CONDITIONAL MODELS FOR 3D HUMAN POSE ESTIMATION

BY ATUL KANAUJIA

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

Dimitris Metaxas

and approved by

_____

_____

_____

_____

New Brunswick, New Jersey

January, 2010

ABSTRACT OF THE DISSERTATION

Conditional Models for 3D Human Pose Estimation

by ATUL KANAUJIA

Dissertation Director: Dimitris Metaxas

Human 3d pose estimation from monocular sequence is a challenging problem, owing to highly articulated structure of human body, varied anthropometry, self occlusion, depth ambiguities and large variability in the appearance and background in which humans may appear. Conventional vision based approaches to human 3d pose estimation mostly employed ”top-down methods”, which used a complete 3d human model, in a hypothesized pose, to explain the configuration of the humans in the observed 2d image. In this thesis, we work with ”bottom-up methods” for human pose estimation, that use low level image features to directly predict 3d pose. The research draws on recent innovations in statistical learning, observation-driven modeling, stable image encodings, semi-supervised learning and learning perceptual representations. We address the problems of *(a)* modeling pose ambiguities due to 3d-to-2d projection and self occlusion, *(b)* lack of sufficient labeled data for training discriminative models and *(c)* high dimensionality of human 3d pose state space. In order to resolve 3d pose ambiguities, we use multi-valued functions to predict multiple plausible 3d poses for an image observation. We incorporate unlabeled data in a semi-supervised learning framework to constrain and improve the training of discriminative models. We also propose generic probabilistic Spectral Latent Variable Models to efficiently learn low dimensional representations of high dimensional observation data and apply it to the problem of human 3d pose inference.

Acknowledgements

First and foremost, I thank my advisor Dimitris Metaxas for continuously encouraging me to do fundamental research rather than incremental work. He had been very supportive for all the research work and project demonstrations I performed during my PhD years at Rutgers University. His passion for work, extremely tolerant attitude and trust on me has always motivated me to work with him.

I thank my co-advisor, Cristian Sminchisescu for guiding me both during my initial and later years of the PhD program. Technical discussions with him were always very thorough and insightful. He was very helpful in guiding me and providing useful feedbacks for improving research talks and presentations. Without his supervision, clear planning and detailed analysis, this research work wouldn't have existed.

I thank my committee members Ahmed Elgammal, Vladimir Pavlovic and Chandra Kambhamettu for taking out time to review the thesis and providing useful feedbacks to improve it. I express my gratitude to Ahmed Elgammal and Vladimir Pavlovic for the technical assistance they provided me during the coursework and research at Rutgers. Special thanks to my colleagues at CBIM, Zhiguo Li, Yuchi Huang, Peng Yang, Chansu Lee and Gabriel Tsechpenakis for helping me in preparing talks, demonstrations and organizing paper discussion groups.

Finally, thanks to my colleagues at ObjectVideo, Mun Wai Lee and Gaurav Aggarwal for always motivating me to wrap up my thesis during my employment. I also thank my manager, Niels Haering for being understanding and allowing me to finish the thesis at my own pace.

Dedication

To my wife Shweta, for putting up long hours with me during my thesis writing, and my parents for their constant support

Table of Contents

List of Abbreviations

ACC Adaptive Combination of Classifiers

ASM Active Shape Models

ARD Automatic Relevance Determination

BME Bayesian Mixture of Experts

CART Classification And Regression Tree

CG Computer Graphics

CCA Canonical Correlation Analysis

CMU Carnegie Mellon University

DOF Degrees Of Freedom

DRR Dimensionality Reduction for Regression

EM Expectation Maximization

GMM Gaussian Mixture Model

GTM Generative Topographic Map

GPLVM Gaussian Process Latent Variable Model

HE Hessian Eigenmaps

HOG Histogram of Oriented Gradients

ID3 Iterative Dichotomiser 3

IRLS Iterative Re-weighted Least Square

ISOMAP ISOmetric MAPping

ISM Implicit Shape Models

kBME Kernel Bayesian Mixture of Experts

KDE Kernel Dependency Estimation

KLT Kanade-Lucas-Tomasi

LLE Locally Linear Embedding

LM-BFGS Limited Memory-Broyden Fletcher Goldfarb Shanno

LTSA Local Tangent Space Alignment

MAP Maximum A Posterior

LE Laplacian Eigenmaps

MAR Marginal Auto-Regressive

MARS Multivariate Adaptive Regression Splines

ME Mixture of Experts

ML Maximum Likelihood

MLP Multi-Layer Perceptrons

MoCap Motion Capture

MSER Maximal Stable Extremal Regions

MSB Multi-Level Spatial Blocks

NN Nearest Neighbor

RVM Relevance Vector Machine

PCA Principal Component Analysis

PGH Pairwise Geometric Histograms

PPCA Probabilistic Principal Component Analysis

RBF Radial Basis Functions

REM Regularized Expectation Maximisation

RKHS Reproducing Kernel Hilbert Space

RCA Relevant Component Analysis

SBL Sparse Bayesian Learning

SCAPE Shape Completion and Animation of PEople

SIFT Scale Invariant Feature Transform

SLVM Spectral Latent Variable Model

SLDS Switching Linear Dynamic System

SNoW Sparse Network of Winnows

SOM Self Organizing Map

SVM Support Vector Machines

VQ Vector Quantization

Chapter 1

Introduction

Widespread installation of surveillance cameras has reinvigorated research in vision problems of object detection, tracking, classification and recognition. One of the core capabilities of intelligent video analytics is detection and tracking of humans in 3D space, recognition of their activities, detection of anomalous behavior and assessment of potential threats they can pose. In order to apply optical technologies to detailed analysis of human behavior, it is not sufficient to coarsely localize the targets in the scene. Rather, to do so, we need to estimate their 3D articulated posture, shape and motion in the scene. Traditionally, the task of 3D pose estimation had been addressed using a multi-camera based 3D motion capture system, that required the subjects to wear specialized suits with precisely placed and calibrated markers, in a carefully controlled laboratory environment. The setup process is time-consuming and costly, and impractical for non-invasive surveillance based applications. Moreover, most of these systems required manual initialization and are not automatic.

Recent research has lead to development of less invasive marker-less 3D human motion capture systems, using multiple, strategically placed, calibrated cameras. However, this approach still requires a well instrumented operating environment, which is difficult to achieve in surveillance systems where the cameras are generally sparse and usually have non-overlapping field of views. The problem is significant both in terms of enhancing applicability and reducing the installation costs of the motion capture technology to gaming and animation industry, intelligent video analytics and surveillance.

In most cases, the only video footages that are available for reconstruction are in monocular image format. Millions of security cameras have already been installed globally for asset protection, airport and port security, border monitoring and law enforcement purposes. For example, the London subway has over 10,000 cameras installed. However, monitoring these

video feeds by human operators is extremely labor intensive and unreliable. Hence it is desirable to develop advanced motion capture techniques to automatically generate 3D motion estimates of the human targets by tracking their movements in these video footages and enable more detailed behavioral analysis.

Monocular motion capture technology is critical for automatic video analysis and has a wide ranging commercial applications including computer animation, realistic gaming, human-computer interface, ergonomics, biomechanical and clinical studies. We identify the following areas where the technology can be applied:

- *Effective automatic video analytics tools* - can help monitor large number of video feeds to detect unusual activities and identify potential threats. Suspicious or hostile human activities may include security boundary intrusion, loitering, climbing of fences, carrying of heavy or large objects and left-bag events. Automating human behavior and shape analysis are key steps towards achieving this goal.

- *Improved human computer interaction* - Automated 3D motion capture will assist in developments of techniques for improved human computer interaction by using more accurate vision based components to recognize different gesture and motion in 3D. This has vast potential use in role-playing games where the movements of the user in the physical domain are appropriately reflected as an action in the virtual environment.

- *Cost effective solution for movement analysis* - Improved 3D motion capture will allow non-invasive techniques for identifying the underlying causes for walking abnormalities in clinical patients. The results of gait analysis have been shown to be useful in determining the best course of treatment in these patients.

- *Intelligent training systems for sporting activities* - The analysis of sports related movements often entails analyzing a variety of highly dynamic movements. Motion analysis provides the tools for the sports medicine and performance professionals to perform accurate functional evaluations/analyses for clinical and research oriented purposes.

- *Realistic animation* - Cost effective solution to importing realistic body movements in animated characters in videos. Human gait modeling can be used to simulate realistic

(a) (b) (c)

Figure 1.1: Current motion capture systems are marker-based and require expensive multi-camera setup, in a well instrumented environment. Applications of 3D motion capture technology include: *(a)* Real-time computerized motion analysis for sports training and performance evaluation. Figure shows a practical application (*courtesy Competitive Edge, Inc.*) that provides 3D swing motion analysis for improving biomechanical efficiency of the human body. *(b) and (c)* Motion capture systems are increasingly used for realistic animations and special effects in the entertainment industry and gaming.

walking styles. Animating a digital character by imitating the actions of a real actor in a movie sequence can be easily achieved if accurate 3D reconstruction from monocular image sequence is feasible.

- *Robotic locomotion* - Design of robot appendages and control mechanisms to allow robots to move fluidly and efficiently

This chapter introduces the thesis, giving overview of the human 3d pose estimation problem, provides a brief background on the problem and the potential applications of this area. We list out the main challenges and open problems that are encountered in this field. We present possible ways of solving these problem and how we address them in this thesis. Finally, we discuss the key contributions of the thesis and the chapters in which each of the topics have been discussed in detail.

1.1 Human Pose Reconstruction using Monocular Image Sequence

In this thesis we take a step towards developing a fully automated solution for marker-less motion capture system from monocular image sequences. We focus on the problem of estimating

3D human pose from monocular image sequence captured from either a stationary or a moving camera. We do not assume cameras to be calibrated in our framework. The algorithms developed in the thesis draws on recent innovations in statistical learning, observation driven modeling, robust image encoding and manifold learning.

1.1.1 Issues and Challenges

Our goal in this thesis, is to develop robust solutions for automatic 3D human pose inference from monocular image sequences, in an uncontrolled environment. While humans are adept in inferring 3D pose of the objects using only relatively low-resolution visual observations and temporal cues, for the vision based systems it is still a challenging task. We identify the following key challenges:

- **Ambiguity due to projection of objects from 3D space to 2D image plane:** Generation of a 2D planar image from 3D scene is modeled as a perspective projection from 3D space to 2D plane. Perspective projection is a non-linear transformation and can be modeled using a pinhole camera model, intrinsic distortion parameters of the camera and rigid transformation between the world and camera. Inverse perspective projection on the other hand, transforms a point on the 2D image to a line vector in 3D and is a one-to-many relation as any point along this 3D line vector would project to the same 2D point on the image. Finding an inverse of a perspective projection transformation is therefore an ill-conditioned problem as it involves learning a one-to-many mapping from 2D image points to multiple plausible configurations in 3D space. The lack of depth information make the human 3D pose estimation problem inherently difficult and obscure.

  Pose ambiguities thus necessitates the use of additional cues from either the learned priors on 3D poses or the temporal dynamics, learned from various typical human activities. Hence, for a 3D pose inference problem, that typically involves optimization of a image matching cost distribution (usually a posterior of pose space), the one-to-many mapping from 2D image to 3D space manifests as multiple modes on this cost surface (the posterior map over the 3D pose state space).

  A variety of techniques have been proposed to adequately handle these multimodalities

in the dataset. The posterior distribution is usually modeled as a multi-modal mixture of Gaussian distribution or as piecewise Gaussian distributions. A number of works have also adopted sequential Monte Carlo based approaches to model multiple modes of the posterior as a set of the samples from the posterior distribution.In order to propagate the multimodal posterior over time, a multi-hypothesis tracking framework is employed. In top-down models, efficient observation likelihood function based on robust image matching techniques can disambiguate many poses. In bottom-up modeling, multi-valued relation from the images descriptor space to 3D human pose space can be learned using multiple regression functions. Improving the image descriptors may further assist in resolving many pose ambiguities. For example, silhouette based descriptors are more ambiguous compared to feature encodings based on the texture in the interior regions of the target.

- **Large variability in shape, appearance and anthropometry of humans**: Humans occur in a variety of complex poses, shapes, anthropometry and clothing appearances. Loose fitting clothes can cause occlusion of body parts and interfere in detection of coherent structures in the image that enable accurate estimation of 3d pose. In real imaging scenarios there may be additional noise, due to specularities, lighting and viewpoint changes. In bottom-up models, large variability in the observations is handled by improving feature extraction techniques and developing robust image descriptors that can discriminate between different poses yet remain invariant to geometric and photometric variations in the image. Finding appropriate descriptors for a given problem is a difficult task and usually involves comparing multiple descriptors and choosing the descriptor that gives best prediction accuracy.

- **Image clutter**: Accurate localization of human targets in a 2D image is a pre-requirement step for 3D pose estimation. Although background subtraction can be used to accurately delineate human targets, it requires modeling of static pixels in the scene and severely restricts the range of applications to scenarios with static background only. In realistic settings, with moving camera and changing background, detection and localization of

humans is even more challenging, and techniques based on human detection by classification are strongly affected by the presence of clutter, transient backgrounds and large intra-class variability of humans. In most cases the targets are coarsely localized as bounding boxes in the image. The extraneous regions around the target in the bounding box tend to influence the descriptors computed over these bounding boxes and significantly increase their variability. Training accurate predictors that are robust to these background noises is a challenging task and requires special mechanism to reduce these perturbations to the image descriptors.

- **Self occlusion and kinematic ambiguities**: Human body has a highly articulated structure and may assume complex poses, occasionally causing a body part to occlude the other, when observed from a single viewpoint. For example common activities like running or walking, when viewed from side, frequently cause self occlusion of the arm and leg joints that are not facing the camera. In addition, rotation symmetries of different body parts make it difficult to estimate all the kinematic parameters of the human pose. For example, rotation of arm segments around its axis (twist), is difficult to estimate from visual cues alone, as the change in the appearance due to twists in the arm may be imperceptible under a standard image resolution. The non-observability of body parts and kinematic ambiguities, make the problem of 3D pose recovery inherently difficult.

  Ambiguities due to self occlusion can be resolved by incorporating motion priors in the pose estimation framework. Most of the joint angles are strongly correlated in various human activities, and configurations of the joints that are not directly observable in the image, can be inferred from the observable joints, by learning correlations between them.

- **High dimensionality**: Human body has highly articulated structure, requiring a number of joints to accurately model the pose in 3D space. A skeletal human body model typically contains ≈ 30 joint parameters to characterize pose in terms of positions and orientations of the skeletal links. Many animation applications may require more detailed articulations (*e.g.* fingers and feet) that have skeletons with ≈ 60 degrees of freedom. Estimation and tracking of the articulated 3d human pose thus requires search in high

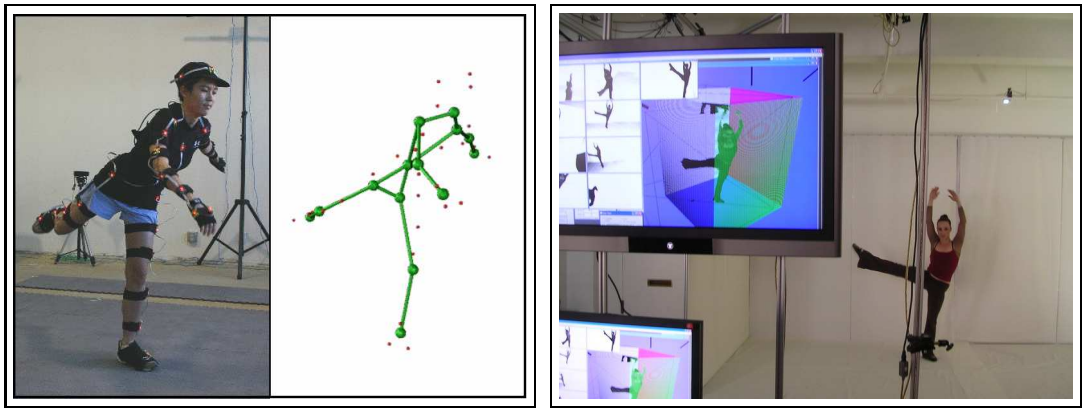


Figure 1.2: Current state of the art motion capture systems *(left)* The subjects need to wear markers (active or passive) that can be readily detected from multiple sensors. The 3D positions of the markers are computed using triangulation and a kinematic skeleton is optimally fitted to the point clouds. Figure here shows a subject with active markers (illuminated LEDs) that are easily identified by the optical system. *(right)* Recent research (courtesy **Organic Motions, Inc.**) have developed marker-less techniques for motion capture that are able to track precise movements of the subjects in their everyday clothing.

dimensional state space for the optimal pose that best explains the observed configuration in the 2D image. Inference must therefore take place over a large space of possible 3D configurations and thus entails use of appropriate optimization techniques that avoid local minima and attain globally optimal 3D pose .

- **Kinematic singularity**: A singular point of an algebraic curve is a point where the curve becomes degenerate (has unpredictable behavior). Ambiguities may as well be introduced due to inability of the framework to uniquely represent a functional mapping (or transformation). At the singular point of a function $\mathbf{y} = f(\mathbf{x})$, the Jacobian matrix $\mathbf{J}(\mathbf{x}) = [\frac{\partial \mathbf{y}}{\partial x_1}, \frac{\partial \mathbf{y}}{\partial x_2}, \cdots, \frac{\partial \mathbf{y}}{\partial x_n}]^T$ is not of full rank and its pseudo-inverse is not defined. In the neighborhood of singular points, even a small change in $f(\mathbf{x})$, $\delta\mathbf{y} = \delta f(\mathbf{x})$ requires an enormous change in $\delta\mathbf{x}$. A typical example of such a shortcoming is kinematic singularity arising due to non-unique decomposition of the 3D rotation angles. A 3D joint angle can be described by multiple compositions of rotations along the 3 orthogonal axes, thus resulting in degenerate solutions.

1.2 Background on Human Pose Reconstruction

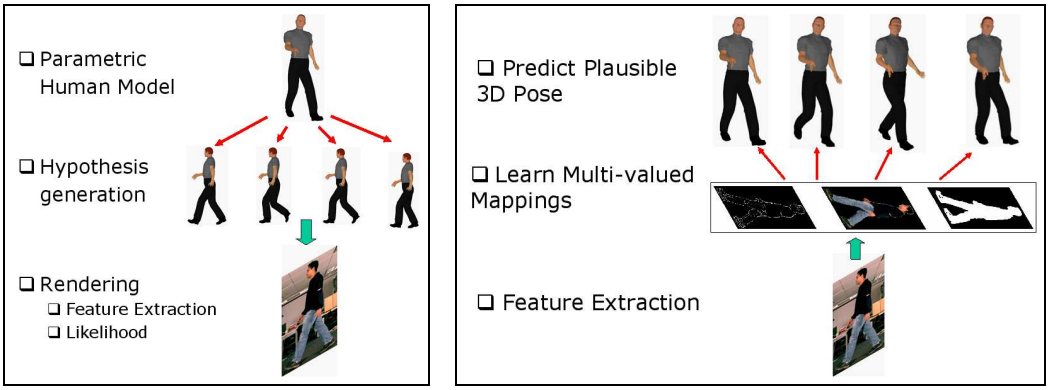
Articulated skeleton of human body is represented using a hierarchical chain of 1D rigid bodies, called links, that are interconnected to one another via joints. The links are free to rotate around the joints about the axes. The Degree of Freedom (DOF) associated with the joint depends on its type (whether pivot, saddle, hinge or ball and socket joint). The dimensionality of the 3d pose state space is thus determined by DOF of each of the joints in the kinematic model. Depending upon the desired application, the skeletal articulation may be either detailed - that can capture subtle movements(e.g. 3D character animations) or low-detailed - that are required to model only representative poses of the application domain (e.g. people in walking or standing pose).

The skeleton structure is organized as a hierarchy, with the root joint having global translation and rotation parameters. The rest of the skeletal segments are obtained by constructing global transformation using all the segments in the hierarchical path that connect this segment to the root. The joint angles are represented in a local coordinate frame relative to the parent joint. This is to avoid the error in a single joint to distort the entire 3D pose. The global transformation is obtained by a series of translation offsets and local rotational transformations of the segments in the hierarchical path connecting the segment to the root.

An additional advantage of representing 3D pose as joint angles (instead of joint locations) is that the motion capture data from one skeleton can be easily imported to another skeleton. This can be directly used for deforming a computer graphic character with animation packages like *Maya* and *Poser*. A potential setback of using joint angles to encode 3D pose is that angular measurements are cyclical and angles separated by 360^o are the same. We overcome this problem by transforming the discontinuous joint angle space to continuous sinusoidal space and representing each joint angle with a pair of sine and cosine values. There exist two paradigms of designing a human 3D pose estimation framework - top-down (generative models) and bottom-up (discriminative) modeling.

1.2.1 Top-down Modeling

A top-down methodology refers to using a semantically, high-level description of the complete human pose to explain lower level image signals. Generative models have been around in



Top-down modeling Bottom-up modeling

Figure 1.3: Comparison of the two paradigms of human 3D pose estimation frameworks, *(Left)* Top-down or generative modeling attempts to model the 2d image generation process from the objects in 3d space. They assume a prior human model for generating hypotheses of 3d poses. These poses are evaluated using a projective camera model and an observation likelihood model. The pose is estimated as the modes of the likelihood cost surface. *(Right)* Bottom-up or discriminative modeling extract features from the images and directly predict 3D poses using a learned predictor model

vision for a long time and a number of approaches exist that estimate the 3D pose by explicitly modeling the 2D image generation process by rendering a synthetic 3D human model. Top-down methods rely on accurate representation of the human body using a kinematic chain based articulated structure and use it to evaluate various 3D pose hypotheses by rendering it to 2D image plane and matching it to the observed images.

Generative methods requires modeling of both 3d pose and 3d shapes of the human body. The 3D human pose is encoded as a vector containing the global rotation of the root joint and a set of relative joint angles for each of the limbs of the articulated kinematic chain model. In the past, most of the generative frameworks have employed simplistic, low-detailed human body shape models. These synthetic models were based on simple geometric shapes such as truncated cones or tapered cylinders with elliptical cross-sections. Although simplistic and computationally efficient for rendering, these models are often poor representation of human 3d shape.

The 3D pose and shape of the human body model are controlled by the parameters for the 3D joint angles and latent parameters of the 3D shapes. The generative tracking frameworks estimate these parameters by searching in the parameter space using stochastic methods and generating several hypotheses for the 3D poses and shapes. These hypotheses are evaluated by

directly rendering the 2D image from the 3D human models synthesized with these parameters and outputting the most probable 3D pose and shape configuration.

High dimensionality of the pose and shape space has motivated the development of more sophisticated algorithms for stochastic search in the parameter space. Unconstrained search algorithms may be sub-optimal as the parameter space is usually high dimensional. Most of these search algorithms therefore exploit kinematic priors, learned over the parameter spaces of typical human poses and shapes to constrain the search to a range of values. These priors may be in the form of either physical joint limits or means and variances of typical human pose joint angles for various activities. Recently a variety of works have also attempted to learn efficient biomechanical priors on human motion, based on joint torques and mechanics.

Efficient techniques such as partitioned sampling has been used in the past to reduce the search space for an articulated kinematic models. These algorithms perform hierarchical search and independently localize the parts of the articulated structure. A variety of MCMC based sampling (importance sampling, layered sampling and Gibbs sampler) methods have been employed to reduce the number of particles required for efficient representation and tracking of the posterior density over the 3d state parameters.

The hypotheses generated in the prediction step are evaluated using the observation likelihood function. Evaluating a hypothesis amounts to calculating image evidence of the synthetic 3D model and entails modeling of the 2D image generation process. Construction of an effective likelihood function involves realistic human body modeling - both 3D shape and kinematic chain models, the camera projection model for rendering the 3D human model to 2D image plane and an efficient image matching function to compute the likelihood cost. Image descriptors such as edges, intensities or silhouette shape may be used to match the rendered 2D configurations to the observed 2D image.

Observation likelihood function therefore forms a key component of the generative modeling and determines the probability of the pose parameters by modeling the image generation process. The accuracy of pose estimation therefore critically depends on how accurately the perspective projection is modeled and whether the image features extracted from the synthetically rendered 2D images generalize to the observed images.

Estimating pose parameters using generative model therefore involves inferencing posterior

distribution over model parameters using the realistic 3D human body model to measure the likelihood of an image and the prior distribution over the body pose parameters. The prior distribution may be obtained as temporal prior (based on learned dynamics of the motion) or as typical shape and appearance parameters (learned from labeled images).

Despite efficient inference algorithms and better extrapolation ability to unseen data, generative algorithms occasionally fail due to specific observation likelihood, bad initialization and inaccurate human 3d shape modeling. Most of the generative frameworks require to be initialized manually. High dimensionality of the parameter search space makes estimation of the optimal pose computationally expensive if the search is initialized from a completely different pose. Generative tracking methods (like Condensation, Kalman Filtering) are usually difficult to recover from error.

1.2.2 Bottom-up Modeling

Bottom-up (discriminative) methods, on the other hand provide a complementary approach for human pose estimation problem and attempt to learn the inverse of the perspective projection. Bottom-up methods learn statistical models to directly predict 3D poses from the descriptor vectors extracted from the observed 2d images. The discriminative models may have an additional intermediate step of first inferring mid level features , such as coherent body part structures, from the low level image signals and using these to infer the 3D poses. These coherent structures may either correspond to different parts of the human in which case, a weak human body model may be used to constrain these parts to form a valid pose configurations. The human body model may be specified as a set of spatial constraints between different body parts and unlike generative models, may not have precise shapes of actual human body parts. Most of these models are based on pictorial structures [70] and trains separate classifiers to detect various human body parts in the image. On the other hand, the low level image descriptors can be used to directly predict the 3D pose states. This is achieved by learning function approximators(regression) that maps points in the 2D image descriptor space to the 3D pose space. Discriminative methods (also referred to as predictive models) treat 3D human pose as a point in high dimensional joint angle space. For a novel input 2D image, it estimates the 3D pose by interpolating between 3D poses corresponding to the similar 2D exemplars already seen during

the training phase. The interpolant may be based on Nearest Neighbor regression or more compact, non-linear parametric regression. The ability of discriminative models to estimate pose for any observed image therefore depends on the amount of training data used. As long as the training exemplars are sufficiently representative of the test dataset, predictive models almost always outperforms the generative models.

The problem of learning mappings from 2D image features to 3D poses is ill-defined and ambiguous. The mapping is one-to-many as several distant 3D poses may render similar 2D images. For instance, 2D projections of a person walking towards the camera and walking away from the camera appear similar to each other. The predictive models therefore uses multiple mapping functions to predict plausible 3D poses for a given 2D visual input.

Furthermore, to reduce the amount of data required for training the discriminative models and take into account the strong correlations between joints, these mappings are typically learned between the input features and low dimensional, compact representations of the 3D human pose. The low dimensional representations are learned by taking into account the correlations between various joints and identifying perceptually meaningful structure in them.

Discriminative modeling may outwardly seem as an over simplistic approach, requiring a labeled training set containing pairs of 2D images and the corresponding 3D pose, and a statistical learning model to learn multi-valued mapping from image descriptor space to 3D pose space. However, it faces a number of challenges primarily due to scarcity of labeled training data for learning a model that can be generalized to varied environment and lack of robust image descriptors that are less affected by perturbations caused by background clutter, viewpoint and lighting changes in the scene. Furthermore, unlike generative methods that are intrinsically regularized, these methods depend on the choice of learning model and the learning algorithm to avoid overfitting and learn models than can be generalized to unseen test dataset.

1.2.3 Combined Approaches

Addressing the deficiencies in the two categories of human 3D pose modeling techniques, many works in the past have also attempted to overcome their drawbacks by integrating these approaches in a common framework that combines their strengths. The two realms of frameworks can be loosely combined where the discriminative framework is used to bootstrap the

generative model. The discriminative models can be used to initialize the search in generative models by providing accurate approximations for the posterior distribution over 3D pose states. The conditional distribution learned in discriminative frameworks provide a useful proposal distribution for sampling plausible poses, whereas generative framework provides a feedback mechanism of re-projecting the reconstructed 3D pose to the 2D image plane and evaluating it according to the specific criteria (matching features based on edges, silhouette shapes etc.). Both discriminative and generative models may be trained independently.

Another way of combining cues from discriminative and generative approaches is by incorporating additional semantic information from the parts detectors of human body. For example, in the framework proposed by Sigal *et.* al[154], the parts detectors provide bottom-up information to assist particle filter based tracking of human 3D pose. The parts detectors enable automatic initialization of the generative model and also assist recovery from occasional tracking failure. The 3D human model is organized as a graphical model where each node represents the limbs and the connection between the nodes encodes the spatial compatibility between pairs of part configurations. The bottom-up cues are incorporated in a similar fashion, by importance sampling from the conditionals of the part detectors, and using it in the non-parametric belief propagation framework to infer the 3D body pose. Although similar to above approach, this uses low level image features to first estimate part components of the human body and then using these intermediate level detections in the graphical model framework for estimating the 3D pose. This approach is more robust to occlusion and local deformations although part detectors are often too noisy and occasionally have spurious detections in the image.

More strongly coupled approaches may learn a joint model that has both discriminative and generative sub-components[165]. Learning the model parameters alternates self-training stages in order to maximize the probability of the observed evidence (images of humans). During one step, the predictive model is trained to invert the generative model using samples drawn from it. In the next step, the generative model is trained to have a state distribution close to the one predicted by the discriminative model. At local equilibrium the two models have consistent, registered parameterizations. During the inference, the pose predictions are driven mostly by the fast discriminative model, but implicitly include generative feedback to ensure consistency.

1.3 Tracking in Low Dimensional Space

Inferring human pose from a single image is difficult due to inherent depth ambiguities and limb foreshortening effects. One way to counter these challenges is by enforcing temporal continuity(tracking) constraint on the 3D pose estimation. Based on the pose estimation results in the past frames, we may give higher probability to a pose that is similar to the pose estimated in the previous frames compared to other distant 3D poses. Due to this additional weight assigned to a set of poses, tracking can disambiguate between several plausible poses from a sequence of observations.

Typically, tracking is done by modeling dynamics of human motions. Humans exhibit complex dynamic behavior that is highly non-linear and non-uniform. A good dynamic model is sufficiently representative of all the possible pose state trajectories and how they evolve over time. Human motion lacks a clear structure and may be composed of multiple basic action units. Transition between these action segments is undefined in most of the cases. Within a category of motion, different subjects have widely varying diversity in styles. Recent research have developed improved dynamical models that have greater descriptive power whilst requiring only limited training exemplars.

Human skeleton has a highly articulated structure with the number of degrees of freedom ranging from 30 to 60, depending on the level of articulation details desired for a given application. More dimensions entail more training samples needed to accurately model the data distribution in the pose state space. In order to alleviate the curse of dimensionality, most of the 3D pose tracking frameworks restrict the pose inference to low dimensional subspace. These techniques learn a low-dimensional representation of 3D human pose by discovering intrinsic dimensionality of the joint angles space. For instance, joint angles of a human running or walking has the intrinsic dimensionality of 1, the phase of the running and walking cycle. This is due to the fact that in typical human activities, the joint angles are strongly correlated, and values of most of the joint angles can be directly inferred from values of only a few key joints of the skeleton, which in most cases, are much lower than the total number of degrees of freedom. However, developing effective techniques for modeling complex nonlinearities of the human pose state space, using only a minimal set of low dimensional, latent parameters

is a challenging task. A variety of methods that attempt to learn these low dimensional, perceptual representations exist in literature and have been used for improving the computational complexity of tracking.

1.4 Overview of Our Approach

Past research on human motion analysis have mainly focused on top-down techniques for 3D human pose estimation and tracking. Most of these techniques often lack generalization over a wide range of human poses, appearances, shapes and backgrounds. A key component of generative framework is the likelihood function which is complex and difficult to model as it requires modeling large variability in pose, shape and appearance of the humans in real scenes. Furthermore, non-linear dynamics, 2D-3D ambiguities, self-occlusion and kinematic singularities may cause the posterior to be multi-modal over the pose state parameters. The multimodal posterior distribution is non-parametrically represented using a discrete set of samples (particle filters) where each sample corresponds to a hypothesized 3D pose. For articulated human body, with high degrees of freedom, the number of samples required to accurately explore the high dimensional state space increases dramatically thereby leading to high inference cost for 3D pose.

Averting these limitations, we adopt discriminative pose estimation framework that provides a direct learning of the mapping from the 2D images to the corresponding 3D pose using a set of labeled exemplars. The widely held belief is that discriminative modeling always outperforms the generative modeling[125]. However, in the absence of sufficient labeled training data, generative models tend to perform better, although it approaches its higher asymptotic error fast, as more labeled data is added to the training set. Further, in the absence of sufficiently large training set, the accuracy of generative model depends on how closely the learned generative model approximates the actual data generation model. If the two models mismatch, the discriminative and generative frameworks tend to have similar accuracy.

In this thesis, we attempt to directly estimate the 3D human pose from the 2D image descriptors in real scenes. We make contributions to discriminative pose estimation framework in the following key research areas:

- Statistical learning algorithms for probabilistic modeling of multi-valued relations
- Probabilistic discriminative framework for conditional density propagation
- Image descriptors for balancing the invariance-selectivity tradeoff
- Semi-supervised learning framework for multi-valued functions
- Probabilistic framework for mapping high dimensional 3D pose state to low dimensional latent space

In the bottom-up approach, the 3D pose of human body is predicted directly from the image descriptors extracted from the visual observations. The image descriptors encode the shape and texture information of the human in the image. The discriminative model use these descriptors as the input to predict a vector of 3D joint angles that encodes 3D human pose configurations in the scene. The imaging sensors are not required to calibrated as the image descriptors are translation invariant. The statistical models used for prediction, are typically trained using a supervised learning framework, on a set of labeled exemplars (pairs of 2D image inputs and 3D joint angle output). Acquiring labeled exemplars is a costly procedure and requires 3D motion capture setup, in a well controlled laboratory settings, so that both the 2D image and the 3D pose can be captured synchronously and accurately. This is not only time consuming but also impractical to many application requiring 3d pose reconstruction as the real scenes rarely contain humans in clothing similar to special marker based costumes required for the 3D motion capture. In the proposed research, we therefore attempt to generate realistic training data by importing real motion capture sequences to a virtual computer graphic(CG) human model and use the rendered images with real backgrounds to train the statistical predictor model. We refer to this data as “quasi-Real” data (see fig. 1.4). Furthermore, we also develop techniques to efficiently restrict the 3D pose inference to a latent, low dimensional, perceptual representation of the original 3D joint angle space.

Our human representation is based on an articulated skeleton with spherical joints that has 56 degrees of freedom, including the global rotation. To gather data, we use Autodesk Maya [2], with realistically rendered computer graphics human 3D surface models. Choice of image descriptors depend on whether the background model in the scene can be learned or not. For

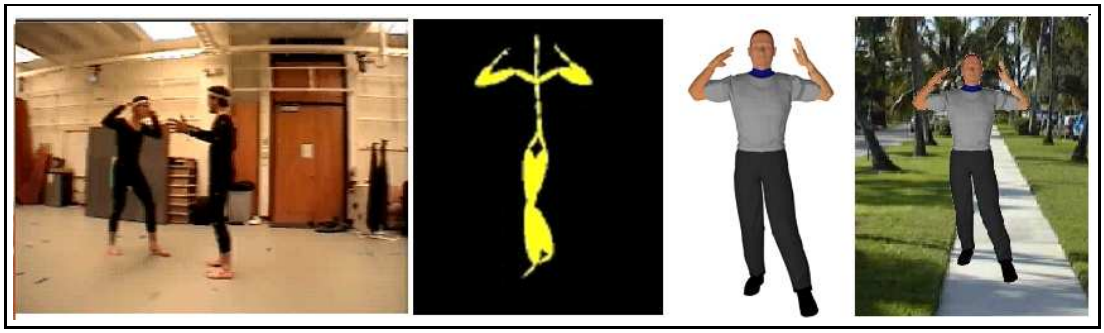


Figure 1.4: Various pre-processing stages of generating training and testing data in our framework. The 3D motion capture sequence from [1] is available in 3D joint angles format (62 degrees of freedom). The motion capture data is imported to a computer graphic(CG) model of standard anthropometry using Maya software package. Realistic training/testing (quasi-Real) data is obtained by rendering the 3D model on a real background image

images captured from a static camera, humans can be easily localized and delineated using background subtraction. The silhouettes obtained from background subtraction are used for extracting the shape descriptors, which is then used as inputs for human 3d pose estimation. For a moving camera, humans are localized as bounding boxes using a classification based human detector[53]. Since the humans are not clearly delineated in the bounding box, shape cues cannot be used in pose estimation(see fig. 1.5). We therefore use texture/appearance based image representations as the input in our discriminative framework. The labeled exemplars are used for training a regression framework, that maps 2D image descriptor vector to a 3D pose vector. In order to resolve pose ambiguities due to perspective projection of 3D humans to 2D images and self-occlusion, we train multiple regression functions to predict multiple plausible 3D poses for a given image observation. We use Mixture of Experts (ME) framework to train a set of experts (regressor). One of the issue in training a discriminative is to avoid overfitting and learn models that can generalize well to an unseen test data. Generative models are implictly regularized as inference assumes a prior human body model, that is used to generate various pose hypotheses to explain the observations. The optimization is therefore constrained over a given model space. For predictive models however, we need an explicit mechanism to avoid overfitting in order to generalize them over a larger test dataset. Therefore, we train Mixture of Expert(ME) model using Bayesian learning paradigm that intrinsically embodies regularization and model selection using Occam's razor. However exact Bayesian inference is intractable and certain approximations are needed to make the computation feasible. The key advantage of

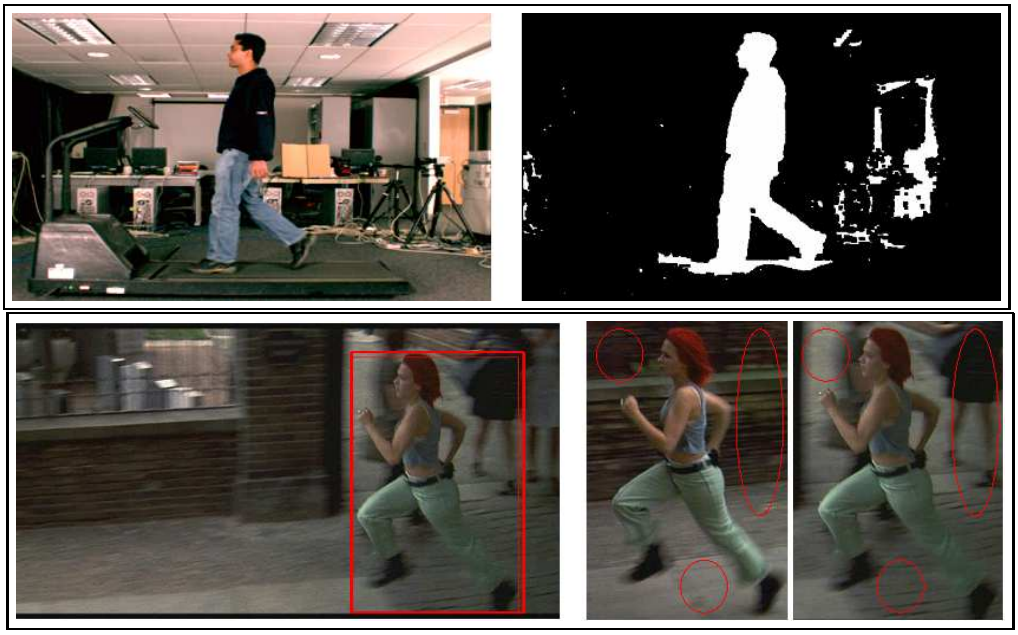


Figure 1.5: Two standard ways of localizing humans in the image: *(Top row)* For a static camera, background subtraction can be used to extract approximate silhouettes of the humans in the image. We use non-parametric background subtraction[64] for obtaining foreground regions. A key challenge to this approach is to identify and eliminate regions due to shadows, that tend to distort the shape of the silhouette and may distort the image descriptor used for 3d pose estimation. *(Bottom row)* Background subtraction cannot be used when the camera is undergoing motion. In such a setting, a classification based human detector can be used to search for humans in the image. Typically, this is done over multiple scales of rectangular bounding box, in order to detect humans at different distance from the camera. In such settings, the extraneous regions in the detected bounding box due to the background, may cause perturbations in the descriptor and lead to inaccurate 3D pose estimation results

Bayesian learning is that it learns models that are much sparser compared to other predictive learning techniques, by selecting only relevant basis functions in the learned mapping.

At each time step of 3D pose estimation from a monocular sequence, we also enforce temporal continuity constraint to assign higher probabilistic weights to pose that are similar to the pose estimated in the previous time step. This not only smoothes the predicted state trajectory but also disambiguates poses where several possible reconstructions are possible. Tracking entails learning a human motion model that predicts the pose in the next frame using the pose estimated in the current frame. Dynamic motion models can be learned from the labeled exemplars from various activity sequences.

One of the shortcoming of discriminative learning methodology is that it should be trained

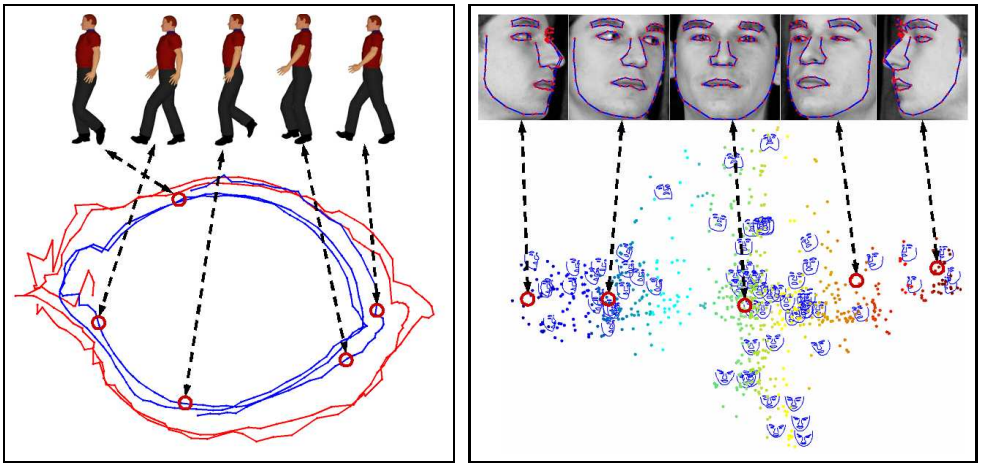


Figure 1.6: Raw sensory signals such as image pixels or joint angle vectors are high dimensional manifestation of lower dimensional and perceptually meaningful variables in the latent space. *(Left)* shows the walking cycle as 2D embedded points in a high dimensional ($\approx$ 60 DOF) joint angles space. *(Right)* 2D embedded space of shapes of the facial features as the head undergoes large rotations. Shape is represented as 150 dimensional vector of co-ordinates of landmark points and lie on a highly non-linear Riemannian manifold. However space of shapes due to the movement of the head is a 2 dimensional embedded space where each variable denote the yaw and pitch of the head.

on sufficiently rich set of labeled examples, in order for it to generalize well on realistic scenarios. One way of attacking this problem is to use unlabeled data to improve the learning of the models. The unlabeled data refers to input images for which 3D ground truth pose is not available. One of the way of incorporating information embedded in the input data is by enforcing the functions to vary smoothly along the instrinsic structure of the inputs. This effectively means that for the input data points that are close to each other on the manifold, their outputs should also be close to each other. This additional constraint can be used to further restrict the parameter space of the optimal pose, thus enabling learning of more accurate discriminative models. Given the high degree of freedoms of the articulated human body, the pose estimation algorithms should be able to effectively search in high dimensional state space and output most optimal 3d pose. Repetitive activities like walking, running and jogging have much lower perceived dimensionality than what is observed from the image pixels or the joint angles. In order to make the pose prediction scalable and support multiple activities, it is natural to reduce the dimensionality by learning representations that reflect perceptual structure of various activities. Recent work on visual tracking has identified the importance of low-dimensional models with intuitive geometric properties and mappings between the latent (low dimensional) and the

ambient (original observed) space. These models use the fact that in various human activities, there exist high degree of correlation between different joints. The low-dimensional representations are obtained by projecting the data to the decorrelated axes obtained using these models. We use Sparse Spectral Latent Variable Models (SLVM) that combine the advantages of spectral embeddings with the parametric latent variable models. SLVM learn stable latent spaces and preserve global or local geometric properties of the modeled data (see fig. 1.6). This offers low-dimensional generative models with probabilistic, bi-directional mappings between latent and original spaces. The learned bi-directional mappings between the latent and the ambient spaces are probabilistically consistent and can be used for efficient visual tracking of human pose in low dimensional space.

