constraints for multi-spanstatistical languagemodeling. In *ICASSP*, volume 2, pages 677–680, 1998.

[27] T. Hofmann. Probabilistic latent semantic analysis. In *Uncertainty in A.I,*, Stockhol, 1999.

[28] F. Monay and D. Gatica-Perez. Plsa-based image auto-annotation: constraining the latent space. In *ACM International Conference on Multimedia*, pages 348–351, 2004.

[29] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google's image search. In *ICCV*, pages 1816–1823, 2005.

[30] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering object categories in image collections. In *ICCV*, volume 1, pages 370–378, 2005.

[31] R. Urtasun, D. J. Fleet, A. Hertzmann, and P. Fua. Priors for people tracking from small training sets. In *ICCV*, Beijing, China, 2005.

[32] R. Urtasun, D. J. Fleet, and P. Fua. 3d people tracking with gaussian process dynamical models. In *CVPR*, pages 238–245, 2006.

[33] K. Moon and V. Pavlovic. Impact of dynamics on subspace embedding and tracking of sequences. In *CVPR*, pages 198–205, June 2006.

[34] N. D. Lawrence and A. J. Moore. Hierarchical gaussian process latent variable models. In *ICML*, pages 481–488. ACM, 2007.

[35] A. Shon, K. Grochow, A. Hertzmann, and R. Rao. Learning shared latent structure for image synthesis. *NIPS*, 2005.

[36] R. Navaratnam, A. Fitzgibbon, and R. Cipolla. Semi-supervised joint manifold learning for multi-valued regression. In *ICCV*, 2007.

[37] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin. Graph embedding and extensions: A general framework for dimensionality reduction. *Pattern Analysis and Machine Intelligence*, 29:40–51, 2007.

[38] K. C. Li. Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, 86:316–327, 1991.

[39] K. C. Li. On principal hessian directions for data visualization and dimension reduction: Another application of stein's lemma. *Journal of the American Statistical Association*, 87:1025–1039, 1992.

[40] R. D. Cook and S. Weisberg. Discussion of li. *Journal of the American Statistical Association*, 86:328–332, 1991.

[41] B. Li, H. Zha, and F. Chiaromonte. Contour regression: A general approach to dimension reduction. *The Annals Of Statistics*, 33:1580–1616, 2005.

[42] K. Fukumizu, F. R. Bach, and M. I. Jordan. Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces. *Journal of Machine Learning Research*, 5:73–99, 2004.

[43] K. Fukumizu, F. R. Bach, and M. I. Jordan. Kernel dimension reduction in regression. Technical report, Department of Statistics, University of California, Berkeley, 2006.

[44] Sajama and A. Orlitsky. Supervised dimensionality reduction using mixture models. In *ICML*, pages 768–775, 2005.

[45] X. Yang, H. Fu, H. Zha, and J. Barlow. Semi-supervised nonlinear dimensionality reduction. In *ICML*, pages 1065–1072, 2006.

[46] N. D. Lawrence. Probabilistic non-linear principal component analysis with gaussian process latent variable models. *Journal of Machne Learning Research*, 6:1783–1816, 2005.

[47] V. L. Girko. Circular law. *Theory of Probability and Its Application*, 29:694–706, 1984.

[48] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Discriminative density propagation for 3d human motion estimation. In *CVPR*, pages 390–397, 2005.

[49] C. Sminchisescu, A. Kanujia, Z. Li, and D. Metaxas. Conditional visual tracking in kernel space. In *NIPS*, 2005.

[50] http://www.hid.ri.cmu.edu/Hid/databases.html.

[51] C. Lee and A. Elgammal. Simultaneous inference of view and body pose using torus manifolds. In *ICPR*, 2006.

[52] L. Sigal, S. Bhatia, S. Roth, M. J. Black, and M. Isard. Tracking loose-limbed people. In *CVPR*, pages 421–428, 2004.

[53] R. D. Cook. Using dimension-reduction subspaces to identify important inputs in models of physical systems. In *Proceedings of Section on Physical and Engineering Sciences*, 1994.

[54] R. D. Cook. *Regression graphics*. Wiley Inter-Science, 1998.

[55] J. Nilsson, F. Sha, and M. I. Jordan. Regression on manifolds using kernel dimensionality reduction. In *ICML*, 2007.