1.5 Thesis Contribution

Learning mappings from 2D images to 3D human pose is an ill-conditioned problem and subject to a number of open challenges namely 2D-3D ambiguities, large within the same pose class variance due to shape and appearance, high degree of articulation of the human body and lack of labeled data for training. In the thesis we have targeted these problems and aimed at developing novel techniques for enhancing human 3d pose estimation in real scenes. Following are the main contributions of the thesis:

Bayesian models for learning multi-valued functions: We develop sparse Bayesian learning framework for training mixture of experts model. Sparse Bayesian learning generates compact and well-regularized predictive models. Mixture of experts framework essentially learns multiple regression functions to map an input to multiple plausible outputs. In addition, there is a ranking function, that estimates the competency of the regressors to predict for the input. The ranking weights(referred to as gates) are themselves observation dependent functions. Bayesian formulation provides a principled way of quantifying the complexity of the models and promote simpler models by penalizing over-parameterized models.

Discriminative 3D human pose tracking: We propose a discriminative framework for the propagation of multi-modal (Mixture of Gaussian) distributions using probabilistic, continuous, temporal chained models. Tracking of 3D pose is useful for an image sequence where

prior estimates of the pose may be used to resolve pose ambiguities in the current frame. The discriminative tracking algorithm, unlike generative methods, does not require modeling of complex likelihood function, that is both computationally expensive and indirect. Rather, the distribution of 3D pose dynamics is directly learned from the sequence of labeled exemplars.

Image descriptors with improved stability to geometric transformations and background clutter: We use multi-level(hierarchical) encodings in our regression framework and compare several of these descriptors for the problem of human 3d pose estimation from monocular sequence. Hierarchical descriptors are coarse-to-fine representations that encode image at multiple levels of semantic information content and invariance to perturbations due to geometric transformation, viewpoint and illumination changes. Further, we preprocess the extracted descriptors using metric learning and canonical correlation analysis to make them robust to changes in the background.

Semi-supervised learning based on Manifold Regularization: In order to make training with diverse, real-world datasets possible, we learn models using both labeled and unlabeled data. We incorporate the unlabeled data using a semi-supervised learning framework based on manifold regularization. In order to use it to learn multi-valued mappings, we generalize semi-supervised learning to mixture of experts model.

Sparse spectral latent variable models: We propose non-linear generative models, referred to as Sparse Spectral Latent Variable Models (SLVM), that combine the advantages of spectral embeddings and parametric latent variable models for learning stable, latent representation of high dimensional data. The latent spaces learned using SLVM, preserve global or local geometric properties of the modeled data. Our model is efficient to learn and probabilistically consistent. Furthermore, the learned bi-directional mapping allow us to map any out-of-sample points in the observation space to the latent space and back.

1.6 Thesis Outline

Introduction This chapter introduces the thesis, giving overview of the human pose estimation problem and the motivation behind it. We provide a brief background on the problem and the potential applications of this area. Further we list out the main challenges and open problems

that are encountered in this research area. We also present possible ways of solving these problem and how we address them in this thesis.

State of the Art In the next chapter we describe recent research work in the field of human 3D pose estimation. A number of works adopt different approaches for solving the same problem(e.g. generative framework) that may perform better in certain scenarios and worse in other. It is useful to understand pros and cons of these techniques and how the proposed framework overcomes the deficiencies existing in them. In addition, we briefly discuss works in the research areas that are not directly related to human pose estimation problem but form an integral component in any 3D human pose reconstruction framework e.g. human detectors and image descriptors.

Bayesian Mixture of Experts Chapter 3 introduces the machine learning models that form the core component of our discriminative learning framework. Mixture of experts is a non-linear, supervised learning framework that learns one-to-many mapping by dividing the input space into multiple sub-domains and fitting regression surfaces in them. These data sub-domains have soft boundaries that is learned as a probabilistic multi-category classifier. Key motivation behind training the Mixture of Experts model using Bayesian framework is that the models trained using maximum-likelihood(ML) learning has tendency to overfit the training data. Sparse Bayesian learning automatically regularize the models and learn simpler models that can generalize well to unseen test data. The content of this chapter is based on the work published in *Discriminative Density Propagation for 3D Human Motion Estimation, CVPR 2005*

Discriminative 3D Human Pose Estimation In chapter 4, we apply the Bayesian mixture of experts model to learn one-to-many mappings between the 2D observation and the 3D human pose. We assume a static camera so that background pixel modeling can be used for extracting silhouettes of the moving human targets. We develop a discriminative tracking technique for propagating multi-modal state condtional distribution across time using continuous temporal chain model. We represent the multi-modal distributions as Gaussian mixture model that is pruned at each time step to avoid exponential increase in the number of components. In addition, we also investigate the techniques for improving the computational efficiency by reducing the dimensions of the input feature space and output joint angle space using Kernel PCA. The content of this chapter is based on the research work published in BM^3E *: Discriminative*

Density Propagation for Visual Tracking, Transactions on PAMI 2007

Hierarchical Models for 3D Human Pose Inference Chapter 5 extends the human pose estimation framework to realistic scenarios where the camera is undergoing motion. The motion of background pixels induced by the camera motion, make the background subtraction inapplicable to such scenarios. We therefore use classification based human detectors to localize humans in the scene and use region based image descriptors to learn efficient models for 3D pose estimation. However, bounding box introduces additional challenges to the pose estimation problem due to background clutter and misalignment. We use hierarchical image descriptors that are more robust to these perturbations and have some degree of invariance to the local deformations and misalignment of the bounding box. These descriptors are complemented with noise suppression mechanism using metric learning techniques based on Canonical Correlation Analysis and Relevant Component Analysis. These refine and further align the image descriptors to minimize within pose class invariance in order to better tolerate deformation, misalignment and clutter in the image. The content of this chapter is based on the work published in *Semi-supervised Hierarchical Models for 3D Human Pose Reconstruction, CVPR 2007*

Sparse Spectral Latent Variable Model In chapter 6, we introduce probabilistic latent variable models that combine the advantages of spectral embeddings and parametric latent variable models. These non-linear generative models, referred to as Sparse Spectral Latent Variable Models (Sparse SLVM), provide stable latent spaces that preserve global and local geometric properties of the observed data. The proposed framework is generic and can be used with any spectral embedding method. We demonstrate the Sparse SLVM model on human pose estimation problem where we restrict the pose estimation to the low-dimensional state space. In this chapter, we also look into the problem of fitting and tracking deformable 2D shapes. In contrast to rest of the thesis, this part of the chapter studies the effectiveness of the SLVM algorithm in modeling nonlinearities of the 2D shape space and learn low-dimensional latent spaces that preserve the geometric structure of the ambient space. We demonstrate the applicability of the framework to various vision problems. The contents of this chapter is based on the work published in *Spectral Latent Variable Models for Perceptual Inference, ICCV 2007*

Conclusions and Perspectives Chapter 7 concludes the thesis with the summary of the work

and a discussion on the results, giving suggestions on improving the results. Usually, any research work solves certain problems and creates many more interesting avenues for conducting further research. We provide the practical implications of the proposed framework and highlight some of the open problems in the field of human 3D motion modeling. We also provide some perspectives on the significance of the problem to the task of improving surveillance and graphics based applications.

1.7 Relevant Publications

The thesis is based on the work published in following conference publications:

- **Spectral Latent Variable Models for Perceptual Inference**, Atul Kanaujia, Cristian Sminchisescu, Dimitris N. Metaxas, *International Conference on Computer Vision 2007*
- **Semi-supervised Hierarchical Models for 3D Human Pose Reconstruction**, Atul Kanaujia, Cristian Sminchisescu, Dimitris N. Metaxas,*Conference on Computer Vision and Pattern Recognition 2007*
- **Learning Ambiguities Using Bayesian Mixture of Experts**, Atul Kanaujia, Dimitris Metaxas, *International Conference on Tools with Artificial Intelligence 2006*
- BM^3E **: Discriminative Density Propagation for Visual Tracking**, Cristian Sminchisescu, Atul Kanaujia, Dimitris Metaxas, *IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007*
- **Discriminative Density Propagation for 3D Human Motion Estimation**, Cristian Sminchisescu, Atul Kanaujia, Zhiguo Li, Dimitris N. Metaxas,*Conference on Computer Vision and Pattern Recognition 2005*
- **Conditional Visual Tracking in Kernel Space**, Cristian Sminchisescu, Atul Kanaujia, Zhiguo Li, Dimitris N. Metaxas,*Neural Information Processing Systems 2005*

Chapter 2

State of the Art

In this chapter we review some of the recently proposed, state of the art techniques for understanding and estimation of human 3D motion from monocular image sequences. Human pose estimation encompass several constituent sub-tasks that include human detection, graphics based human body modeling, robust image encoding, dimensionality reduction and statistical modeling of human motion dynamics. As discussed in the previous chapter, each of these problems is a research field in itself and a vast literature exist on several aspects of these areas. In the following sections we will review relevant works in each of these areas.

2.1 Human Detection

The foremost step in 3D human pose estimation is the detection and localization of humans in a scene. For a static camera with known background model, we can use background subtraction to identify moving regions in the scene. Background subtraction can cause many issues, due to shadows and multiple occluding objects, that are difficult to resolve. Recent techniques proposed for background subtraction include non-parametric kernel density estimation[64], temporal variation of intensity distributions[97], online auto-regressive modeling[119], modeling complex dependencies between the location and color of the image pixels[150] and intensity and texture modeling using GMM[178]. In the presence of camera motion conventional method for statistical background modeling cannot be used. Moving camera induces 2D motion to the pixels in the scene, making it impossible to learn intensity distributions of the background pixels and identify targets by background subtraction. The solution to this problem is to use discriminative patterns in the visual observations to distinguish targets from other static and dynamic objects in the environment. The problem of vision based target detection is considerably

more difficult for human targets due to variability in poses, clothing appearance, anthropometry and range of scales in which they may appear in the image. Detection by classification is perhaps the most popular approach that has been extensively investigated in the literature. This approach detects the targets by evaluating the likelihood of finding an object over a grid of bounding box, regularly placed on the image. Some of the recently proposed human detection techniques include [183, 54, 202, 189, 109, 11, 42].

Discriminative classification methods[11, 116, 189, 73, 144, 118, 129, 130, 54, 146, 183, 202] for classifiers are preferred over generative, parts based detectors [68, 108] as they are more accurate, assuming they are trained on sufficiently representative examples. Compared to generative models that attempt to model the distributions of the individual classes (humans and non-humans) and the underlying complex observational dependencies, discriminative classifiers aim to learn the optimal classification boundary separating the two classes. Among discriminative classifiers[54, 202], Support Vector Machine and Boosting[183, 116, 129, 189] have been extensively exploited for learning person detectors in the past, as they have the ability to select relevant descriptors in the high dimensional input feature space. Support Vector machines[41, 90] have been extensively applied for training binary classifiers as they generalize well to varied data set. Papageorgiou and Poggio [130] used Haar wavelets to extract the patterns of the targets and the non-targets, and trained support vector machines to classify them into two categories. This work was extended by [54] that used Histogram of Oriented Gradient features to learn the classifier. Mohan *et al.*[118] used an adaptive combination of classifiers (ACC) that are composed of distinct example-based component classifiers. Each of the component classifiers is a Support Vector Machine that are trained to detect object parts. Neural Networks have also been employed in learning the discriminative boundary between two classes[144]. Less conventional techniques include Sparse Network of Winnows (SNoW)[11] that uses a neural network based linear functions over the feature space. The Winnow learning algorithm is used to select only the relevant set of features for learning the classifier.

2.2 Human 3D Pose Modeling

As discussed in the previous chapter, there exist two main approaches for human pose estimation from video imagery - top-down, generative model and bottom-up, predictive models. A top-down methodology refers to using a high-level description of the complete human pose to explain lower-level image signals. This normally involves hypothesizing the 3D human pose and using a synthetic human body model to measure the likelihood of 2D human configuration in an image. Bottom-up methods, on the other hand, start with lower-level image features and use these to predict higher-level 3D pose information in the form of a pre-specified set of 3D human body parameters(either joint angles or joint centers). Both the methodologies require two critical components - accurate synthetic models for shape and pose of the articulated human body structure and robust algorithms for computing encodings that compactly represent the semantic contents of the image. In the following subsections we list state of the art techniques in these two areas.

2.2.1 Modeling Articulated Human Structure

Accurate representation of human body involves realistic modeling of its articulated skeletal structure and the 3D shape of the body parts. The shape, size and relative proportions of various body components should be in accordance with the standard anthropometric norms, in order to accurately model the variability in human shapes encountered in realistic scenarios. In addition, appearance of the body parts may be useful to get an accurate estimate of the matching cost when used in a top-down pose estimation framework. The accuracy of top-down models for 3D pose estimation critically depends on how accurate is the image generation process using the synthetic human model. Depending on the targeted application(animation or human detection) human models with different levels of details may be used.

For human detector applications, that identify humans in the image by detecting different body parts and recognizing coherent structure between them, a detailed 3D human pose reconstruction may not be required and a coarse 2D parts based human models may usually suffice.3D human modeling is more suitable for applications that are processing image stream with human targets of sufficient resolution (> 120 pixels height) and that require more detailed

behavioral analysis such as detecting abnormal walking style or suspicious activity and detecting concealed objects or anomalous shapes. In such scenarios, detailed visual changes such as the effect of limb foreshortening due to changes in the depth, should be adequately modeled in order accurately infer 3D poses.

3D human model representation of the humans has a number advantages over 2D human pose estimation such as modeling self occlusion, body parts foreshortening and 3D shape deformation in a principled fashion. However it is in general a more difficult and an ill-conditioned problem. Especially in a single camera scenario where lack of observability may cause ambiguities and multiple 3D configuration may generate similar appearing 2D observations.

A variety of 3D and 2D human models have been used in the literature: Kinematic 3D Human model [4, 40, 46, 74, 96, 185, 164], Scaled Prismatic 2D Model[4, 59, 132] and part-based 3D human model [57, 93, 106, 110]. Models based on kinematic tree are by far the most widely used 3D model. Typical 3D human model are composed of a kinematic tree based skeletal structure, with bones represented as rigid links. The links are connected to each other via joints of varying degrees of freedom(1-3) in a hierarchical tree structure. The root joint of the skeleton defines the global reference of the skeleton with rotation and translation values in the world co-ordinate frame. The orientation and locations of rest of the joints and link segments are defined in the reference frame of the parent node of the skeletal hierarchy. The 3D motion data is imported to the skeleton as angles for each of the degree of freedom of the skeletal segments. The joint angles may be parameterized as Euler angles, Quaternions or Exponential maps. Each segment is transformed using a global transformation in the world coordinate frame and local transformation in the reference frame relative to the parent node joint.

One of the requirement of generative modeling is to accurately parametrize the human body 3D shape and to simulate the image generation process (under the assumption of known camera parameters and projection model) that resulted in the observed image. It is therefore critical to estimate the shape of human body parts in order to accurately model this process. Conventional shape models were based on relatively simple 3D geometric structures such as ellipsoids[47, 115, 40], cylinders[51, 58, 137, 152, 154, 141], tapered cones[191, 106, 157, 3], super-quadrics[170, 75, 163] and other geometric primitives[63, 55, 75]. Most of these models

are crude representations of human shape and do not accurately represent the variability of shapes in different poses. Instead of coarsely representing the 3D shape of the human body parts using simple 3D shapes (cylinders, cuboid or sphere), a more detailed 3D mesh surface, composed of interconnected polygons can be used.

Recently, more accurate shape models based on 3D laser scans have been employed to learn 3D shape model of humans[19, 159, 17]. Sigal et al.[159] used SCAPE (Shape Completion and Animation of PEople) [14] model to represent articulated and non-rigid deformation of the human body under pose and anthropometric variations. The rich and natural range of body shape variations are captured as low dimensional linear subspace learned using Principal Component Analysis(PCA). The 3D mesh based shape model consisted of 25,000 polygons and provide a detailed representation of human shapes for improved likelihood modeling. Gall et. al[86, 85, 80] recently proposed a model-based approach for 3D human shape estimation, that used a multi-layer framework of global and local optimization, to fit a detailed 3D human shape model to the image and marker positions obtained from motion capture system. However it did not use any prior knowledge of the dynamics to estimate the pose and shape of the target. The framework used a 3D human shape model of 5000 triangles. In order to estimate the 3D shape of a human in a generic pose configuration, the 3D shape surface needs to deform under the influence of the rotation and translation of the joints. This is achieved by associating different regions of 3D mesh to different skeletal joints (a method known as skin binding) and estimating transformations using the transformation of the joints. Other works that have employed detailed 3D mesh based shape models include [43, 37, 94].

2.2.2 Image Descriptors

Image descriptors are compact encodings of the shape and appearance of the objects in the image and forms an intermediate step towards inferring high level semantic details from low level image pixels. Generative framework for human 3D pose estimation requires modeling of the complex likelihood function to probabilistically rank various hypothesized poses by matching them against the observed image. This involves realistic rendering of 3D human model to 2D image plane and using an appropriate *image features* and distance function to estimate similarity (or dissimilarity) to the observed image. Typical image features are based on

shape and appearance that can be efficiently extracted and has a smooth matching cost function. Common features include silhouettes[121, 19, 4, 111, 65, 76, 138, 58], contours[72, 140] and edges[157, 168, 121, 154, 106]. Appearance features based on skin color pixels have been employed as well in human pose estimation[106, 107]. For shape based features, chamfer distance transform has been widely used for matching silhouettes[159]

Bottom up approaches, on the other hand, directly predict 3D human poses using image descriptors which are compact summarizations of the low level pixels in the image. These descriptors should be able to implicitly encode semantic information of various human body parts and their relative location with respect one another in a vector form so that they can be used to train exemplar based discriminative framework for estimating 3D human pose. Success of discriminative framework critically depends on these features and how invariant they are to geometric distortions, illumination and viewpoint changes. Image descriptors should be distinctive enough to differentiate between different poses, yet invariant to within the same pose class variance - people in similar stances, but differently proportioned, or photographed on different backgrounds.

The image descriptors can be categorized into two groups, based on the representation. The first group is the sparse representations that include the constellation model [42] and Implicit Shape models (ISM) [109]. Sparse feature representations are computed using a bag of features model [11, 116, 136, 109] over a set of interest points detected in the image. The interest points are detected using standard algorithms (e.g. Harris Corners, Maximally Stable Extremal Regions, Scale Invariant Feature Transform (SIFT), etc.) that encapsulate local image information and are assumed to be stable across viewpoint changes, illumination and scale variations.

Sparse image descriptors typically encode the semantic information of pose configuration of the humans in an observed image using various co-occurrence statistics of local patches. These local patches are either sampled randomly or are cluster centers of the codebook obtained by agglomerative clustering of patches obtained from the training exemplars. A codebook based on clustering local patches allows the descriptors to be robust to variations in shape and appearances of the image features around these interest points. These sparse feature representation

are therefore more robust to partial occlusion as loose spatial relation between the parts generally enforces weaker constraints on detection of all the parts in the image. More complex algorithms have also utilized the spatial relation between codebook parts, in addition to their shape and appearance.

In order to improve detection of targets that have discriminative parts, always occurring in fixed spatial locations relative to each other, many works[42, 11, 69, 73, 116] try to detect these coherent structures in the image. The encodings for the parts may be either local appearance (e.g. intensity histogram) or shape information (SIFT descriptors [112] or shape context [122]). A number of works [42, 68, 69] have also employed descriptors based on constellations of parts, that learn probabilistic representations of the various parts of the human targets. These representations aim to model the variations in shape, appearance and relative scaling of different parts of the human target. Different parts are automatically detected as local patch around the interest points or as a regions using entropy based methods like MSER (Maximal Stable Entropy Region). The spatial configuration is learned as a distribution on relative angles with respect to a particular landmark point(or region). More recently Leibe *et al.* [109] proposed Implicit Shape Models (ISM) for detecting pedestrians. ISM consists of a vocabulary of prototypical object parts' local appearances and a non-parametric spatial probability distribution that specifies where each entry of the codebook is likely to occur on the object.

The other category of descriptors is based on the dense representation. Unlike histogram based sparse descriptors, these descriptors are computed as a large vector, obtained by concatenating local patch features over a dense grid of points, typically on a bounding box enclosing the target[54, 183, 202, 189, 130, 145]. Similar to sparse representations, the local patch features encode shape and appearance information. However, strict spatial ordering of the dense feature representation makes them more discriminative albeit less invariant to the background clutter and partial occlusion. Examples of dense feature representations include Haar wavelets, covariance descriptors and Histogram of Oriented Gradients [54]. Viola *et al.* [189] used Haar features that encode intensity difference between two regions. Dalal *et al.* [54] used histogram of gradient orientations over a dense grid of fixed sized, overlapping blocks to encode the targets in the image. Zhu *et al.* [202] extended this approach by allowing variable sized blocks

in the descriptors. They also enhanced the computation speed of Histogram of Oriented Gradients (HOG) features such that they could be computed using integral images. More recently, [145] used mid-level edge information (called shapelet features) for densely encoding low-level gradient information to discriminate humans from non-humans. The mid-level information is obtained by automatically selecting a subset of salient gradient features that are relevant for detecting the target.

It is apparent that these two categories of image descriptors lie at the extremes of the range of descriptors that vary in discriminative power(selectivity) and their invariance to background clutter, geometric deformations and viewpoint changes. Clearly, there exist a tradeoff between the selectivity and invariance of the image descriptors, and it is not sufficient to encode an image at any single level of this tradeoff. More recently, hierarchical image descriptors [12, 95, 147] were proposed to efficiently represent image at multiple levels of abstraction and better tolerate the intra-pose class variability. Multilevel encodings are in general more stable and invariant to geometric transformations, local deformations, clutter and misalignments in the training and test set, as the image is encoded at several levels of abstraction, with the higher levels being semantically more informative and lower levels being more discriminative. Studies in object recognition [147, 103, 127, 9] have also demonstrated the effectiveness of multilevel image encodings by achieving substantial performance gains for the tasks of image classification and retrieval.

2.3 Human Pose and Shape Estimation

In the literature, there exist two main approaches for human pose estimation from video imagery - top-down, generative models and bottom-up, predictive models. A top-down methodology refers to using a synthetic human body model (with hypothesized poses and shape based on standard anthropometry) to explain the pose configurations of humans in the image. This involves rendering the model to 2D image plane using appropriate camera modeling(orthographic or fully perspective) and measuring the likelihood of the observation. Bottom-up methods, on the other hand, use less abstract, lower-level image signals to directly generate 3D pose hypotheses. They learn regression functions to predict 3D pose of the human targets using the

image descriptors as inputs.

Past research on human motion analysis has mainly focused on top-down modeling techniques which search in high dimensional space for the optimal articulated poses that best explains the observed human configuration in the image. Only recently, predictive models have been used[138, 5, 164] to directly estimate human pose from low-level image descriptors.

Research areas in top-down (generative modeling) human pose reconstruction involve techniques like modeling 3D human body pose and shape, accurate modeling of observational cost using robust visual cues and modeling temporal dynamics of human motion. Whereas research in discriminative human pose reconstruction relates to areas like statistical models for learning multi-valued mappings, design of robust image descriptors, correlation analysis for dimensionality reduction and semi-supervised learning.

2.3.1 Generative Algorithms

Generative algorithms attempt to model the joint distribution over the input and the output points, and estimates the state conditional using Bayes' rule. Generative modeling requires construction of an accurate observation likelihood or the cost function which essentially models the probability of the observation conditioned on the hypothesized state of the output. Under a uniform state prior distribution, pose estimation involves complex sampling or non-linear optimization methods for inferring the peaks of the likelihood function. If a prior model on the states is available, the optimization is done on the posterior distribution to predict the output states as the MAP (maximum aposterior) estimates. The prior model essentially constrains the search in the parameter space of the output state and yields more plausible estimates of the output. A variety of research works in the past have proposed algorithms for accurate modeling of the state prior [39, 82, 57, 154, 161, 185, 88] or improved features (based on silhouette, edge distance transform, texture or natural image statistics) for accurate modeling of the observation likelihood[151, 141, 172]. Most of the generative probabilistic tracking frameworks are based on propagating a single Gaussian distribution (Kalman filtering) or mixture of Gaussians propagation using particle filters [83, 57, 48, 169, 170, 173, 154]. Although generative models can flexibly reconstruct complex human motions and implicitly handle problem constraints, the inferencing may be expensive due to computationally expensive observation

likelihood step that involves rendering a 2d image from a hypothesized 3d pose and matching it to the observed image. Inaccurate observation likelihood may occasionally lead to uncertain inferences[57, 151, 170, 163].

2.3.2 Discriminative Algorithms

The drawbacks in generative framework motivates the complementary approach of discriminative modeling [138, 121, 148, 182, 10, 65], that attempts to directly predict state distributions from the observed image features. Several methods exist for discriminative pose prediction [138, 9, 164, 165, 166, 156] and are primarily targeted towards accurately modeling of 3d pose ambiguity in the data and developing robust image encodings to resolve these ambiguities while preserving their discriminative power. Discriminative approaches is not without its own difficulties: background clutter, occlusion and depth ambiguities make the observations-to-state mapping multi-valued and not amenable to simple functional prediction. Although the mapping from 2d images to 3d pose is multi-valued, several authors demonstrated good practical performance using single hypothesis methods[148, 121, 182, 10, 65]. The discriminative methods differ in their organization of the training set and in the prediction on the unseen test dataset: some construct data structures for fast nearest-neighbor retrieval [148, 182, 121], others learn robust regression models [10, 65]. Inference involves fast computation of the index to the best matching 3D pose[148], affine matching of projection of the 3d joint centers[121].

A number of authors have also pursued the problem of learning the mapping from 2d image space to 3d pose space using multiple functions. Amongst these, Rosales & Sclaroff [138] take a notably different approach, by accurately modeling the joint distribution using a mixture of perceptrons. A related method proposed by Grauman *et al* [76], who model the joint distribution of a 3d pose and multiple silhouettes obtained from multiple viewpoints, using a mixture of probabilistic PCA[180]. More recent work by Agarwal & Triggs [7] use multiple functions, learned in a clusterwise regression framework, to map 2d observations to 3d poses. Clusterwise regression[56, 138] attempts to model joint distribution over the inputs and the outputs and assigns fixed prior weights to each of the mapping function. In contrast, the recent proposed implementation of multi-valued mappings[167, 165] uses direct modeling of the conditional state distribution using multiple functions, the weights of which input dependent(Mixture of

Experts model).

2.3.3 Combined Approach

A variety of approaches have also tried to combine the generative and discriminative learning methods[138, 159, 165, 19, 154]. The two realms of frameworks can be loosely combined by allowing the discriminative framework to bootstrap the generative model[138, 159]. The discriminative models can be used to initialize the parameter search in generative models. Pose prediction in generative framework is typically done using MAP (Maximum A Posterior) estimate of the posterior distribution over pose states. The multimodal posterior distribution is represented as particles (or samples) from a proposal distribution that approximates the posterior. The probabilistic mapping functions learned using discriminative learning provides an accurate proposal map for sampling plausible poses.

However, instead of sampling entire articulated structure at the same time, it is also possible to learn individual part detectors and using them to infer semantically higher level pose information from the image. More recent works based on parts based human models[99, 154, 157, 3, 19, 106, 190] employ importance samples, as bottom-up cues, from the conditionals of the part detectors, to initialize and generate pose hypotheses for the top-down framework. Parts based models have been widely used for 2D detection of articulated human body parts and are generally based on pictorial structures[70, 67].

2.4 Statistical Learning in Discriminative Framework

Learning in discriminative framework is mostly supervised and exemplar based, where a set of labeled 2D images and corresponding 3D human pose are used to learn interpolants (linear or kernel) from the image descriptors space to joint angle space. Discriminative (feed-forward) predictors offer the promise of speed, full automation and complete flexibility in selecting the image descriptor but have to model multi-valued image-to-3D relations. This is a core issue in any predictive framework as the mapping from image to 3D pose is one to many and several 3D poses may get projected to similar appearing 2D images.

A second difficulty for reliable pose prediction is the design of image descriptors that are

distinctive enough to differentiate between various poses, yet invariant to within the same pose class variance - people in similar stances, but differently proportioned, or photographed on different backgrounds.

Finally due to reliance of discriminative modeling on the training set, generalization to very different poses, body proportions, or scenes where people are filmed against background clutter, may be problematic. The construction of realistic pose labeled human databases (images of humans and their 3D poses) is inherently difficult because no existing system can provide accurate 3D ground truth for humans in real-world, non-instrumented scenes. Current solutions rely either on motion acquisition systems like Vicon, but these operate in engineered environments, where subjects wear special costumes and markers and the background is simplified, or on quasi-synthetic databases, generated by CG characters, animated using motion capture, and placed on real image backgrounds.

In the following, we discuss some of the state-of-the-art techniques addressing these challenges.

2.4.1 Learning Multi-Valued Relation

Multi-valued mapping is typically handled using multiple regression functions that map 2D observations to multiple distant 3D poses. The final prediction is obtained using some form of ranking mechanism for these function, based on the likelihood of the observation. Rosales & Sclaroff [138] made an initial effort to learn this multi-valued relation using a set of multi-layer perceptrons(MLP). Mixture of experts provides a useful framework to learn multi-valued relation [87, 92, 194, 30] and is composed of a set of experts that are local, contextual function approximators. This model extends clusterwise regression [56, 138] which is maximum likelihood methodology for simultaneously clustering a dataset into multiple clusters and fitting multiple regression functions to each of the clusters. As discussed earlier, clusterwise regressions lacks an accurate mechanism to assign weights to the functions for an unseen test data.

2.4.2 Stable Image Descriptors

Image descriptors are compact representations of semantic contents of an image and are used as inputs to the discriminative models. Descriptors can be made more robust to background clutter using distance metric learning and correlation analysis. Learning metric for clustering and image classification has been studied by [20, 198, 149] and essentially aim to learn a subspace in which the Euclidean distance between the projections of the descriptors corresponding to the same (equivalence) class is minimized. Most of these methods differ in their treatment of equivalence constraints and the optimization performed for maximizing the intra-class similarity. Some methods constrain the problem using only similar (image) instances, referred to as *Chunklets*, others build contrastive cost functions based on both similar and dissimilar class constraints, or learn projections that maximize mutual within-class correlation. Metric learning and correlation analysis can be useful for suppressing noise and discovering intrinsic, latent shared structure in the data. They are appropriate for our problem where image descriptors are affected by differences in the background and *within the same pose class* variations.

2.4.3 Semi-supervised Learning

In order to address the challenge of scarcity of datasets containing humans in varied environment and diverse poses, we may utilize both the labeled and unlabeled exemplars to train models using semi-supervised learning methods. There is substantial work in semi-supervised learning [44]. Literature on semi-supervised learning methods include Transductive learning ([188],[91]), Semi-supervised SVMs[27], Manifold Regularization[23], Co-training[35] and gradient based regularization [38]. Transductive SVM uses a joint optimization of the SVM objective function over the binary valued labels of the unlabeled exemplars. This is achieved by using inductive SVM to label the unlabeled dataset and iteratively switching the labels at each step of solving the SVM quadratic program. Semi-Supervised SVMs (S3VM) [27, 71] incorporate information from unlabeled data by including the minimum hinge-loss for the two choices of labels for each unlabeled example. This is solved as a mixed-integer program for linear SVMs.

A number of *Graph Based Approaches* have been proposed in the literature. In Zhu et al.

[203], unlabeled examples are labeled using transductive learning and are used to label the test examples using nearest neighbor method. In Chapelle et al. [45], test points are approximately represented as a linear combination of training and unlabeled points in the feature space induced by the kernel. In Co-training[35] algorithm, multiple weak learners are trained on labeled examples and used to label the unlabeled examples. These exemplars are used to train other weak learners. Recent work in human tracking [123] showed promising results when learning mixtures of joint human poses and silhouettes, based on Expectation Maximization applied to partially labeled data.

2.4.4 Large Scale Learning of Conditional Models

Potential success of discriminative models critically depends on the dataset used for training the models. It is therefore necessary to train the framework in all possible human poses in order to sufficiently generalize it to novel test cases. Due to high dimensionality of the pose state space, the amount of data needed to accurately learn the models also scales exponentially. A significant downside of existing conditional algorithms is their scalability. A number of works [148, 36, 186] aim at reducing the computational time and memory required to train large scale discriminative models. Shakhnarovich et. al.[148] used 1,775,000 2D image-3D pose pairs to train a nearest neighbor based framework for efficient hashing based lookup. Bo *et.* al[36] used fast conditional models, learned using forward feature selection and bound optimization to train multi-valued predictors with data set of the order of $100,000$. Urtasun *et.* al[186] an online sparse probabilistic regression scheme based on Gaussian processes is used for efficient inference of complex, high-dimensional, and multimodal mappings between the 2D image and the 3D pose space. They used training set of size 10^5 that contained both synthetic and real data exemplars.

2.5 Dimensionality Reduction

Human skeleton has highly articulated structure with typical number of joints in any skeletal representation ranging from 20 to 30, depending on the degree of accuracy desired. Different joints may have 1-3 Degrees of freedom thus generating $40 - 60$ dimensional vector as an

adequate representation for most of the human poses. Despite of high dimensionality of human pose, in most of the human motions (like walking, running, jumping *etc*.), various joint angles are strongly correlated. These correlations are also evident in their 2D observations obtained by perspective projection of these poses on the image plane. In order to exploit correlations among observations and among 3D pose state variables, we can learn perceptual embeddings in the low dimensional space that preserve the geometric properties and restrict visual inference to it.

Recent work on visual tracking has identified the importance of latent variable models for learning low-dimensional representation that preserve intuitive geometric properties in the observation space. More recent work in dimensionality reduction has focused on algorithms based on spectral manifold learning. Spectral methods can model intuitive local or global geometric constraints [142, 175, 21, 60, 195]. Learning of spectral embedding typically involves construction of a data affinity matrix based on the locally linear manifold assumption. The task of finding the embeddings reduces to an eigen-decomposition problem that can be solved in polynomial time and is local-optima free. Spectral methods however lack a probabilistic framework and cannot be used to map out-of-sample points to the learned space. Furthermore, there is also no clear way to map points from the latent space to the ambient space. More complex models have emerged that complement these spectral methods with mappings for out-of-sample points from observed space to the latent space[25].A variety of non-linear latent variable models exist *e.g*. mixtures of factor analyzers or PPCA [179]. Most of the methods can model complicated non-linear structure but, do not provide global latent coordinate systems or latent spaces which preserve local or global geometric properties of the data. Regular grid-based methods like the Generative Topographic Mapping(GTM) [31] are not appropriate for structured problems, where the latent space distribution is unlikely to be uniform and when the latent spaces are higher than 2-3d. Although GTM[31] is a powerful non-linear method, it cannot unfold many convoluted manifolds (*e.g*. spirals, rolls) due to its fixed, data independent embedding grid in latent space, and its tendency to getting stuck at local optima in training. Kernel Dependency Estimation(Kernel PCA + Pre Image learning)[18, 196] is another non-linear dimensionality reduction framework that is able to learn mapping between the ambient space and the latent space given a definition of kernel function that serve as a similarity measure on the input or

output space. It suffers the same drawback and fails to preserve geometric structure across the ambient space and latent space. The Gaussian Process Latent Variable Model(GPLVM) [100] is a non-linear PCA technique based on a zero mean unit Gaussian prior in latent space and a Gaussian Process mapping to ambient(data) space. It is a competitive model primarily targeting data reconstruction error, but not designed to enforce the constraints to preserve geometric structure in the latent space. The latent prior is data independent and geometric properties of ambient data are not explicitly preserved. Memisevic [114] models the joint density using a separable product of non-parametric kernel density estimates and computes an embedding by optimizing a mutual information criterion over latent space coordinates, similar in spirit to GPLVM. There exist a number of works that have employed these models for human 3d pose inference. Elgammal & Lee [65] proposed a framework to directly infer 3D body pose from the human silhouettes by first projecting the visual inputs to low dimensional activity manifold and use a Radial Basis Function(RBF) network to predict the 3D pose from it. However their framework is not probabilistic and also there are no mappings from the observed space to the latent space. Sminchisescu & Jepson [161] use spectral embeddings (*e.g*. Laplacian Eigenmaps) to learn low-dimensional generative models that are consistent during inference. They build a continuous generative model for 3d human pose inference in the reduced space and learn RBF mappings from the latent space to the ambient space. Urtasun *et al* [185] use Scaled GPLVM to track people in golfing and walking sequences. They learn low-dimensional representation of 3d human pose state space and use a generative model with non-linear continuous mapping between the latent space and the full pose space, to track the 3D human pose in the latent space. They use WSL tracker [89] to track the joint centers in the 2D image sequence.

One of the key motivation behind dimensionality reduction is to improve computational cost of training regression models, where both inputs and outputs may have high dimensionality. While in practice learning mappings between low dimensional embeddings of inputs and outputs improves the computational cost, the low dimensional subspace for both input and output points are not guaranteed to preserve the correlations between them. Recent works [117, 158, 78] have proposed algorithms that learn low dimensional representations of data while preserving the statistical correlation between the input and the output points. Minyoung

et. al[117] proposed a non-linear extension of their DRR (dimensionality reduction for regression) that exploits the covariance operators(kernel gram matrix) in RKHS to estimate the variance of the inverse regression for estimating the central subspace. The central subspace is the low dimensional minimal subspace that preserves the correlation between the input and output points. Sigal *et.* al[158] proposed latent variable model based on shared kernel information that defines a multimodal density over the input feature space, the shared latent space and the output space. The advantages of this model is that it has lower complexity for learning and the conditional distribution can be easily used to condition on any combination of the input, output and the shared latent space.

2.6 Tracking and Dynamics

In most cases, the human pose needs to be estimated for a sequence of images rather than a single image and therefore needs to be tracked. Tracking serves multiple purpose of smoothing the estimated pose for a sequence of time steps and also facilitates resolving pose ambiguities by assigning larger weights to poses closer to pose predicted in the previous time step. Tracking plays a critical role in 3D pose estimation from monocular image sequence primarily due to lack of observability and ambiguity which is significantly more severe compared to multi-sensor scenarios.

Tracking human motion requires modeling dynamics of human activities and a variety of tracking frameworks exist in literature. Most tracking models are based on first and second-order Markov models[49, 89] or auto-regressive(AR) processes[128]. Agarwal and Triggs[6] use a second order AR dynamical model for tracking 3D human pose. Repetitive and cyclic motions like running and walking may be modeled using Switching Linear Dynamic System (SLDS) [128, 131, 132, 84] which is a second order stochastic process and allows more complex activities to be modeled, although they are computationally expensive to learn for high dimensional output states. A variety of non-parametric models based on particle filtering also exist for tracking[141, 154, 153, 105]. These however require a large training dataset and also does not infer the probabilistic density function.

Most of the recent approaches track high dimensional human state space by learning low dimensional subspace using methods such as PCA[34, 62], LLE, Isomap or Laplacian Eigenmaps[65, 162, 193, 134, 120, 187], and tracking the low-dimensional representation of 3D pose in the learned latent space. To reconstruct the 3D pose in original ambient space, we require an additional step to map the points from the latent space to ambient space. Sminchisescu and Jepson [162] use spectral methods(Laplacian Eigenmaps) to learn low dimensional subspace and an RBF mapping to reconstruct poses from latent positions.

Several methods have also been proposed to learn low dimensional subspaces of dynamic models. The motivation behind these methods is to exploit the dynamic nature of the data in addition to only correlations. As a result, the learned subspaces exhibit both temporal smoothness and the periodic characteristics of the motion they model. Urtasun et. al [187] proposed a latent variable dynamical model(Gaussian Process Dynamic Model) that explicitly models the dynamics in the latent space. Moon and Pavlovic[120] used a hybrid framework of Gaussian Process Latent Variable Model(GPLVM) and Marginal Auto-Regressive(MAR) model to learn the non-linearly embedded subspace of stable auto-regressive sequences. The learned subspace is used as a prior distribution to estimate the generative Nonlinear Dynamic System models used for tracking 3D human poses.

Chapter 3

Bayesian Mixture of Experts

3.1 Introduction

One-to-many mappings, commonly refered to as multi-valued functions, are a common occurrence in computer vision. In this chapter we describe Bayesian Mixture of Expert model for learning compact models for multi-valued functions. The contents of this chapter are based on the work, BM^3E *: Discriminative Density Propagation for Visual Tracking*, Cristian Sminchisescu, Atul Kanaujia, Dimitris Metaxas, *IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007* and the technical report *Learning to Reconstruct 3D Human Motion from Bayesian Mixture of Experts, A Probabilistic Discriminative Approach*, C. Sminchisescu, Atul Kanaujia, Zhiguo Li, Dimitris Metaxas, *Technical Report, University of Toronto, Oct. 2004*.

The name "multi-valued functions" is a misnomer as functions are one-to-one or many-to-one relations. Multi-valued functions refer to one-to-many relations that usually arise from the inverse of non-injective functions like trigonometric, hyperbolic, exponential and even powered functions. In computer vision, perspective projection is one such example function which does not have an inverse function since an object when observed from different viewpoints may render 2D images that are similar in appearance. The mapping from observed 2D images to 3D shapes is therefore a one-to-many relation and cannot be functionally approximated. In probabilistic settings, this is equivalent to saying that for an object conditional probability of 3D pose for a given observed 2D image is multimodal and several probable solutions exist for it. In context of human 3D pose estimation problem from monocular image sequences, this essentially means that several 3D human pose states may have similar 2D image observations both in terms of shape and appearance. Modeling relation between the observed 2D images and the corresponding 3D poses therefore produces highly multimodal distributions [153, 163, 161].

Deterministic human pose estimation algorithms [39, 124, 160, 148, 5, 4, 65] attempt to directly estimate the conditional distribution $p(\mathbf{x}|\mathbf{r})$ of 3D pose state from the set of training exemplars $\mathcal{D} = \{(\mathbf{r}^{(i)}, \mathbf{x}^{(i)}) \mid i = 1 \ldots N\}$, where $\mathbf{r}^{(i)}$ are the observations and $\mathbf{x}^{(i)}$ are the 3D pose states. The labeled exemplars are assumed to be noisy samples from the *joint distribution* of predictor and response variables $p(\mathbf{x}, \mathbf{r})$.

Note that it is difficult to avoid ambiguity in 3d pose estimation as it is intrinsic to the structure of the problem. More sensitive descriptors may be able to discriminate between some of the ambiguous poses but may not generalize well across perturbations caused due to illumination changes, disproportionate body parts or viewpoint changes. In order to resolve ambiguities, the multi-valued mappings should be accurately modeled in any pose estimation framework.

A number of methods have been proposed in the literature that target the problem of 3D human pose estimation by learning these mappings [148, 15, 121, 182, 5, 4]. Mappings may be learned either as non-parametric nearest neighbor regression model[148, 15, 182, 121] by constructing efficient data structures for faster retrieval or as a parametric regression model [5, 4, 65] of linearly weighted sum of, generally non-linear and fixed, basis vectors. Training nearest neighbor regression thus involves learning locality sensitive indexing of the nearest-neighbors of the observed input and predicting the output as weighted combination of the outputs corresponding to them. Learning parametric regression models typically involves optimization of regularized cost function to estimate model parameters (weights associated to the basis vectors). For an observed input covariate, the output is obtained as weighted combination of the basis vectors[5, 4, 65], or affine reconstruction from joint centers [104, 174, 121]. Among discriminative methods, a notable exception is [138], who clustered their dataset into soft partitions and learned functional approximations (*e.g.* perceptrons or regressors) within each of the clusters. However, clusterwise functional approximation [133, 56, 138] is only going halfway towards a multi-valued inversion, because inference is not straightforward. The problem is that the framework essentially models the 2D-3D relation using a joint distribution and not a conditional distribution. Therefore, for new inputs, cluster / perceptron membership probabilities cannot be computed as during (supervised) learning, because the outputs are missing. The learned mixture coefficients are not useful because they are fixed and obtained as averages over the training set. Therefore it is not clear what approximator or set of approximators to use for

any new observation. Various post-hoc strategies based on finding input cluster neighbors may be used, but these fall out of the estimated model that is not optimized to consistently compute such queries. On the other hand averaging across different cluster predictors can give poor results (see 3.2 for a discussion). Nevertheless, clusterwise regression [133, 56, 138] is useful as a proposal mechanism, *e.g*. during generative inference based on quadrature-style Monte-Carlo approximations and indeed this is how it has been primarily used [138]. A related method has been proposed by [76], where a mixture of probabilistic PCA is fitted to the joint distribution represented as silhouette features in multiple views paired with their 3D poses. Reconstruction is based on the MAP estimates. In this imaging setting the state conditional could be unimodal, but missing data makes inference (*i.e*. conditional computation) non-trivial, demanding in principle, an application of Bayes' rule and marginalization (see our §3.5).

In this chapter, we describe formulation and evaluation of Bayesian conditional mixture of experts that allows flexible discriminative modeling. The proposed framework is motivated by the fact that many vision problems like 3D pose estimation and tracking involve the recovery of inverse, intrinsically multi-valued mappings. Conditional distributions of ambiguous static or dynamic covariates are therefore multimodal and require multiple function approximators to map similar input observations to multiple plausible but perceptually different outputs. Our algorithms are based on hierarchical mixture of experts [87, 92, 194, 30] and joint mixture of experts [199, 184]. Mixture of Experts models are elaborated versions of clusterwise or switching regression [133, 56, 138], where the expert mixture proportions (called gates) are themselves observation-dependent predictors. These gates distributions represent the competence of the experts to predict for an input observation and are modeled as normalized softmax activation functions. Inference is simple and produces multimodal state conditionals. Learning is different from [194] in that we use sparse greedy approximations, and differs from [30, 184] in that we use *type-II maximum likelihood* Bayesian approximations [113, 180] and not structured variational ones.

3.2 Overview of Bayesian Learning Framework

Bayesian learning intrinsically embodies regularization and model selection using Occam's razor[113] [180].'Occam's razor' is the principle that states that simpler models should be preferred over unnecessary complex models. Selecting a model that fits best to the training data does not guarantee best performance on the test data. Best fit learning leads to implausibly detailed and over-parameterized models that interpolate and generalize poorly. Bayesian learning theory provides us with the framework to quantify the complexity of the models and systematically promote simpler models by penalizing the over-parameterization. Bayesian learning is a three level process.

In the *first level* the model is fit to the observed data by maximizing posterior distribution over the model parameters $\mathbf{\Theta}$.

$$p(\mathbf{\Theta}|\mathcal{D},\alpha,\beta,\mathcal{M}) = \frac{p(\mathcal{D}|\mathbf{\Theta},\beta,\mathcal{M})p(\mathbf{\Theta}|\alpha)}{p(\mathcal{D}|\mathcal{M})} \tag{3.1}$$

The normalizing constant is called the evidence of the model $\mathcal{M}$ and is not required for fitting a given model $\mathcal{M}$ to the data set $\mathcal{D}$. The first term on right hand side(likelihood) is the loss function and second term(prior distribution) is the smoothing factor. α and β are the scale parameters of these distributions. Assuming the distributions to be gaussians with appropriate normalization factors, in the current settings:

$$p(\mathcal{D}|\mathbf{\Theta},\beta,\mathcal{M}) = \frac{e^{-\beta L_{\mathbf{\Theta}}(\mathcal{D})}}{(2\pi/\beta)^{N/2}} \tag{3.2}$$

$$p(\mathbf{\Theta}|\alpha,\mathcal{M}) = \frac{e^{-\alpha P(\mathbf{\Theta})}}{\int e^{-\alpha p(\mathbf{\Theta})} d\mathbf{\Theta}} \tag{3.3}$$

where $L_{\mathbf{\Theta}}$ is the loss function to be minimized and $P(\mathbf{\Theta})$ is the penalty term for penalizing complex models with larger $|\mathbf{\Theta}|$ (smoothing). The posterior obtained is a joint function of scale parameters α, β for the loss function and smoothing prior respectively. Given α and β, most probable Θ_{MP} can be obtained by maximizing the posterior distribution (3.1).

The *Second level* of bayesian learning involves model selection by estimating the most probable scale parameters α_{MP} and β_{MP} by maximizing the posterior distribution:

$$p(\alpha,\beta|\mathcal{D},\mathcal{M}) \propto p(\mathcal{D}|\alpha,\beta,\mathcal{M})p(\alpha|\mathcal{M})p(\beta|\mathcal{M}) \tag{3.4}$$

For uniform prior distributions of α and β, maximizing (3.4) is equivalent to maximizing the evidence $P(\mathcal{D}|\alpha, \beta, \mathcal{M})$. This evidence maximization procedure is called *Type II Maximum Likelihood* maximization and can be used to compute most probable α_{MP} and β_{MP} as the modes of the likelihood distribution obtained by marginializing out the parameters Θ:

$$p(\mathcal{D}|\alpha, \beta, \mathcal{M}) = \int p(\mathcal{D}|\mathbf{\Theta}, \beta, \mathcal{M}) p(\mathbf{\Theta}|\alpha, \mathcal{M}) d\Theta \tag{3.5}$$

The posterior distributions for model parameters $\mathbf{\Theta}$ is approximated as $P(\mathbf{\Theta}|\mathcal{D}, \alpha, \beta, \mathcal{M}) \approx P(\mathbf{\Theta}|\mathcal{D}, \alpha_{MP}, \beta_{MP}, \mathcal{M})$ which can be used with (3.1), to estimate most probable parameters $\mathbf{\Theta} = \mathbf{\Theta_{MP}}$ (mode of the posterior distribution(3.1)), α_{MP} and β_{MP} using iterative evidence maximization[113].

The *third stage* of Bayesian Framework allows us to quantitatively rank different models by comparing different basis functions, the regularization prior distributions (with the scale parameter α) and the noise models(with the scale parameter β). Different priors and basis functions corresponds to different hypothesis about the unknown data generation process and can be compared by evaluating evidence. In the past, [113][124] have proposed Gamma distribution for the prior for scale parameters α and β. Using gamma priors causes posterior distribution of scale parameters to concentrate at large values for inputs which contribute little towards the data interpolant to be predicted. The $\mathbf{\Theta}$ parameters corresponding to these low relevance inputs can be pruned. The parameter set $\mathbf{\Theta}$ so obtained is sparser compared to those obtained by *Maximum Margin* approaches like SVM. This formulation is a form of *Automatic Relevance Determination* and has been applied in a variety of optimization methods in Gaussian process like settings.

3.3 Mixture of Experts Framework

The mixture of experts(ME) model [87, 92, 194, 30] consists of a gate distribution that groups data points into multiple clusters and a set of experts that are local, contextual function approximators fitted to each of the clusters. ME model extends clusterwise regression [56, 138] which is a maximum likelihood methodology for simultaneously fitting multiple regression functions and clustering a dataset into M clusters. For the observation-state(predictor-response) pairs

$\mathcal{D} = \{(\mathbf{r}^{(1)}, \mathbf{x}^{(1)}), (\mathbf{r}^{(2)}, \mathbf{x}^{(2)}), \cdots, (\mathbf{r}^{(N)}, \mathbf{x}^{(N)})\}$ and a desired value of M, clusterwise regression assumes that the output state $\mathbf{x}^{(i)}$ is distributed as mixture of Gaussian distributions :

$$\mathbf{x}^{(i)} \sim \sum_{m=1}^{K} g_m p_m(\mathbf{x}^{(i)}|\mathbf{r}^{(i)}, \mathbf{W}_m, \sigma_M) \tag{3.6}$$

where g_m are fixed cluster proportions that are estimated using maximum likelihood learning in , *Expectation Maximization* framework.

Adaptive mixture of experts[87] is a competitive learning model, consisting of multiple regression functions and an associated weight that is input dependent. These probabilistic weights, henceforth referred to as 'gates', model the mixing proportions of different clusters. The idea behind the gates is to make a stochastic decision about which single expert to use for each new input rather than linearly combining the expert outputs with fixed weights. The experts transform their inputs into output predictions that are combined in a probabilistic mixture model.

For modeling a conditional distribution, the 'inputs' can be observations $\mathbf{r}^{(n)}$ when modeling $p(\mathbf{x}^{(n)}|\mathbf{r}^{(n)})$, states $\mathbf{x}^{(n)}$ when modeling $p(\mathbf{x}^{(n)}|\mathbf{x}^{(n-1)})$ or observation-state pairs $(\mathbf{x}^{(n-1)}, \mathbf{r}^{(n)})$ for $p(\mathbf{x}^{(n)}|\mathbf{x}^{(n-1)}, \mathbf{r}^{(n)})$. The 'output' is the state, $\mathbf{x}^{(n)}$ throughout. The model is consistently parameterized and has both gate and expert parameters that are jointly estimated during learning. Whereas some of the inputs multiple plausible outputs may be possible and will have different predictions from the experts, less ambiguous inputs will have all the experts predicting similar outputs. All the gates outputs for the ambiguous inputs will typically have large values, while for unambiguous inputs, most of these gate values will be 0 in order to assign higher probability to a particular expert. Formally this is described by:

$$p(\mathbf{x}|\mathbf{r}, \mathbf{W}, \mathbf{\Omega}, \boldsymbol{\lambda}_i, \boldsymbol{\beta}_i) = \sum_{i=1}^{M} g(\mathbf{r}|\boldsymbol{\lambda}_i, \boldsymbol{\beta}_i) p(\mathbf{x}|\mathbf{r}, \mathbf{W}_i, \mathbf{\Omega}_i^{-1}) \tag{3.7}$$

where the gates and the expert distributions are:

$$g(\mathbf{r}|\boldsymbol{\lambda}_i, \boldsymbol{\beta}_i) = \frac{f(\mathbf{r}|\boldsymbol{\lambda}_i, \boldsymbol{\beta}_i)}{\sum_{k=1}^{M} f(\mathbf{r}|\boldsymbol{\lambda}_k, \boldsymbol{\beta}_k)} \tag{3.8}$$

$$p(\mathbf{x}|\mathbf{r}, \mathbf{W}_i, \mathbf{\Omega}_i) = \mathcal{N}(\mathbf{x}|\mathbf{W}_i \boldsymbol{\phi}(\mathbf{r}), \mathbf{\Omega}_i^{-1}) \tag{3.9}$$

Here $\mathbf{r}$ are input or predictor variables, $\mathbf{x}$ are outputs or responses, g are *input dependent* positive gates, computed as functions $f(\mathbf{r}|\boldsymbol{\lambda}_i, \boldsymbol{\beta}_i)$, parameterized by the weights $\boldsymbol{\lambda}_i$ (f should

produce probabilities as gates g within $[0, 1]$). By construction g are normalized to sum to 1 for any given input $\mathbf{r}$. Also p are Gaussian distributions (3.9) with covariances $\mathbf{\Omega}_i^{-1}$, centered at 'expert' predictions, here kernel (ϕ) regressors with weights $\mathbf{W}_i$. The parameters of the model including experts and gates, are collectively stored in $\Theta = \{(\boldsymbol{\alpha}_i, \mathbf{W}_i, \mathbf{\Omega}_i, \boldsymbol{\lambda}_i, \boldsymbol{\beta}_i) | i = 1 \dots M\}$.

Learning the mixture of experts essentially involves optimization of the parameters Θ by maximization of the log-likelihood of a data set, $\mathcal{D} = \{(\mathbf{r}^{(i)}, \mathbf{x}^{(i)}) \mid i = 1 \dots N\}$ *i.e.* the accuracy of predicting $\mathbf{x}$ given $\mathbf{r}$, averaged over the data distribution.

Learning in mixture of experts, as originally proposed by Jordan *et. al*[92], is also based on Expectation-Maximization framework coupled with *Iteratively Re-weighted Least Square(IRLS)* algorithm for gates estimations. Our algorithm is based on the original ME learning algorithm and proceeds as follows. In the E-step we estimate the posterior:

$$h(\mathbf{x}, \mathbf{r} | \mathbf{W}_i, \mathbf{\Omega}_i, \boldsymbol{\lambda}_i, \boldsymbol{\beta}_i) = \frac{g(\mathbf{r} | \boldsymbol{\lambda}_i, \boldsymbol{\beta}_i) p(\mathbf{x} | \mathbf{r}, \mathbf{W}_i, \mathbf{\Omega}_i^{-1})}{\sum_{j=1}^{M} g(\mathbf{r} | \boldsymbol{\lambda}_j, \boldsymbol{\beta}_j) p(\mathbf{x} | \mathbf{r}, \mathbf{W}_j, \mathbf{\Omega}_j^{-1})} \tag{3.10}$$

h gives the probability that the data is mapped using the i^{th} expert, and can be obtained only for labeled exemplars as it requires knowledge of both inputs and outputs. The M-step involves solving the optimization problems for experts and gates. The gate parameters $(\boldsymbol{\lambda}_i, \boldsymbol{\beta}_i)$ are estimated by essentially training a multi-category classifier with inputs as $\mathbf{r}$ and output as h. The expert parameters $(\mathbf{W}_i, \mathbf{\Omega}_i)$ are estimated by fitting regression functions in the inputs, re-weighted by the expert membership probabilities h. The reweighing essentially enables fitting the regression to the input domain only.

3.3.1 Learning Mixture of Experts Model

Mixture of experts (ME) models may differ in the way the conditional distribution $p(\mathbf{x}|\mathbf{r}, \Theta)$ for the output is computed. The probability model for ME is given in eqn. 3.7 and involves two conditional distributions, the gates $g_i = p(\mathbf{i}|\mathbf{r}, \boldsymbol{\lambda}_i, \boldsymbol{\beta}_i)$ and the experts $p(\mathbf{x}|\mathbf{r}, \mathbf{W}_i, \mathbf{\Omega}_i^{-1})$.

The gates distribution plays a key role in the learning of mixture of experts model and is essentially a multi-category classifier. The gates may be discriminatively modeled as a softmax function [87, 92, 194, 30] or generatively modeled as a naive Bayes classifier [199, 184]. Discriminative modeling leads to nested EM algorithm as the cost function has intractable form and is typically optimized using numerical optimization methods(like *Iterative Re-weighted*

Least Square). Generative modeling uses Bayes' rule to express the gate distribution as :

$$g_i = p(\mathbf{i}|\mathbf{r}, \boldsymbol{\lambda}_i, \boldsymbol{\beta}_i) = \frac{p(\mathbf{i})p(\mathbf{r}|\mathbf{i}, \boldsymbol{\lambda}_i, \boldsymbol{\beta}_i)}{\sum_{j=1}^{M} p(\mathbf{j})p(\mathbf{r}|\mathbf{j}, \boldsymbol{\lambda}_j, \boldsymbol{\beta}_j)} \tag{3.11}$$

This models the inverse distribution $p(\mathbf{r}|\mathbf{i}, \boldsymbol{\lambda}_i, \boldsymbol{\beta}_i)$ and does not require costly nested loop EM algorithm, as discussed in detail in the §3.5.

Next section introduces the Bayesian Mixture of Experts model. In the thesis we have used conditional mixture of experts models, discriminative gate distribution, and propose a learning framework based on Bayesian model selection and regularization. We have also implemented the Mixture of Experts model with generative gates and discuss it in more detail in the §3.5.

3.4 Bayesian Conditional Mixture of Experts

A full Bayesian treatment requires computing posterior distributions over a set of parameters (weights of regressor or classifier) and associating a set of hyperparameters to control the prior distributions. As in most cases the exact computations are intractable, we rely on approximations and design iterative Bayesian EM algorithms, based on type-II maximum likelihood(ML-II) [113, 180]. ML-II optimization is a Bayesian model selection technique with greedy (expert and gate weights) subset selection. This strategy aggressively sparsifies the experts and the gates by eliminating inputs with small weights after each iteration [180, 102]. As in many Bayesian settings [113, 180, 30], the weights for the experts $\mathbf{W}_i$ and the gates $\boldsymbol{\lambda}_i$(3.7), are controlled by hierarchical priors, typically Gaussians with 0 mean, and having inverse variance hyperparameters $\boldsymbol{\alpha}_i$ ($\boldsymbol{\beta}_i$ for the gates) controlled by a second level of Gamma distributions. This gives an automatic relevance determination mechanism [113, 180] that avoids overfitting and encourages compact models with fewer non-zero weights for efficient prediction. *Inference* in these model is straightforward using (3.7). The result is a conditional mixture distribution with components and mixing probabilities that are input-dependent. We now provide the details of the sparse Bayesian learning applied to mixture of experts and the associated inference.

3.4.1 BME Formulation

In the conditional mixture of experts model, the data generation process assumes N data points are produced by one of M experts, selected in a stochastic manner. This is modeled by indicator

(hidden) variables $\mathcal{Z} = \{z_i^{(n)} | i = 1 \dots M, n = 1 \dots N\}$ where $z_i^{(n)}$ is 1 if the output data point $\mathbf{x}^{(n)}$ has been produced by expert i and zero otherwise. The parameters and hyperparameters of the model are denoted by Θ, where $\Theta = \{(\mathbf{W}_i, \mathbf{\Omega}_i, \boldsymbol{\alpha}_i, \boldsymbol{\lambda}_i, \boldsymbol{\beta}_i) \mid i = 1 \dots M\}$, and $\boldsymbol{\lambda}_i$, $\mathbf{W}_i$ are the individual gate and expert predictor parameters respectively. We omit the bias terms for clarity. The conditional probability of the output $\mathbf{x}^{(n)}$ (of dimension D) for an observed input $\mathbf{r}^{(n)}$ (of dimension d) is a mixture model with M components:

$$p(\mathbf{x}^{(n)}|\mathbf{r}^{(n)}, \Theta) = \sum_{i=1}^{M} p(z_i^{(n)}|\mathbf{r}^{(n)}, \boldsymbol{\lambda}_i) p(\mathbf{x}^{(n)}|\mathbf{r}^{(n)}, \mathbf{W}_i, \mathbf{\Omega}_i^{-1}) \tag{3.12}$$

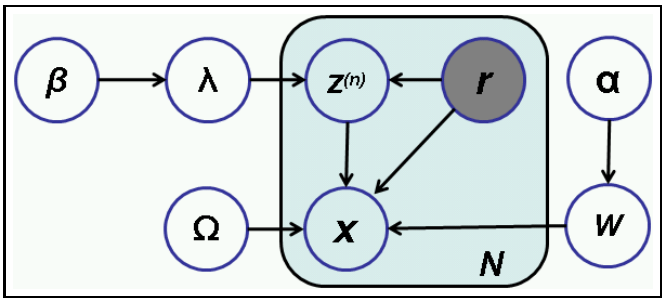


Figure 3.1: Graphical model of mixture of experts model[30]. We show here a detailed dependency graph of the parameters and the hidden variables in the BME model. Shaded nodes indicate instantiated variables that are observed and used as inputs in the conditionals Bayesian Mixture of Experts model.

The probability of each expert is a Gaussian centered at its prediction $\mathbf{W}_i \boldsymbol{\phi}(\mathbf{r}^{(n)})$, where $\boldsymbol{\phi}$ is a vector of kernel functions:

$$\chi_i^{(n)} = p(\mathbf{x}^{(n)}|\mathbf{r}^{(n)}, \mathbf{W}_i, \mathbf{\Omega}_i^{-1}) = \mathcal{N}(\mathbf{x}^{(n)}|\mathbf{W}_i \boldsymbol{\phi}(\mathbf{r}^{(n)}), \mathbf{\Omega}_i^{-1}) \tag{3.13}$$

The conditional (prior) probability of selecting expert i, given the input *only*, is implemented using softmax function. This ensures that the expert outputs are probabilistically consistent (positive and sum to 1), for any given input:

$$g_i^{(n)} = p(z_i^{(n)} = 1|\mathbf{r}^{(n)}, \boldsymbol{\lambda}_i) = \frac{e^{\boldsymbol{\lambda}_i^\top \boldsymbol{\phi}(\mathbf{r}^{(n)})}}{\sum_{k=1}^{M} e^{\boldsymbol{\lambda}_k^\top \boldsymbol{\phi}(\mathbf{r}^{(n)})}} \tag{3.14}$$

The conditional (posterior) probability $h_i^{(n)}$ of selecting expert i, given *both* the input $\mathbf{r}^{(n)}$ *and* the output $\mathbf{x}^{(n)}$, is:

$$h_i^{(n)} = p(z_i^{(n)} = 1|\mathbf{x}^{(n)}, \mathbf{r}^{(n)}, \mathbf{W}_i, \boldsymbol{\lambda}_i, \mathbf{\Omega}_i) = \frac{g_i^{(n)} \chi_i^{(n)}}{\sum_{k=1}^{M} g_k^{(n)} \chi_k^{(n)}} \tag{3.15}$$

The posterior is only available during learning. For inference (prediction) based on (3.12), the learned prior (3.14) *i.e*. the gate distribution, is used. The gate and expert weights have zero centered Gaussian priors, with variance controlled by a second level of Gamma hyper-priors. This avoids overfitting and provides an automatic relevance determination mechanism, encouraging compact models with fewer non-zero expert and gate weights, for efficient prediction [113, 124, 180, 30]:

$$p(\boldsymbol{\lambda}_i|\boldsymbol{\beta}_i) = \prod_{k=1}^{d} \mathcal{N}(\lambda_i^k|0, \frac{1}{\beta_i^k}) \tag{3.16}$$

$$p(\mathbf{W}_i|\boldsymbol{\alpha}_i) = \prod_{j=1}^{D}\prod_{k=1}^{d} \mathcal{N}(w_i^{jk}|0, \frac{1}{\alpha_i^k}) \tag{3.17}$$

$$p(\boldsymbol{\alpha}_i) = \prod_{k=1}^{d} \text{Gamma}(\alpha_i^k|a, b) \tag{3.18}$$

$$p(\boldsymbol{\Omega}_i^{-1}) = \text{Gamma}(\boldsymbol{\Omega}_i^{-1}|c, d) \tag{3.19}$$

$$p(\boldsymbol{\beta}_i) = \prod_{k=1}^{d} \text{Gamma}(\beta_i^k|a, b) \tag{3.20}$$

$$\text{Gamma}(v|a, b) = \frac{b^a v^{(a-1)} e^{-bv}}{\Gamma(a)} \tag{3.21}$$

The parameters (a, b) are set to $a = 10^{-2}$ and $b = 10^{-4}$ to give broad hyper-priors [30, 113, 124, 180].The likelihood of the incomplete dataset $\mathcal{D}$ can be written as:

$$\ell(\mathcal{D}|\Theta) = \prod_{n=1}^{N}\sum_{k=1}^{M} p(z_k^{(n)} = 1|\mathbf{r}^{(n)}, \boldsymbol{\lambda}_k) p(\mathbf{x}^{(n)}|\mathbf{r}^{(n)}, \mathbf{W}_i, \boldsymbol{\Omega}^{-1}) \tag{3.22}$$

This has an inconvenient form and is difficult to maximize due to the summation. Instead, a more convenient distribution to optimize is the likelihood of the complete dataset that can be

written as:

$$\ell_c(\mathcal{D}, \mathcal{Z}|\Theta) = \prod_{n=1}^{N} p(z_k^{(n)} = 1|\mathbf{r}^{(n)}, \boldsymbol{\lambda}_k) p(\mathbf{x}^{(n)}|\mathbf{r}^{(n)}, \mathbf{W}_i, \boldsymbol{\Omega}^{-1}) \tag{3.23}$$

$$= \prod_{n=1}^{N} \prod_{k=1}^{M} \{p(z^{(n)}|\mathbf{r}^{(n)}, \boldsymbol{\lambda}_k) p(\mathbf{x}^{(n)}|\mathbf{r}^{(n)}, \mathbf{W}_i, \boldsymbol{\Omega}^{-1})\}^{z_k^{(n)}} \tag{3.24}$$

$$\tag{3.25}$$

Jordan *et. al* used maximum likelihood learning based on complete likelihood optimization to train ME model[92]. However as discussed above, models obtained using ML learning often fail to generalize well. In the next section, we propose a learning framework based on optimization of the posterior distribution over the unknown parameters Θ. Within the optimization framework we also use Bayesian Model selection to avoid overfitting.

3.4.2 Posterior Formulation and Bayesian Inference

Having formulated the prior distribution and the likelihood, Bayesian inference proceeds by computing the posterior over the unknown parameters of the model. The complete posterior can be factorized as:

$$p(\mathbf{W}, \boldsymbol{\lambda}, \boldsymbol{\Omega}, \boldsymbol{\alpha}, \boldsymbol{\beta}|\mathcal{Z}, \mathcal{D}) = \frac{p(\mathcal{Z}, \mathcal{D}|\mathbf{W}, \boldsymbol{\lambda}, \boldsymbol{\Omega}, \boldsymbol{\alpha}, \boldsymbol{\beta}) p(\mathbf{W}, \boldsymbol{\lambda}, \boldsymbol{\Omega}, \boldsymbol{\alpha}, \boldsymbol{\beta})}{p(\mathcal{Z}, \mathcal{D})} \tag{3.26}$$

Here $\mathcal{Z} = \{\left(\mathbf{r}^{(n)}, \left[z_1^{(n)}, \cdots, z_M^{(n)}\right]\right) | n = 1 \cdots N\}$ denotes the set of the data pair of the input variable and the corresponding indicator variable (denoting the expert used for mapping the data point). The posterior distribution in this form is analytically intractable as it is difficult to compute the normalizing integral:

$$p(\mathcal{Z}, \mathcal{D}) = \int \{p(\mathcal{Z}, \mathcal{D}|\mathbf{W}, \boldsymbol{\lambda}, \boldsymbol{\Omega}, \boldsymbol{\alpha}, \boldsymbol{\beta}) p(\mathbf{W}, \boldsymbol{\lambda}, \boldsymbol{\Omega}, \boldsymbol{\alpha}, \boldsymbol{\beta})\} d\mathbf{W} d\boldsymbol{\alpha} d\boldsymbol{\Omega} d\boldsymbol{\beta} \tag{3.27}$$

We therefore factorize the posterior distribution using the chain rule and the fact that the posterior of $\mathbf{W}$ and $\boldsymbol{\lambda}$ are conditionally independent given respective hyperparameters(as depicted in the graphical model fig.3.1) as:

$$p(\mathbf{W}, \boldsymbol{\lambda}, \boldsymbol{\alpha}, \boldsymbol{\Omega}, \boldsymbol{\beta}|\mathcal{Z}, \mathcal{D}) = p(\mathbf{W}|\boldsymbol{\alpha}, \boldsymbol{\Omega}, \mathcal{Z}, \mathcal{D}) p(\boldsymbol{\alpha}, \boldsymbol{\Omega}|\mathcal{Z}, \mathcal{D}) p(\boldsymbol{\lambda}|\boldsymbol{\beta}, \mathcal{Z}) p(\boldsymbol{\beta}|\mathcal{Z}) \tag{3.28}$$

For Mixture of Experts learning, we compute the M posterior distributions for the weights of each of the experts and the gate function. The expert parameters are the weights and the variance of the Gaussian centered at the prediction of the kernel regressor $\{\mathbf{W}, \mathbf{\Omega}\}$(3.13). Gate parameters are the weights $\boldsymbol{\lambda}$ (3.14) of the kernelized inputs to the softmax activation gate functions. The posterior distribution is still intractable due to non-Gaussian posterior distributions for the hyperparameters $\{\boldsymbol{\alpha}, \mathbf{\Omega}, \boldsymbol{\beta}\}$. We are therefore forced to make following key assumptions to facilitate the analytical computation of posterior:

1. For the joint posterior distribution over weights for the experts, we assume the following conditional independence given the indicator variables $\mathcal{Z}$:

$$p(\mathbf{W}, \boldsymbol{\alpha}, \mathbf{\Omega}|\mathcal{D}, \mathcal{Z}) = \prod_{i=1}^{M} p(\mathbf{W}_i, \boldsymbol{\alpha}_i, \mathbf{\Omega}_i|\mathcal{D}, \mathcal{Z}) \tag{3.29}$$

   This is a reasonable assumption as the experts are learned locally within each of the clusters and $\mathcal{Z}$ effectively encodes the soft clustering of the dataset in M clusters. Under this assumption we can estimate the hyperparameters for each expert independently of the others given the values of indicator variables $\mathcal{Z}$.

2. We approximate the hyperparameter posteriors $p(\boldsymbol{\alpha}, \mathbf{\Omega}|\mathcal{D}, \mathcal{Z})$ and $p(\boldsymbol{\beta}|\mathcal{Z})$ by delta function at their modes $\delta(\boldsymbol{\alpha}_{MP}, \mathbf{\Omega}_{MP})$ and $\delta(\boldsymbol{\beta}_{MP})$. This assumption is based on the fact that for predictive modeling, the point estimate of these hyperparameters are representative of their posterior distribution. For the mixture of expert prediction:

$$\begin{aligned} p(\mathbf{x}^*|\mathcal{D}, \mathcal{Z}) = & \int p(\mathbf{x}^*|\mathbf{W}, \boldsymbol{\lambda}, \boldsymbol{\alpha}, \mathbf{\Omega}, \boldsymbol{\beta}) p(\mathbf{W}|\boldsymbol{\alpha}, \mathbf{\Omega}, \mathcal{D}, \mathcal{Z}) p(\boldsymbol{\alpha}, \mathbf{\Omega}|\mathcal{D}, \mathcal{Z}) \\ & p(\boldsymbol{\lambda}|\boldsymbol{\beta}, \mathcal{Z}) p(\boldsymbol{\beta}|\mathcal{Z}) d\mathbf{W} d\boldsymbol{\lambda} d\boldsymbol{\alpha} d\mathbf{\Omega} d\boldsymbol{\beta} \qquad (3.30) \\ \approx & \int p(\mathbf{x}^*|\mathbf{W}, \boldsymbol{\lambda}, \boldsymbol{\alpha}, \mathbf{\Omega}, \boldsymbol{\beta}) p(\mathbf{W}|\boldsymbol{\alpha}_{MP}, \mathbf{\Omega}_{MP}, \mathcal{D}, \mathcal{Z}) \delta(\boldsymbol{\alpha}_{MP}, \mathbf{\Omega}_{MP}|\mathcal{D}, \mathcal{Z}) \\ & p(\boldsymbol{\lambda}|\boldsymbol{\beta}_{MP}, \mathcal{Z}) \delta(\boldsymbol{\beta}_{MP}|\mathcal{Z}) d\mathbf{W} d\boldsymbol{\lambda} \qquad (3.31) \end{aligned}$$

   In the above equations the predictions obtained by marginalizing over the hyperparameters are near-identical to those obtained by setting the hyperparameters to their most probable values. The most probable values may be obtained as MAP estimate of the hyperparameters. For the i^{th} expert, Maximum APosteriori(MAP) estimate of the hyperparameters is obtained by maximizing the likelihood(for the non-informative priors

$p(\boldsymbol{\alpha}_i), p(\boldsymbol{\Omega}_i)$) :

$$p(\boldsymbol{\alpha}_i, \boldsymbol{\Omega}_i|\mathcal{D}, \mathcal{Z}) \propto p(\mathcal{D}|\boldsymbol{\alpha}_i, \boldsymbol{\Omega}_i, \mathcal{Z})p(\boldsymbol{\alpha}_i)p(\boldsymbol{\Omega}_i) \tag{3.32}$$

In the Bayesian learning framework, $p(\mathcal{D}|\boldsymbol{\alpha}_i, \boldsymbol{\Omega}_i, \mathcal{Z})$ is known as the marginal likelihood and its maximization is called *Type II maximum likelihood*[113, 28]. Notice that the indicator variables $\mathcal{Z}$ are required in marginal likelihood distribution and assigns higher weights to the samples $\mathcal{D}$ belonging to the associated cluster i and lower weights to rest of the samples.

We can now readily formulate the sparse Bayesian learning for Mixture of Experts by inferencing the posterior distribution of the experts and gates parameters separately.

Inference of Expert Conditionals

The conditional likelihood for the expert distribution has the form:

$$p(\mathcal{D}|\mathbf{W}_i, \boldsymbol{\Omega}_i, \mathcal{Z}) = \prod_{n=1}^{N} p(\mathbf{x}^{(n)}|\mathbf{r}^{(n)}, \mathbf{W}_i, \boldsymbol{\Omega}^{-1}, z_i^{(n)}) \tag{3.33}$$

$$= \prod_{n=1}^{N} p(\mathbf{x}^{(n)}|\mathbf{r}^{(n)}, \mathbf{W}_i, \boldsymbol{\Omega}^{-1})^{z_i^{(n)}} \tag{3.34}$$

$$= \prod_{n=1}^{N} \frac{(2\pi)^{-\frac{(N+1)}{2}}}{|\boldsymbol{\Omega}_i|^{-\frac{1}{2}}} \exp\{-\frac{z_i^{(n)}}{2}(\mathbf{x}^{(n)} - \mathbf{W}_i^T\boldsymbol{\phi}(\mathbf{r}^{(n)}))^T\boldsymbol{\Omega}_i^{-1}(\mathbf{x}^{(n)} - \mathbf{W}_i^T\boldsymbol{\phi}(\mathbf{r}^{(n)}))\} \tag{3.35}$$

Notice that the effect of soft clustering using the indicator variables $z_i^{(n)}$ is to simply reweigh the input-output vectors $(\mathbf{x}^{(n)}, \mathbf{r}^{(n)})$ by the coefficient $\sqrt{z_i^{(n)}}$. This enforces locality of the experts by not allowing the training to be affected by data points belonging to other clusters. The weight posterior for the i^{th} expert can be obtained as:

$$p(\mathbf{W}_i|\boldsymbol{\alpha}_i, \boldsymbol{\Omega}_i, \mathcal{D}, \mathcal{Z}) = \frac{p(\mathcal{D}|\mathbf{W}_i, \boldsymbol{\Omega}_i, \mathcal{Z})p(\mathbf{W}_i|\boldsymbol{\alpha}_i)}{p(\mathcal{D}|\boldsymbol{\alpha}_i, \boldsymbol{\Omega}_i, \mathcal{Z})} \tag{3.36}$$

The normalizing distribution is referred to as marginal likelihood in relation to Bayesian models and its maximization is a model selection technique known as *Type II Maximum Likelihood*. For the expert weight parameters, this can be exactly computed in a closed form by marginalizing out the weight parameters in the likelihood, as both the likelihood and the prior are normal

distributions:

$$p(\mathcal{D}|\boldsymbol{\alpha}_i, \boldsymbol{\Omega}_i, \mathcal{Z}) = \int p(\mathcal{D}|\boldsymbol{\Omega}_i, \mathbf{W}_i, \mathcal{Z}) p(\mathbf{W}_i|\boldsymbol{\alpha}_i) d\mathbf{W} \tag{3.37}$$

The weights posterior can be analytically computed as normal distribution:

$$p(\mathbf{W}_i|\boldsymbol{\alpha}_i, \boldsymbol{\Omega}_i, \mathcal{D}, \mathcal{Z}) = (2\pi)^{-(N+1)/2} |\Sigma_{\mathcal{Z}}|^{-1/2} \exp\{-\frac{1}{2}(\mathbf{W}_i - \mu_i)^T \Sigma_{\mathcal{Z}}^{-1} (\mathbf{W}_i - \mu_i)\} \tag{3.38}$$

The covariance and mean of the Gaussian distribution:

$$\Sigma_{\mathcal{Z}} = (\boldsymbol{\Omega}_i^{-1} \mathcal{R}_{\mathcal{Z}} \left\{ \boldsymbol{\phi}(\mathbf{r}^{(n)}) \right\}^T \mathcal{R}_{\mathcal{Z}} \left\{ \boldsymbol{\phi}(\mathbf{r}^{(n)}) \right\} + \mathbf{A}_i) \tag{3.39}$$

$$\mu_i = \boldsymbol{\Omega}_i^{-1} \Sigma_{\mathcal{Z}} \mathcal{R}_{\mathcal{Z}} \left\{ \boldsymbol{\phi}(\mathbf{r}^{(n)}) \right\}^T \mathcal{R}_{\mathcal{Z}} \left\{ \boldsymbol{\phi}(\mathbf{r}^{(n)}) \right\} \tag{3.40}$$

where $\mathbf{A}_i = diag\{\alpha_{i,1}, \alpha_{i,2}, \cdots, \alpha_{i,K}\}$ and $\mathcal{R}_{\mathcal{Z}}\{\mathbf{y}^{(n)}\}$ are the reweighted inputs according to the clustering as denoted by the indicator variables $\mathcal{Z}$. The weights of the inputs are automatically learned during the optimization and are discussed in more detail in the §3.4.5.

Inference of Gates Distributions

The role of the gate distribution is to do soft clustering of the dataset and allow points to lie in multiple regions. The key advantage of using soft partitioning is to ameliorate the severe variance error that is particularly prevalent in divide and conquer techniques used for training decision trees like CART, MARS and ID3. The gate is a multi-category classifier with the likelihood as a multinomial distribution:

$$P(\mathcal{Z}|\boldsymbol{\lambda}) = \prod_{n=1}^{N} \prod_{i=1}^{M} \rho_i \left\{ \boldsymbol{\phi}(\mathbf{r}^{(n)}) \right\}^{z_i^{(n)}} \tag{3.41}$$

with canonical link functions as

$$\rho_j\{\mathbf{f}\} = \frac{e^{-f_j(\mathbf{y})}}{\sum_i^M e^{-f_i(\mathbf{y})}} \tag{3.42}$$

where the mappings $f_j(\mathbf{y}) = \sum_n^N \lambda_{j,n} \Phi(\mathbf{y}, \mathbf{y}^{(n)}) = \boldsymbol{\lambda}_{\mathbf{j}}^{\mathbf{T}} \boldsymbol{\phi}(\mathbf{y})$, are the kernel basis interpolant at N training points. We assume independent weight priors for the M gates:

$$p(\boldsymbol{\lambda}|\boldsymbol{\beta}) = \prod_{i=1}^{M} p(\boldsymbol{\lambda}_{\mathbf{i}}|\boldsymbol{\beta}_{\mathbf{i}}) \tag{3.43}$$

Computing the exact weights posterior for the gate function is analytically intractable as the likelihood is non-Gaussian and the normalization factor cannot be obtained by marginalizing out nuisance parameters (as in the case of expert distribution in the equation 3.36).

Hence we use Laplace approximation to estimate the marginalization integral in the equation 3.47. Laplace method approximates $\int f(\mathbf{y})d\mathbf{y}$ by a Gaussian distribution centered at the modes $\hat{\mathbf{y}}$ of $f(\mathbf{y})$ and with the covariance computed as the hessian of $\ln\{f(\mathbf{y})\}$ at $\hat{\mathbf{y}}$. If we define:

$$\mathcal{H}(\boldsymbol{\lambda}, \mathcal{Z}, \boldsymbol{\beta}) = -\ln\{p(\mathcal{Z}|\boldsymbol{\lambda}, \boldsymbol{\beta})p(\boldsymbol{\lambda}|\boldsymbol{\beta})\} \tag{3.44}$$

then we can re-write the posterior in the form:

$$\begin{aligned} p(\boldsymbol{\lambda}|\mathcal{Z}, \boldsymbol{\beta}) &\propto \exp\{-\mathcal{H}(\boldsymbol{\lambda}, \mathcal{Z}, \boldsymbol{\beta})\} \\ &\simeq \exp\{-\mathcal{H}(\boldsymbol{\lambda}_{MP}, \mathcal{Z}, \boldsymbol{\beta})\}\exp\{-\frac{1}{2}(\boldsymbol{\lambda} - \boldsymbol{\lambda}_{MP})^T\mathbf{A}(\boldsymbol{\lambda} - \boldsymbol{\lambda}_{MP})\} \end{aligned} \tag{3.45}$$

where $\mathbf{A}$ is the curvature of the posterior and is computed as the hessian:

$$\mathbf{A} = \nabla_{\boldsymbol{\lambda}}\nabla_{\boldsymbol{\lambda}}(\ln\{\mathbf{p}(\boldsymbol{\lambda}|\mathcal{Z}, \boldsymbol{\beta})\})|_{\boldsymbol{\lambda}_{MP}} \tag{3.46}$$

Note that the approximation is nothing but expanding the logarithm of the integrand using Taylor series and retaining terms to second order. Also note that the first order term vanishes at the modes $\boldsymbol{\lambda}_{MP} = \{\boldsymbol{\lambda}_{(1,MP)}, \cdots, \boldsymbol{\lambda}_{(M,MP)}\}$. For computing the marginal likelihood of the hyperparameters $\boldsymbol{\beta}$ we again make Laplace approximation:

$$\begin{aligned} p(\mathcal{Z}|\boldsymbol{\beta}) &= \int p(\mathcal{Z}|\boldsymbol{\lambda})p(\boldsymbol{\lambda}|\boldsymbol{\beta})d\boldsymbol{\lambda} \\ &= \int \exp\{-\mathcal{H}(\boldsymbol{\lambda}, \mathcal{Z}, \boldsymbol{\beta})\}d\boldsymbol{\lambda} \\ &\simeq \exp\{-\mathcal{H}(\boldsymbol{\lambda}_{MP}, \mathcal{Z}, \boldsymbol{\beta})\}\int \exp\{-\frac{1}{2}(\boldsymbol{\lambda} - \boldsymbol{\lambda}_{MP})^T\mathbf{A}(\boldsymbol{\lambda} - \boldsymbol{\lambda}_{MP})\}d\boldsymbol{\lambda} \\ &= \exp\{-\mathcal{H}(\boldsymbol{\lambda}_{MP}, \mathcal{Z}, \boldsymbol{\beta})\}(2\pi)^{N/2}|\mathbf{A}|^{-1/2} \end{aligned} \tag{3.47}$$

The mode $\boldsymbol{\lambda}_{MP}$ is obtained by optimization of the weights posterior using *Iterative Re-weighted Least Square*, the details of which are provided in the Appendix A. The size of error bars matrix $\mathbf{A}$ scales with the number of experts M and may cause the optimization to become computationally expensive. We therefore assume block diagonal form for the covariance matrix $\mathbf{A}$ and independently optimize the posterior for multiple gate parameters:

$$\boldsymbol{\lambda}_{MP} = \arg max_{[\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \cdots, \boldsymbol{\lambda}_M]} \left\{ \sum_{i=1}^{M}\sum_{n=1}^{N} z_i^{(n)} \log \rho_i \left\{ \boldsymbol{\phi}(\mathbf{r}^{(n)}) \right\} - \sum_{i=1}^{M} \frac{1}{2}\boldsymbol{\lambda}_i^T\mathbf{B}_i\boldsymbol{\lambda}_i \right\} \tag{3.48}$$

where $\mathbf{B}_i = diag\{\boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_K\}$ are the hyperparameters corresponding to weight parameters of the gate function $\boldsymbol{\lambda}_i$ corresponding to i^{th} expert. The above approximation allows us to simplify the complex weight posterior as a multivariate Gaussian. The approximation is accurate as we expect the log-posterior to be unimodal due to negative definite Hessian (see appendix A).

3.4.3 Optimizing the Hyperparameters

Proceeding further with the sparse Bayesian learning, we follow common optimization framework for the expert and gate distribution under the assumption of independent hierarchical priors(both for the gates and the experts). The weights corresponding to large hyperparameters are removed from the set of basis vector as they have their posterior concentrated at zero and therefore deemed irrelevant to the learned mapping function. The MAP estimates of the hyperparameters corresponding to i^{th} expert $\boldsymbol{\alpha}_{i,MAP}, \boldsymbol{\Omega}_{i,MAP}$ and the corresponding gate function $\boldsymbol{\beta}_{i,MAP}$, that maximize the marginal likelihood $p(\mathcal{D}|\boldsymbol{\alpha}_i, \boldsymbol{\Omega}_i)$ and $p(\mathcal{Z}|\boldsymbol{\beta}_i)$ respectively (eqn. 3.37) cannot be obtained in a closed form, and is estimated iteratively (by differentiating the marginal likelihoods and equating them to zero).

$$\alpha_{i,j}^{(k+1)} = \frac{\left[1 - \alpha_{i,j}^{(k)} \Lambda_{(j,j)}\right] + 2a}{\mu_i^2 + 2b} \tag{3.49}$$

$$(\boldsymbol{\Omega}_i^{-1})^{new} = \frac{\|\mathbf{x} - \mathbf{W}_i^T \Phi(\mathbf{r})\|^2 + 2d}{\left[N - \sum_j (1 - \Lambda_{(i,i)} \alpha_{i,j})\right] + 2c} \tag{3.50}$$

where (a, b) and (c, d) are the parameters of the gamma priors over $\boldsymbol{\alpha}_i$ and $\boldsymbol{\Omega}_i$ respectively. The learning algorithm repeatedly applies the hyperparameter estimates to prune off the weights $W_{(i,j)}$ that have large $\alpha_{(i,j)}$.

The learning of Bayesian mixture of experts model proceeds by iteratively clustering the dataset and estimating the parameters for the experts and gate distribution. We use regularized Expectation-Maximization framework to estimate the model parameters. In the next subsection we give a brief overview of proximal point algorithms on which the regularized EM framework is based.

3.4.4 Regularized Expectation Maximization

Mixture of Experts model is trained using maximum likelihood framework based on Expectation Maximization(EM). Given a set of labeled exemplars $\mathcal{D} = \{(\mathbf{r}^{(n)}, \mathbf{x}^{(n)}) \mid i = n \dots N\}$, EM algorithm tries to estimate the unknown(missing) indicator variables $\mathcal{Z} = \{(\mathbf{r}^{(n)}, z_i^{(n)}) \mid i = 1 \dots M, n = 1 \dots N\}$ by maximizing the expected value of the augmented log likelihood over both observed and the missing variables $\ell_c(\mathcal{D}, \mathcal{Z}) = \log(p(\mathcal{D}, \mathcal{Z}|\Theta))$. Expectation step consists of estimation of the expected value of the hidden variables $\mathcal{Z}$ and the complete log likelihood $Q(\Theta, \Theta^k) = E\left[\ell_c(\mathcal{D}, \mathcal{Z}|\Theta) \mid \mathcal{D}, \Theta^k\right]$ using the value of the parameters Θ at the k^{th} iteration. In the M-step, we compute the new parameter estimates, Θ^{k+1} by maximizing the expected value of the complete log likelihood $Q(\Theta, \Theta^k)$ with respect to Θ. The incomplete log likelihood is expressed as:

$$\log(p(\mathcal{D}|\Theta)) = E\left[\log(p(\mathcal{D}, \mathcal{Z} \mid \Theta))|\mathcal{D}, \Theta^k\right] - E\left[\log(\frac{p(\mathcal{D}, \mathcal{Z}|\Theta)}{p(\mathcal{D}|\Theta)})|\mathcal{D}, \Theta^k\right] \tag{3.51}$$

$$\ell(\mathcal{D}) = Q(\Theta, \Theta^k) + H(\Theta, \Theta^k) \tag{3.52}$$

where $H(\Theta, \Theta^k) = -E\left[\log(p(\mathcal{Z}|\mathcal{D}, \Theta))|\mathcal{D}, \Theta^k\right]$. It follows from the Jensen's inequality and the fact $H(\Theta, \Theta^k) \geq 0$ that increase in the expected value of the complete log likelihood $\ell_c(\mathcal{D}, \mathcal{Z})$ also increases the incomplete likelihood $\ell(\mathcal{D})$. The standard EM algorithm has many drawbacks, prominent being its slow rate of convergence towards the end and tendency to overfit the data. We introduce a regularized EM (REM) algorithm to avoid overfitting and learn sparser mixture of experts models. The key motivation behind the regularized EM is to penalize complex models and promote compact models that have lesser tendency to overfit the data.