[56] C. K. I. Williams and C. E. Rasmussen. Gaussian processes for regression. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8*, pages 514–520, Cambridge, MA, 1996. The MIT Press.

[57] N. D. Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. In *Advances in Neural Information Processing Systems 16*. The MIT Press, 2004.

[58] MSU data are produced by Remote Sensing Systems and sponsored by the NOAA Climate and Global Change Program. Data are available at www.remss.com.

[59] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *Pattern Analysis and Machine Intelligence*, 23(6):643–660, 2001.

[60] K. C. Lee, J. Ho, and D. Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *Pattern Analysis and Machine Intelligence*, 27(5):684–698, 2005.

[61] A. Asuncion and D. J. Newman. UCI machine learning repository, 2007.

[62] Y. LeCun. Mnist handwritten digit database, http://yann.lecun.com/exdb/mnist/.

[63] http://www.cs.toronto.edu/∼roweis.

[64] F. Black and M. Scholes. The pricing of options and corporate liabilities. *The Journal of Political Economy*, 81:637–654, 1973.

[65] J. C. Cox, S. A. Ross, and M. Rubinstein. Option pricing: A simplified approach. *Journal of Financial Economics*, 7:229–263, 1979.

[66] A. M. Malz. Do implied volatilities provide early warning of market stress? *Journal of Risk*, 3, 2001.

[67] M. Britten-Jones and A. Neuberger. Option prices, implied price processes, and stochastic volatility. *Journal of Finance*, 55(2):839–866, 2000.

[68] R. Cont and J. D. Fonseca. Dynamics of implied volatility surfaces. *Quantitative Finance*, 2, 2002.

[69] M. Fengler, W. Härdle, and C. Villa. The dynamics of implied volatilities: A common principal components approach. *Review of Derivatives Research*, 6:179–202, 2003.

[70] S. Gonçalves and M. Guidolin. Predictable dynamics in the s&p 500 index options implied volatility surface. *Journal of Business*, 2006.

[71] M. R. Fengler, W. K. Härdle, and E. Mammen. A semiparametric factor model for implied volatility surface dynamics. *Journal of Financial Econometrics*, 5:189–218, 2007.

[72] F. Audrino and D. Colagelo. Forecasting implied volatility surfaces. University of St. Gallen Department of Economics working paper series 2007 with number 2007-42, 2007.

Vita

Kooksang Moon

1991	Graduated from Chungdong High School, Seoul, Korea.
1992 - 1996	**B.S.** in Electrical Engineering, Yonsei University, Seoul, Korea.
1999 - 2001	**M.S.** in Computer Science and Engineering, The State University of New Your (SUNY) at Buffalo, Buffalo, New York.
2003 - 2006	Graduate Assistant, Computer Science, Rutgers University, Piscataway, New Jersey.
2006 - 2008	Teaching Assistant, Computer Science, Rutgers University, Piscataway, New Jersey.
2008	Graduate Assistant, Computer Science, Rutgers University, Piscataway, New Jersey.
2009	**Ph.D.** in Computer Science, Rutgers University, Piscataway, New Jersey.

Publications

2005	Estimation of Human Figure Motion Using Robust Tracking of Articulated Layers Kooksang Moon and Vladimir Pavlovic, *IEEE Workshop on Vision for Human-Computer Interaction*, (2005).
2006	Impact of Dynamics on Subspace Embedding and Tracking of Sequences Kooksang Moon and Vladimir Pavlovic, *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, (2006).
2007	Graphical Models for Human Motion Modeling Kooksang Moon and Vladimir Pavlovic, *chapter in Human Motion Capture: Modeling, Analysis, Animation, Metaxas, Rosenhahn and Kleete Eds.*, Springer, (2007).
2008	Monocular 3D Human Motion Tracking Using Dynamic Probabilistic Latent Semantic Analysis Kooksang Moon and Vladimir Pavlovic, *Canadian Conference on Computer and Robot Vision*, (2008).

2008 Regression Using Gaussian Process Manifold Kernel Dimensionality Reduction
Kooksang Moon and Vladimir Pavlovic, *IEEE Int. Workshop on Machine Learning for Signal Processing*, (2008).