We reformulated the learning of Mixture of Experts using *proximal point algorithm*[29]. For a concave objective function $\mathcal{F}(\Theta)$ a generalized proximal point algorithm is defined by the iteration:

$$\Theta^{(k+1)} = \arg\, max_{\Theta}\left[\mathcal{F}(\Theta) - \psi^{(k)}\mathbf{d}(\Theta, \Theta^{(k)})\right] \tag{3.53}$$

where $\mathbf{d}(\Theta, \Theta^{(k)}) \geq 0$ is a non-negative distance-like penalty function and $\psi^{(k)}$ is a sequence of positive numbers. Expectation-Maximization is a special case of proximal point algorithm

[50] with $\psi^{(k)} = 1$ and a Kullback-type proximal penalty:

$$\begin{aligned}\Theta^{(k+1)} &= \arg\, max_{\Theta} Q(\Theta, \Theta_k) \\ &= \arg\, max_{\Theta} \left\{ \log(p(\mathcal{D}, \mathcal{Z}|\Theta)) + E\left[\log(\frac{p(\mathcal{D}, \mathcal{Z}|\Theta)}{p(\mathcal{D}|\Theta)}) | \mathcal{D}, \Theta^{(k)} \right] \right\}\end{aligned} \tag{3.54}$$

This is equivalent to maximizing the following equation:

$$\Theta^{(k+1)} = \arg\, max_{\Theta} \left\{ \log(p(\mathcal{D}, \mathcal{Z}|\Theta)) + E\left[\log\frac{p(\mathcal{Z}|\mathcal{D}, \Theta)}{p(\mathcal{Z}|\mathcal{D}, \Theta^{(k)})} | \mathcal{D}, \Theta^{(k)} \right] \right\} \tag{3.55}$$

Each iteration of EM is guaranteed to increase the complete log likelihood $\ell_c(\mathcal{D}, \mathcal{Z})$ (and the incomplete log likelihood $\ell(\mathcal{D})$). Regularized EM algorithm is defined by the iteration in the M step as:

$$\begin{aligned}\Theta^{(k+1)} &= \arg\, max_{\Theta} \left[\ell_c(\mathcal{D}, \mathcal{Z}) - \Phi\left(\Theta, \Theta^{(k)}\right) \right] \\ &= \arg\, max_{\Theta} \left[\ell(\mathcal{D}) - \Phi\left(\Theta, \Theta^{(k)}\right) - E\left[\log\frac{p(\mathcal{Z}|\mathcal{D}, \Theta^{(k)})}{p(\mathcal{Z}|\mathcal{D}, \Theta)} | \mathcal{D}, \Theta^{(k)} \right] \right]\end{aligned} \tag{3.56}$$

The penalty term satisfies the condition $\Phi(\Theta, \Theta^{(k)}) \geq 0$ and is iteration dependent, non-negative value. We can immediately prove the convergence of the Regularized EM by adding another penalty term $\Phi(\Theta, \Theta^{(k)})$ to the term $\psi^{(k)}\mathbf{d}(\Theta, \Theta^{(k)})$ in the (3.53). In [135] Rockafeller showed that superlinear convergence is achieved when the sequence $\psi^{(k)}$ converges to 0. Note that for the k^{th} iteration:

$$\begin{aligned}\ell^{(k+1)}(\mathcal{D}) - \ell^{(k)}(\mathcal{D}) \geq\ & \Phi\left(\Theta^{(k+1)}, \Theta^{(k)}\right) - \Phi\left(\Theta^{(k)}, \Theta^{(k)}\right) \\ & + E\left[\log\frac{p(\mathcal{Z}|\mathcal{D}, \Theta^{(k)})}{p(\mathcal{Z}|\mathcal{D}, \Theta^{(k+1)})} | \mathcal{D}, \Theta^{(k)} \right] - E\left[\log\frac{p(\mathcal{Z}|\mathcal{D}, \Theta^{(k)})}{p(\mathcal{Z}|\mathcal{D}, \Theta^{(k)})} | \mathcal{D}, \Theta^{(k)} \right]\end{aligned} \tag{3.57}$$

Where $E\left[\log\frac{p(\mathcal{Z}|\mathcal{D}, \Theta^{(k)})}{p(\mathcal{Z}|\mathcal{D}, \Theta^{(k)})} | \mathcal{D}, \Theta^{(k)} \right] = 0$ and $E\left[\log\frac{p(\mathcal{Z}|\mathcal{D}, \Theta^{(k)})}{p(\mathcal{Z}|\mathcal{D}, \Theta^{(k+1)})} | \mathcal{D}, \Theta^{(k)} \right] \geq 0$. The additional penalty term $\Phi\left(\Theta^{(k)}, \Theta^{(k)}\right) = 0$. In order to have monotonic convergence of the log likelihood using the optimization equation (3.56), the choice of the penalty term should be such that $\Phi\left(\Theta^{(k+1)}, \Theta^{(k)}\right) \geq 0$. For the penalty function, we may use a weight decay prior (corresponding to ridge regression)[194], to regularize the estimated function. Instead we employ hierarchical priors with automatic relevance determination (ARD) mechanism in our framework that promotes compact models, by selecting only relevant basis vectors from the training

set. As we show in the next section, the choice of the penalty function, satisfies the requirement for monotonic increase of the likelihood function and thus leads to faster convergence of the expectation maximization iterations.

3.4.5 Regularized Expectation Maximization for Bayesian Mixture of Experts

We use zero-mean Gaussian distribution over the parameters as the prior that constitute a quadratic penalty term in the cost function being optimized in the M-step. The prior distributions on the smoothing penalty parameters of the i^{th} expert and the corresponding gate function are conditioned on the hyperparameters $\Theta = \{(\mathbf{W}_i, \boldsymbol{\lambda}_i | \boldsymbol{\Omega}_i, \boldsymbol{\alpha}_i, \boldsymbol{\beta}_i)\}$, and are estimated using the most probable values of the hyperparameters. The most probable values of the hyperparameters $\boldsymbol{\Omega}_i^{MP}, \boldsymbol{\alpha}_i^{MP}, \boldsymbol{\beta}_i^{MP}$ are obtained by maximizing their marginal likelihood(*type-II maximum likelihood*). The regularization penalty for the M-step is a quadratic term that is independent of the previous iteration and has the form $(\mathbf{W}^{(k)})^T \mathbf{A}^{(k)} \mathbf{W}^{(k)}$ where the most probable hyperparameters $\mathbf{A}^{(k)} = \text{diag}\{\alpha_1, \alpha_2, \cdots, \alpha_N\}$ in the k^{th} iteration is obtained by optimizing the marginal likelihood at every M-step. It is straight forward to prove the monotonic increase in the likelihood as the $(\mathbf{W}^{(k+1)})^T \mathbf{A}^{(k+1)} \mathbf{W}^{(k+1)} - (\mathbf{W}^{(k)})^T \mathbf{A}^{(k)} \mathbf{W}^{(k)} \geq 0$.

Because the regularization term biases the searching space to some extent, we expect the REM algorithm to also converge faster than the standard EM algorithm. We confirm this in the experiments on the toy dataset discussed in the §3.4.6. In addition to faster convergence rate, the hierarchical priors lead to a special case of EM algorithm that promotes *sparsity* of weights at every EM iteration of the learning algorithm.

The Bayesian mixture of experts is trained by iteratively estimating the expected values of the missing data (the indicator variables $\mathcal{Z}$) in the E Step followed by the maximization of the penalized likelihood, which in our formulation is the posterior over the weights parameters of the BME. The complete posterior over all the parameters $\Theta = \{\mathbf{W}, \boldsymbol{\lambda}, \boldsymbol{\Omega}, \boldsymbol{\alpha}, \boldsymbol{\beta}\}$ is intractable and does not have a convenient form that can be analytically optimized. Instead, we optimize the posterior over the weight parameters with MAP estimates of the hyperparameters $\{\boldsymbol{\Omega}^{MP}, \boldsymbol{\alpha}^{MP}, \boldsymbol{\beta}^{MP}\}$ obtained at each M-step. The posterior over the expert and gate weight

parameters can be rewritten as:

$$\begin{aligned} & p(\mathbf{W}, \boldsymbol{\lambda} | \boldsymbol{\Omega}^{MP}, \boldsymbol{\alpha}^{MP}, \boldsymbol{\beta}^{MP}, \mathcal{Z}, \mathcal{D}) \\ \propto & \left\{ \prod_{i=1}^{M} p(\mathbf{W}_i | \boldsymbol{\alpha}_i^{MP}, \boldsymbol{\Omega}_i^{MP}, \mathcal{D}, \mathcal{Z}) \right\} p(\boldsymbol{\lambda} | \boldsymbol{\beta}^{MP}, \mathcal{Z}) \\ = & \prod_{i=1}^{M} \left\{ p(\mathcal{D} | \mathbf{W}_i, \boldsymbol{\Omega}_i^{MP}, \mathcal{Z}) p(\mathbf{W}_i | \boldsymbol{\alpha}_i^{MP}, \boldsymbol{\Omega}_i, \mathcal{Z}) \right\} p(\mathcal{Z} | \boldsymbol{\lambda}) p(\boldsymbol{\lambda} | \boldsymbol{\beta}^{MP}) \\ = & \prod_{n=1}^{N} \left\{ \prod_{i=1}^{M} \left\{ p(\mathbf{x}^{(n)} | \mathbf{r}^{(n)}, \mathbf{W}_i, \boldsymbol{\Omega}_i^{MP})^{z_i^{(n)}} p(\mathbf{W}_i | \boldsymbol{\alpha}_i^{MP}, \boldsymbol{\Omega}_i, \mathcal{Z}) \right\} \prod_{i=1}^{M} p(z_i^{(n)} | \mathbf{r}^{(n)}, \boldsymbol{\lambda}_i) p(\boldsymbol{\lambda}_i | \boldsymbol{\beta}_i^{MP}) \right\} \end{aligned} \tag{3.58}$$

In the above equations we have factorized the posterior over the expert weights $\mathbf{W}_i$ and the gate parameters $\boldsymbol{\lambda} = \{\boldsymbol{\lambda}_i \mid i = 1 \dots M\}$ into product of likelihoods and priors. The likelihood function for the gates is expressed as $p(z_i^{(n)} | \mathbf{r}^{(n)}, \boldsymbol{\lambda}_i) = \rho_i \left\{ \boldsymbol{\phi}(\mathbf{r}^{(n)}) \right\}^{z_i^{(n)}}$. Having defined the posterior distribution, we can now formulate the expectation maximization algorithm.

Expectation Step: The expectation step involves computing the expected values of the missing variables $E[z_i^{(n)}] = p(z_i^{(n)} = 1 | \mathbf{x}^{(n)}, \mathbf{r}^{(n)})$. We can estimate this conditional distribution using Bayes' rule

$$p(z_i^{(n)} = 1 | \mathbf{r}^{(n)}, \mathbf{x}^{(n)}, \mathbf{W}_i, \boldsymbol{\lambda}_i, \boldsymbol{\Omega}_i) = \frac{p(\mathbf{x}^{(n)} | \mathbf{W}_i, \mathbf{r}^{(n)}, z_i^{(n)} = 1, \boldsymbol{\Omega}_i^{-1}) p(z_i^{(n)} = 1 | \mathbf{r}^{(n)}, \boldsymbol{\lambda}_i)}{\sum_{m=1}^{M} p(\mathbf{x}^{(n)} | \mathbf{W}_m, \mathbf{r}^{(n)}, z_m^{(n)} = 1, \boldsymbol{\Omega}_m) p(z_m^{(n)} = 1 | \mathbf{r}^{(n)}, \boldsymbol{\lambda}_m)} \tag{3.59}$$

Note here that the expected values of $z_i^{(n)}$ may be non-integral as they represent the soft probability weight that the point $(\mathbf{x}^{(n)}, \mathbf{r}^{(n)})$ is mapped using the i^{th} expert and follows the constraint $\sum_{i=1}^{M} E[z_i^{(n)}] = 1$.

Maximization Step: The maximization step estimates the parameters $\mathbf{W}_i$ and $\boldsymbol{\lambda}_i$ for each expert i and the gate function respectively. It uses the soft clustering weights (expected values of the hidden variables $E[z_i^{(n)}]$) obtained in the E-step to compute the expected value of the

posterior. We maximize the log of the posterior as it is more analytically tractable:

$$\log\{p(\mathbf{W},\boldsymbol{\lambda},\boldsymbol{\Omega},\boldsymbol{\alpha},\boldsymbol{\beta}|\mathcal{Z},\mathcal{D})\} \propto \sum_{i=1}^{M}\sum_{n=1}^{N} E[z_i^{(n)}]\log\{p(\mathbf{x}^{(n)}|\mathbf{W}_i\mathbf{r}^{(n)},\boldsymbol{\Omega}_i^{-1})\}+ \sum_{i=1}^{M}\log\{p(\mathbf{W}_i|\boldsymbol{\alpha}_{i,MP},\boldsymbol{\Omega}_{i,MP})\}+ \sum_{i=1}^{M}\sum_{n=1}^{N} z_i^{(n)}\{\log\{p(z_i^{(n)}|\mathbf{r}^{(n)},\boldsymbol{\lambda}_i)\} + \sum_{i=1}^{M}\log\{p(\boldsymbol{\lambda}_i|\boldsymbol{\beta}_{i,MP})\} \tag{3.60}$$

The update equations for $\mathbf{W}_i$ and $\boldsymbol{\lambda}_i$ can be obtained by differentiating the log posterior with respect to these parameters and equating them to 0. Note, that terms for the experts and the gate distributions can be optimized separately to obtain the iterative updates for their respective parameters:

$$\mathbf{W}_i^{(k+1)} = \arg\, max_{\mathbf{W}_i}\{\sum_{n=1}^{N} E(z_i^{(n)})\log\,\{p(\mathbf{x}^{(n)}|\mathbf{r}^{(n)},\mathbf{W}_i^{(k)},z_i^{(n)},\boldsymbol{\Omega}_i)\} - (\mathbf{W}_i^{(k)})^T\boldsymbol{\alpha}_i\mathbf{W}_i^{(k)}\} \tag{3.61}$$

$$\boldsymbol{\lambda}_i^{(k+1)} = \arg\, max_{\boldsymbol{\lambda}_i}\{\sum_{n=1}^{N} E(z_i^{(n)})\log\,\{p(z_i^{(n)}|\mathbf{r}^{(n)},\boldsymbol{\lambda}_i^{(k)})\} - (\boldsymbol{\lambda}_i^{(k)})^T\boldsymbol{\beta}_i\boldsymbol{\lambda}_i^{(k)}\} \tag{3.62}$$

For estimating the expert weights $\mathbf{W}_i$ we optimize the equation (3.61). This is weighted generalized least square problem and can be solved by reweighting the data terms $\{\mathbf{x}^{(n)},\boldsymbol{\phi}(\mathbf{r}^{(n)})\}$ as $\left\{\sqrt{E(z_i^{(n)})}\mathbf{x}^{(n)},\sqrt{E(z_i^{(n)})}\boldsymbol{\phi}(\mathbf{r}^{(n)})\right\}$ and fitting the regressor to the data terms. This effectively reduces the influence of the data points that are mapped by different experts such that the regression function fits locally in the associated cluster. The posterior distribution for the expert weights can be exactly obtained(analytically) as Gaussian distribution. The Bayesian model selection step for selecting a subset of relevant weights for the experts is carried out by maximizing the marginal likelihood of the hyperparameters obtained by analytically integrating out the weight parameters. The weights corresponding to the hyperparameters $\{\boldsymbol{\alpha}_{MP},\boldsymbol{\Omega}_{MP}\}$ having high value are deemed irrelevant, and are not used for learning the expert mapping. This implicitly penalizes more parametrized models by reducing the number of weight parameters in the expert kernel mapping. The maximization of the marginal likelihood is an iterative process and involves selection of relevant weights using the MAP estimates of the hyperparameters, followed by kernel ridge regression to estimate the weight parameters of the experts.

The gate weights are estimated using a doubly looped iterative process by maximizing the (3.62) which is equivalent to multi-category classification problem with inputs as the observation variables $\mathbf{r}^{(n)}$ and targets as the soft clustering weights $E(z_i^{(n)})$ that are computed during the E-step. The posterior optimization is intractable analytically and requires Laplace approximation to express the distribution as Gaussian. The mode of the posterior is obtained using *Iterative Re-weighted Least Square*(IRLS), as discussed in detail in §3.4.2 and appendix A. The computational cost for the exact updates for the gates parameters increases exponentially with the number of experts. In principle, we may employ faster methods based on forward selection for learning experts and gate distributions. Forward basis selection [66, 181] algorithms commence with an empty basis set and iteratively optimize the cost function (*Type II Maximum likelihood*) by sequentially selecting basis functions. This is in contrast to backward elimination methods that start with the large basis set containing all the exemplars and iteratively eliminate non-relevant basis functions using automatic relevance determination (ARD) mechanism. The decision to add a basis function to the basis set is based on the contribution of the basis function towards minimizing the cost. If adding a basis function improves the overall likelihood, it is included in the basis set. Similarly, for every addition to the basis set, the contributions of the rest of the bases functions are evaluated. The bases functions rendered irrelevant due to addition are deleted from the basis set. Both addition and deletion of a basis function may increase the likelihood objective function. The candidate basis functions may be chosen randomly or sequentially and are added to the basis set using an automatic relevance determination mechanism. The iterations are repeated until the change in the objective function (marginal likelihood) is less than a fixed threshold.

3.4.6 Evaluation of BME on Toy Dataset

We explain the Bayesian mixture of experts modeling through an illustrative toy example. Our dataset consists of about 250 values of x generated uniformly in $(0, 1)$ and then evaluated as $r = x + 0.3\sin(2\pi x) + \epsilon$, with ϵ drawn from a zero mean Gaussian with standard deviation 0.05 (also shown in [30]). Notice that $p(x|r)$ is multimodal and for the input r around 0.5, there are 3 modes of the conditional distribution. In fig. 3.2 we show different models that use linear and Gaussian kernel experts, as well as different gating functions, as presented in §3.4.1. First row

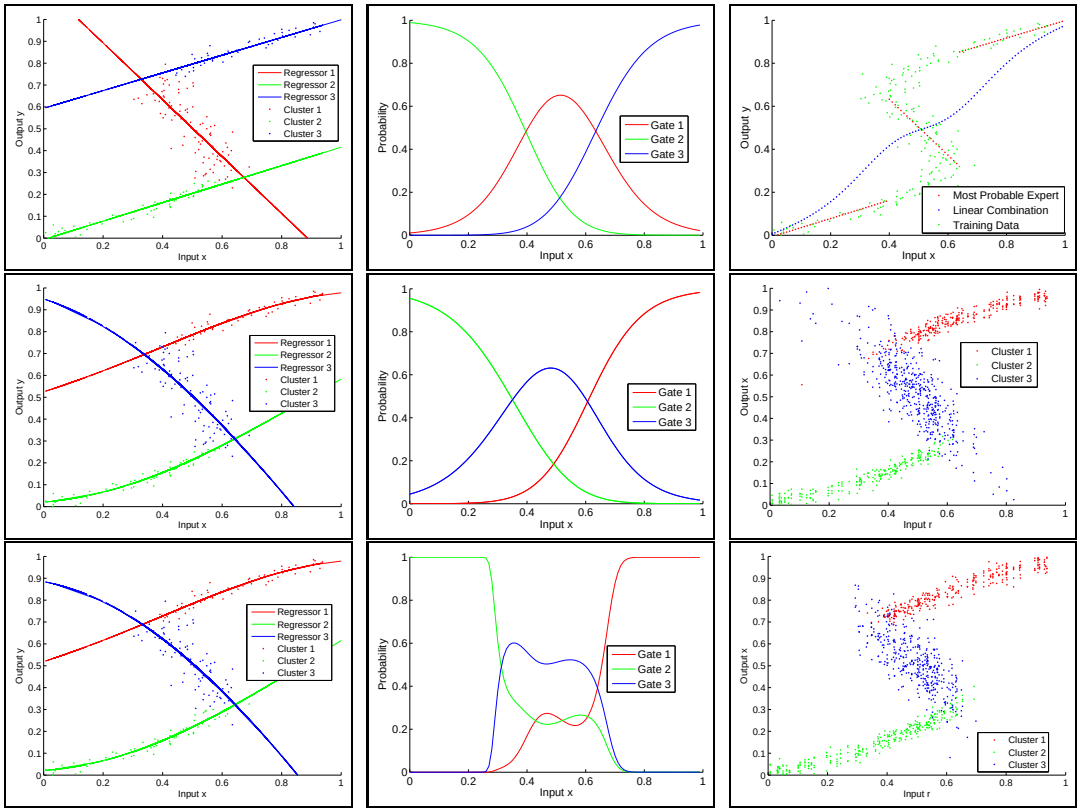


Figure 3.2: Experts and gates fitted to a synthetic toy dataset using different Bayesian Mixture of Experts models. (*top row*) Linear experts fitted using conditional BME and the corresponding gate distributions. The gate function is a softmax function with log linear inputs. On the right we show final prediction as weighted linear combination of expert predictions and most probable experts. (*middle row*) Kernel experts fitted to the multimodal toy dataset. The gate function is the softmax function. The output on the right are obtained by sampling the gate distribution followed by the sampling from the chosen expert distribution. (*bottom row*) Kernel experts and gate functions with kernelized inputs that generate more accurate gate distributions at the tails. The output sampled from these gates exhibit lower variance

shows the Bayesian Mixture of Experts with three linear experts and log-linear gate distribution fitted to the multimodal dataset. The plot on the right shows the predictions, both as the most probable expert outputs and as the gate-weighted linear combination of expert outputs. Last two rows show the results of BME using kernel experts and softmax gate functions. Last row has the softmax gate activation function with kernel inputs. Notice the improved gate variance at the tails due to kernel inputs (as opposed to linear inputs in the softmax gate function). Right column compares the data generated by sampling gate distributions.

Fig. 3.3 shows the plot for conditional likelihoods with the EM iterations for the five different mixture of experts implementations. For the experts we used linear regression learned

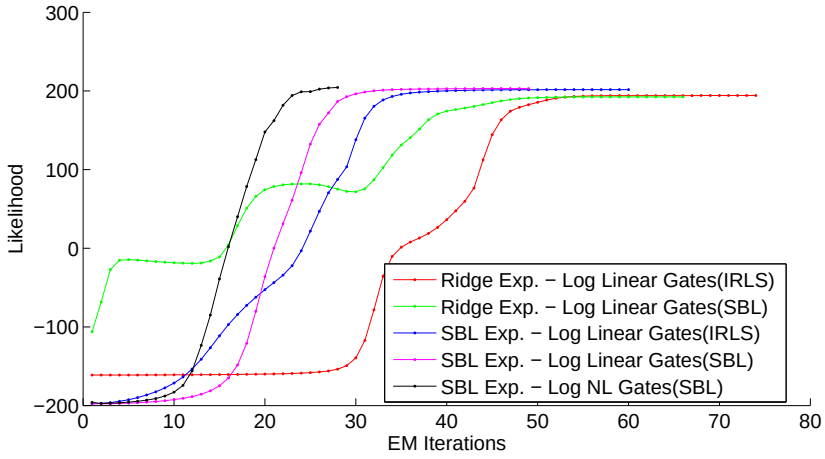


Figure 3.3: Change in likelihoods with the EM iterations for various implementations of mixture of experts models. In the figure we compare the learning rate of mixture of experts model where the kernelized experts are learned either using ridge regression(Ridge Exp.) or sparse Bayesian learning(SBL). The gates are learned using IRLS algorithm or sparse Bayesian learning(SBL) and may have log linear or log nonlinear model. The convergence rate is maximum for the ME with kernel experts and log nonlinear gates.

using least square estimate (standard EM) and the kernel basis implementation learned in Bayesian framework (regularized EM). For the Gates distribution we used the softmax function learnt using Iterative Re-weigted Least Square(IRLS) method, Bayesian softmax function and Bayesian multinomial log-nonlinear function(3.42), learnt using penalized likelihood maximization (MAP estimate). The plots clearly shows the improvement in the rate of increase of likelihood with iterations due to regularized EM algorithm and non-linear parameter search in Bayesian framework.

3.5 Mixture of Experts Based on Joint Density

Learning gate parameters of mixture of experts(ME) model is computationally expensive and a variety of extensions to ME model have been proposed in the past. Amongst most notable and relevant alternative to the ME implementation used in this thesis, was proposed by Xu *et.* al[199]. Their formulation estimate the conditional density $p(\mathbf{x}|\mathbf{r}, \Theta)$ by modeling the joint distribution over inputs and outputs $p(\mathbf{x}, \mathbf{r}|\Theta)$ and then obtain the conditional using Bayes'

rule.

$$p(\mathbf{x}|\mathbf{r},\Theta) = \frac{p(\mathbf{r},\mathbf{x}|\Theta)}{\int p(\mathbf{r},\mathbf{x}|\Theta)d\mathbf{x}} \tag{3.63}$$

This implementation leads to a different parametric form for the gates and essentially eliminates the double loop optimization used for learning gates parameters in the Bayesian Mixture of Experts. In the new architecture, optimization of gate parameters can be done in closed form, leading to significantly faster convergence of the Expectation Maximization. Rosales *et.* al[139] have also employed this architecture of ME in their implementation of Specialized Mapping Architecture. Assume for generality, a full covariance mixture model of the joint distribution over input-output pairs $(\mathbf{x},\mathbf{r})$, given by:

$$p\left(\begin{pmatrix}\mathbf{r}\\ \mathbf{x}\end{pmatrix}\Big|\Theta\right) = \sum_{i=1}^{M}\rho_i\mathcal{N}\left(\begin{pmatrix}\mathbf{r}\\ \mathbf{x}\end{pmatrix}\Big|\begin{pmatrix}\boldsymbol{\mu}_i^r\\ \boldsymbol{\mu}_i^x\end{pmatrix},\begin{pmatrix}\boldsymbol{\Sigma}_i^{rr} & \boldsymbol{\Sigma}_i^{rx}\\ \boldsymbol{\Sigma}_i^{xr} & \boldsymbol{\Sigma}_i^{xx}\end{pmatrix}\right) \tag{3.64}$$

Note here that the joint Gaussian distribution3.64 can be decomposed as[139]:

$$\mathcal{N}\left(\begin{pmatrix}\mathbf{r}\\ \mathbf{x}\end{pmatrix}\Big|\begin{pmatrix}\boldsymbol{\mu}_i^r\\ \boldsymbol{\mu}_i^x\end{pmatrix},\begin{pmatrix}\boldsymbol{\Sigma}_i^{rr} & \boldsymbol{\Sigma}_i^{rx}\\ \boldsymbol{\Sigma}_i^{xr} & \boldsymbol{\Sigma}_i^{xx}\end{pmatrix}\right) \tag{3.65}$$

$$= \mathcal{N}(\mathbf{r}|\boldsymbol{\mu}_i^r,\boldsymbol{\Sigma}_i^{rr})\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_i^x + \boldsymbol{\Sigma}_i^{xr}(\boldsymbol{\Sigma}_i^{rr})^{-1}(\mathbf{r}-\boldsymbol{\mu}_i^r), \boldsymbol{\Sigma}_i^{xx} - \boldsymbol{\Sigma}_i^{xr}(\boldsymbol{\Sigma}_i^{rr})^{-1}\boldsymbol{\Sigma}_i^{rx}) \tag{3.66}$$

Using the decomposition(3.65) and the Bayes' rule(3.63), we can express the conditional distribution $p(\mathbf{x}|\mathbf{r},\Theta)$:

$$\begin{aligned} &= \frac{\sum_{i=1}^{M}\rho_i\mathcal{N}(\mathbf{r}|\boldsymbol{\mu}_i^r,\boldsymbol{\Sigma}_i^{rr})\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_i^x + \boldsymbol{\Sigma}_i^{xr}(\boldsymbol{\Sigma}_i^{rr})^{-1}(\mathbf{r}-\boldsymbol{\mu}_i^r), \boldsymbol{\Sigma}_i^{xx} - \boldsymbol{\Sigma}_i^{xr}(\boldsymbol{\Sigma}_i^{rr})^{-1}\boldsymbol{\Sigma}_i^{rx})}{\sum_{i=1}^{M}\rho_i\mathcal{N}(\mathbf{r}|\boldsymbol{\mu}_i^r,\boldsymbol{\Sigma}_i^{rr})} \\ &= \sum_{i=1}^{M}\frac{\rho_i\mathcal{N}(\mathbf{r}|\boldsymbol{\mu}_i^r,\boldsymbol{\Sigma}_i^{rr})}{\sum_{i=1}^{M}\rho_i\mathcal{N}(\mathbf{r}|\boldsymbol{\mu}_i^r,\boldsymbol{\Sigma}_i^{rr})}\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_i^x + \boldsymbol{\Sigma}_i^{xr}(\boldsymbol{\Sigma}_i^{rr})^{-1}(\mathbf{r}-\boldsymbol{\mu}_i^r), \boldsymbol{\Sigma}_i^{xx} - \boldsymbol{\Sigma}_i^{xr}(\boldsymbol{\Sigma}_i^{rr})^{-1}\boldsymbol{\Sigma}_i^{rx}) \end{aligned} \tag{3.67}$$

For the mixture of experts model based on the above formulation, we need to enforce additional constraints that $\mathbf{x} = \mathbf{W}^T\mathbf{r}$. We can now associate the conditionals in the (3.5) to the components of the mixture of experts model - gates and the expert conditionals. Specifically the mixture of experts model is learned as:

$$p(\mathbf{x}|\mathbf{r},\Theta) = \sum_{i=1}^{M} g(\mathbf{r}|\boldsymbol{\delta}_i)\mathcal{N}(\mathbf{x}|\mathbf{W}_i\mathbf{r},\boldsymbol{\Omega}_i^{-1}) \tag{3.68}$$

where the parameter for the gate distributions include the component mixing proportions ρ_i and the mean and variances of the Gaussians $(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$. For the expert conditionals $\mathcal{N}(\mathbf{x}|\mathbf{W}_i\mathbf{r}, \boldsymbol{\Omega}_i^{-1})$ learned in sparse Bayesian settings, the parameters include $(\mathbf{W}_i, \boldsymbol{\alpha}_i, \boldsymbol{\Omega}_i)$, where $\boldsymbol{\alpha}_i$ are the covariance parameters of the hierarchical priors. The gate distribution is modeled as:

$$g_i^{(n)} = \frac{\rho_i \mathcal{N}(\mathbf{r}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i^{-1})}{\sum_{k=1}^{M} \rho_k \mathcal{N}(\mathbf{r}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k^{-1})} \tag{3.69}$$

The learning proceeds in a similar way as the Bayesian Mixture of Experts except the inference of gate distributions, which can be obtained in a closed form. To estimate the joint model, we introduce hidden variables $z_i^{(n)}$ with similar interpretation as for the conditional in §3.4.1. During the M-Step the posterior distribution over the hidden variables is computed based on (3.10) and (3.69):

$$\begin{aligned} h_i^{(n)} &= p(z_i^{(n)} = 1|\mathbf{x}^{(n)}, \mathbf{r}^{(n)}, \mathbf{W}_i, \boldsymbol{\lambda}_i, \boldsymbol{\Omega}_i, \rho_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \\ &= \frac{\rho_i \mathcal{N}(\mathbf{r}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i^{-1}) \mathcal{N}(\mathbf{x}|\mathbf{W}_i\mathbf{r}, \boldsymbol{\Omega}_i^{-1})}{\sum_{k=1}^{M} \rho_k \mathcal{N}(\mathbf{r}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k^{-1}) \mathcal{N}(\mathbf{x}|\mathbf{W}_k\mathbf{r}, \boldsymbol{\Omega}_k^{-1})} \end{aligned} \tag{3.70}$$

Note here that the gate distributions essentially denote the prior distributions in the above formulation:

$$g_i^{(n)} = p(z_i^{(n)} = 1|\mathbf{r}^{(n)}, \boldsymbol{\lambda}_i = (\rho_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)) \tag{3.71}$$

In the E-Step, the mixing proportions, means and covariance of the expert conditionals are obtained [92, 199]:

$$\rho_i = \frac{\sum_{n=1}^{N} h_i^{(n)}}{N} \tag{3.72}$$

$$\boldsymbol{\mu}_i = \frac{\sum_{n=1}^{N} h_i^{(n)} \mathbf{r}^{(n)}}{\sum_{n=1}^{N} h_i^{(n)}} \tag{3.73}$$

$$\boldsymbol{\Sigma}_i = \frac{\sum_{n=1}^{N} h_i^{(n)} (\mathbf{r}^{(n)} - \boldsymbol{\mu}_i)(\mathbf{r}^{(n)} - \boldsymbol{\mu}_i)^\top}{\sum_{n=1}^{N} h_i^{(n)}} \tag{3.74}$$

Fig. 3.4 shows the experts and gate distributions based on the conditional models estimated from the joint distribution (as described in §3.5). The right column shows the data generated from the model by sampling the gate distributions. Notice that the estimates for the experts are similar, but the gates are somewhat different from the conditional models estimated directly (fig. 3.2). Although the mixture of experts model based on the joint distributions produce well-fitted models for the toy dataset, we observed that these models are unstable when trained on

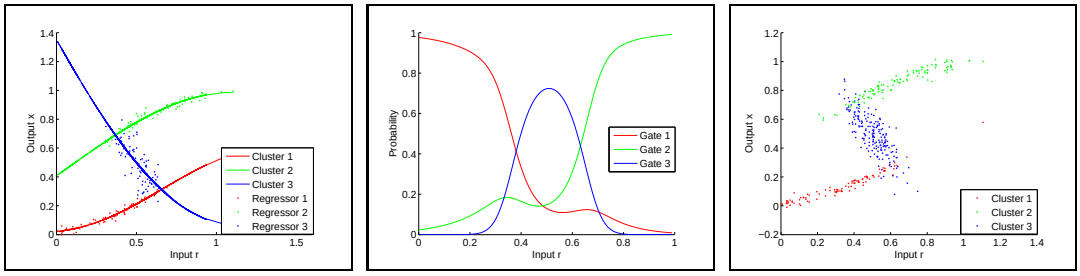


Figure 3.4: *(left)* The experts and *(middle)* the gate distributions of mixture of experts model learned by modeling the joint distribution over the input and output points.*(right)* shows the plot of data points obtained by sampling the gates and the expert distributions

high dimensional input-output pairs. We therefore do not pursue these models in training our discriminative framework.

3.6 Discussion

We have introduced a probabilistic framework for learning mixture of experts model in Bayesian framework. Further, we use Bayesian mixture of experts(BME) to model multimodal distributions obtained from inverse of many-to-one mappings. Modeling perceptual data such as inferencing 3D human pose from 2D images is one such example where multiple human pose states may generate similar 2D image observations under perspective projection. Strictly speaking, the inverse mapping from the observations to states is multi-valued and cannot be functionally approximated. BME allows us to learn these multi-modal distributions using multiple functions and probabilistically selecting the experts based on the observation. In addition to class of algorithms that directly model the conditional distribution of output given the input, we also develop Mixture of Experts models that model the joint distribution. Both the algorithms, as discussed in §3.4.1 and §3.5, can be formulated to train using sparse Bayesian learning and are useful for estimating compact conditional mixture of experts models.

Chapter 4

Discriminative 3D Human Pose Reconstruction

4.1 Introduction

While humans are adept in inferring 3D states of the objects using only relatively low-resolution visual observations, for the vision based systems, it is still a challenging task. From computational point of view, it is an ill-posed problem. There is loss of depth information due to perspective projection of 3D objects to 2D image. Inferring 3D states from 2D images is therefore a challenging problem and involves learning an inverse mapping that is one-to-many, as several distant 3D poses may generate similar 2D visual observations. Furthermore, for larger degree of articulation of the objects, the computational cost increases exponentially with the number of connected parts.

In this chapter we apply Bayesian mixture of experts learning to estimate and track articulated 3D human pose from monocular image sequences. The contents of this chapter are based on the publication, BM^3E *: Discriminative Density Propagation for Visual Tracking*, Cristian Sminchisescu, Atul Kanaujia, Dimitris Metaxas, *IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007.*

Human body has highly articulated structure with high degrees of freedom. The skeleton structure is organized as a hierarchy, with the root joint having global translation and rotation parameters. The rest of the skeletal segments are obtained by constructing global transformation using all the segments in the hierarchy that connect this segment to the root. The joint angles are represented in a local coordinate frame relative to the parent joint. This is to avoid the error in a single joint to distort the rest of the human pose. The global transformation is obtained by a series of translation offsets and local rotational transformations of the segments in the hierarchical path connecting the segment to the root.

Typically, human 3D pose is represented as relative joint angles of the links with respect to parent link and global 3D location of the root joint, coded as K-dimensional vector $\mathbf{x}\epsilon\mathcal{R}^K$. Representing 3D pose as joint angles (instead of joint locations) has an additional advantage that the motion capture data from one skeleton can be easily imported to another skeleton and used to deform a computer graphic character in animation software like Maya and Poser. A potential setback of using joint angles to encode 3D pose is that angular measurement is cyclical and angles separated by 360^o are the same. We overcome this problem by transforming the discontinuous joint angle space to continuous sinusoidal space and representing key joint angles with a pair of sine and cosine values.

The task of 3D human pose estimation is formulated as: given an observation (image descriptor) vector $\mathbf{r}\epsilon\mathcal{R}^L$, we want to infer the vector of joint angles $\mathbf{x}$ using statistical learning models. Assuming the vector spaces for poses $\mathcal{R}^K$ and observation $\mathcal{R}^L$ to be continuous, we aim to learn probabilistic conditional $p(\mathbf{x}|\mathbf{r})$ and use it for inferring 3D poses from the 2D observations. For a sequence of observations we propagate the filtered conditional $p(\mathbf{x}_i|\mathbf{R} = \{\mathbf{r}_1, \cdots, \mathbf{r}_i\})$ over time and use it to estimate the 3D pose using all the past observations.

As discussed in chapter 1, there exist two broad paradigms of 3D pose estimation techniques - Top-down models (Generative models) and bottom-up models (Discriminative or Predictive models). Both the techniques have their strengths and weaknesses.

Generative models directly learn the joint distribution $p(\mathbf{x}, \mathbf{r}) = p(\mathbf{r}|\mathbf{x})p(\mathbf{x})$ from the prior distribution over 3D joint angles space $p(\mathbf{x})$ and likelihood function $p(\mathbf{r}|\mathbf{x})$. The prior distribution explicitly models the poses and joint angles that humans can assume during various activities. The likelihood function is the probability of observing the image given the projection of 3D human model with the estimated pose state(joint angles) to 2D image plane. The likelihood function typically uses low-level, shapes (or appearance) based image features to determine similarity between the rendered 2D image and the observation. One of key challenges to generative approaches is modeling the complex likelihood function. A variety of low level features such as chamfer distance transform, edges and contours can be used to estimate the similarity. The posterior distribution over the 3D joint angle space is a highly multimodal distribution due to similarity of limbs to the background and to each other, self

occlusion, kinematic singularities and depth ambiguities. Hence there is a strong need to develop robust observation likelihood functions that are able to differentiate between various 3D poses that may generate similar 2D projections. A number of generative approaches have been proposed[39, 82, 57, 154, 161, 185, 88, 83, 48, 170, 173] in the past. Due to inherent ambiguity in the pose prediction, a few of these approaches[152, 154, 173, 57] represent the posterior as a set of samples that are temporally propagated using particle filters and resolved over multiple time steps by estimating optimal path trajectory.

Discriminative models provides a complementary approach that directly learn the conditional distribution $p(\mathbf{x}|\mathbf{r})$ directly from a set of labeled exemplars. Specifically, this involves reformulating the task of 3D human pose prediction as a regression problem whereby the mapping from 2D images to 3D human is learned as a mapping function, $\mathcal{F}(\mathbf{r}) : \mathcal{R}^L \rightarrow \mathcal{R}^K$ fitting a set of labeled exemplars. Despite of its apparent simplicity, it is not devoid of challenges, prominent being the inherent multi-valuedness of the mapping function. Furthermore, there is also a need for sufficiently representative, labeled training dataset to accurately learn the 2D-to-3D mapping. Fig 4.1 illustrates various frequently encountered ambiguities Learning a regression

Figure 4.1: **(a)**180^o Flipping ambiguity **(b)** Leg assignment ambiguity**(c)** Arm flipping ambiguity

function from 2D images to human poses in 3D is a non-trivial task and involves appropriate choice of image descriptors that can compactly summarize the semantic content of observed image. In addition, unlike generative models that are implicitly regularized, discriminative models need to be explicitly penalized for over-parameterization, in order to avoid overfitting in the absence of sufficient training data.

Nevertheless, discriminative modeling have been increasingly gaining popularity due to their simplicity and improved performance in scenarios where ample labeled training data is available. Discriminative models have been used in the past [138, 121, 148, 182, 10, 65] for

directly predicting human pose states from the visual observations. Some of these methods [10, 65] use single regressors to learn the mappings whereas [148, 182, 121] used a large database of exemplar poses to predict states using fast nearest neighbor search. Rosales and Sclaroff[138] used several forward mappings to learn the one-to-many function. They modeled the relationship between observations and states using either a joint distribution over states and observations, or as a conditional distribution. They fit a mixture of Gaussian model where each Gaussian is a distribution around the mapping functions. The priors for each of the mapping function is assumed to be constant. Similar approaches were adopted by [7] and also by [76] that modeled the joint distribution of silhouettes and poses using mixture of probabilistic PCA. In this chapter, we propose a 3D pose estimation framework using bottom-up (discriminative)

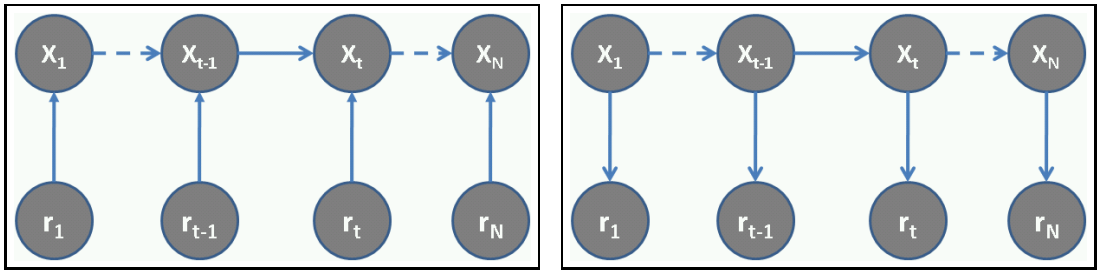


Figure 4.2: Bayesian network depicting density propagation in a *(left)* discriminative chain model and *(right)* generative chain model. Notice the reverse arrows in the discriminative model, that represent marginal independence of the 2 parent nodes with a common child node.

modeling. The mapping from 2D images to 3D pose is multi-valued and therefore we fit multiple regressors using the framework of *Bayesian Mixture of Experts(BME)*. BME provides an efficient model to learn these mappings in a probabilistic consistent framework based on automatic relevance determination(ARD) and expectation maximization (EM). Specifically, we use sparse Bayesian learning (SBL) for training the experts and the gate functions. As discussed in the previous chapter, SBL uses automatic relevance determination mechanism to select sparse basis set by optimizing the marginal likelihood over hyperparameters (*type-II Maximum Likelihood*). The learned models are compact and regularized. BME clusters the joint space of input features and the 3D poses into sub-domains and fits a mapping function $\mathcal{F}(\mathbf{r}) : \mathcal{R}^L \rightarrow \mathcal{R}^K$ within each of these domains. These domains are automatically learned based on the posterior distribution that represents the "goodness of fit" of an expert to the data points in the domain. In a typical 3D pose tracking framework, the pose variable $\mathbf{x}_t$ at each time step t is predicted

using the filtered distribution $p(\mathbf{x}_t|\mathbf{R}_t = \{\mathbf{r}_1, \mathbf{r}_2, \cdots, \mathbf{r}_t\})$. In the proposed discriminative tracking framework, we model this conditional as a mixture of Gaussian. At every time step we obtain this conditional by marginalization of the learned distribution $p(\mathbf{x}_t|\mathbf{r}_t, \mathbf{x}_{t-1})$. In order to avoid exponential increase in the number of hypotheses due to filtering, the low probability components are pruned off at each time step. Human body has highly articulated structure, with various connected components having high degrees of freedom (typically ≈ 60 DOF). The 3D pose state, which is typically represented using 3D joint angles, therefore has high dimensionality.

Learning a multi-valued mapping from the input image feature space to joint angle space involves learning multiple mappings for each of the joint angles either assuming conditional independence of output variates given the input features or modeling a broad conditional distribution to directly predict the entire joint state vector using the input feature. The former approach is computationally expensive and completely ignores the correlations between joints, whereas the latter approach require a large set of training exemplars to accurately learn a mapping function for directly predicting the entire human 3D pose state vector.

A number of human activities has much lower intrinsic dimensionality compared to the DOF of the joint angle state. This is due to strong correlation between the joint angles. For instance activities like running and walking will always have the two leg/arm joint angles moving coherently with respect to each other. For computational efficiency, we propose a framework to train discriminative models for estimating low-dimensional representations of human 3D pose states and 2D image features. The low-dimensional state space is induced by kernel transformation followed by de-correlation using PCA[196, 18]. Both the input features and the output states are projected to high dimensional feature space using kernel mapping. The low dimensional input and output features are obtained by applying linear principal component analysis(PCA) to the kernelized data points. For error analysis and reconstruction, the prediction in the feature space are projected back to original space using the learned pre-image.

We demonstrate our methods on real and motion capture-based test sequences, and give comparisons with the nearest neighbor and single-regressor based methods. In the next section we describe the details of discriminative propagation of the filtered conditional distribution over multiple time steps. In §4.3 we discuss the visual inferencing framework in kernel induced

space. §4.4 discusses the image descriptors used to encode the 2D image observations. We apply the discriminative pose estimation methodology on some real and synthetic sequences, and discuss the results in the §4.6. Finally we conclude the chapter with some discussion on the strengths and weaknesses of the framework in §4.7.

4.2 Discriminative Density Propagation

As discussed in the previous chapter, the conditional distribution $p(\mathbf{x}_t|\mathbf{r}_t)$ is learned as a mixture of Gaussians in an expectation maximization algorithm. The mixture of experts model is an effective framework for modeling one-to-many mappings and allows us to probabilistic weight multiple plausible outputs for a given input. However it is not sufficient to resolve ambiguities using observation alone, as similar image projections may be generated by different 3D poses(reffig. 4.1). Most of these 3D poses produce subtle differences in the 2D images that are often overlooked by the learned predictors and hence are difficult to disambiguate.

In practice, the 3D states are predicted using all the observation seen till current time step t, using the filtered conditional $p(\mathbf{x}_t|\mathbf{R}_t)$. This conditional distribution is obtained by marginalizing over states $\mathbf{x}_{t-1}$ in the previous time step:

$$\begin{aligned} p(\mathbf{x}_t|\mathbf{R}_t) &= \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t, \mathbf{x}_{t-1}|\mathbf{R}_{t-1}, \mathbf{r}_t) d\mathbf{x} \\ &= \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{R}_{t-1}, \mathbf{r}_t) p(\mathbf{x}_{t-1}|\mathbf{R}_{t-1}, \mathbf{r}_t) d\mathbf{x} \\ &= \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t) p(\mathbf{x}_{t-1}|\mathbf{R}_{t-1}) d\mathbf{x} \end{aligned} \tag{4.1}$$

As opposed to generative chain model fig. 4.2(a) where we model the observational likelihood $p(\mathbf{r}|\mathbf{x})$ under the assumption that observation are conditionally independent given the current state, in discriminative density propagation framework we exploit the learned the distributions, $p(\mathbf{x}_t|\mathbf{r}_t)$ and $p(\mathbf{x}_t|\mathbf{r}_t, \mathbf{x}_{t-1})$ to analytically compute the filtered conditional. The discriminative density propagation assume dependencies between the state and observation variables as depicted by the graphical model shown in fig. 4.2(b). In this Bayesian network, the nodes are conditionally independent of the ancestral nodes given the parents i.e.

$$\Rightarrow p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{R}_{t-1}, \mathbf{r}_t) = p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t) \tag{4.2}$$

For nodes with 2 parents $\mathbf{x}_t$, the node variables $\mathbf{x}_{t-1}$ and $\mathbf{r}_t$ are marginally independent.

$$\Rightarrow p(\mathbf{x}_{t-1}|\mathbf{R}_{t-1}, \mathbf{r}_t) = p(\mathbf{x}_{t-1}|\mathbf{R}_{t-1}) \tag{4.3}$$

Notice the difference in the direction of arrows of the generative and discriminative chain models in the fig. 4.2. For initializing the discriminative tracker, we assume that in the first frame the visual inputs are unambiguous so that $p(\mathbf{x}_1|\mathbf{R}_1) = p(\mathbf{x}_1|\mathbf{r}_1)$ is an accurate predictor of 3D pose. The filtered density is propagated for the subsequent time steps(4.1) using the learned distribution $p(\mathbf{x}_t|\mathbf{r}_t, \mathbf{x}_{t-1})$. In practice, the distribution $p(\mathbf{x}_t|\mathbf{r}_t)$, $p(\mathbf{x}_t|\mathbf{r}_t, \mathbf{x}_{t-1})$ and $p(\mathbf{x}_{t-1}|\mathbf{R}_{t-1})$ are mixture of Gaussians. The distribution $p(\mathbf{x}_t|\mathbf{r}_t, \mathbf{x}_{t-1})$ is learned with inputs $\mathbf{y}_t = [\mathbf{r}_t \quad \mathbf{x}_{t-1}]$ as

$$\mathcal{G}(\mathbf{x}_t, \mathbf{\Omega}) = p(\mathbf{x}_t|\mathbf{y}_t = [\mathbf{r}_t \quad \mathbf{x}_{t-1}]) = \sum_{i=1}^{M} p(z_i|\mathbf{y}_t)\mathcal{N}_i(\mathbf{x}_t|\mathbf{W}_i\Phi(\mathbf{y}_t), \mathbf{\Omega}_i^{-1}) \tag{4.4}$$

The marginalization in the eqn. 4.1 is carried out analytically:

$$p(\mathbf{x}_{t-1}|\mathbf{R}_{t-1}) = \mathcal{G}(\mathbf{x}_{t-1}, \mathbf{P})$$

$$p(\mathbf{x}_t|\mathbf{y}_t) = \mathcal{G}(\mathbf{A}\mathbf{y}_t, \mathbf{Q}) \text{ where } \mathbf{A} = \frac{d\mathbf{F}}{d\mathbf{y}}|_{\mathbf{x}_{t-1}}$$

$$\int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t)p(\mathbf{x}_{t-1}|\mathbf{R}_{t-1})d\mathbf{x} = \mathcal{G}(\mathbf{A}\mathbf{y}_t, \mathbf{A}\mathbf{P}\mathbf{A}^T + \mathbf{Q})$$

where $\mathbf{F}$ is the continuously differentiable, non-linear expert mapping function. The analytical integration approximates the non-linear mapping by linearization using the Jacobian matrix evaluated at the inputs. Here the temporal prior distribution $p(\mathbf{x}_{t-1}|\mathbf{R}_{t-1})$ is available from the previous time step. The resulting mixture of Gaussian has M^2 components and are reduced to M-component approximation by pruning off low weights Gaussian components. The weights of the Gaussian components are obtained from the gate distributions and may not cover all the modes of the posterior. In the proposed framework, we may also opt for cluster based approaches to prune off Gaussian components.

4.3 Bayesian Mixtures of Experts over Kernel Induced State Spaces (kBME)

For computational efficiency we reduce the dimensions of the input feature space and the output 3D joint angle space using kernel PCA. The proposed formulation is based on kernel dependency estimation(KDE) [196, 18] which uses kernelPCA to de-correlate the inputs $\mathbf{r}$ and the

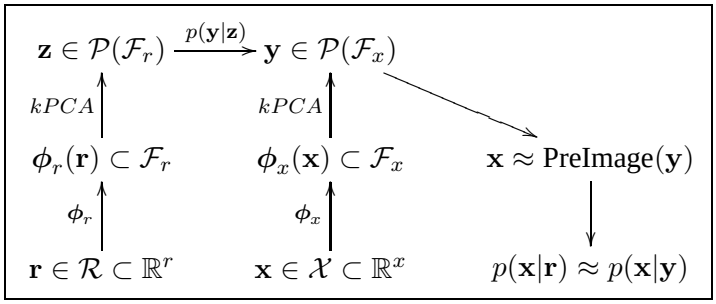


Figure 4.3: The kernel dependency estimation framework for visual inferencing of 3D poses in low dimensional space. Both the inputs $\mathbf{r}$ and the outputs $\mathbf{x}$ are first projected to the feature space, denoted as $\mathcal{F}_r$ and $\mathcal{F}_z$, using the rbf kernels $\boldsymbol{\phi}_r$ and $\boldsymbol{\phi}_x$ respectively. In the feature space, these points are de-correlated using PCA and projected along the learned orthogonal dimensions. This yields the low-dimensional representations of input and output in the learned subspace $\mathcal{P}(\mathcal{F}_r)$ and $\mathcal{P}(\mathcal{F}_x)$ as $\mathbf{z}$ and $\mathbf{y}$ respectively. A conditional Bayesian mixture of experts is learned to map the points in the $\mathbf{z} \in \mathcal{P}(\mathcal{F}_r)$ to $\mathbf{y} \in \mathcal{P}(\mathcal{F}_x)$ as $p(\mathbf{y}|\mathbf{z})$. For visual inference, we can directly track the points in the low-dimensional subspace by discriminatively propagating the filtered conditional $p(\mathbf{y}|\mathbf{Z})$ in the kernel PCA subspace of 3D human poses. The above figure depicts modeling of the conditional $p(\mathbf{y}_t|\mathbf{z}_t)$ for the time-step t. We can similarly learn the conditional $p(\mathbf{y}_t|\mathbf{z}_t, \mathbf{y}_{t-1})$ to predict $\mathbf{y}_t$ at each time-step using the filtered conditional. In order to estimate the error in joint angles and visualize a predicted pose, we project the points in the low-dimensional space to original joints state space using a learned *PreImage* function.

outputs $\mathbf{x}$ in some Hilbert space $\mathcal{F}_r$ and $\mathcal{F}_x$ respectively, and use the low-dimensional representations to learn the mappings.

The key motivation behind the KDE framework is that learning a mapping to a high dimensional output space is not only computationally expensive but also sub-optimal.A simple thought experiment will illustrate the sub-optimality of original framework. Learning a mapping $f(\mathbf{r}, \alpha) : \mathcal{R}^L \rightarrow \mathcal{R}^K$ essentially involves estimating mapping parameters by minimization of loss function

$$\hat{\alpha} = \text{ arg } min_\alpha \sum_{(\mathbf{r}_i, \mathbf{x}_i)} \mathcal{L}(\mathbf{x}_i, f(\mathbf{r}_i, \alpha)) \tag{4.5}$$

Assume, that the joint angles i and j are strongly correlated while the joint angles k is uncorrelated to either of the joints i or j. The loss function $\mathcal{L}$ effectively gives equal weights to the 3D poses differing in the any of the three joints. For two poses $\mathbf{x}_1$ and $\mathbf{x}_2$ that differ from the groundtruth pose at uncorrelated joint k and at the two correlated joints i and j respectively, the loss function gives undue advantage to the pose $\mathbf{x}_1$, which under ideal circumstances, should be treated equally to the pose $\mathbf{x}_2$. This is sub-optimal and may not train accurate predictive models. The loss function $\mathcal{L}$ may be thought of as a kernel function defined in output space that measures

similarity between two 3D pose states and effectively maps the outputs to a high dimensional Hilbert space. The principal components that are extracted in this feature space are essentially de-correlated axes of variations. The points in the feature space are projected onto the low dimensional subspace formed by these principal components. The mapping can then be learned for each of the orthogonal dimensions of transformed output space. This effectively allows us to minimize the loss function along the de-correlated modes of variations and facilitate learning of more accurate models. Fig. 4.3 shows the framework for visual inferencing in kernel induced low dimensional space. The input features $\mathbf{r}$(2D image observations) are projected to Hilbert space $\mathcal{F}_r$ using the kernel mapping Φ_r and used to learn the principal subspace $\mathcal{P}(\mathcal{F}_r)$ using PCA. The inputs in the transformed space $\mathbf{z}$ are obtained by projecting points $\Phi_r(\mathbf{r})$ on the principal components. Similarly, the output points $\mathbf{x}$ are de-correlated to $\mathbf{y}$ by first projecting them to Hilbert space $\mathcal{F}_x$ and learning principal subspace $\mathcal{P}(\mathcal{F}_x)$. Even though the input $\mathbf{z}$ and the output $\mathbf{y}$ points are de-correlated, the mapping between them is still multi-valued. We use conditional Bayesian Mixture of Experts (BME)(chapter 3) to learn the mapping between the low-dimensional points $\mathbf{z}$ and $\mathbf{y}$ in the reduced features $\mathcal{P}(\mathcal{F}_r)$ and $\mathcal{P}(\mathcal{F}_x)$ respectively.

$$p(\mathbf{y}|\mathbf{z}) = \sum_{i=1}^{M} g(\mathbf{z}|\lambda_i)\mathcal{N}(\mathbf{y}|\mathbf{W}_i\Phi(\mathbf{z}), \mathbf{\Omega}_i^{-1}) \tag{4.6}$$

where $g(\mathbf{z}|\lambda_i)$ is the gate distribution to select appropriate expert and $(\mathbf{W}_i, \mathbf{\Omega}_i)$ are the parameters of the i^{th} expert. The weights of the experts and of the gates, $\mathbf{W}_i$ and $\boldsymbol{\lambda}_i$, are controlled by hierarchical priors, typically Gaussians with 0 mean, and having inverse variance hyperparameters controlled by a second level of Gamma distributions. We learn this model using a double-loop EM and employ *type II Maximum Likelihood* optimization [113, 180] with greedy weight subset selection.

Visual inference in the low dimensional kernel space is done in the same fashion as in the original state space. Specifically, we learn the conditional distributions $p(\mathbf{y}_t|\mathbf{z}_t)$ and $p(\mathbf{y}_t|\mathbf{z}_t, \mathbf{y}_{t-1})$ using the low-dimensional inputs $\mathbf{z}_t$ and outputs $\mathbf{y}_t$ over multiple time steps t. The 3D pose inference at each time step is done using the filtered density $p(\mathbf{y}_t|\mathbf{Z}_t = \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_t\})$ which is propagated in the same fashion as (4.1)

$$p(\mathbf{y}_t|\mathbf{Z}_t) = \int_{\mathbf{y}_{t-1}} p(\mathbf{y}_t|\mathbf{y}_{t-1}, \mathbf{z}_t)p(\mathbf{y}_{t-1}|\mathbf{Z}_{t-1})d\mathbf{y} \tag{4.7}$$

Both the conditionals $p(\mathbf{y}_t|\mathbf{z}_t)$ and $p(\mathbf{y}_t|\mathbf{z}_t, \mathbf{y}_{t-1})$ are learned using M Gaussian components using Bayesian Mixture of Experts. We integrate M^2 pairwise products of Gaussians analytically. The means of the expanded posterior are clustered and the low probability Gaussian components are pruned off.

For error reporting and reconstruction, we need to obtain the 3D pose in the original joint space for the point $\mathbf{y}$ in the low-dimensional, kernel induced space. This is done by back-projecting $\mathbf{y}$ to a point $\mathbf{x}$ in original space and is referred to as the pre-image of $\mathbf{y}$. Typically, the pre-image $\mathcal{P}(\Phi_x(\mathbf{x}))$ is obtained by minimizing the cost function using various non-linear optimization methods:

$$\mathbf{x} = \arg\, min_{\mathbf{x}} \|\mathbf{y} - \Phi_x(\mathbf{x})\| \tag{4.8}$$

The optimization techniques in general have tendency to getting stuck at local optimum. We use the algorithm proposed by Bakir et. al [18] which approximates the pre-image mapping by fitting a regression function $\mathcal{P} : \mathcal{F}_x \rightarrow \mathcal{R}^L$ to directly predict the pre-image using the low-dimensional representations as input. However, the Reproducing Kernel Hilbert Space(RKHS) $\mathcal{F}_x$ is an infinite dimensional space as infinite number of samples $\mathbf{x}$ can be used to generate the set of exemplars for training the regression mapping $(\mathbf{x}, \Phi_x(\mathbf{x}))$. A way to get around this problem is to assume that a finite set of training samples $\mathbf{r}_i, \mathbf{x}_i$ form a low dimensional subspace in their RKHS $\mathcal{F}_r$ and $\mathcal{F}_x$ respectively. This subspace can be obtained using PCA and selecting finite number of principal components that define the subspace. Thus, the mapping is learned from the low-dimensional, kernel induced space $\mathbf{y} = \mathcal{P}(\mathcal{F}_x(\mathbf{x}))$ to the original output space $\mathbf{x}$, as shown in the fig. 4.3. Following [196, 197], we use a sparse Bayesian kernel regressor to learn the pre-image, with the training data as $(\mathbf{x}_t, \mathbf{y}_t)$

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x}|A\boldsymbol{\phi}_y(\mathbf{y}), \boldsymbol{\Sigma}^{-1}) \tag{4.9}$$

with parameters and covariances $(A, \boldsymbol{\Sigma}^{-1})$. Since temporal inference is performed in the low dimensional kernel induced state space, the pre-image function needs to be calculated only for visualization or error reporting. Transforming the solution from the reduced feature space $\mathcal{P}(\mathcal{F}_x)$ to the output space $\mathcal{X}$ gives (by covariance propagation) a Gaussian mixture with elements, coefficients $g(\mathbf{z}|\boldsymbol{\lambda}_i)$ and components $\mathcal{N}(\mathbf{x}|\mathbf{A}\boldsymbol{\phi}_y(\mathbf{W}_i\boldsymbol{\phi}(\mathbf{z})), \mathbf{A}\mathbf{J}_{\boldsymbol{\phi}_y}\boldsymbol{\Omega}_i^{-1}\mathbf{J}_{\boldsymbol{\phi}_y}^T\mathbf{A}^T + \boldsymbol{\Sigma}^{-1})$ where $\mathbf{J}_{\boldsymbol{\phi}_y}$ is the Jacobian of the mapping $\boldsymbol{\phi}_y$.

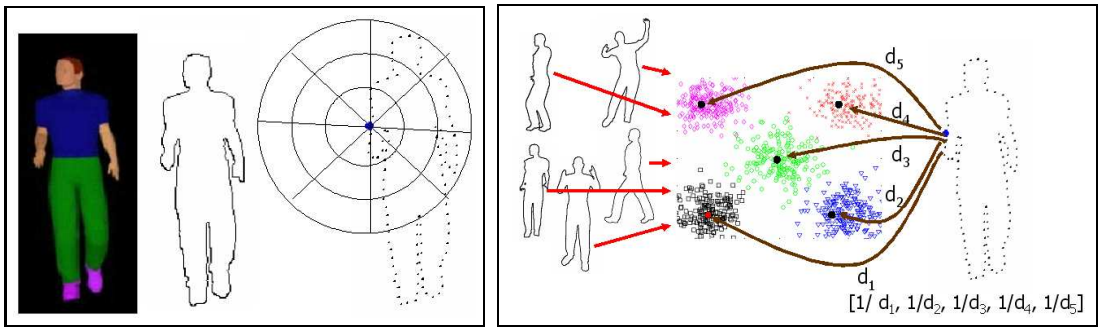


Figure 4.4: *(Left)* shows the computation of the shape context histogram. For an input image, we extract the outer contour of the silhouette. The contour is sub-sampled into fixed number of points that are used to vote into angular and radial bins. *(Right)* shows the process of dimension reduction using vector quantization. A set of representative shapes are used to construct a codebook of the shape context histograms of the points from the outer contour of the silhouettes. An image is encoded by soft voting to the histogram bins obtained from the cluster centers of codebook.

4.4 Image Descriptors

The image descriptors are compact representations of semantic contents of an image and describe low-level features such as shape, color or texture of the objects in the image. Image descriptors lie in a smooth continuous space such that humans appearing in similar configurations should have the descriptors that are close to each other. This will enable learning of smooth regressor functions that can efficiently map these representations to corresponding 3D pose configurations. However finding an appropriate image descriptors specific to a given domain is a challenging task and usually require trial and evaluation procedure to determine most competitive representations.

One of the key challenges to discriminative learning is the need for large labeled dataset for supervised learning of the 2D-to-3D mappings. Labeled data consists of 2D images and the corresponding 3D poses and is typically acquired using an expensive motion capture system. Therefore, we generate labeled exemplars using synthetic Computer Graphic(CG) human 3D model of standard anthropometry and regular clothing to render realistic 2D observations. We import motion capture data to the rigged human character to generate representative 3D pose and the corresponding 2D observations. We use Maya software to do the rendering. In order for such an approach to be practical, the synthetic human model should be of standard anthropometry and representative of the class of typical humans shapes. This will ensure that

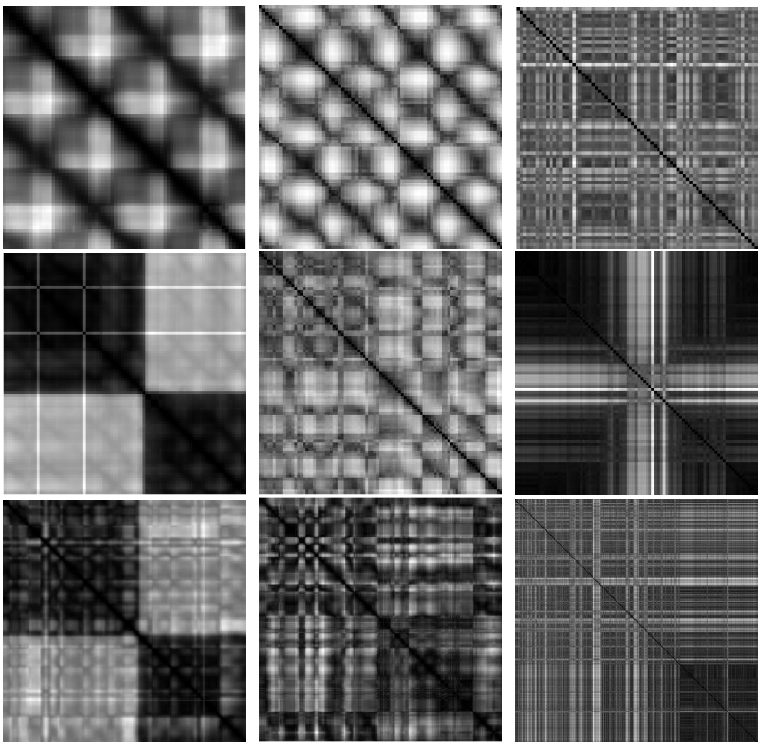


Figure 4.5: Affinity matrices computed using Euclidean distances between temporally ordered 3D pose sequence and the corresponding image descriptors. Left columns show the Joint Angles affinity matrix, middle column is due to Shape Context(SC) of the silhouette contour and right column is obtained from the Pairwise Geometric Histogram(PGH) of the internal edges of the image silhouettes. *(Top row)* walking sequence observed from lateral view. Note here that for the image descriptors based on shape context of silhouettes, the walking cycle appears to have twice frequency due to left-right leg ambiguity. *(Middle row)* Complex walking sequence in which the subject walks towards the camera and turns back. The root joint angles undergo 180^o change causing large difference in the Euclidean distances of the 3D joint vector. However due to forward/backward ambiguity the image descriptor do not behave in a similar way; *(Bottom row)* Conversation sequence of human moving hands and limbs irregularly. The affinity matrices of the image descriptors are less intuitively related to that of 3D poses.

the descriptors generated from synthetic model are similar to real image sequences. Hence it is critical to have visual descriptors that can be used to train robust mappings applicable to real scenarios.

In addition, the image descriptors should be sufficiently discriminative to resolve the pose ambiguities yet invariant to apparent variations due to synthetic pose rendering and disproportionate body parts. Our visual descriptors are based on the silhouettes obtained from statistical background subtraction that uses non-parametric density modeling of the foreground and background pixels[64]. We extract the outer contour of the silhouette and encode the shape using

shape context descriptor, as proposed by Belongie and Malik[24, 121, 10].

Shape context is a robust way to encode the shape as local histogram of edge information. These are less affected by the local artifacts dues to shadows, similarity to background and occlusion, compared to global shape representations using shape moments [39, 15]. Shape context essentially encodes distributions of relative locations of the sampled points and is a highly discriminative descriptor. For the contours, the local histograms are computed for each of the point sampled at a regular distance on the contour. For an edge based shape context, we can randomly sample points from the edges and use them for generating the histogram.

However shape context for two different images cannot be directly compared using Euclidean metric and instead uses χ^2 test statistics to match two points. A known framework to bring the descriptors to a common basis set for comparison is using vector quantization(VQ). VQ essentially learns a codebook from the representative points in the training dataset and encodes the shape as histogram of co-occurrence statistics of the codebook points. The co-occurrence statistics of a contour is obtained as the soft votes by the points sampled from the contour, for each of the entry in the codebook. The entries in the codebook are obtained by k-means clusterings of the shape context histograms of the training examples. In addition, Vector Quantization allows us to get rid of the noise in the observations due to local artifacts in the silhouettes and also reduce dimensionality of the input features.

Fig. 4.4 depicts the steps involved in the vector quantization of the shape context descriptors. We used shape context histogram [24, 121, 10] with 5 radial bins, 12 angular bins, with bin size automatically adapting to accommodate scale change in the silhouette size. We use $K = 60$ clusters for generating the codebook from the sampled points of the training images. We also investigated the encodings for internal edges using pairwise geometric histograms(PGH). PGH uses angle and distance between edges to generate histogram based descriptor[13]. In order to do so, it approximates the internal edges as piecewise linear curves. The histogram is generated by accumulating the votes for the relative angle and distances between every pair of lines into angular and distance bins. However, we observed that the using internal edge information as visual cues gave much worse results compared the shape context histogram. This may be due to less generalized patterns observed due to the internal edges in the synthetic image sequences that fail to apply well to the real sequences, thus causing inconsistencies in 3D pose predictions.

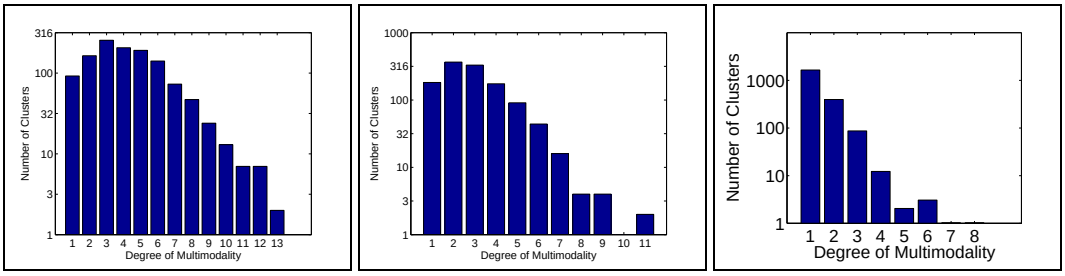


Figure 4.6: Multimodality analysis for a large database of 8262 samples of 2D images and the corresponding 3D human poses. In the plots we show the frequency histogram of associations between the clusters of the 3D poses and the 2D image descriptors. The histogram value of the K^{th} bin denotes the number of associations in which K clusters of 3D poses correspond to the 2D image descriptors belonging to a single cluster. The y-axis shows the ńumber of clusteránd is plotted on log-scale. For generating the plots, we normalized the input and output vectors to have 0 mean and standard deviation as 1 for each dimension. *(Left)* Multimodality analysis of data points $(\mathbf{x}, \mathbf{r})$ (1209 clusters). *(Middle)* Analysis of $(\mathbf{x}_t, [\mathbf{x}_{t-1} \quad \mathbf{r}_t])$ (1203 clusters). We cluster the input features as the concatenated vector $[\mathbf{x}_{t-1} \quad \mathbf{r}_t]$, and the outputs as the joint angle vectors $\mathbf{x}_t$. *(Right)* Multimodality analysis of the points $(\mathbf{y}_t, [\mathbf{z}_t \quad \mathbf{y}_{t-1}])$ in the kernel induced feature space for the training set. The input $\mathbf{z}_t$ dimension is 25, the output $\mathbf{y}_t$ dimension is 6, both reduced using kernel PCA. We cluster independently in $(\mathbf{z}_t, \mathbf{y}_{t-1})$ space and $\mathbf{y}_t$ space, using many clusters (2100).

4.5 Human Body Pose Dataset

Acquiring a rich dataset containing all the variabilities that are typically encountered in real scenario, is difficult. Therefore we generate training data using packages like Maya (Alias Wavefront) with realistically rendered computer graphics human surface models and animated using real human motion capture data[1]. A number of authors [138, 148, 65, 10, 182] in the past have also adopted this approach. Our human pose state $(\mathbf{x})$ is based on an articulated skeleton with spherical joints that has 56 d.o.f. including global translation (the same model is shown in fig. 4.10 and used for all reconstructions). The database consists of 8262 individual pose samples, obtained from motion sequence clips of different human activities including walking, running, turns, jumps, gestures in conversations, quarreling and pantomime. For obtaining the 2D observations we computed the visual descriptors on the rendered image sequence for each of these activities. We conducted an exploratory data analysis on the degree of multivaluedness existing in the dataset of 8262 pairs of pose states and image observations. For each

Sequence	$p(\mathbf{x}_t\|\mathbf{r}_t)$			$p(\mathbf{x}_t\|\mathbf{x}_{t-1},\mathbf{r}_t)$		
	NN	RVM	BME	NN	RVM	BME
NORMAL WALK	4 / 20	2.7 / 12	2 / 10	7 / 25	3.7 / 11.2	2.8 / 8.1
COMPLEX WALK	11.3 / 88	9.5 / 60	4.5 / 20	7.5 / 78	5.67 / 20	2.77 / 9
RUNNING	7 / 91	6.5 / 86	5 / 94	5.5 / 91	5.1 / 108	4.5 / 76
CONVERSATION	7.3 / 26	5.5 / 21	4.15 / 9.5	8.14 / 29	4.07 / 16	3 / 9
PANTOMIME	7 / 36	7.5 / 53	6.5 / 25	7.5 / 49	7.5 / 43	7 / 41
Normal walk	15.8 / 179.5	9.54 / 72.9	7.41 / 128.5	5.79 / 164.8	8.12 / 179.4	3.1 / 94.5
Complex walk	17.7 / 178.6	15 / 179.8	8.6 / 178.8	17.8 / 178.6	9.5 / 179.9	7.7 / 134.9
Running	20.1 / 178.2	10.6 / 76.8	5.9 / 177.4	9.3 / 64.9	8.64 / 76.8	3.3 / 59.5
Conversation	12.9 / 177.4	12.4 / 179.9	9.9 / 179.7	12.8 / 88.8	10.6 / 179.9	6.13 / 94.3
Pantomime	20.6 / 177.4	17.5 / 176.4	13.5 / 178.5	21.1 / 177.4	11.1 / 119.9	7.4 / 119.2
Dancing	18.4 / 179.9	20.3 / 179.9	14.3 / 179.9	25.6 / 179.9	14.9 / 149.8	6.26 / 124.6

Table 4.1: Root Mean Square(RMS) errors per joint angle (average error / maximum joint average error) in degrees for various sequence of motions using the two conditional models, $p(\mathbf{x}_t|\mathbf{r}_t)$ and $p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{r}_t)$. We compare the prediction errors for three different algorithms - Nearest Neighbor(NN) regression, Relevance Vector Machine(RVM)[180] and Bayesian Mixture of Experts(BME). For the BME, we use prediction from the most probable experts for the test input. Note here that no probabilistic tracking is performed in these experiments. *(top table)* shows result obtained by training separate activity models for each sequence and testing on motions in their class (BME uses 5 Gaussian kernel experts). *(bottom table)* shows results obtained by training one single BME model on the entire database of 8262 exemplars. BME model in these experiments is used 10 sparse linear experts while RVM used one sparse linear expert. In all tests, accuracy is reported w.r.t. the most probable expert for BME.

2D image observation we estimated the number of different 3D pose configurations that might have generated it. In order to get a realistic estimate of the multimodality in the data we want to estimate only distant poses that correspond to a given 2D observation. We do this by finding correspondences between the clusters as against the individual data points. Each 2D input cluster represents a perturbed observation data $\mathbf{r}_i$ and take into account the variations due to local shape distortions and shadow artifacts. Similarly, a 3D pose cluster represents a set of nearby pose configurations $\mathbf{x}_i$ and account for slight changes in viewing direction, anthropometry and joint angles. For each of the observation cluster, we define the degree of multimodality as the number of different pose clusters corresponding to each of the constituent observations. The plots of the histogram are shown in the fig. 4.6. The clustering also ensures that 3D poses lying in different clusters differ from each other substantially. We also do the similar analysis with

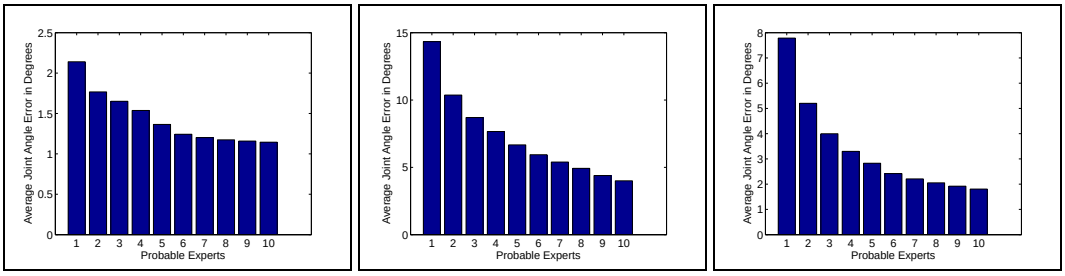


Figure 4.7: Pose reconstruction error in the 'best k' experts ($k = 1 \ldots 10$) for single BME model, trained on a synthetic database of 8262 exemplars. In the plots we show the prediction accuracy w.r.t. the k most probable experts. The prediction accuracy is measured by choosing the k most probable experts and computing the error using the expert prediction that is closest to the ground truth. *(Left)* and *(Middle)* train and test errors for dancing. *(Right)* test errors for a person walking towards the camera, turning 180^o and going back. As illustrated in these plots, the most probable experts may not always be reliable, but prediction from the top most probable experts are indeed more accurate.

the inputs as $(\mathbf{r}_t, \mathbf{x}_{t-1})$ and the outputs as $\mathbf{x}_t$. Working with the previous state and the current observation (fig. 4.6b) does not eliminate ambiguity but somewhat reduces it due to additional temporal information in the inputs. For an unbiased similarity measure between the inputs $(\mathbf{r}_t, \mathbf{x}_{t-1})$, we center and whiten (zero mean and unit variance) the vectors.

We also conducted multi-modal data analysis in the kernelized input $\mathbf{z}_t$ and the output $\mathbf{y}_t$ space (fig. 4.6c). We used 6 dimensional kernel PCA output space and 25 dimensional kernel PCA input space. The ambiguity is severe enough to cause significant errors and therefore needs to be adequately handled using multi-valued functions.

4.6 Results

We show results on real and artificially rendered motion capture-based test sequences. We compare the pose estimation accuracy with the existing methods: nearest neighbor regression and single regression based on Relevance Vector Machine[180]. For Kernel Dependency Estimation we compare the results with PCA with varying number of dimensions.

The prediction error is reported in degrees (for mixture of experts, this is w.r.t. the most probable one and normalized per joint angle, per frame. We also report maximum joint angle prediction error averaged over all the joints and frames of the sequence.

Figure 4.8: Pose reconstruction results for a test synthetic conversation sequence. 3D joint angle estimates were obtained using $p(\mathbf{x}_t|\mathbf{r}_t)$.*(First row)* Original input image sequences. *(Second row)* Pose reconstruction obtained using the most probable expert of BME model and rendered from the same viewpoint as the input. *(Third row)* Pose reconstruction of the same shown from a different viewpoint. Notice that some of the estimated 3d poses are substantially different from the test data(last column). Perturbations in the image descriptors can at times cause unrealistic poses to be predicted in discriminative models. To a large extent such gross prediction errors can be eliminated in generative based 3d pose estimation frameworks provided the observation likelihood is robustly modeled.

We test several human activities obtained by animating a 3D human model using motion-capture data from [1]. The sequences are artificially rendered using Maya software[2] with an ambient light source to create ideal lighting conditions.These conditions allow us to compare different algorithms based on how accurately they model the 2D-3D relation, by factoring out noise in the observed image sequences due to changes in shape, appearance, body proportions of the subjects and illumination. We show the results of experiments in the table 4.1.

We run two comparisons, one by training separate models for each activity class and testing on it (top half of table 4.1), the other by training one global model on the entire database and using it to track all motion types (the bottom half of 4.1). Training and testing is performed on motions on the different trials of the same motion, performed by different actors.

Training on separate activities: (top half of table 4.1) We use several training sets: walking oblique w.r.t. to the image plane (train 300, test 56), complex walk towards the camera and

Figure 4.9: 3D pose estimation results for the conversation sequence using the conditional distribution $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t)$ learned using BME. *(First row)* Test image sequence *(Second row)* Reconstruction result using the most probable experts seen from the same viewpoint as the test image data. *(Third row)* Most probable reconstruction as seen from a different viewpoint. *(Fourth Row)* 2^{nd} most probable reconstruction seen from the same viewpoint as shown also on first and second rows. Note that the pose prediction from the 2^{nd} most probable experts is not too far from the most probable predictions. Prediction in the last column shows the forward backward ambiguity.

turning back (train 900, test 90), running parallel to the image plane (train 150, test 150), conversation involving some hand movement and turning (train 800, test 160), pantomime (1000 train, 100 test).

In the table 4.1(top half), we compare the prediction errors for Nearest Neighbor (NN), Relevance Vector Machine(RVM) and Bayesian Mixture of Experts(BME) using the learned conditional distributions $p(\mathbf{x}_t|\mathbf{r}_t)$ and $p(\mathbf{x}_t|\mathbf{r}_t, \mathbf{x}_{t-1})$. We show the average joint angle error per frame and the maximum joint angle error for the entire sequence. For computing the prediction error using the conditional $p(\mathbf{x}_t|\mathbf{r}_t, \mathbf{x}_{t-1})$, the models were initialized from the ground truth pose. Notice that Bayesian Mixture of Experts model consistently outperforms NN and RVM in terms of the average joint angle error. The BME was trained with 5 kernel experts. We used

Figure 4.10: *(Left)* Reconstruction of a walking sequence using $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t)$. Top row shows original sequence images; Bottom row shows the reconstructed poses seen from the same viewpoint. *(Right)* 3d pose estimations for a 'running' sequence using $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t)$. First row shows the original image sequence. Second row shows the reconstruction seen from a different synthetic viewpoint. In the 3D model animation module based on Maya software package, we did not enforce no self-intersection constraint on the surface mesh deformation due to which we observe penetration of the limbs into body for some of the results (running sequence, 2^{nd} image in the bottom row

Radial Basis Function(RBF) as the kernel. Bayesian Mixture of Experts involves learning of both the expert regressors (used for pose prediction) and the gate distribution function (used to select the most appropriate expert). Due to inaccuracies of the gate function, the most probable experts may not be always the best expert. Fig. 4.7 shows the prediction accuracies using best experts in the most probable k-experts, where we vary k from 1 to 10. It should be noted that in most cases there is always some expert which predicts the 3D pose accurately. However, due to wrong choice of the expert, we may end up with wrong prediction with higher pose estimation error.

Training on multiple activities:: We have also evaluated the framework by training different models on multiple activities. Our dataset was composed of 8262 labeled exemplars (2D images with corresponding 3D pose) from a variety of activities. We used 7238 examples to learn the conditional distribution $p(\mathbf{x}_t|\mathbf{r}_t)$ and 7202 samples to train the conditional $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t)$. We tested on six motion types - normal walk - 55 Frames, complex walk - 100 frames, running - 150 frames, conversation – 100 frames, pantomime – 200 frames, dancing 270 frames. For

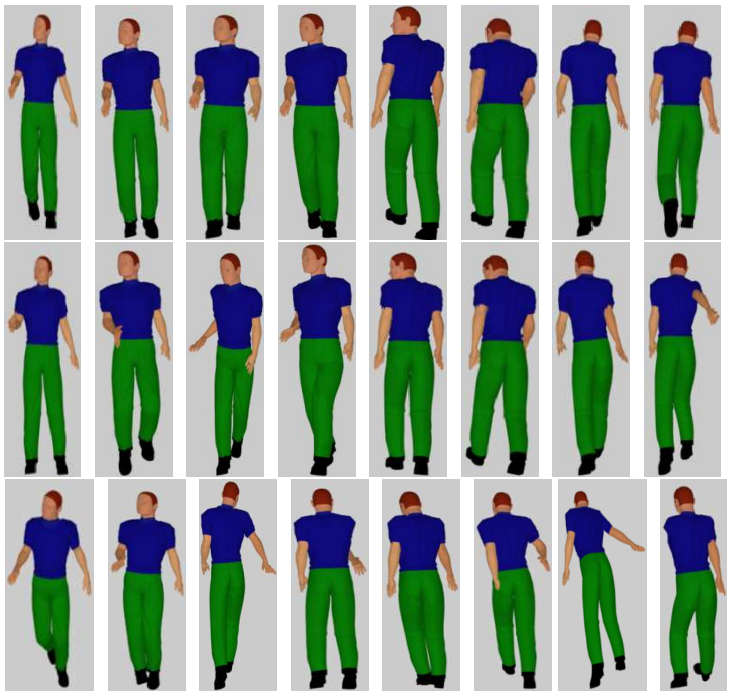


Figure 4.11: Tracking a 'complex walk' sequence in which a person walks towards the camera and turns back. We predict the 3D poses using $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t)$. *(First row)* Test image sequence. *(Second row)* 3d pose reconstruction using the most probable experts as seen from the same viewpoint. *(Third row)* reconstruction using the second most probable expert prediction (notice 180^o turn ambiguities) as well as forward-backward flipping ambiguities of the arms and legs[170, 171].

these experiments *only* we used conditional models based on 10 linear (as opposed to Gaussian kernel) experts and a 200d shape context feature vector made of two 100d histograms computed separately for the points sampled at regular distance from the contour, and the points sampled randomly from the internal edge features. Results are shown in the bottom-half of table 4.1. Notice that both average and maximum joint angle errors have increased for all the models. This is due to increased variability in the dataset, that would in general lead to less accurate trained models. Also notice that NN and RVM regressors perform more worse when trained larger dataset as they cannot explicitly learn the multi-valued mappings. For all the learned models, dancing and pantomime had highest prediction error due to more varied poses and complex dynamics that is more difficult model compared to repetitive activities like walking and running.

In Fig. 4.7 we show the average joint angle errors from the best of most probable k-experts. The best of the k-experts are chosen based on how close the prediction of the expert is to the

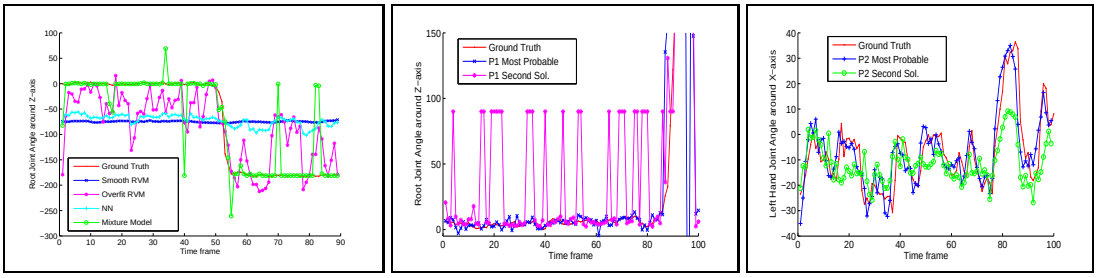


Figure 4.12: Joint angle predictions for a complex walking sequence *(left)* as shown in the fig. 4.11 and conversation sequence*(middle)* and *(right)*, shown in the fig. 4.8 and fig. 4.9 respectively. It is difficult to resolve the multimodality in *(left)* using a single regressor based on RVM or Nearest Neighbor regression. Note that the multimodality in left plot is due to forward-backward ambiguity. In many such cases the correct pose will be either the first or the second most probable mode. Also notice the 90^o ambiguities in the conversation sequence *(middle)* are apparent in the second most probable mode.

ground truth prediction. We used the same dataset to plot the fig. 4.7. For comparison, we show the errors both on the training and testing dataset in fig. 4.7*(left)* and*(middle)* respectively. Reconstruction error using best 3-experts gives substantially more accurate results compared to using the most probable experts. This also reflect the interdependency of the experts and the gate distribution models, where errors may be introduced not only due to the inaccurate mapping functions but also due to inaccurate choice of experts.

Fig. 4.8 shows the pose estimation for a conversation sequence using the conditional $p(\mathbf{x}_t|\mathbf{r}_t)$. We predict using the most probable expert and show the reconstruction from a different viewpoint on a synthetic 3D human model using Maya software. Fig. 4.9 shows the same using the conditional $p(\mathbf{x}_t|\mathbf{r}_t, \mathbf{x}_{t-1})$. The second and third rows shows the prediction using most probable experts while the last row shows the prediction using less probable experts. Notice, that different experts predict distant 3D poses for the same input.

Fig. 4.10 shows the reconstruction results on the synthetic sequences for walking and running. Top row shows the original sequence, while the bottom row shows the predicted poses from a different viewpoint. For generating the reconstruction results, no self intersection constraints were applied during the deformation of the 3D surface mesh model under the influence of skeleton joints. Therefore in some of the results, body parts were shown to penetrate each other.

Joint angle predictions: Fig. 4.12*(left)* shows plots for the root joint angles for the complex

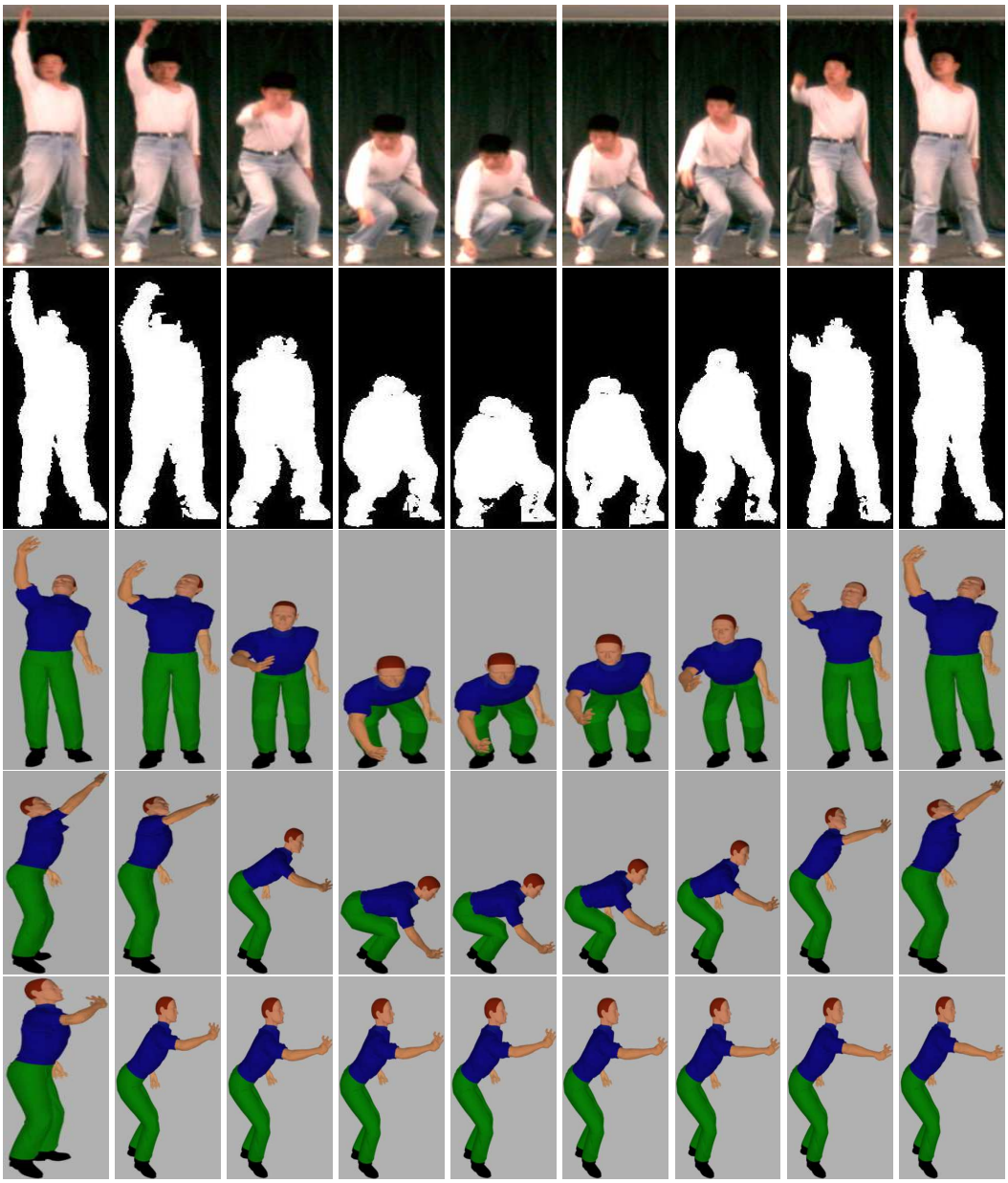


Figure 4.13: *(First row)* Real test image sequence of a subject mimicking a bending pickup sequence used for learning the BME model. *(Second row)*Image silhouettes of the real sequence, *(Third row)* 3D pose reconstruction as seen from the same viewpoint used for training BME,*(Fourth row)* 3d pose reconstruction from a novel viewpoint. Notice that despite noisy silhouettes, our probabilistic tracker based on Bayesian mixture of expert (BME) can reconstruct the 3D pose with a reasonable perceptual accuracy *(Fifth row)* A unimodal reression model based on (4.1) fails to accurately reconstruct the sequence for the bending pickup sequence and gets stuck at the mean pose.

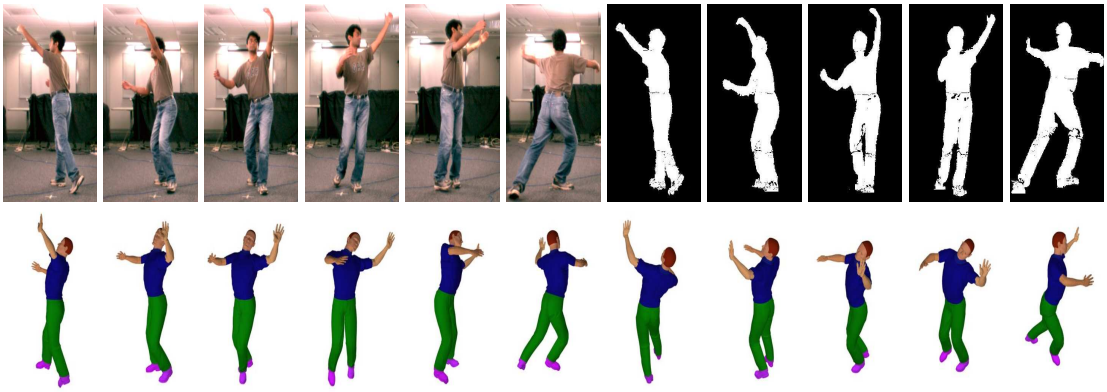


Figure 4.14: Tracking and 3d pose reconstruction of a dancing sequence using the filtered conditionals. *(Top row)* shows original images and silhouettes (the algorithms use both the silhouette contour and the internal image edges); *(Bottom row)* shows reconstructions from training (left) and a novel, synthetic viewpoint (right).

walk sequence in which the subject walks towards the camera and turns back. The root joint angle undergoes a 180^o rotation. Here we show the prediction from $P1 = p(\mathbf{x}_t|\mathbf{r}_t)$. Single hypothesis predictor, outputs angles that are either average between the two widely different poses or that are far from the ground-truth. The Mixture model is able to learn the multi-valued mapping more accurately. Fig. 4.12*(middle)* and *(right)* shows the plots for the root joint angle and the left hand joint angle using $P1 = p(\mathbf{x}_t|\mathbf{r}_t)$ and $P2 = p(\mathbf{x}_t|\mathbf{r}_t, \mathbf{x}_{t-1})$ respectively. Here we show that the most probable expert is more accurate than the second most probable expert prediction, although at times, second most probable solution is nearer to the groundtruth.

Real Image Sequences - Bending and Picking up, Dancing and Walking: We test the framework on real sequences captured in laboratory settings. The subjects mimicked the activities for which the 3D motion capture data was available [1].

For the reconstruction using the filtered density, we track using discriminative density propagation(4.1) with 5 mode posteriors $p(\mathbf{x}_t|\mathbf{R}_t)$. The BME conditionals $p(\mathbf{x}_t|\mathbf{r}_t, \mathbf{x}_{t-1})$ was also based on 5 experts, with RBF kernels and the degree of sparsity varying between 5%-25%. Fig. 4.13 shows the reconstruction results of the bending and pickup sequence, from a 2s video filmed at 60 fps, where a subject mimics the act of picking an object from the floor. For this sequence, we trained the model on synthetically rendered bending and pickup sequence. The subject performed the sequence in a similar fashion as the training sequence.

We also experimented with the single hypothesis tracker, based on RVM and propagated

Figure 4.15: Reconstruction of a walking sequence observed from lateral view *(First row)* observed test images of a walking subject ; *(Second row)* extracted silhouettes from the real sequence; *(Third row)* 3D pose reconstruction as seen from the same viewpoint that is used in the training; *(Fourth row)* Image rendering of 3D pose from a novel viewpoint.

using (4.1). However due to complexity of the motion, the single hypothesis tracker fails to track the complete sequence and gets stuck at the mean pose after some initial accurate predictions(last row of the fig. 4.13).

For the multi-hypotheses tracker, we predict using the most probable expert of the filtered conditional at each time step. Fig. 4.16*(left)* plots number of modes of the multiple hypothesis tracker for the right should joint and the right upper limb joint at each time step. Although we propagate the posterior using $M = 5$ components, at each time step, the filtering step generates M^2 components. Typically, the number of modes are lesser as some of the component Gaussians are close to each other. Fig. 4.16*(middle)* plots the minimum and maximum angle

difference between the modes of the filtered density for the root joint. Fig. 4.16*(right)* plots the mixing coefficients of the multiple components of the filtered conditional. Fig. 4.14 shows the

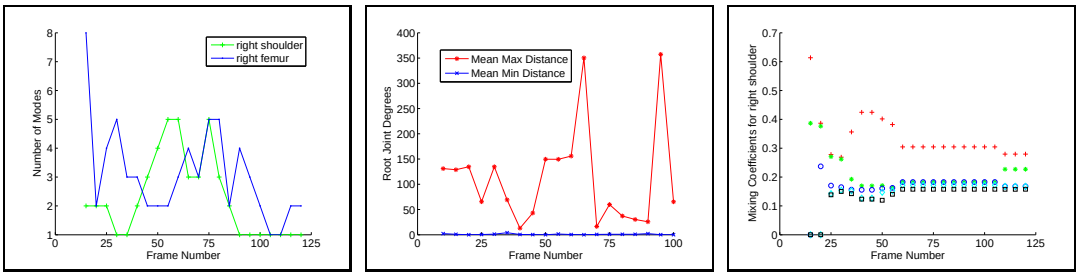


Figure 4.16: Quantitative 3d joint prediction results for the bending-pickup sequence shown in the fig. 4.13. *(Left)* shows the number of modes for the right femur joint and right shoulder joint angles. *(Middle)* shows the maximum and minimum distance between the modes of the root joint angles. *(Right)* shows the mixing proportions of the right shoulder during tracking.

reconstruction of the dancing sequence, where the subject undergoes a complete 360^o turn. The model was trained on a similar sequence, synthetically rendered using Maya software. Notice, the complexity of the poses made imitation of the sequence substantially difficult, and hence different from the training sequence. We show the reconstruction from 2 different viewpoints. For this sequence, we used shape context from both the outer contour and the internal edges to predict the pose. The silhouettes, obtained from background subtraction was used to mask out the background edges. Although overall predicted 3D poses appear to be similar to the pose in the observed images, the clear bias of the discriminative modeling towards the training set is quite apparent in the results. To illustrate the multimodalities in the dataset, we plot the filtered conditional of joint angles of the dancing sequence in fig. 4.17. We show the filtered density for the right shoulder, right thigh and right foot joint. Notice, the clear multimodalities of the filtered density, with modes differing in angles $> 40^o$. Fig. 4.15 shows the results of the 3D pose reconstruction for the walking sequence, observed from side.

4.6.1 Reconstruction Results in Low Dimensional Kernel Induced Space

For evaluating 3D pose estimation in low dimensional Kernel Induces Space, we project the points to high dimensional feature space using radial basis functions as kernels. The input space is reduced to 50d space while the output space is reduced to 6d using the PCA in the kernel feature space. Number of dimensions were selected by choosing the minimum dimensions that

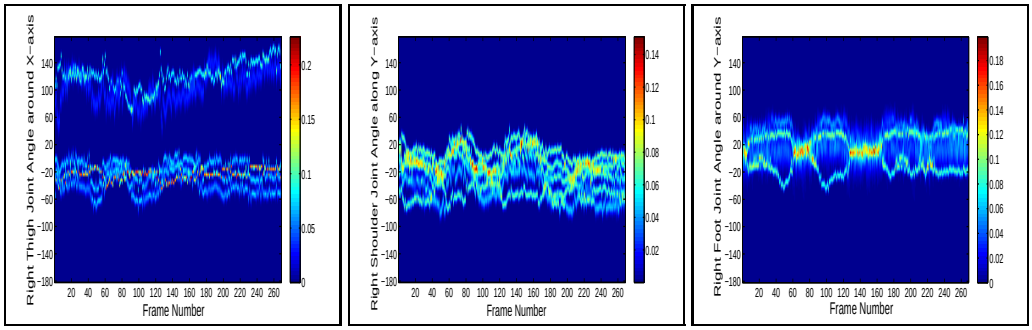


Figure 4.17: Illustration of the discriminative tracking of the dancing sequence, shown in the fig. 4.14, with the filtered conditional(4.1) composed of 5 Gaussian components. Time is along the horizontal axis, filtered density at time-step on the vertical (showing one selected variable), probability is color coded. Notice different types of multimodality of the filtered conditionals, including well separated paths *(left)*, bundles *(middle)* and merge/splits *(right)*.

gave sufficiently low reconstruction error for the test data. The kBME conditionals are trained as discussed in §4.3.

In order to demonstrate the improvements in computational time, we evalute kerel PCA framework trained on a walking sequence of 2000 frames and tested on the similar sequence of 750 frames. We reduce the dimensionality of the output state space using KPCA and learn the conditional distribution $p(\mathbf{y}_t|\mathbf{r}_t)$ using kBME with 3 experts. We used the shape context descriptor as the input image encodings. For learning the pre-Image we used single regression function based in RVM[180]. Fig. 4.18 compares the prediction time with different kernel PCA dimensions. Notice that the prediction time with 6d kernel PCA is reduced by 50% of the same in original joint space, without significantly affecting the joint angle prediction accuracy. Fig. 4.18*(middle)* shows the average reconstruction error by projecting the joint angle vectors to low dimensional kernel space and then backprojecting to original space using the learned pre-Image mapping. In fig. 4.19*(left)*, we compare the accuracy of kernel BME(kBME) trained on different number of output state dimensions in reduced kernelized space, on the dancing sequence with 50d observation descriptor.

Kernel based BME outperforms other low dimensional models on the dancing sequence with highly non-linear movement of various body parts. It substantially improves over PCA based models as linear methods cannot accurately learn nonlinearities in the human motions. Fig. 4.20 shows the human pose reconstruction for a jumping sequence using 4 and 6 output dimensions in the kernelPCA state space. While 4 dimensions are too low, 6 dimensions are

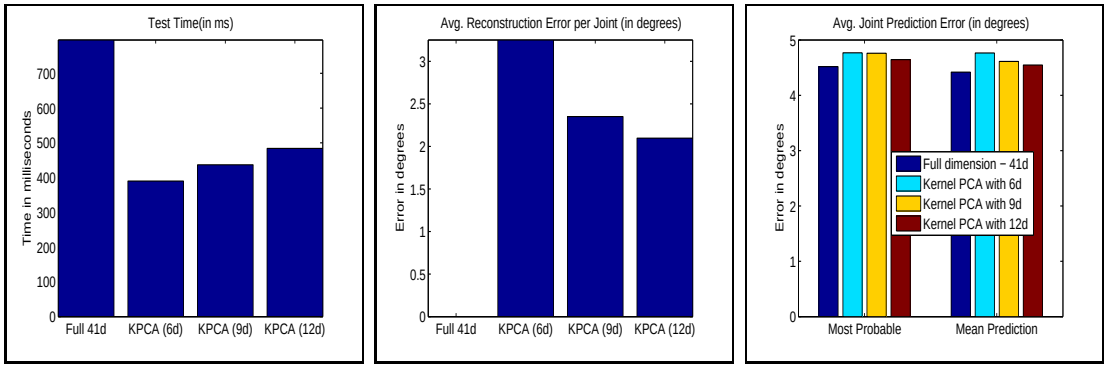


Figure 4.18: Evaluation of kernelPCA on a walking sequence using the conditional $p(\mathbf{y}_t|\mathbf{r}_t)$, *(left)* prediction times using BME with output dimensions reduced using kernelPCA with varying dimensions, *(middle)* average reconstruction error per joint of the original joint space using kernelPCA with varying dimensions of the kernel induced output space, *(right)* average 3d pose prediction error (in original joint space) using the most probable expert and weighted mean of all the experts of kBME. Note that there is no significant degradation in pose estimation accuracy due to dimensionality reduction using kernelPCA. As we increase the number of dimensions of the kernelPCA, the accuracy improves.

sufficient to summarize the highly correlated 56 dimensional joints state space using the kernelPCA state space. Table 4.6.1 compares the average prediction errors in joint angles on a synthetic test data using low-dimensional kBME with KDE-RR(Kernel Dependency Estimation - Ridge Regression), KDE-RVM, RVM and BME. Notice that prediction accuracy of Kernel BME closely matches the BME. We do additional analysis on the accuracy of kBME model in predicting pose using the most probable expert. The gate distribution effectively decides what experts should be used for a given input data. In most cases the most probable expert is indeed the best predictor of pose (as depicted in the fig. 4.19*(middle)*) using $p(\mathbf{y}_t|\mathbf{z}_t)$. Fig. 4.19*(right)* shows the number of times the most probable experts for $p(\mathbf{y}_t|\mathbf{z}_t, \mathbf{y}_{t-1})$ are best predictors given the most probable predictions from the previous time step. The probability that the most probable expert is indeed the best predictor is high and effectively corroborate the consistency of Bayesian mixture of experts model. Fig. 4.21 shows the 3D pose prediction for two people in the sequence. For each subject we used 6 dimensional kernelPCA to represent the 56d 3D pose. We used BME with 5 experts to learn the conditional distribution $p(\mathbf{y}_t|\mathbf{z}_t)$ and $p(\mathbf{y}_t|\mathbf{z}_t, \mathbf{y}_{t-1})$. The 3D pose is estimated using the filtered conditionals in the low dimensional kPCA space and mapped to original 3D pose state space using the pre-image learned using a BME with 3 experts.

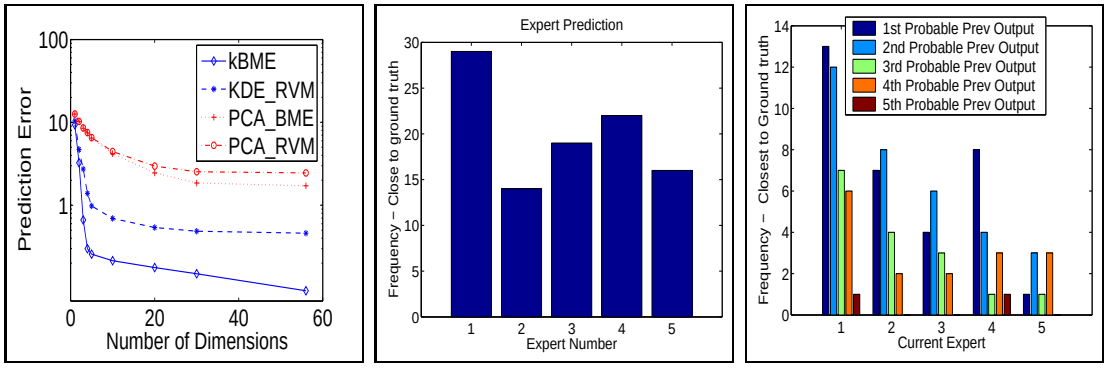


Figure 4.19: Evaluation of dimensionality reduction methods using kernel BME *(Left)* Comparison of 3D joint angles prediction accuracy for a synthetic dancing sequence. We compare prediction accuracy for the Bayesian Mixture of Experts(BME) with the single regression model based on RVM[180]. For learning low dimensional space, we compare kernel PCA with linear PCA algorithm. kBME is KDE with BME as discussed in §4.3 and KDE-RVM is a Kernel Dependency Estimator (KDE) with a Relevance Vector Machine (RVM). PCA-BME and PCA-RVM are the models in which low-dimensional subspace is learned using PCA while prediction is done using BME and RVM respectively. For all the learning models were trained on 300 labeled exemplars from a synthetic dancing sequence. *(Middle)* Histogram showing the prediction accuracy of various experts of a kBME predictor. Here we show the number times the k^{th} most probable expert (as encoded by the gate distribution of BME) is closest to the ground truth, for the same dancing sequence. Notice in the plot that the prediction from the most probable expert is indeed most frequently the most accurate prediction. However, inaccuracies in the gate distributions may occasionally cause the less probable experts to be better. *(Right)* Histograms showing the similar distribution of probability mass for the pose prediction using the conditional $p(\mathbf{y}_t|\mathbf{y}_{t-1}, \mathbf{z}_t)$ across 2 consecutive time steps, for the conversation sequence. As evident from the plots, the prediction from $p(\mathbf{y}_t|\mathbf{y}_{t-1}, \mathbf{z}_t)$ when conditioned on the most probable prediction from the previous time step, is indeed the prediction that is most frequently closest to the ground truth. This illustrates the consistency of the pose estimation framework.

	KDE-RR	RVM	KDE-RVM	BME	kBME
Walk and turn	10.46	4.95	7.57	4.27	4.69
Conversation	7.95	4.96	6.31	4.15	4.79
Run and turn left	5.22	5.02	6.25	5.01	4.92
Walk and turn back	7.59	6.9	7.15	3.6	3.72
Run and turn	17.7	16.8	16.08	8.2	8.01

Table 4.2: Comparison of average joint angle prediction error for different models - KDE-RR is a KDE model with a ridge regression (RR) as the predictor, KDE-RVM uses a single regression based on Relevance Vector Machine(RVM), BME model predicts the 3d joints in the original space of 3D joint angles(56d) and kBME refers to the BME model, predicting low dimensional representations of 3d joint angles using KDE, as discussed in §4.3. In these experiments we used single kernel regressor to learn pre-images of the kernel PCA subspace. All the low-dimensional models had 6 output dimensions. Testing was done on 100 video frames for each sequence, on artificially generated image sequences, not in the training set.

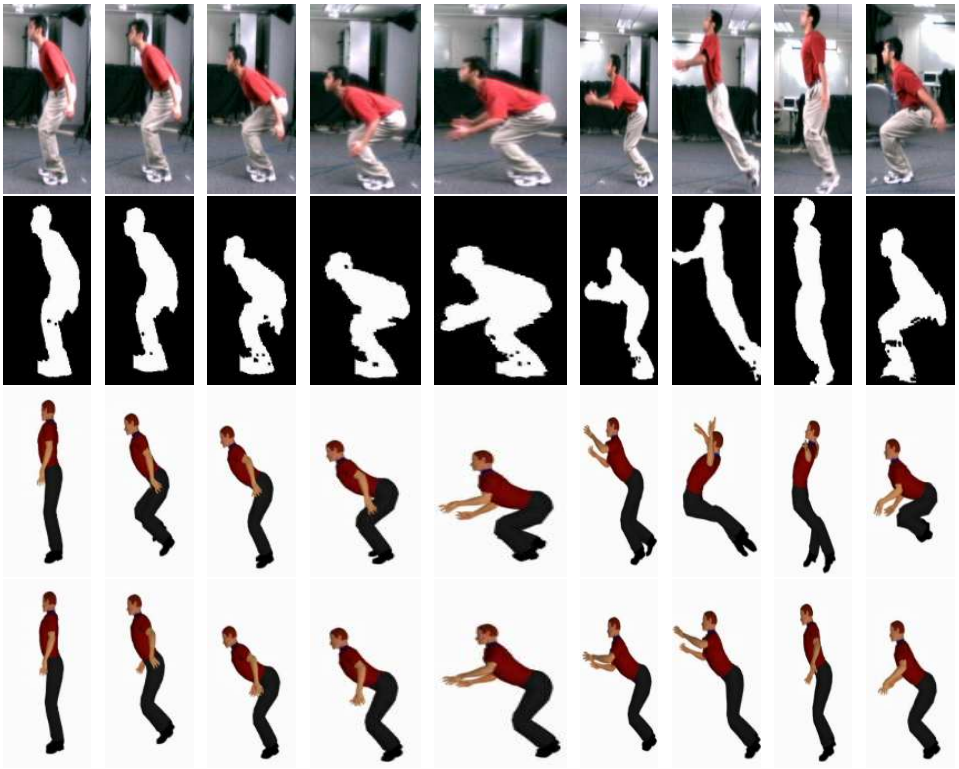


Figure 4.20: Reconstruction of a jumping sequence. *(First row)* Observed image sequence of a subject imitating a jumping sequence used for training. *(Second row)* The silhouettes of the image sequence used for feature extraction. *(Third row)* The 3d pose reconstruction using 4 dimensions in the kernel induced latent space. *(Fourth row)* The reconstruction using 6 dimensions. Notice, that 4d are insufficient to accurately model the nonlinearities of the 56d poses in the jumping sequence, often leading to high prediction errors and unrealistic reconstructions.

Figure 4.21: 3D human pose reconstruction for two clearly separated human targets performing sequences - washing window and bending. For each of the 2 people, we used 6d low-dimensional, kernel induced state space. *(Top row)* Observed image sequence. *(Bottom row)* 3d pose reconstruction as seen from the same viewpoint as of the observed images.

4.7 Discussion

In this chapter, we have proposed a framework for discriminative propagation of multiple prediction hypotheses, as obtained from multi-valued mappings learned using Bayesian Mixture of Experts. We apply the framework to a challenging problem of 3D human pose estimation by learning mappings from 2D shape based features to human body joint angles. We use coarse features based on shape context histogram so that the models trained on synthetic sequences can be easily generalized to real sequences, and at the same time, able to discriminate a pose from another. The filtered density, obtained by analytical marginalization over all possible poses in the previous time-step, is used to predict human pose in the current frame. Furthermore, we applied kernel dependency estimation framework to reduce the dimensionality of the 3D pose and improve the computational efficiency of the framework. We experiment with a number of synthetic and real sequences and demonstrated the feasibility and applicability of the proposed framework to the problem of human pose estimation from monocular image sequences.

Chapter 5

Hierachical Models for 3D Human Pose Inference

5.1 Introduction

In this chapter we propose hierarchical image descriptors that are more tolerant to perturbations due to background clutter, geometric transformations, lighting changes and misalignment. The contents of this chapter are based on the work *Semi-supervised Hierarchical Models for 3D Human Pose Reconstruction,* Atul Kanaujia, Cristian Sminchisescu, Dimitris N. Metaxas,*Conference on Computer Vision and Pattern Recognition 2007*.

In the previous chapter, we demonstrated that discriminative modeling can be efficiently used to reconstruct 3D human pose from 2D images for generic human activities, and provide a useful alternative to generative framework. The learning involves optimization of an objective function to infer the conditional distribution $p(\mathbf{x}|\mathbf{r})$, that is used to directly predict pose $\mathbf{x}$ using the image descriptors $\mathbf{r}$ as the inputs. Despite its apparent simplicity, the framework faces a number of challenges due to inherent ambiguity in 2D-to-3D mapping, background clutter, large variability in shape and appearance of the targets and illumination changes. Learning inverse of perspective projection is inherently ambiguous, and therefore necessitates use of multiple mapping functions from the input image space to output pose space.

In typical laboratory settings, background clutter can be removed by modeling pixel distributions of static regions of the scene. Under controlled lighting conditions, these are used to extract silhouettes that delineates the human from the background. This enables robust analysis of the observed foreground region as it discards the undue noise due to the background clutter. However in realistic scenarios obtaining exact human silhouette using background modeling may not be feasible due to the camera motion, rapid illumination variations or fast changing backgrounds. In such scenarios, humans are localized in a scene using a human detector. A human detector is a binary classifier that detects humans by classifying bounding boxes as

humans and non-humans. These bounding boxes are extracted on a regular grid and across multiple scales to detect objects of different sizes and at varying distance from the camera[53]. In chapter 3 and chapter 4, we used shape descriptors computed for the silhouettes as the inputs for 3D pose estimation. This framework can be easily extended to handle any inputs and predict pose from the descriptors computed over a bounding box that are obtained from the human detector. However since the bounding box typically enclose extraneous regions around the target, the image descriptors exhibit undue noisy variations due to changes in the background. A key challenge here is therefore in designing robust image descriptors that are sufficiently discriminative to differentiate between various body poses yet are invariant to within pose class variations due to background clutter, misalignment of bounding box and different human body proportions. The image descriptors should vary smoothly with the pose configurations of the human in the image and should remain invariant to changes in the background due to motion or clutter. In order to make such a framework scalable and generalizable to any human activity, we need to overcome the following key hurdles:

- *Invariance of the image descriptors to intra-pose class variability* - Humans appear in a variety of shapes and appearances owing to their highly articulated body structure and variations in the clothing styles. Different subjects have body parts of varying anthropometry. In addition, there may be perturbations in the visual descriptors due to illumination and viewpoint changes. In order to apply discriminative framework for pose prediction in realistic scenarios, we should be able to train a predictive model that can discriminate between different poses of the same subject yet remain invariant across different subjects in same pose.

- *Noisy data and extraneous information in visual inputs* - Humans may appear in varying environment. In a typical discriminative framework, humans are localized in the image sequence as coarse bounding boxes, obtained from a human detector. The visual inputs are in the form of an image descriptor vector computed over this bounding box that encloses the human. This introduces extraneous information in the inputs in the margins and may influence the pose prediction from the trained model. Ideally, we would want the discriminative learning framework to automatically identify the relevant features in

the inputs and ignore the noise introduced in it by the background clutter. However there is no guarantee that the learning method is actually doing that.

- *Large labeled dataset for training* - In order to learn accurate predictive models, it is critical that the training dataset is sufficiently representative of the image sequences captured under realistic scenarios. Discriminative models have been shown [126] to outperform generative models in the presence of sufficiently large set of exemplars. Typically, these models are trained in a supervised fashion on a labeled dataset consisting of pairs of 2D images and the corresponding 3D joint angle measurements of humans in a sufficiently representative set of poses. Obtaining these joint angle measurements is however a cumbersome process and requires expensive motion capture equipments (like VICON). In order to operate accurately, these motion capture systems require controlled laboratory settings with ideal illumination conditions, that is difficult to simulate in a real outdoor environment.

In this chapter we propose techniques targeted towards solving these challenges. The proposed framework brings together several innovations from the past research in object recognition, distance metric learning, correlation analysis and semi-supervised learning.

5.1.1 Overview of the Approach

The predictive models are trained by importing the motion capture data into a computer graphic(CG) model and rendering the sequence with the CG model placed in the realistic outdoor images[9, 165], as shown in fig. 5.1. We refer these sequences as *Quasi-Real* data. In order to design image descriptors that can balance their *invariance* to intra-class variability (different humans in similar stance) with the high *selectivity*(to discriminate between different poses), we develop hierarchical image descriptors that encode image at multiple levels of this tradeoff. We use hierarchical image descriptors[147, 9, 103, 127] for training of conditional models for human pose inference. These descriptors allow us to encode an image at multiple degrees of invariance(or selectivity), thus making them robust to perturbations due to geometric deformations, viewpoint changes and illumination variations. The second difficulty due to background clutter can be alleviated to a large extent by using statistical methods of *distance metric learning*

and *correlation analysis*, to selectively assign large weights to the relevant dimensions of the image descriptor. The image descriptor is made invariant to background clutter by projecting them into a subspace such that the distance between descriptors of humans in similar poses but with different backgrounds, is minimized. The lack of sufficient training data can be resolved by incorporating additional information from the unlabeled data in the learning framework. This is known as *semi-supervised learning* and has been applied in the past [91], to improve the learning of conditional distributions. It should be noted that learning in generative framework implicitly uses the marginal distribution over inputs $p(\mathbf{r})$ to estimate the joint distribution $p(\mathbf{x}, \mathbf{r})$. The marginal distribution $p(\mathbf{r})$ can be estimated from the unlabeled data and is disregarded in the discriminative learning framework, where only the conditional distribution $p(\mathbf{x}|\mathbf{r})$ is optimized.

Semi-supervised learning methods use both the labeled and unlabeled data to improve the inference of the conditional distribution. We follow the framework of manifold regularization to impose smoothness constraint on the conditional distribution $p(\mathbf{x}|\mathbf{r})$ such that it varies smoothly along the geodesics in the intrinsic geometry of $p(\mathbf{r})$. In effect, it ensures that the predictions from the inputs $\mathbf{r}$ close in the input space should be close to each other. We extend this framework for learning multi-valued predictors that uses multiple regression mappings from 2D image descriptors $\mathbf{r}$ to 3D human pose $\mathbf{x}$. Notice, that the semi-supervised learning critically depends on the perceptually similar inputs to be close to each other in the input space. A pre-processing step that brings the inputs closer using distance metric learning and correlation analysis, is therefore appropriate in such a learning framework. In the following sections, we discuss each of these methodologies in detail.

Figure 5.1: Images from our training and testing database, where a virtual, Computer Graphic(CG) human model, is placed in real-world images. The CG model has an anthropometry of an average sized human body. The clothing is also typical to most humans, in order to improve the generalization of the model.

5.2 Hierarchical Image Encodings

Image descriptors are a collection of pixelwise summarizations to characterize the objects depicted in the image. Typically, the summarization is a histogram characterizing specific statistics (such as gradient orientations, color and edges) over a local neighborhood of an interest point detected in the image. Most descriptors are used to uniquely describe an interest point characterizing an object by encoding the texture of the local patch surrounding it. These descriptors should therefore be distinctive and at the same time robust to changes in viewing angle and misalignment, in order to consistently recognize the class of an object in the image.

Most of the earlier approaches to texture based object recognition may be grouped into two broad categories - *Dense Grid* based and *Sparse Keypoints* based approaches. The former approach computes local descriptors for all the patches on a regular dense grid of pixels and encodes the image as a concatenation of these descriptors. The strict ordering of the grid make these features highly selective in uniquely characterizing the object class in an image. However object recognition also require these descriptors to be invariant to viewpoint changes and geometric deformations. Although these features are robust to global 2D transformations such as scaling, rotation and translations, their performance degrades substantially in the presence of in-plane rotation and viewpoint changes.

The class of sparse keypoints based descriptors are obtained as an aggregate of co-occurrence statistics of codebook patches (obtained either as randomly sampled patches or cluster centers of patches randomly extracted from the training images) at sparsely detected interest points in the image. Recent works by [103, 200]demonstrated the *bag-of-feature* approach wherein descriptors are computed at affine-invariant interest points of the image. Many works have achieved promising results for object recognition using these sparse representations. These global histogram based features however uses both background and foreground interest points for learning the codebook. Even though the background is not entirely uncorrelated with the foreground for the task of object recognition, it introduces additional noise in the descriptors if the goal is to predict poses using the computed descriptor. These descriptors tend to be more influenced by the background clutter compared to dense grid based descriptors. However they are more robust to viewpoint changes and in-plane rotation.

Evidently, there exist inherent tradeoff between the discriminative power and degree of invariance of the descriptors. It is more appropriate to view these categories of descriptors as two extremes of a continuum of invariance(or selectivity). Various descriptors that demonstrate different degree of invariance (or selectivity) lie on this continuum and it is desirable to combine the strengths of both the categories of descriptors for more effective human pose inference.

In order to select the most competitive representation of the image, we therefore encode the image at multiple degrees of invariance(or selectivity) using hierarchical image descriptors. These descriptors progressively relaxes(or constrains) spatial and geometric constraints to achieve varying degree of discriminative power of descriptors at each level. Multi-level descriptors may be categorized into 2 broad classes based on how multiple levels of encoding are constructed from the image:

- Hierarchically Structured – Each level is generated from the preceding level by accumulating semantic information over the local neighborhood. The semantic information is encoded as histogram of certain statistics that are progressively made coarser(or finer) using larger(smaller) bins, at each level of the multi-level encoding. From implementation point of view, the image is encoded only once at the lowermost level. Higher levels are constructed from the levels below that are semantically more informative but more coarser.

- Independently Structured – Each level is independently encoded from the image by computing local statistics over progressively larger spatial domains. Specifically, the image is processed independently for constructing each level of the descriptor. The semantic information is accumulated in a local region that is progressively enlarged to make higher levels more invariant to geometric distortion but less discriminative.

The final descriptor is a vector obtained by concatenating the descriptors at each level. Hierarchical descriptors have been extensively applied for object recognition [127, 103, 147] due to their ability to discriminate between different classes (*selectivity*) yet remaining invariant to geometric deformation and photometric variations(*invariance*).

We employ hierarchical descriptors for estimating 3D pose of humans in the image using the predictive framework discussed in chapter 3 and chapter 4. We compare several hierarchical

descriptors, each exhibiting outstanding performance and tolerance to geometric distortions and misalignment. These descriptors also differ in the manner in which semantic information is encoded at different levels. In the rest of this section we describe the algorithms for constructing various hierarchical descriptors we used in our framework.

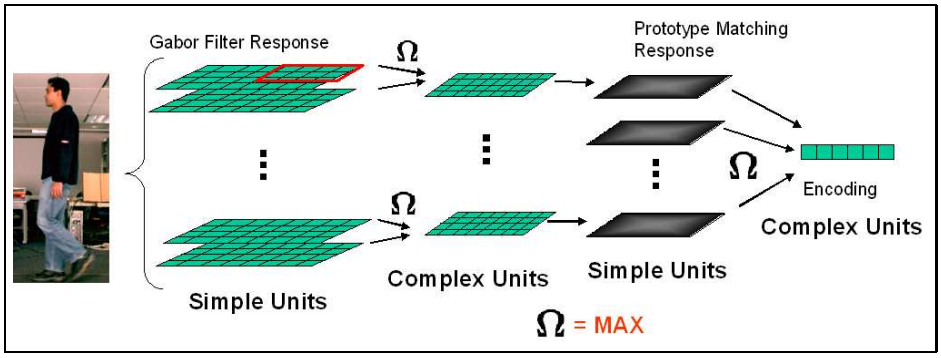


Figure 5.2: HMAX features are constructed as alternating layers of simple and complex units. Simple units computes the response of image from the oriented filters or co-occurrence statistics of prototype patches. Complex units pool the response over a local neighborhood in order to improve invariance of the descriptors to local misalignment

HMAX Visual processing in primates for recognizing an object in a scene is far superior to the state-of-the-art machine vision systems. Recently Serre *et.* al.[147] proposed hierarchical descriptors that tries to emulate the stages of object recognition in humans and primate visual cortex. HMAX uses max pooling to modulate the invariance of the encodings. The extraction steps for HMAX features is illustrated in the fig. 5.2. It consists of alternate layer of simple and complex units, where simple units are obtained either by convoluting with a bank of oriented filters or by template matching with the learned prototypes. In our framework, we cluster the randomly sampled patches from the training data and use cluster centers as the codebook prototypes. However the prototypes could as well be obtained by random sampling or from interest point detector of the training images. The complex units are obtained by max pooling the responses over a local neighborhood. Whereas the simple units encodes the semantic information as the response of image pixels to oriented filters(or matching to prototypes) that determines the selectivity of the descriptor, the MAX operation improves invariance for the descriptor to local deformation and viewpoint changes by pooling the maximum response in a local neighborhood. This construction mechanism thus exhibit balanced tradeoff between selectivity and invariance. The hierarchy may consist of any number of these alternating layers

with each higher level constructed from the preceding level in a bottom up fashion. The outputs from the complex units of the final layer are used as encodings.

Hyperfeatures are hierarchically structured features, as proposed by Agarwal *et.* al.[8], that formalizes the idea of encoding image at multiple levels of abstraction, where each level is constructed from the preceding level in the bottom up fashion, similar to HMAX. The lowermost level is computed as SIFT descriptors on a dense grid of multi-scale image pyramid. The higher levels are constructed by accumulating and averaging the co-occurrence statistics of prototypes (obtained as k-means cluster centers) over a local neighborhood. This vector quantization followed by local aggregation(local histogramming) effectively integrates higher order texton style representation in a local neighborhood, making the descriptor robust to misalignment and geometric deformation. Thus the higher levels, although coarser, are semantically more informative. The codebook for each level of the hierarchy is used to encode image by accumulating the prototype votes over the entire multi-scale image pyramid. The descriptors obtained using this global histogramming from each level are concatenated and used as the hierarchical encodings. Although similar to HMAX in organization, Hyperfeatures encodes the image at different degrees of invariance(or selectivity) as opposed to the former which employs the multi-level structure of convolution and max pooling to achieve a balance between the same.

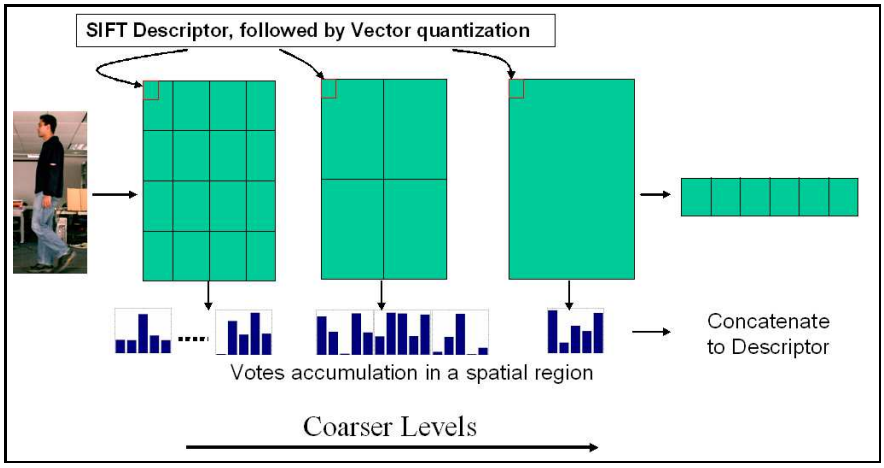


Figure 5.3: Spatial pyramid features are obtained as concatenation of co-occurrence statistics of prototype patches computed over progressively larger partitions of the image. The co-occurrence statistics is computed over a regular grid and accumulated over the partition

Spatial Pyramid is a multi-Level encoding as proposed by Lazebnik *et.* al.[103] that is

computed as bag-of-features histogramming over progressively less localized but spatially ordered image partitions. These descriptors are computed over multiple levels of spatial partitioning of the 2D image. Lower levels are more locally partitioned and hence less invariant to geometric deformation. Within each of the partitions, vector quantization is performed over either sparsely detected interest points or densely sampled regular grid locations(fig. 5.3). In contrast to the pyramid match kernels (Grauman *et.* al.[77]), where the descriptor is computed by histogramming co-occurrence statistics at progressively finer bins, this approach aggregates the features at multiple levels of spatial resolutions. Spatial pyramid improves the selectivity of the descriptor by enforcing spatial ordering between partitions. While within each partition, the co-occurrence statistics of prototypes are aggregated to maintain some level of invariance to local misalignment. Each of the levels are independently encoded directly from the image. The image descriptor is obtained as a concatenation of descriptors computed at each level of the hierarchical encoding, where each level may have different weights assigned to it.

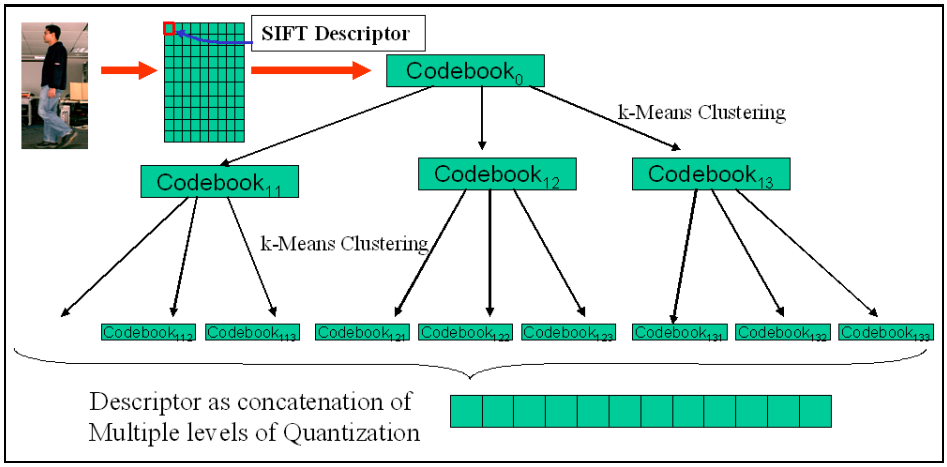


Figure 5.4: Vocabulary tree encodes the image at multiple levels of resolution in the feature space by repetitively clustering the SIFT descriptors of patches into K clusters and subclustering each cluster further into K clusters. The cluster centers of the tree formed are used to compute co-occurrence statistics of the patches computed on a regular grid locations of the image.

Vocabulary tree is derived from the multi-level coarse-to-fine descriptors introduced by Nister *et.* al.[127] for object recognition. Vocabulary tree encodes the image using multi-resolution histograms where each histogram bin is recursively partitioned into multiple bins. The recursion generates a tree structure (fig. 5.4 with each non-root node representing a bin

from the histogram computed at the parent level. Features falling in the bins corresponding to each internal node are further vector quantized to encode them using more discriminative visual vocabulary. In the vocabulary tree proposed in [127], the encoding of a patch was obtained as a path from the root node to leaf of the local patch SIFT descriptors at sparsely detected MSER interest points. Although effective for object recognition, it cannot be used to compare two encodings using Euclidean distance metric, as the paths from the root node to the leaf node may vary. This may cause similar images (with similar human poses) to have very different encodings. Therefore, in our framework, the image is encoded as a vector obtained by concatenating histograms for all the nodes present in the tree, with entries corresponding to the nodes not in the path set to zero. Instead of computing SIFT descriptors for the detected interest points, we compute it for local patches at regular dense grid locations for all the levels of multi-scale image pyramid. Each of the local patch is encoded using vector quantization at every node of the tree. For the multi-scale image pyramid, the hierarchical encoding is obtained as aggregate of all the encodings of the local patches. Vocabulary tree is constructed hierarchically in a top-down fashion with lower tree levels more discriminative but less invariant to due to finer resolution histogram bins.

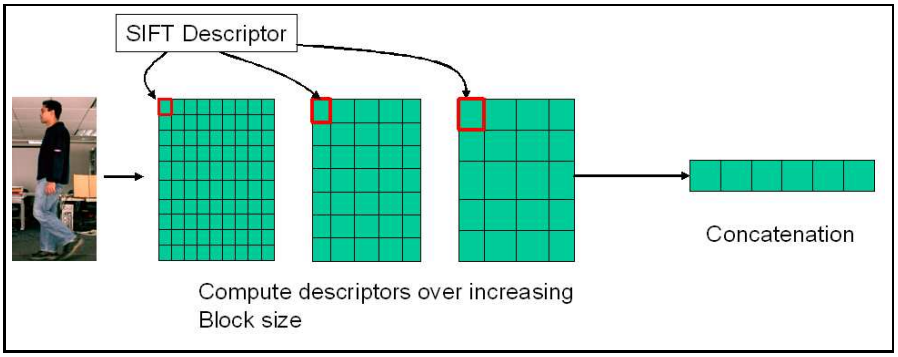


Figure 5.5: Multi-Level Spatial Block(MSB) computes encodings at each level by computing local SIFT descriptors with increasing block size, on patches centered at regular grid locations.

Multi-Level Spatial Blocks(MSB) These multi-level encodings are obtained by computing the dense grid based local SIFT descriptors over progressively larger neighborhood. SIFT descriptors, as proposed by Lowe *et.* al.[112], has been widely applied for the task of feature matching and object recognition. The SIFT descriptor is constructed as a concatenation of gradient orientation histograms computed over the block of regularly placed cells. The

cells accumulate the gradient orientation information over a small region of pixels and uses bilinear interpolation to soft vote the to quantized orientation bins of the histogram, thus enhancing the invariance to affine transformation and misalignment. The Gaussian smoothing of the entire block assigns higher weights to gradients near the block center. For the Multi-level Spatial Blocks(MSB), higher levels are encoded using SIFT blocks with larger cell size. Larger block size causes more smoothing(due to larger Gaussian scale) thereby avoiding noisy details from being encoded. Whereas Larger cell size aggregates the gradient orientations over larger pixels thus making it more invariant to the local geometric deformations. The blocks are locally normalized in order to make the descriptor invariant to illumination changes. Each level of the MSB can be computed independently, with varying degree of selectivity and invariance(fig. 5.5).

Spectral clustering methods are useful techniques for visualization of high dimensional data by extracting low dimensional, perceptual representations that preserve the topology of the points in the original ambient space. Fig. 5.8 shows the 3D spectral embeddings(Isomap) of hierarchical encodings for the walking sequence of a synthetic computer graphics(CG) model. The sequence involved 2 full walking cycles viewed from the side. It is evident that left-right leg assignment ambiguities exist for the Hyperfeatures, Vocabulary tree and Spatial pyramid features. While HMAX and MSB representations are able to efficiently distinguish between them by mapping the observations to different points in the latent space.

5.3 Metric Learning and Correlation Analysis

Two similar human poses in the scene may have different image descriptors due to misalignment of the bounding box and different backgrounds Moreover, different anthropometry of the human targets may cause body parts to appear at different locations with respect to the bounding box. Although multi-level/hierarchical encodings improve the tolerance of the image descriptors to local misalignment of bounding box and geometric deformations, they are still affected by the background clutter and variations in body proportions of the human target. It is difficult to explicitly model these variations in the training of the discriminative framework, primarily due to lack of representative dataset containing sufficient variability in body proportions. In

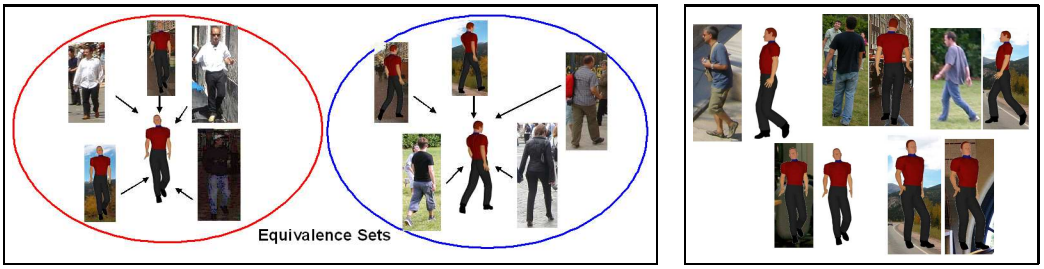
addition, the background clutter is difficult to remove from the bounding box due to the large variability in the aspect ratios of different human body poses (e.g. compare a human standing straight with the human standing with arms wide open). These make the trained models difficult to generalize on the data set with unseen subjects or backgrounds. Fig. 5.9 illustrates this on an image sequence of synthetic and real human model walking parallel to the image plane. The figure on the right shows the Isomap embeddings of the MSB image descriptors computed for image sequence with clean background, cluttered background and real human sequence. For comparison, we have temporally aligned the walking cycles of the real image sequence with the synthetic human sequence. Notice that although the embeddings obtained using Isomap preserve the topology of the points in the learned latent space, two perceptually similar poses are mapped to latent points that are very far from each other. This indicates that simply changing the background of the human target, may cause the image descriptors to appear different. Predictors, that use these descriptors for estimating poses may easily get confused due to this and output a completely different pose.

A careful analysis of computation of these descriptors indicate that specific regions of the bounding box contribute (votes for) to fixed set of spatial bins of the image descriptor histograms. It therefore appears reasonable to assume that the noise introduced due to background clutter can be mitigated by downweighting those histogram bins where the background image regions has voted. Linear predictors implicitly assume Euclidean metric in input space, whereas kernel methods use an explicit metric induced by the selected kernel, both of which assign equal weights to all the histogram bins of the descriptor. In either case, there is no guarantee that an Euclidean metric with covariance estimated from the learning algorithm, would provide the best invariance with respect to the task e.g. classification, the invariance to within the same pose class. Hence there is a need for problem dependent metric that can be used to compare the image descriptors using predefined notion of similarity/dissimilarity.

In this section we discuss learning techniques to build a metric - or alternatively, to compute representations with implicit Euclidean metric, for a desired level of invariance to changes in characteristics of the scene. For the task of 3D human pose reconstruction, we would like the models trained on synthetic datasets to generalize well on the real image sequences. In order to make the descriptors invariant to human targets or background clutter, we define a

invariance class which is composed of images of real and synthetic humans having different body proportions in the same pose but with a different backgrounds. We learn metric that maximizes similarity between the descriptors computed for each of images in these invariance class. It should be noted that learning a metric amounts to learning a full covariance matrix that downweights the unwanted variance in the data using a linear transformation. This is equivalent to learning a linear subspace in which within class variance is minimized. In practice we need to train with only a few qualitatively different poses in order to learn a useful metric that can be used with any pose.

There exist extensive literature on distance metric learning with more recent works include Wagstaff *et.* al.[192], Xing *et.* al.[198] and Hillel *et.* al.[20]. In this section we present 2 techniques, intended towards learning a task dependent metric for improving invariance of the image descriptors - Relevant Component Analysis and Canonical Correlation fig. 5.6. The first approach learns a linear subspace from a set of pre-specified equivalence classes, such that the distance between the projected points belonging to the same equivalence class is minimized. The second approach computes a pair of independent subspaces such that the correlation between a set of paired images that are known to be strongly correlated is maximized when they are projected in it.



(a) (b)

Figure 5.6: This figure illustrates the training of distance metric learning framework for *(a)* Relevant Component Analysis(RCA) and *(b)* Canonical Correlation Analysis(CCA). RCA is trained using a number of equivalence sets each composed of people in similar poses but different backgrounds and body proportions. CCA requires a number of pair of images that are known to be strongly correlated.

5.4 Relevant Component Analysis (RCA)

Relevant Component Analysis learns a full covariance matrix by minimizing the spread within each invariance class of images containing humans in similar pose but different background. The images in these classes are related to each other using equivalence relation and hence these chunks of data points are refered to as equivalence class. RCA uses pairwise similarity between the data points as the side information and learns a linear transformation to assign lower weights to the dimensions having high in-class variability and vice versa. This effectively down-scales the global unwanted variability within the data by assigning higher weights to relevant dimensions of image descriptor and lower weights to dimensions that are due to background noise. The new feature space obtained from RCA reveals the intrinsic structure of the data points such that more robust models can be trained using them. Learning in RCA is done by minimizing the within-class covariance with an additional constraint to prevent the trivial solution of shrinking of the entire subspace. The cost function is written as:

$$\min_{\mathbf{D}} \frac{1}{U} \sum_{j=1}^{k} \sum_{i=1}^{U_j} ||\hat{\mathbf{r}}_{ji} - \mathbf{m}_j||_{\mathbf{D}}, \text{ s.t. } |\mathbf{D}| \geq 1 \tag{5.1}$$

where U is the total number of examples and U_j is the number of examples in the j^{th} equivalence class. Also $\mathbf{m}_j$ is the mean of the j^{th} equivalence class with centered data points as $\hat{\mathbf{r}_{ji}} = \mathbf{r}_{ji} - \mathbf{m}$. The minimization of the (5.1) can be effectively achieved using a single matrix computation of the in-class covariance matrix $\mathbf{C}$[20]. This is computed in a closed form as:

$$\mathbf{C} = \frac{1}{U} \sum_{j=1}^{k} \sum_{i=1}^{U_j} (\hat{r}_{ji} - m_j)(\hat{r}_{ji} - m_j)^T \tag{5.2}$$

The estimated covariance matrix can be used to compute the Mahalanobis distance $\mathbf{D}$. Alternatively, the covariance matrix can be diagonalized using whitening transformation to obtain a linear subspace in which the similarity between the data points belonging to the same class is maximized. This also provides a technique for dimensionality reduction by using eigenvectors corresponding to the largest eigenvalues of the whitening transformation matrix.

5.5 Canonical Correlation Analysis(CCA)

Correlation analysis adopts a different approach to enhance similarity between the image descriptors of the humans in similar poses but taken under different backgrounds and having different body proportions. As opposed to reducing distance between these image descriptors, it aims to learn a pair of linear subspace such that their projections are maximally correlated.

Correlation analysis is not a novel concept and had been widely used in past, in a variety of forms for detecting semantically similar regions. Normalized cross correlation analysis formalizes the notion of detecting regions that correlates maximally with the mask and can been used for template matching and object detection. Ordinary correlation analysis however, largely depends on the basis system in which the variates are defined. The 2 variables may be highly correlated but their relationship may not be visible in the current subspace. Canonical Correlation Analysis, as introduced by [81] aims to find principle subspace in which the shared structure between the 2 classes is evident. Given two sets of vectors $\mathbf{r}$ and $\mathbf{u}$, as samples: $\mathbf{S} = ((\mathbf{r}_1, \mathbf{u}_1), (\mathbf{r}_2, \mathbf{u}_2), \ldots, (\mathbf{r}_n, \mathbf{u}_n))$, and their projection on two arbitrary directions, $\mathbf{w}_r$ and $\mathbf{w}_u$, with $\mathbf{S}_r = (\langle \mathbf{w}_r, \mathbf{r}_1 \rangle, \ldots, \langle \mathbf{w}_r, \mathbf{r}_n \rangle)$, and $\mathbf{S}_u = (\langle \mathbf{w}_u, \mathbf{u}_1 \rangle, \ldots, \langle \mathbf{w}_u, \mathbf{u}_n \rangle)$, CCA maximizes the cost:

$$f = \max_{\mathbf{w}_r, \mathbf{w}_u} \frac{\langle \mathbf{S}_r \mathbf{w}_r, \mathbf{S}_u \mathbf{w}_u \rangle}{||\mathbf{S}_r \mathbf{w}_r|| ||\mathbf{S}_u \mathbf{w}_u||} = \tag{5.3}$$

$$\max_{\mathbf{w}_r, \mathbf{w}_u} \frac{\mathbf{w}_r \top \mathbf{C}_{ru} \mathbf{w}_u}{\sqrt{\mathbf{w}_r^\top \mathbf{C}_{rr} \mathbf{w}_r \mathbf{w}_u^\top \mathbf{C}_{uu} \mathbf{w}_u}} \tag{5.4}$$

with $\mathbf{C}_{rr}$ and $\mathbf{C}_{uu}$ *within-sets* covariance matrices and $\mathbf{C}_{ru} = \mathbf{C}_{ur}^\top$ *between-sets* covariances. As clear from the eqn. (5.3), CCA is invariant to the affine transformation of the variables. The optimization can be computed in the closed form using the eigenvalue decomposition problem as following:

$$\mathbf{C}_{rr}^{-1} \mathbf{C}_{ru} \mathbf{C}_{uu}^{-1} \mathbf{C}_{ur} \mathbf{w}_r = \lambda^2 \mathbf{w}_r \tag{5.5}$$

$$\mathbf{C}_{uu}^{-1} \mathbf{C}_{ur} \mathbf{C}_{rr}^{-1} \mathbf{C}_{ru} \mathbf{w}_u = \lambda^2 \mathbf{w}_u \tag{5.6}$$

It should be noted that only one of the eigenvalue problem needs to be solved. As the covariance matrix may be singular, $\mathbf{C}^{-1}$ may not be always possible for the eqn. (5.5). This near collinearity phenomenon tends to make the solution highly sensitive to the random variations

of the 2 set of variates. In order to avoid the overfitted solution we penalize the norms of the learned basis vectors. The optimization eqn.(5.3) can be reformulated as:

$$\max_{\mathbf{w}_r,\mathbf{w}_u} \langle \mathbf{S}_r\mathbf{w}_r, \mathbf{S}_u\mathbf{w}_u \rangle \text{ s.t. } \mathbf{w}_r^T\mathbf{C}_{rr}\mathbf{w}_r + \gamma||\mathbf{w}_r||^2 = 1 \text{ and } \mathbf{w}_u^T\mathbf{C}_{uu}\mathbf{w}_u + \gamma||\mathbf{w}_u||^2 = 1 \quad (5.7)$$

The regularization coefficient γ can be set using cross-validation. Above optimization yields the following closed form eigen-decomposition problem:

$$(\mathbf{C}_{rr} + \gamma\mathbf{I})^{-1}\mathbf{C}_{ru}(\mathbf{C}_{uu} + \gamma\mathbf{I})^{-1}\mathbf{C}_{ur}\mathbf{w}_r = \lambda^2\mathbf{w}_r \quad (5.8)$$

$$(\mathbf{C}_{uu} + \gamma\mathbf{I})^{-1}\mathbf{C}_{ur}(\mathbf{C}_{rr} + \gamma\mathbf{I})^{-1}\mathbf{C}_{ru}\mathbf{w}_u = \lambda^2\mathbf{w}_u \quad (5.9)$$

λ^2 is the squared canonical correlation and the eigenvectors $\mathbf{w}_r$ and $\mathbf{w}_u$ are the canonical correlation basis vectors. The larger the eigenvalues λ, the greater the correlation between the 2 projected set of variables. The minimum of the dimensions of $\mathbf{r}$ and $\mathbf{u}$ is the maximum correlation basis vectors that can be used for CCA. Large problems can be solved efficiently using predictive low-rank decomposition with partial Gram-Schmidth orthogonalization. Non-linear extensions to CCA can be obtained using the standard kernel trick of projecting the data points to high dimensional feature space [79] and performing CCA in it.

5.6 Semi-supervised Learning using Manifold Regularization

Semi-supervised learning employs both the labeled and the unlabeled data points in the training of models. Labeled examples are pair of data $(\mathbf{x}, \mathbf{r})$ sampled from the joint distribution $p(\mathbf{x}, \mathbf{r})$ whereas unlabeled data are $\mathbf{r}$ drawn from the marginal distribution $p(\mathbf{r})$. Marginal distribution $p(\mathbf{r})$ can be used as a prior, to additionally constrain the parameter search in a discriminative learning framework. It should be noted that the need for semi-supervised learning is more profound for a discriminative framework that directly learns the conditional distribution $p(\mathbf{x}|\mathbf{r})$ and completely ignores the marginal distribution $p(\mathbf{r})$ which is available in the form of unlabeled data. This is in contrast to generative framework which learns the joint distribution $p(\mathbf{x}, \mathbf{r})$ and implicitly models the marginal distribution $p(\mathbf{r})$.

A number of works exist in literature that incorporate the unlabeled data to improve the learning of statistical models. Recent works on semi-supervised learning include Transductive

learning ([188],[91]), Semi-supervised SVMs[27], Manifold Regularization[23], Co-training[35] and gradient based regularization [38]. More notable among these is the semi-supervised learning framework proposed by Belkin *et.* al.[23] which incorporates additional information about the geometric structure of the marginal distribution $p(\mathbf{r})$ to regularize the function to be learned. In the following sections, we extend the manifold regularization framework to mixture of experts training.

5.6.1 Manifold Regularization

Manifold regularization(MR) was introduced to incorporate the geometric information of the unlabeled data for improving the function learning. Statistical learning involves modeling functional relationship between predictor variables $\mathbf{r}$ and response variables $\mathbf{x}$. Functions f can either be data interpolant(regression) or a boundary between two classes(classification). Typically, statistical learning involves supervised training using a set labeled exemplars (x_i, r_i) that are noisy samples from the joint distribution $p(\mathbf{x}, \mathbf{r})$ Manifold Regularization applies additional constraints to the learning in order to control the complexity of function with respect to the geometry of the input points. MR is based on the assumption that the conditional distribution of the dataset $p(\mathbf{x}|\mathbf{r})$ varies smoothly along the geodesics in the intrinsic geometry of the marginal distribution $\mathcal{P}_{\mathbf{r}}$. This constraint is different from the smoothness constraint to control the complexity of the model in the ambient space which does not take into account the topology of the input data $\mathbf{r}$. This smoothness constraint can be enforced by optimizing the objective function with an additional regularization term that penalizes the gradient of the mapping function f along the geometry of the marginal distribution. Effectively, it penalizes large changes of the function f in the region near the manifold $\mathcal{M}$ of the input points $\mathbf{r}$.

In most cases where the analytical form of the marginal density $\mathcal{P}_{\mathbf{r}}$ is not known, support of the marginal distribution can be approximated using graph Laplacian $\mathbf{L}$. Graph Laplacian is a discrete approximation of a continuous manifold and is constructed by approximating geodesics in the intrinsic geometry using Euclidean distance in a local neighborhood.

The manifold regularization term can be constructed from the both labeled and unlabeled

exemplars, for the linear mapping function $f(\mathbf{r} : \mathbf{W}) = \mathbf{W}^T\mathbf{r}$ using the graph Laplacian as:

$$\mathcal{R}_i = \int_{\mathcal{M}} ||\nabla \mathbf{f}||^2 d\mathcal{P}_\mathbf{r} \approx \sum_{u,j}^{U} (f(\mathbf{r}_u) - f(\mathbf{r}_j))^2 N_{uj} \tag{5.10}$$

The regularization term can be re-written as:

$$\mathcal{R}_i = \sum_{u,j=1}^{U} (\mathbf{W}_i\mathbf{r}_u - \mathbf{W}_i\mathbf{r}_j) N_{uj} (\mathbf{W}_i\mathbf{r}_u - \mathbf{W}_i\mathbf{r}_j)^\top = \mathbf{W}_i\mathbf{R}^\top\mathbf{L}\mathbf{R}\mathbf{W}_i^\top \tag{5.11}$$

For the non-linear mapping function, the data points $\mathbf{r}_u$ are projected to high dimensional feature space using the kernel map as $\mathbf{K}(\mathbf{r}_j) = [K(\mathbf{r}_j, \mathbf{r}_1), \ldots, K(\mathbf{r}_j, \mathbf{r}_l), \ldots, K(\mathbf{r}_j, \mathbf{r}_{l+u})]^T$ and the function is estimated as $f(\mathbf{r} : \mathbf{W}) = \mathbf{W}^T\mathbf{K}(\mathbf{r})$. Notice here, that both labeled and unlabeled points can be used as initial set of basis vectors. The regularization term can then be obtained as:

$$\mathcal{R}_i = \sum_{u,j=1}^{U} (\mathbf{W}_i\mathbf{K}(\mathbf{r}_u) - \mathbf{W}_i\mathbf{K}(\mathbf{r}_j)) N_{uj} (\mathbf{W}_i\mathbf{K}(\mathbf{r}_u) - \mathbf{W}_i\mathbf{K}(\mathbf{r}_j))^\top = \mathbf{W}_i\mathbf{K}^\top\mathbf{L}\mathbf{K}\mathbf{W}_i^\top \tag{5.12}$$

where U is the size of the entire training set, that includes both labeled and unlabeled points. $\mathbf{N}$ is a matrix of graph weights N_{ij} that depends on the geodesic distance of the neighbor j to the data point i along the manifold of input points. $\mathbf{R}$ is a $\dim(\mathbf{r}) \times U$ matrix that stores all the input vectors $\mathbf{r}$ in the training set and $\mathbf{K}$ is a $U \times U$ is the kernel matrix. Graph Laplacian $\mathbf{L}$[1]$= \mathbf{D} - \mathbf{N}$ with $\mathbf{D}$ as a diagonal matrix containing elements $D_{ii} = \sum_{j=1}^{U} N_{ij}$ can be directly constructed from the training vectors. The additional penalty term controls the complexity of the function in the intrinsic geometry in the same way as weights prior control the complexity in the ambient space. The function learning (regression or classification) can be formulated as the following optimization problem:

$$f^* = \text{argmin}_f \left[\mathcal{L}(\mathbf{x}, \mathbf{r}|\mathbf{W}) + \gamma_A \|\mathbf{W}\|^2 + \gamma_I \mathbf{W}_i\mathbf{K}^\top\mathbf{L}\mathbf{K}\mathbf{W}_i^\top \right] \tag{5.13}$$

The regularization coefficients γ_A and γ_I can be estimated using cross-validation.

[1]This is (typically) a sparse graph construction, obtained by connecting each training point to its k-nearest neighbors and computing local Gaussian distances to them. A global regularizer based on geodesic distances can also be used.

5.6.2 Semi-supervised Sparse Bayesian Classification

Bayesian learning has been widely applied to training of regressors and classifiers that are sparse and have lower tendency to overfit the data. This is primarily due to the property of automatic relevance determination(ARD) mechanism that allows us to prune off irrelevant weight parameters of the mapping function during learning

In this section we extend manifold regularization framework to sparse Bayesian learning[180] Using the proposed framework we can iteratively estimate both the intrinsic geometry regularization coefficients γ_I and the ambient space regularization coefficients γ_A. Here we only present the formulation for the binary classifier and can easily extend the framework to regression as well. For a two-class classifier with the inputs $\mathbf{r}_i$ and the class labels $\mathbf{x}_i$, the non-linear classification boundary is represented as $f(\mathbf{r} : \mathbf{W}) = \mathbf{W}^T\mathbf{K}(\mathbf{r})$. For linear boundaries the function takes the following form $f(\mathbf{r} : \mathbf{W}) = \mathbf{W}^T\mathbf{r}$ with the kernel mapping $\mathbf{K}$ replaced by the original input vectors $\mathbf{r}$. The binary classification function is learned as the logistic sigmoid function $\sigma(f) = 1/(1 + e^{-f})$, with the likelihood as a binomial distribution:

$$p(\mathbf{x}|\mathbf{r}, \mathbf{W}) = \prod_{n=1}^{N} \sigma\{\mathbf{K}(\mathbf{r}_n), \mathbf{W}\}^{x_n} \left[1 - \sigma\{\mathbf{K}(\mathbf{r}_n), \mathbf{W}\}\right]^{1-x_n} \tag{5.14}$$

Here the target labels $\mathbf{x}$ lie in the set $\{0, 1\}$.

In the sparse Bayesian learning framework we explicitly define a prior distribution on the weights parameters $\mathbf{W}$, that is controlled by the hyperparameters $\boldsymbol{\alpha}_i$ corresponding to each weight parameter $\mathbf{W}_i$. This prior is referred to as the *ambient* prior as it controls the complexity of the classifier in the ambient space.

$$p_a(\mathbf{W}) = \prod_{i=0}^{N} \mathcal{N}(W_i|0, \frac{1}{\alpha_i}) \tag{5.15}$$

However in order to ensure that classification function is smooth along the support $\mathcal{M} = supp\{P(\mathbf{r})\}$ of the marginal distribution (manifold of the input points) we introduce additional Gaussian prior to penalize large change in the function $\mathbf{f}$ along the intrinsic geometry of the manifold $\mathcal{M}$(approximated as graph laplacian $\mathbf{L}$). We refer to this prior as *intrinsic* prior and is defined as:

$$p_i(\mathbf{W}) \propto \exp\{-\gamma_I \sum_{u,j}^{U} (f(\mathbf{r}_u) - f(\mathbf{r}_j))^2 N_{uj}\} = \exp\{-\gamma_I \mathbf{W}^T\mathbf{K}\mathbf{L}\mathbf{K}^T\mathbf{W}\} \tag{5.16}$$

Here, the similarity values N_{uj} and the regularization coefficients γ_I are same as discussed in the previous section. As $\mathbf{L}$ may be singular, the precision matrix $\mathbf{K}^T\mathbf{L}\mathbf{K}$ can be singular. This may cause the corresponding prior to be improper thus playing no role in the estimation of the $\mathbf{W}$ by maximizing the weights posterior. In order to avoid this, we define a joint prior on the weights parameters using both the ambient prior and the intrinsic geometry prior:

$$p(\mathbf{W}|\gamma_I, \boldsymbol{\alpha}) \propto \exp\{-\{\gamma_I \mathbf{f}^T\mathbf{L}\mathbf{f} \ + \mathbf{W}^T\mathbf{A}\mathbf{W}\}\} = \exp\{-\mathbf{W}^T\Psi(\gamma_I, \boldsymbol{\alpha})\mathbf{W}\} \tag{5.17}$$

where $\mathbf{A} = diag\{\alpha_1, \alpha_2, \ldots, \alpha_N\}$ and $\Psi(\gamma_I, \boldsymbol{\alpha}) = \mathbf{K}^T\mathbf{L}\mathbf{K} + \mathbf{A}$. $\mathbf{K}$ is the kernel matrix obtained from both labeled and unlabeled data points. The term $\mathbf{A}$ with non-negative diagonal terms acts as an additional regularization term in the precision matrix. The set of hyperparameters include the parameters $\{\boldsymbol{\alpha}, \gamma_I\}$. We define hierarchical priors on these hyper-parameters (referred to as hyper-priors) as Gamma distribution:

$$p(\boldsymbol{\alpha}) = \prod_{i=0}^{N} Gamma(\alpha_i|a, b) \tag{5.18}$$

where

$$Gamma(t|a, b) = \Gamma(a)^{-1} b^a \alpha^{a-1} e^{-b\alpha} \tag{5.19}$$

In our formulation, we set the parameters of the gamma distribtution to zero i.e. $a = b = 0$ to yield non-informative gamma hyper-priors. Use of individual hyper-parameters α_i with ARD typically controls the posterior probability for each of the weights W_i and enables weights pruning.

The weights posterior $p(\mathbf{W}|\mathbf{x}, \boldsymbol{\alpha}, \gamma_I) \ \propto \ p(\mathbf{x}|\mathbf{W}, \boldsymbol{\alpha}, \gamma_I)p(\mathbf{W}|\boldsymbol{\alpha}, \gamma_I)$ is iteratively maximized by first estimating the most probable values of the hyper-parameters and using them to estimate the posterior distribution of the weight parameters. Most probable values of the hyper-parameters are obtained by maximizing the marginal evidence of the hyper-parameters (*Type-II Likelihood*) and pruning off irrelevant weights whose posterior is mostly concentrated at zero i.e. the most probable value of the variance hyperparameter is very small. Marginal evidence for the hyper-parameters $\boldsymbol{\alpha}$ and γ_I is computed as:

$$p(\mathbf{x}|\boldsymbol{\alpha}, \gamma_I) = \int p(\mathbf{x}|\mathbf{W}, \boldsymbol{\alpha}, \gamma_I)p(\mathbf{W}|\boldsymbol{\alpha}, \gamma_I)d\mathbf{W} \tag{5.20}$$

The marginal evidence eqn. (5.20) can be exactly computed for bayesian regression as both the distribution terms in the integrand are Gaussian. For binary classification, the likelihood

is a binomial distribution and the marginalization is analytically intractable. We therefore use Laplace approximation that estimates the integral as a local Gaussian approximation of the integrand over the neighborhood of the mode. The integrand in this case is the weights $\mathbf{W}$ posterior. The modes of the non-Gaussian posterior distribution can be obtained by conveniently optimizing its logarithm using *Iterative Reweighted Least Square(IRLS)*:

$$\log\left[p(\mathbf{x}|\mathbf{W},\boldsymbol{\alpha},\gamma_I)p(\mathbf{W}|\boldsymbol{\alpha},\gamma_I)\right] = \sum_{n=1}^{N}\left[x_n \log(\sigma\{\mathbf{K}_L,\mathbf{W}\}) + (1-x_n)\log(1-\sigma\{\mathbf{K}_L,\mathbf{W}\})\right] \tag{5.21}$$

$$-\frac{1}{2}\mathbf{W}^T\mathbf{A}\mathbf{W} - \gamma_I\mathbf{W}^T\mathbf{K}^T\mathbf{L}\mathbf{K}\mathbf{W} \tag{5.22}$$

Here $\mathbf{K}_L$ is the kernel matrix for only labeled examples whereas $\mathbf{K}$ is the kernel matrix using both labeled and unlabeled examples. It should be noted that both $\mathbf{K}$ and $\mathbf{K}_L$ use same set of basis functions which include both labeled and unlabeled data points. The optimization of the marginal evidence eqn.(5.20) w.r.t hyperparameters $\boldsymbol{\alpha}$ and γ_I yields closed form updates for the hyperparameters that are used to iteratively prune off the weights. The details of derivation of the results are provided in Appendix B. The covariance of the fitted Gaussian is estimated as hessian of the weights posterior:

$$\boldsymbol{\Sigma} = (\mathbf{K}_L^T\mathbf{B}\mathbf{K}_L + \mathbf{A} + \gamma_I\mathbf{K}^T\mathbf{L}\mathbf{K})^{-1} \tag{5.23}$$

where $\mathbf{B} = diag\{b_1, b_2, \ldots, b_N\}$ and $b_i = \sigma\{\mathbf{W}^T\mathbf{K}_L(r_i)\}(1-\sigma\{\mathbf{W}^T\mathbf{K}_L(r_i)\})$. The weights are estimated as

$$\mathbf{W} = \boldsymbol{\Sigma}\mathbf{K}_L^T\mathbf{B}\mathbf{x} \tag{5.24}$$

The hyper-parameters α_i and γ_I are estimated as:

$$\alpha_i^{(k+1)} = \frac{\alpha_i^{(k)}T_{ii}}{\Sigma_{ii} + W_i^2} \text{ where } \mathbf{T} = (\mathbf{A} + \gamma_I\mathbf{K}^T\mathbf{L}\mathbf{K})^{-1} \tag{5.25}$$

$$\gamma_I^{(k+1)} = \frac{\gamma_I^{(k)}\mathbf{Tr}\left[\mathbf{T}(\mathbf{K}^T\mathbf{L}\mathbf{K})\right]}{\mathbf{Tr}\left[\boldsymbol{\Sigma}(\mathbf{K}^T\mathbf{L}\mathbf{K})\right] + \mathbf{W}^T(\mathbf{K}^T\mathbf{L}\mathbf{K})\mathbf{W}} \tag{5.26}$$

The weights that have their posterior probability concentrated at zero are subsequently pruned off at every iteration. At each iteration, both the ambient regularization coefficients α_i and the manifold regularization coefficient γ_I are re-estimated.

5.6.3 Semi-supervised Learning for Bayesian Mixture of Experts

We apply the semi-supervised learning framework to the training of mixture of experts(ME) model, by incorporating additional information of the geometry of the input data. The geometry of the marginal distribution of the input data can be used to improve the learning of the experts and the gate parameters of the ME model. In our framework, we learn Mixture of Experts(ME) model using sparse bayesian learning and refer to this model as Bayesian Mixture of Experts (BME). Manifold Regularization can be applied to efficiently estimate parameters of the BME model[2]. The Bayesian mixture of experts(BME) model is composed of a gate distribution $\mathbf{g}$(essentially a multi-category classifier) that clusters the dataset into M classes, and a number of expert regressors $\mathbf{f}$ that are locally fitted within each of the clusters. The gate distribution is learned as multiple one-against-rest binary classifiers. The expert conditional distribution is a Gaussian with mean as the kernel regressors outputs and scale parameter inferred from data itself.

$$p(\mathbf{x}|\mathbf{r}) = \prod_{i=1}^{M} [\mathbf{g}_i(\mathbf{r})\mathbf{f}_i(\mathbf{x}|\mathbf{r})]^{z_i} \tag{5.27}$$

$$\mathbf{g}_i(\mathbf{r}) = \frac{\exp(\boldsymbol{\lambda}_i^\top \mathbf{r})}{\sum_k \exp(\boldsymbol{\lambda}_k^\top \mathbf{r})} \tag{5.28}$$

$$\mathbf{f}_i(\mathbf{x}|\mathbf{r}) = \mathcal{G}(\mathbf{x}|\mathbf{W}_i\mathbf{r}, \boldsymbol{\Omega}_i^{-1}) \tag{5.29}$$

where z_i is the indicator variable that assumes value 1 if i^{th} expert is used for mapping the data $\mathbf{r}$ to $\mathbf{x}$. The learning of the parameters $\Theta_i = \{\Lambda_i, \mathbf{W}_i\}$ is accomplished using Expectation-Maximization algorithm. In the E-step the data set is grouped into different clusters using the expected value of the likelihood function. In the M-step the parameters for the gate distribution and the expert regressors are estimated by maximizing the expectation of the likelihood function. For the test inputs, the estimated gate distribution is used to decide, which expert is best suited for the prediction. The output is either a weighted linear combination of outputs from all the experts or the most probable expert output.

Manifold regularization can be naturally extended to improve the learning of BME model by putting additional constraints on the function parameters, in order to ensure smoothness along the manifold of the input data. For BME, we simultaneously train 2 functions - the

[2]Refer to chapter 3 for the detailed formulation of the BME model

expert function and the gate function. In BME the gate function is a multi-category classifier that clusters the entire dataset into multiple categories and the expert function is locally fitted within each of these categories. Therefore the expert mappings should be constrained to respect the domain boundaries as represented by the gate distribution and should vary smoothly along the geodesics of input geometry belonging to the same category only. For training the experts and the gates, it is equivalent to enforcing the following two constraints:

- Local Expert assumption: The experts conditional distribution $p(\mathbf{x}|\mathbf{r})$ should be smooth for the input points $\mathbf{r}$ that are close in the intrinsic geometry of the expert inputs $p(\mathbf{r})$. In effect this means that the expert prediction for similar inputs should be similar.

- Expert Ranking assumption: Inputs $\mathbf{r}$ that are close in the intrinsic geometry $p(\mathbf{r})$, should have similar expert ranking, as predicted by the gate distribution $\mathbf{g}_i(\mathbf{r})$.

The need for an additional expert ranking assumption arise due to the locality constraint of the expert regressors. The expert regressors are learned using only those inputs $\mathbf{r}_i$ for which the gate distribution g_i assigns higher weights. The inputs that have low confidence of belonging to the same cluster are ignored when fitting the expert. In most cases, the marginal distribution over inputs are not available and it is approximated by graph Laplacian $p(\mathbf{r}) \approx \mathcal{P}_{\mathbf{r}}$ constructed from the discrete samples of the training data that include both labeled and unlabeled input data. Using the graph Laplacian, we can estimate a similarity measure between any two pair of points i and j as N_{ij} if they lie within the ϵ-ball neighborhood. These constraints manifests as an additional regularization term in the objective function that is optimized to learn the regression function(or classification boundary). The additional regularization term ensures smoothness along the intrinsic geometry of the inputs and has the same form as in (5.11) and (5.12). However, for learning expert functions, we need to enforce expert ranking constraint to ensure that only inputs belonging to the same cluster are used for learning the expert. This is accomplished by re-weighting the inputs (and the outputs) by the confidence value as represented by the posterior distribution h, estimated in the Expectation step as:

$$E[z_i] = h_i(\mathbf{x}, \mathbf{r}|\mathbf{W}, \mathbf{\Omega}, \boldsymbol{\lambda}, \boldsymbol{\beta}) = \frac{\mathbf{g}(\mathbf{r}|\boldsymbol{\lambda}_i, \boldsymbol{\beta}_i) p_i(\mathbf{x}|\mathbf{r}, \mathbf{W}_i, \mathbf{\Omega}_i^{-1})}{\sum_{m=1}^{M} \mathbf{g}(\mathbf{r}|\boldsymbol{\lambda}_m, \boldsymbol{\beta}_m) p(\mathbf{x}|\mathbf{r}, \mathbf{W}_m, \mathbf{\Omega}_m^{-1})} \tag{5.30}$$

The posterior h_i is the responsibility that $\mathbf{r}$ is mapped to $\mathbf{x}$ using the i^{th} expert mapping. It can be shown by maximizing the regularized likelihood, that the expectation of the indicator

variable z_i is equal to the responsibilities h_i. The similarity measure between any two inputs i and j, N_{ij} should be low not only when they are far away on the intrinsic geometry but also when the expert ranking(or the expert responsibilities) as represented by the posterior h are different. We estimate the similarity measure as $\mathcal{N}_{ij} = N_{ij} \min(h(\mathbf{x}_i, \mathbf{r}_i), h(\mathbf{x}_j, \mathbf{r}_j))$. The manifold regularization term for the expert learning can therefore be reformulated as:

$$\mathcal{R}_i = \mathbf{W}_i \begin{bmatrix} E[\mathbf{Z}_i^{(L)}]\mathbf{R}^{(L)} & E[\mathbf{Z}_i^{(U)}]\mathbf{R}^{(U)} \end{bmatrix}^T \mathcal{L} \begin{bmatrix} E[\mathbf{Z}_i^{(L)}]\mathbf{R}^{(L)} & E[\mathbf{Z}_i^{(U)}]\mathbf{R}^{(U)} \end{bmatrix} \mathbf{W}_i^T \quad (5.31)$$

where $E[\mathbf{Z}_i^{(L)}]$ is the matrix of expected value of the indicator variables corresponding to the labeled inputs $\mathbf{R}^{(L)}$ that is computed in the expectation step eqn. (5.30). For the unlabeled inputs $\mathbf{R}^{(U)}$ the expectation value cannot be computed using the equation (5.30) as the ground truth labels are not available for them. Therefore for the unlabeled data, we use the expected value of the labels as represented by the gate distribution $E[\mathbf{Z}_i^{(U)}] \approx \mathbf{g}_i(\mathbf{R}^{(U)})$, which is learned at every M-Step of the Expectation-Maximization algorithm. This is a reasonable approximation, as the gate distribution models the expectation value $E[\mathbf{Z}^{(L)}]$ as a function of input variables $\mathbf{r}$[3]. The graph Laplacian $\mathcal{L} = \mathcal{D} - \mathcal{N}$ is also re-estimated at every iteration of the Expectation-Maximization(EM) using the re-weighted inputs where $\mathcal{D}_{ii} = \sum_{j=1}^{U} \mathcal{N}_{ij}$.

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

Figure 5.7: Effect of background clutter and misalignment on sparse and dense image encodings. The figure shows a sample image from three sets of test images, (a) shows a well-aligned bounding box with a synthetic CG avatar in a clean background, (b) shows the same CG model in the same pose with a real background, (c) shows the same but with the misaligned bounding box. The plot in (d) compares the prediction error of BME model when it is trained on sparse histogram based descriptors with the model trained on dense, grid based descriptors. Notice that whereas grid based descriptors are more robust to background clutter, less spatially constrained histogram based descriptors have better performance when the bounding box is not well aligned.

[3]Described in more detail in chapter 3

5.7 Experiments

We evaluate our discriminative framework for human pose inference on both synthetic and real data. In this section we report results from quantitative comparison between the proposed hierarchical descriptors based on 3D pose prediction accuracy. Fig. 5.7 illustrates the effect of background clutter and misalignment of the bounding box on the 3D pose inference using bayesian mixture of experts (BME) model. Here we compare the prediction accuracy of the BME models trained on sparse histogram based SIFT descriptors and grid based block SIFT descriptors, computed densely on a bounding box. Histogram SIFT features are obtained as sparse co-occurrence statistics of codebook patches on a regularly spaced grid over a bounding box. Dense features are obtained by concatenating the SIFT descriptors computed over a local patch at regularly spaced grid centers. We train a BME model on a training image sequence of bending and pick up containing both clean and clutter background, but with well aligned bounding boxes. The plots show the average prediction error per joint angle and clearly demonstrate that under similar training conditions prediction accuracy using sparse global histogram based descriptors is worse compared to dense local grid based descriptors. However global histograms are more robust to misalignment of the bounding box.

Multilevel encodings intend to overcome these deficiencies by representing the image at multiple levels of abstraction, with lower levels being more selective but spatially restrictive and less invariant. The higher levels obtained from the lower levels are coarser but semantically more informative.

5.7.1 Hierarchical Encodings

We use 5 different hierarchical encodings, of roughly the same dimensionality in our experiments. Some of the descriptors required a pre-processing step of codebook generation using representative images from the training set. The representative images are obtained by subsampling the image sequences at regular intervals such that the poses in the consecutive images are substantially different. HMAX(C2 features) used 4 levels with codebooks computed using patch sizes $[4, 8, 12, 16]$. We randomly sampled patches from codebook images and used the 400 cluster centers for each patch size to compute $1600 = 4$ image descriptor.

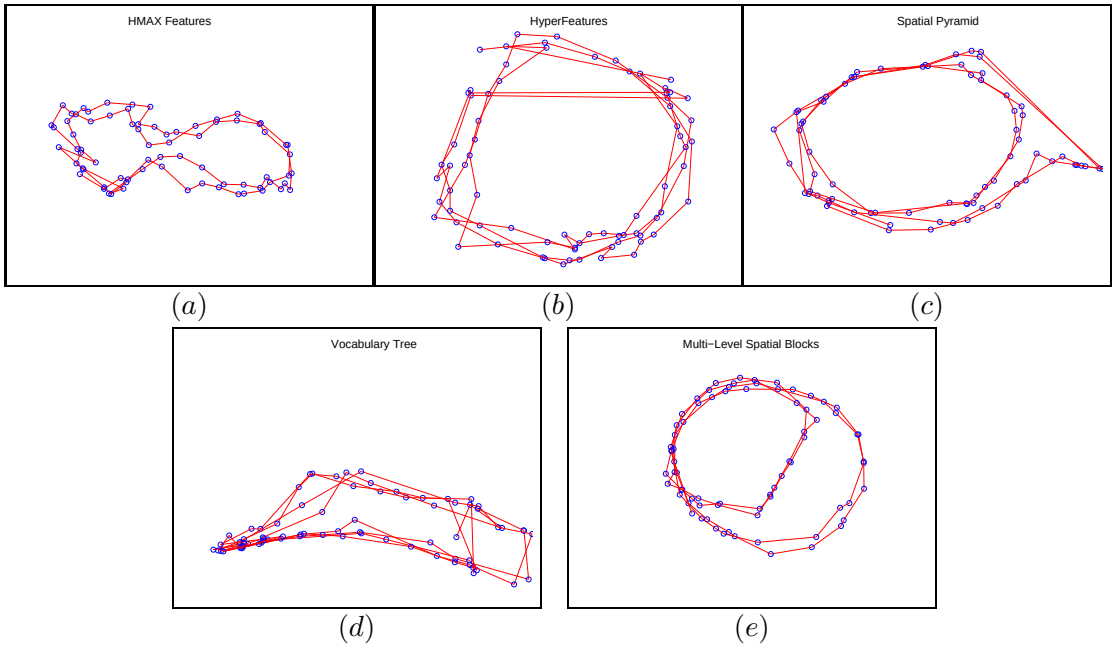


Figure 5.8: Isomap embeddings of the multi-level features for a walking sequence of a synthetic CG model with clean background and viewed from the side. Computation of shortest path neighborhood graph used 5 nearest neighbors, (a) C2 Features (HMAX) (b) HyperFeatures (c) Spatial Pyramid (d) Vocabulary Tree (e) Multi-Level Spatial Block. Clearly, whereas some of the of the descriptors (like HMAX and MSB) are able to disambiguate left leg front pose from the right leg front pose, the other descriptors show 2 full walking cycles as 4 half cycles.

Spatial Pyramid use 3 levels of spatial partitioning of the image window. Within each partition, the SIFT descriptors were computed over local patches computed at regularly spaced grid, with 6×6 pixel cells, 4×4 cells per block, 10 pixel patch overlap. We used 4 angular bins of unsigned gradient orientations from $0 - 180^o$ with bilinear interpolation. The descriptor is obtained as concatenation of the histograms computed for each patch and is a vector of size 1400.

Hyperfeatures also used 3 levels of abstraction where each level consisted of scale-space pyramid of 2, 4 and 6 ($[15/64/63/65/121/4]$) levels. Higher level of abstraction are constructed by accumulating and averaging histogram bins in a local neighborhood (9 neighboring patches) of the adjacent scale-space levels. Each pyramidal level is vector quantized using 800, 400 and 200 cluster centers obtained by clustering local patches at each levels of the scale-space. The descriptor is obtained as a feature vector of size 1400, by concatenating the histograms computed at each level.

Vocabulary tree uses 5 levels of histogram quantization with a branching factor of 4. The

local patch descriptors are obtained as SIFT descriptors, computed over a regularly spaced grid, with 4×4 blocks and 4×4 pixels per cell. The descriptor is obtained as a feature vector of size 1365 by concatenating histograms computed at the nodes of the tree.

Multilevel Spatial Blocks(MSB) is computed as a concatenation of SIFT descriptors computed over multiple levels of grid with varying number of blocks, where each block is composed of 4×4 cells of varying size. We use 3 levels with cell size as 12×12, 24×24 and 48×48 pixels.

Fig. 5.8 shows the Isomap embeddings of the 5 image hierarchical descriptors, for a walking sequence of a synthetic CG model with clean background when viewed from the side. Low dimensional embeddings provide a convenient way to visualize high dimensional features. The sequence consist of 2 full walking cycles. We used 5 Nearest Neighbors for computing the neighborhood graph. From the embeddings, it is clear HMAX features can easily discriminate the left/right leg front ambiguity whereas for Hyperfeatures and Vocabulary tree, it is difficult to differentiate one half walking cycle from the other. For multi-level spatial block and spatial pyramid, the difference is evident but not substantial.

We use hierarchical descriptors as inputs to our discriminative learning framework that uses metric learning for noise suppression and semi-supervised learning to train efficient models for 3D human pose inference. The 3D human pose is encoded as high dimensional, 3D local joint angles of the articulated skeleton. We use Bayesian Mixture of Experts (BME) in our framework to learn the multi-valued mappings from the descriptors encoding 2D image observations to the 3D joints space.

5.7.2 Metric Learning and Correlation Analysis

Before the image features are used to train mixture model, we ensure that they are not unnecessarily influenced by the noise due to background clutter. We use RCA and CCA to learn a subspace in which different poses with different background are close to each other. For RCA, we estimate the Mahalanobis distance on a number of equivalence sets of image descriptors extracted from the group images with same pose but different background. For this we take 750 different poses and render them on various realistic backgrounds (henceforth referred to as quasi-real). This step does not require 3D ground truth pose and estimates a common subspace

in which points belonging to same set(called *Chunklet*) are close to each to other. The subspace is obtained as most relevant orthogonal basis vectors. We use only first most relevant vectors in order to suppress the noise. Hence this technique also reduces the dimensions of the input feature. Fig. 5.9 shows the Isomap embeddings of the Multi-Level Spatial Block descriptors before and after applying RCA.

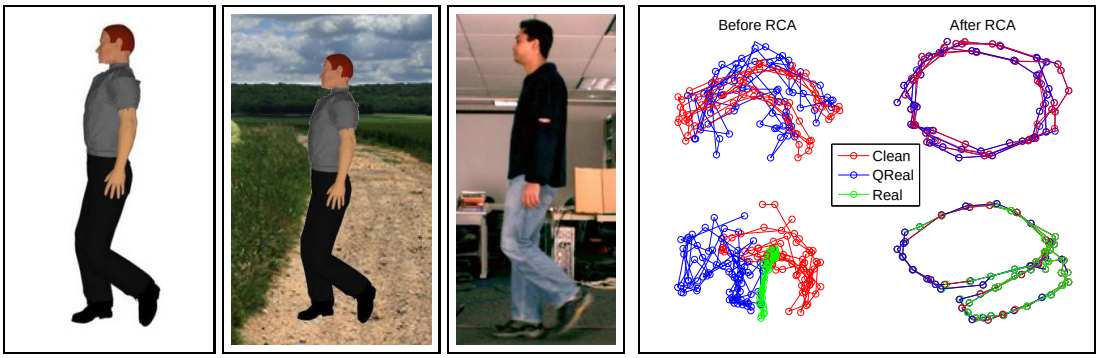


Figure 5.9: In this figure we compare the effect of RCA on the ISOMAP embeddings of hiearchical feature (Multilevel Spatial Blocks) of a walking sequence viewed from the side. We use 3 different image sequences viewed from the same viewpoint, with different degrees of realism - synthetic model with clean background(*Clean*), synthetic model with realistic background(*QReal*) and real sequence(*Real*) of a human subject. We manually aligned the 3D poses by dropping intermediate frames in order to compensate for different walking speeds. On the right we show the embeddings of the sequences before and after metric learning with RCA. The inclusion of pairwise clean and real images of similar poses as chunklets in RCA significantly improves the descriptor invariance to clutter. This *does not* introduces walking half cycle ambiguities, the bottom-rights shows the 2d projection of a somewhat twisted (but not self-intersecting) 3d loop.

For CCA we estimate two different subspaces for a pair of inputs in which the correlation between the two inputs is maximized. We obtain the pair of images in the same fashion as for RCA. Both RCA and CCA require some form of regularization to improve the generalization of the estimated subspace. In each case, regularization with a scale identity matrix usually helps performance. The behavior of CCA is illustrated in fig. 5.10. The figure compares how the canonical correlation varies with the regularization coefficient γ in the equation (5.8) and (5.9), for the training data set. The 45^o orientation denotes higher correlation. The eigenvectors with small eigenvalues λ^2 are least correlated (shown as the projections on last few canonical correlation basis vectors in fig. 5.10(c)). Also notice the effect of regularization on the spread of the projections of the training data on the learned subspace. The subspace obtained from regularized CCA are able to generalize more to the test data. Fig. 5.10*(d)* and *(e)* compares the

projection of the test data on the subspaces obtained from CCA using different regularization coefficient. Notice how the similarity in the correlation of the training and testing data due to larger regularization coefficient enables learning of more robust models. After metric learning,

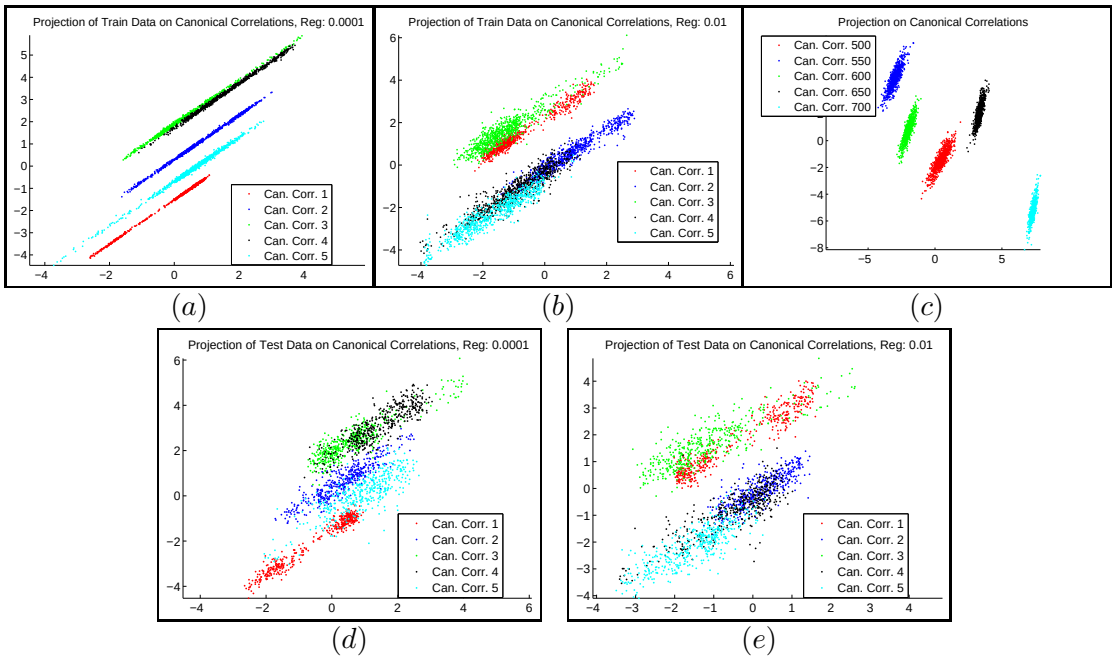


Figure 5.10: Projection of the training and test set on different canonical correlations. *(a)* and *(b)* show the top most correlated components, differently colored (these correspond to the largest eigenvalues), for two levels of regularization - we plot pairs of vector components, hence good correlations (*i.e.* similar values) are achieved when their slope is close to 45^o. *(c)* shows un-correlated directions corresponding to the low eigenvalues – notice the deviation from 45^o. *(d)* and *(e)* compares the coefficients of the test data corresponding to the largest eigenvalues for the two levels of regularization. Notice the similarity between the test and training data of the correlated components for subspace obtained with higher correlation coefficient.

the dimensionality of the image encoding is changed to (this was used for training multi-valued predictors for the experiments with each of the corresponding features): HMAX – 1174, Hyperfeatures – 1073, Spatial Pyramid – 1076, Vocabulary Tree – 1059, Multilevel Spatial Blocks – 1048.

Evaluation on the Dataset: For quantitative experiments we use our own database consisting of $3 \times 3247 = 9741$ quasi-real images, generated from a computer graphics(CG) human model of standard anthropometry and realistically rendered on different image backgrounds. We obtained 3247 different 3d poses from the CMU motion capture database [1] and these

were rendered from different viewpoints to create 3 sets of datasets - *Clean*, *Clutter1* and *Clutter2*. *Clean* dataset is obtained by rendering the synthetic CG model with clean background. *Clutter1* is obtained by rendering the synthetic model with background images that are used for training the model whereas *Clutter2* refers to images with unseen background, although in both the training and testing images, the rendered CG model is randomly placed on the image so that there is very little chance of generating a test data exactly resembling the training data. The dataset is composed of a variety of motion sequences including walking - viewed frontally and laterally, bending-pickup, running, dancing, conversation and a pantomime. We collect three test sets of 150 poses for each of the five motion classes. The motions executed by different subject are not in the training set. In all cases, a 320x240 bounding box of the model and the background is obtained, possibly using rescaling. There is significant variability and lack of centering in this dataset because certain poses are vertically (and horizontally) more symmetric than others (*e.g*. compare a person who picks an object with one who is standing, or pointing an arm in one direction).

We train BME model (a conditional Bayesian mixture of experts) with 5 experts on the entire dataset. Our settings is arguably more complex and varied compared to experiments demonstrated by activity-oriented models [9, 164]. The BME model uses linear experts with sparsity priors in order to generalize better. The sparsity of the experts lied in the range $15\% - 45\%$. The greedy feature selection method based on *Automatic Relevance Determination*(ARD) complement the relevant feature selection done by RCA / CCA. For the quantitative experiments, the 56d human joint angles were reduced to 8d using PCA. Although this introduces some error but it's fast and mapping to the joint angle ambient space is exact. Typically, with 8d PCA subspace, the reconstruction error by back-projecting the points from the subspace to original space is low ($\approx 2^o$). Cumulative results from our tests on the quasi-real databases, on clean background and *Clutter2* are shown in fig. 5.11. In the plots we provide the average joint angle error rates for the 5 experts used in BME, in order to factor out the errors due to inaccurate gate function. In general, performance on *Clutter1* is worse than clean dataset and better than on *Clutter2*, but the problem is arguably simpler. The first two rows give cumulative prediction error per joint angle over all the motion sequences, for different multilevel encodings and the two metric learning methods. In our experiments HMAX works best, followed closely by the

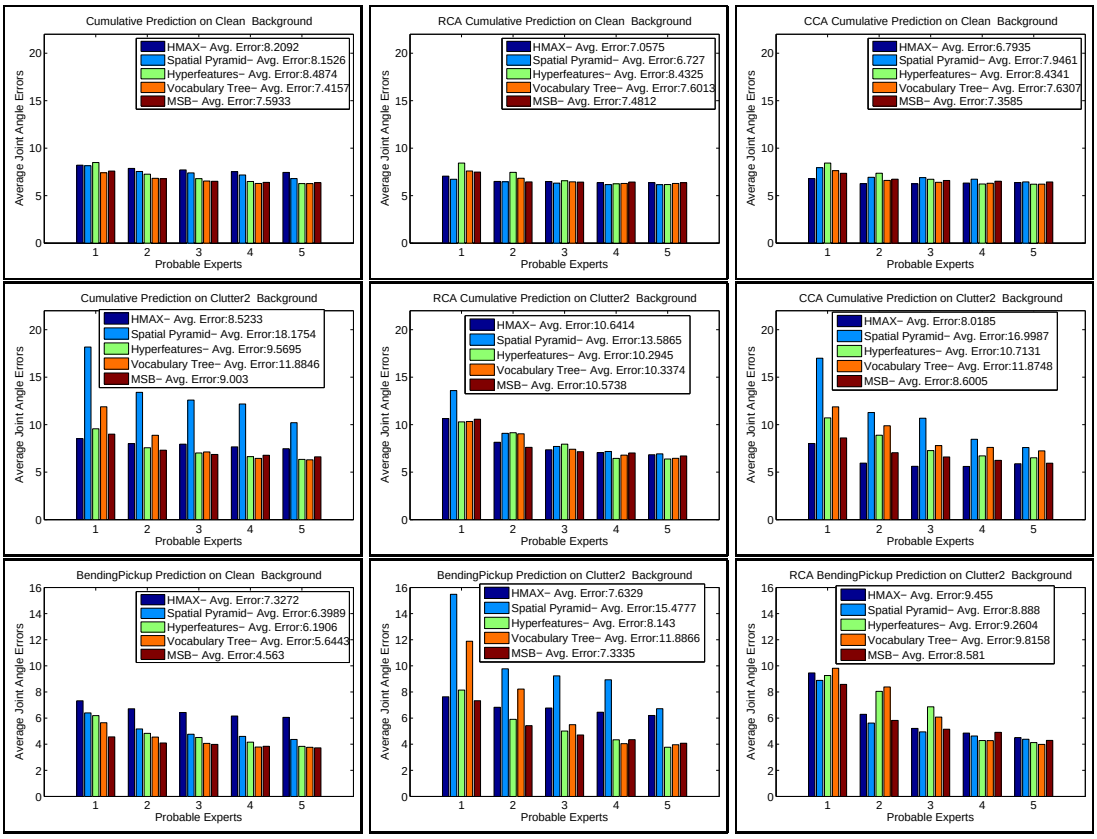


Figure 5.11: *(Top row)* and *(Middle row)* Cumulative 3D pose estimation accuracy for 5 different hierarchical descriptors: HMAX, Hyperfeatures, Spatial Pyramid, Vocabulary Tree, Multilevel Spatial Blocks (MSB). We evaluated 5 different motion sequences : Bending pickup, walking, running, dancing and pantomime. Here we plot the root mean square error of the predicted joint angles using K most probable experts. The K^{th} bar was obtained by selecting the expert prediction closest to the ground truth among the ones predicted by the most probable K experts. We also show the effect of the 2 metric learning techniques (RCA and CCA) on the prediction accuracy. We show the plots of *clean* and *clutter2* datasets. *(Bottom row)* Joint angle prediction errors for the bending and picking motion sequence. A single conditional BME model was trained on the entire dataset.

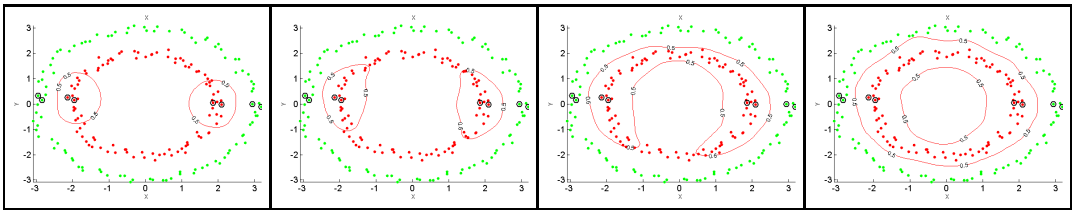
Multi-level Spatial Block and Hyperfeatures. The effect of metric learning on these features was marginal. One reason to explain this could be that these descriptors are robustly encoded to be invariant to noise and perturbations in the image. Also, some degree of noise is removed by feature selection mechanism in learning the expert regressions using *Automatic Relevance Determination* mechanism. Rather, we observed slight drop in performance due to metric learning for these features. For the features based on spatially localized histograms (Spatial Pyramid) and globally computed histograms, the change in the background tends to influence the descriptors more. In spatial pyramid, background regions in the image bounding box get encoded as an image partition whereas in Vocabulary trees, the bottom-most nodes of the tree may be contain smaller clusters due to the noise in the background. Relevant Component Analysis(RCA) improves these descriptors substantially, as shown in the fig. 5.11. The last column in first two rows shows the results using Canonical Correlation Analysis(CCA). The bottom row in fig. 5.11 shows error rates for the bending and picking up test sequence. Note, in our experiments single global model was trained on the entire dataset (and not on separate activities). In general the error rates are higher for *Clutter1* and *Clutter2* sequences. The use of RCA improves the prediction accuracy for the bending pickup sequence substantially. An alternative to RCA/CCA is to use problem dependent kernels, *e.g.* histogram intersections [103], with good resistance to noise and image mismatches. In principle, our kernel-based multi-valued predictors can use histogram kernels to further improve the stability of the descriptors.

5.7.3 Manifold Regularization

We evaluate semi-supervised learning based on manifold regularization on both synthetic and real dataset. Fig. 5.12 shows the improved binary classification decision boundary obtained using manifold regularization, for a 2 circle toy example. The dataset has only 8 labeled points, shown as black circles. Notice how the decision boundary of the classifier automatically adjusts along the geometry of the input distribution, as the intrinsic geometry regularization coefficient γ_I is increased.

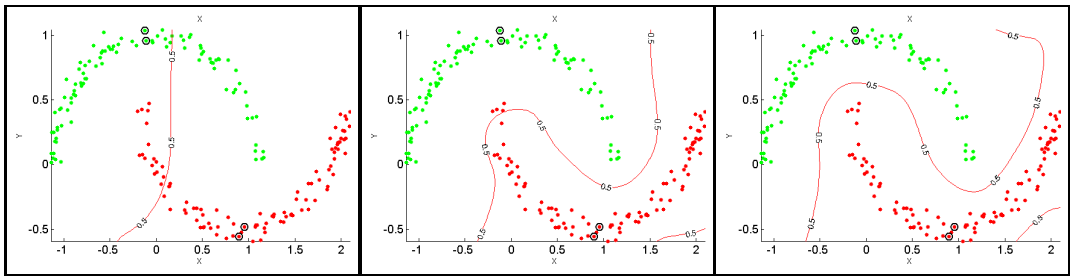
Fig. 5.13 shows the two moons dataset with 200 examples and only 2 labeled points each belonging to different class. Also shown is the decision surface obtained by simultaneously estimating the intrinsic geometry regularization coefficient γ_I and the sparsity of the Bayesian

classifier. The degree of sparsity achieved depends on the initial value of the manifold regularization coefficient γ_I. The sparsity denote the percentage number of basis vectors used in the learned interpolant.



$\gamma_I = 0.00001$, $\gamma_I = 0.001$ $\gamma_I = 0.01$ $\gamma_I = 0.1$

Figure 5.12: Manifold regularization on the dataset containing 200 points sampled from 2 concentric circles. Dataset contained only 8 labeled points - shown as circled data points. In the figure we demonstrate the automatic adjustment of the decision boundary as the intrinsic geometry regularization coefficient is increased from 0.00001 to 0.1



$\gamma_I = 0.0001$, $Sparsity = 18.91\%$ $\gamma_I = 0.001$, $Sparsity = 23.38\%$ $\gamma_I = 0.01$, $Sparsity = 26.37\%$

Figure 5.13: Effect of manifold regularization on the classification boundary for 2-moons dataset. We use sparse bayesian learning with manifold regularization to train the classifier. The dataset contained 4 labeled examples, shown as circles in the figure. The level of sparsity achieved depends on the initial value of the manifold regularization coefficient.

We evaluate the effect of manifold regularization on mixture of experts learning model for a synthetic datasets with one-to-many mapping. The dataset is obtained by sampling points from the inverse of the function $\mathbf{r} = \mathbf{x} + 0.3sin(2\pi\mathbf{x}) + \epsilon$, where ϵ is a zero mean Gaussian with standard deviation 0.05. In effect, manifold regularization for the BME model improves the estimation of the expert regressors and the classification decision boundary of the gating function using the intrinsic geometry of the input data. In the plots shown in fig. 5.14 and fig. 5.15, we illustrate 2 scenarios - adding labeled data points to the training set and adding unlabeled data points to the training set. Fig. 5.14*(top row)*shows the effect of adding labeled data points on the 3 kernel expert regressors. Circled points denote the labeled data points. Notice how

the regressors fit to the dataset as the number of labeled points are increased. Fig. 5.14*(bottom row)* shows the effect of manifold regularization on the gating distribution, where the abscicca denotes the probability weights. Fig. 5.15 shows the effect of unlabeled exemplars for the training the mixture of experts model using manifold regularization. Fig. 5.15*(top row)* shows the improved kernel experts as the number of unlabeled data points increased from 0 to 196. Fig. 5.15*(bottom row)* shows the same for the gating distribution. In the plots we keep ambient regularization and intrinsic geometry regularization coefficient fixed while varying only the number of labeled/unlabeled exemplars for learning the experts and the gating distributions.

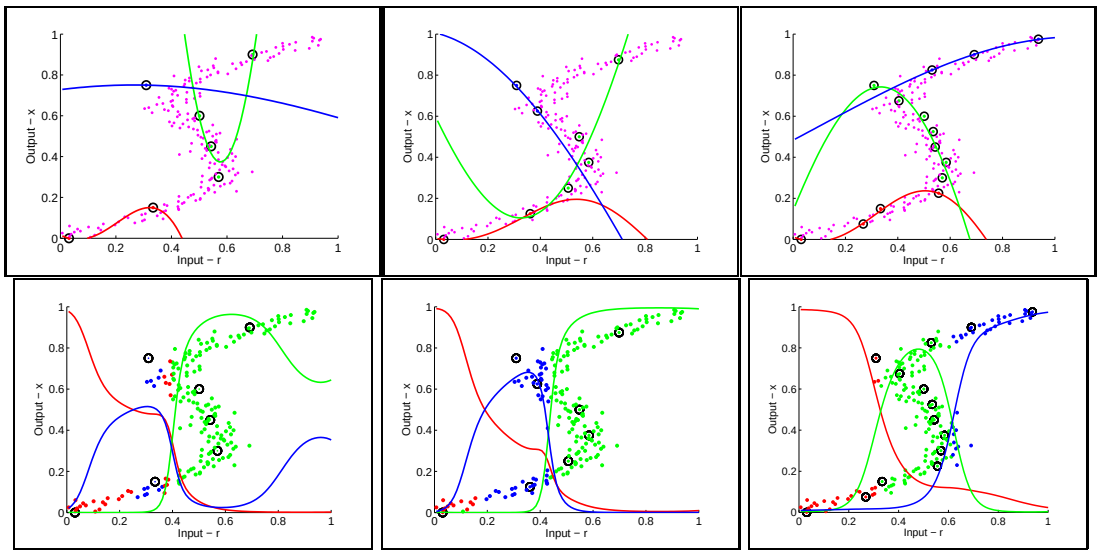


Figure 5.14: Training Bayesian mixture of experts using manifold regularization, with varying number of labeled points (shown as circles) and fixed number of unlabeled points(shown as colored points), *(Top row)* The expert regression functions, *(Bottom row)* the gating distributions of the mixture of experts model.

We also ran experiments using the manifold regularization framework fig. 5.16, where we trained several BME models(with 5 linear regression experts) on a small dataset of a synthetic CG model in cluttered background. We used 30 observations and progressively added $0 - 270$ unlabeled data points to improve the learning. In the figure we show average joint angle prediction error. The BME model was trained on the two sequences of running and bending, with observed images encoded using Multilevel Spatial Block(MSB) descriptors. Clearly, the addition of unlabeled data improves performance. Fig. 5.16*(a)* shows the improvement in prediction due to manifold regularization as the number of unlabeled data points are increased.

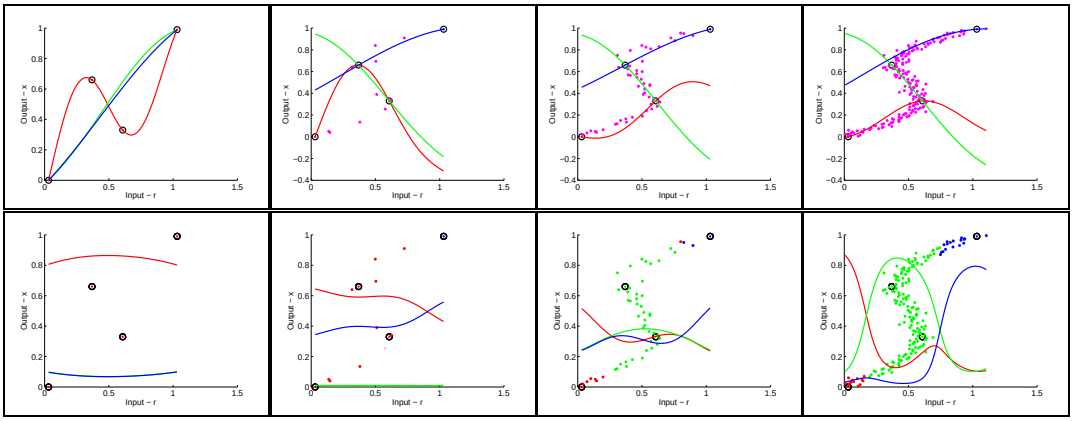
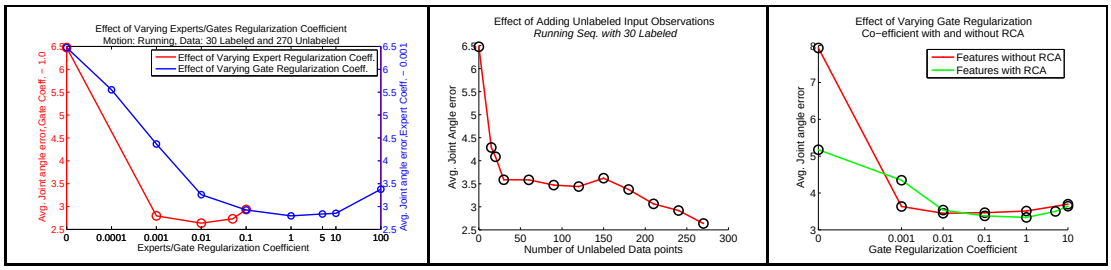


Figure 5.15: Training Bayesian mixture of experts using manifold regularization, with varying number of unlabeled points and fixed number of labeled points(shown as 4 circled points). *(Top row)* the expert regression functions, *(Bottom row)* the gating distributions of the mixture of experts model. Notice in (c) that only a few unlabeled data points are enough to train correct regression and gating functions. Adding more unlabeled points improves the accuracy (d) of the BME model.

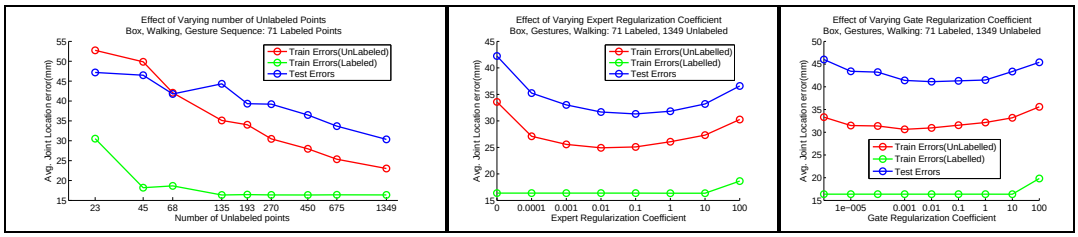
Fig. 5.16*(b)* shows the effect of varying the gate and expert regularization coefficient keeping the other coefficient constant. We use 270 unlabeled samples and 30 labeled samples from the running sequence. Note that large coefficients train oversmooth expert and gating functions and degrades the performance. Fig. 5.16*(c)* shows the effect of varying gate regularization coefficient on the prediction accuracy of models with and without distance metric learning(RCA).

HumanEva Dataset: We have also run experiments using the manifold regularization framework fig. 5.17, where we trained several Mixture of Experts models on HumanEva dataset for a subject performing 3 activities - Boxing, Walking and Gestures. The Mixture of Experts model consisted of 4 experts with kernel functions. The training dataset consisted of 71 labeled exemplars and 1349 unlabeled image observations. The test set contained 1351 data points. In all the experiments we fixed the ambient regularization coefficient to 0.0001. The observed images were encoded using Multilevel Spatial Block(MSB) with 3 levels of hierarchy. Fig. 5.16 shows the average joint location error plots for various models trained using semi-supervised learning. In the figure we show the error plots on the labeled training exemplars, unlabeled training data points and test data points. Fig. 5.17*(a)* shows the improvement in prediction accuracy as the unlabeled data points are progressively added to the training set. As we increase the number



(a) (b) (c)

Figure 5.16: Improved prediction accuracy using semi-supervised learning of the BME model(with 5 linear regression experts), *(a)* shows the effect of adding unlabeled points for learning the BME model on a running sequence. The models are trained on 30 labeled (2D descriptors, 3D pose) pairs, with number of unlabeled exemplars progressively increased from 0 to 270, *(b)* shows the improvement as the intrinsic geometry regularization coefficient is varied for the expert and the gating function. The model was trained on 30 labeled and 270 unlabeled exemplars of the running sequence. The accuracy improves as the coefficient is increased till the plot levels off, following which the performance starts degrading due to over smooth functions. In the fig. *(c)* we compare the improvements achieved due to manifold regularization, when the BME model is trained on the descriptors with and without relevance component analysis(RCA). Here we train on bending and pickup sequence, and vary the intrinsic geometry regularization coefficient of the gating function. Notice that larger improvements is achieved with the descriptors with distance metric learning(RCA).



(a) (b) (c)

Figure 5.17: Improved prediction accuracy using semi-supervised learning of the BME model(with 4 kernel regression experts and non-linear gating distribution),*(a)* shows the effect of adding unlabeled points for learning the BME model on 3 HumanEva[155] sequences - Boxing, Walking and Gestures, performed by the same subject. The models are trained on 138 labeled (2D descriptors, 3D pose) pairs, with number of unlabeled exemplars progressively increasing from 23 to 1349, *(b)* shows the improvement as the intrinsic geometry regularization coefficient is varied for the expert regression models. The gate manifold regularization coefficient was fixed to 0.001 *(c)* shows the same as the regularization coefficient is varied for the gating functions with expert regularization coefficient set to 0.001. The model was trained on 71 labeled and 1349 unlabeled exemplars of the Gesture sequence. The accuracy improves as the coefficient is increased till the plot levels off, following which the performance starts degrading due to underfitting (oversmooth models). We show the joint location prediction accuracy on the training examples (both labeled and unlabeled) and the test examples.

of unlabeled data from 23 to 1349, the average joint location error for all the three evaluation datasets decreases. Fig. 5.17*(b)* and *(c)* shows the effect of varying the gate and expert manifold regularization coefficients while keeping the other ambient regularization coefficients constant to 0.0001. We use 71 labeled exemplars and 1349 unlabeled examples both sampled from the 3 activity sequences of boxing, walking and gestures. Notice that in both the set of learned models, the prediction accuracy on the test and unlabeled data initially decreases and then starts increasing. Large coefficients cause the models to underfit the data and degrades its performance. An interesting observation is that prediction accuracy on the labeled training set remains unchanged initially as we increase the influence of manifold regularization term. This indicates that even though the model was trained on extremely sparse training set of exemplars, there is no significant overfitting in the model due to the ambient regularization term in the object cost function. Manifold regularization thus affects only the ability of the model to generalize.

For qualitative experiments we use images from a movie (Run Lola Run) and the INRIA pedestrian database [53] to evaluate the discriminative pose inference framework. In fig. 5.18 we show the human pose prediction based on real images. These are all automatic 3d reconstructions of fast moving humans in non-instrumented, complex environments. We detect humans in the image sequence using a person detector based on SVM classifier[53]. The detector generates the bounding boxes around the human targets. For training, we use 2000 walking and running labeled poses (quasi-real data of our model placed on real backgrounds, rendered from 8 different viewpoints) with an additional 1000 unlabeled (real) images of humans running and walking in cluttered scenes. The solutions are not entirely accurate in a classical alignment sense but the 3d reconstruction have reasonable perceptual accuracy.

5.8 Discussion

In this chapter we have proposed a discriminative framework for human pose inference that integrates various techniques of robust image encoding, distance metric learning and semi-supervised learning. We propose solution for three key challenges faced by the current discriminative frameworks, *(a)* balancing same pose variations with discriminative power using

Figure 5.18: 3d Pose reconstruction results obtained on images from the movie 'Run Lola Run' (leftmost 5 images) and the INRIA pedestrian dataset (rightmost 3 images) [53]. *(Top row)* shows the observed test images, *(Bottom row)* shows 3d reconstructions using our framework

robust hierarchical encodings, *(b)* reducing noise due to background clutter in the region of interest using distance metric learning and canonical correlation analysis *(c)* improve training of the framework by semi-supervised learning on the dataset augmented with unlabeled exemplars. Hierarchical encodings represent the image observation at multiple levels of the trade-off between discriminative power and invariance to aberrations in the image. Distance metric learning and correlation analysis enable computation of visual descriptors that are invariant to changes in the background. Finally, we use semi-supervised learning based on manifold regularization to train models that are smooth not only in the ambient space but also along the intrinsic geometry of the image descriptor space. This allows us to learn improved discriminative models by incorporating additional information from unlabeled exemplars. We provide quantitative results on both low dimensional synthetic dataset and high dimensional human pose inference to support the proposed framework. Empirically, we also observe that a combined system improves the quality of 3d human pose prediction in images and video.

Chapter 6

Sparse Spectral Latent Variable Models

6.1 Introduction

In this chapter, we introduce a generic algorithm to combine parametric latent variable models and spectral embedding methods in a probabilistically consistent framework referred to as *Sparse Spectral Latent Variable Models*. The contents of this chapter are based on the work *Spectral Latent Variable Models for Perceptual Inference*, Atul Kanaujia, Cristian Sminchisescu, Dimitris N. Metaxas, *International Conference on Computer Vision 2007*

In any machine learning task, it is assumed apriori that the features, encoding the visual stimuli, are perceptually meaningful so that they can be used to train a classifier or a regression model. Extracting these meaningful features from raw sensory inputs such as image pixels is a difficult task and involves a combination of evolution, development and learning process. For example, a 128×64 pixel image of a person walking (or running) can be thought of as point in a 8192 dimensional observation space, however the perceptually meaningful structure of these visual stimuli is of much lower dimensionality, only 1 - encoding the phase of the walking(or running) cycle. Another example is the shape contours of the facial features when observed from different viewpoints. In this case, these points, that correspond to the shape contours, lie on a two-dimensional manifold and is parameterized by viewing angle.

The goal in these problems is therefore, to discover the low dimensional yet perceptually meaningful structure in these observations, and use them to learn efficient models for recognition, classification and a variety of other imagery task. The manifold structure is typically much low dimensional compared to high dimensional observations, and is useful for analysis and visualization of large volumes of data.

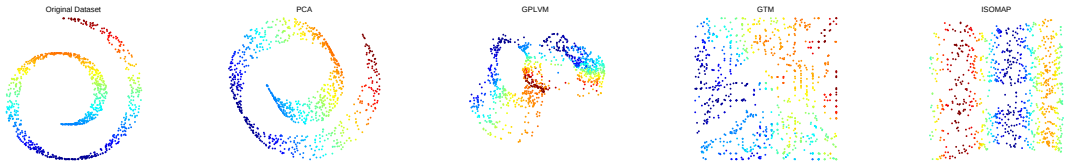
A key difficulty of dimensionality reduction techniques is that the geometry of these observations usually exhibit non-linear structure. Fig. 6.1 illustrates this using a toy dataset of Swiss roll. The color coded data points are embedded on a non-linear 2D planar structure surrounded by a 3D ambient space. Linear techniques like Principal Component Analysis(PCA) and Factor Analysis(FA) are unable to unfold such a highly non-linear structure fig. 6.1(b). A variety of non-linear dimensionality reduction techniques have been proposed in the literature. More prominent methods include Self Organizing Map(SOM)[98], the Generative Topographic Mapping(GTM)[32] and Gaussian Process Latent Variable Model[100]. GTM tries to fit a predefined grid based topology in the latent space to the observed data points using greedy optimization techniques. The top down structure of the learning algorithm makes it difficult to unfold the coarse, non-linear structure of the observed dataset, as the optimizers may get stuck in severe local minima. Fig. 6.1(d) shows the fitting obtained from the GTM. Notice that the associations between the observed points and the learned latent co-ordinates are entangled.

GPLVM is also based on learning non-linear structures of the manifolds by fitting a non-linear map from the low-dimensional, latent space points to the points in the ambient space. This is achieved by using different covariance functions, in the form of Gaussian processes, that enables modeling of non-linear structure of the manifold. This cannot be optimized is a closed form and therefore requires a gradient based optimization of the objective function that effectively minimizes the reconstruction error of the points in the observed space obtained by mapping the points in the latent space. In order to unfold the non-linear embedded structures in the high dimensional observed space, we need learning algorithms that can preserve the topology of the observed data points. The GPLVM technique does not enforce this criteria of preserving the local and global geometric structure and faces setback due to the tendency of the optimizer to get stuck at the local minima. Fig. 6.1(c) shows the inaccurate embeddings obtained from GPLVM with back constraints[100, 101]. One promising class of methods for learning perceptual representation is using spectral methods[143, 176] that learns the globally optimal solution in a bottom-up fashion. These class of methods - referred to as spectral methods - first learn the topological structure of the manifold in the observed space and then learns the metric map that respects this topology. Examples of spectral methods are Isometric feature mapping(Isomap)[176], Locally linear embeddings(LLE)[143], Local Tangent Space

Alignment(LTSA)[201], Laplacian Eigenmaps[22] and Hessian Eigenmaps[61]. The topology is approximated as a connected graph between pair of sampled points in the observed space under the key assumption that the distance between the points in the observation space is an accurate measure of the distance in the embedded manifold *only locally*. Whereas *global spectral methods* like Isomap represents the manifold metric between every pair of points (both local or global) as shortest paths in the connected graph, neighborhood preserving, *local spectral methods* like LLE and LTSA ignore the widely separated points and use only locally linear fitting to learn the global manifold structure. The connected graph clearly respects the topology far better than the regular grid based graph used with SOM and GTM. The low dimensional embeddings are then obtained by orthogonalizing the manifold metric space by solving an eigen-decomposition problem.

Although effective in learning the low-dimensional representations of a complex manifold fig. 6.1(e), spectral methods lack a clear probabilistic framework, with no straightforward method to project out-of-sample data points onto the embedded space. In order to do so, it requires re-estimation of the connected graph and therefore re-computation of the embedded space. Approximate methods for extending spectral methods to handle unseen data points do exist in literature [26]. These methods estimate kernel functions that approximates the connected graph matrix and effectively solve an eigen-decomposition on it to learn the embeddings as eigenfunction (as opposed to eigenvectors). On the other hand there exist a variety of methods for non-linear latent variable models, including mixture of PCA[179], mixture of Factor Analyzers and GTM that are probabilistic but are unable to preserve the geometric properties of the data in the learned latent space.

In this chapter, we introduce a rather general framework to combine parametric latent variable models and spectral embedding methods in a probabilistically consistent fashion. The method preserves the geometric characteristics of the data in the ambient space and can be efficiently used not only to map unseen inputs to the latent space but also to reconstruct data points in the original space for an unseen point in the latent space. The method is trained on top of embeddings obtained from the spectral embedding methods and is able to support complex visual inferencing tasks such as human 3D pose reconstruction and non-linear active shape models. We refer the proposed framework as Sparse Spectral Latent Variable Model (SLVM).



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

Figure 6.1: (a) The original dataset in 3D consisting of 1000 points sampled from a non-linear manifold - Swiss Roll (b) Learned embedding in 2D using PCA, (c) Gaussian Process Latent Variable Model with Back Constraints, (d) Generative Topographic Map (e) Isometric Feature Mapping(ISOMAP)

In principle, any of the embedding methods can be used to initialize the learning. We chose spectral embedding methods for obtaining the prior distribution as these provide a globally optimal solution to the problem of discovering non-linear degrees of freedom underlying the complex natural observations.

6.2 Latent Variable Models

In this section we provide the details of the proposed framework for learning low-dimensional representations using Spectral Latent Variable Models. For a given set of vectors in the observation space $\mathcal{Y} = \{\mathbf{y}_i | i = 1 \cdots N\}$, we want to obtain a low dimensional representation in the latent space $\mathcal{X} = \{\mathbf{x}_i | i = 1 \cdots N\}$. We refer the observation space as the ambient space that surrounds the low-dimensional latent space.

In the following formulation, we denote the latent space points as a d-dimensional vector $\mathbf{x}$ and the ambient data points as vectors $\mathbf{y}$ with dimensionality D. Typically D is much greater than the latent space dimension d.

The goal of latent variable model is to estimate the marginal distribution $p(\mathbf{y})$ in the ambient space, in terms of latent variables $\mathbf{x}$. This is accomplished using a generative form by modeling the joint distribution over ambient and latent variables as $p(\mathbf{y}, \mathbf{x}) = p(\mathbf{y}|\mathbf{x})p(\mathbf{x})$. We define a mapping function from the latent space to the ambient space f($\mathbf{x}$; $\mathbf{W}$). Assuming that the observed data points $\mathbf{y}$ are obtained from this mapping with Gaussian noise (zero mean), we model this relationship as a conditional distribution $p(\mathbf{y}|\mathbf{x}, \mathbf{W})$. The ambient marginal can then

be obtained by marginalizing out the latent variables using the integral:

$$p(\mathbf{y}|\mathbf{W}) = \int p(\mathbf{y}|\mathbf{x}, \mathbf{W})p(\mathbf{x})d\mathbf{x} \tag{6.1}$$

Here, the $\mathbf{W}$ define the parametric mapping function f($\mathbf{x}$; $\mathbf{W}$). In principal, the mapping function can be non-parametric(e.g. Gaussian process as in GP-LVM[100].

In practice, it is convenient to define a likelihood function over the ambient data points and estimate the latent points by maximizing it [32, 179]:

$$\mathcal{L}(\mathbf{W}) = \log \prod_{i=1}^{N} p(\mathbf{y}_i|\mathbf{W}) = \log \prod_{i=1}^{N} \int p(\mathbf{y}_i|\mathbf{x}, \mathbf{W})p(\mathbf{x})d\mathbf{x} \tag{6.2}$$

The mapping function can be linear with a Gaussian noise model:

$$\mathbf{y} = \mathbf{W}\mathbf{x} + \mathcal{N}(0, \sigma) \tag{6.3}$$

For a prior distribution as zero mean Gaussian distribution with a unit covariance, maximizing the log likelihood eqn.6.2, leads to models like probabilistic PCA[179](or Factor Analysis depending upon the Gaussian noise model for p($\mathbf{y}|\mathbf{x}, \mathbf{W}$)) where the matrix $\mathbf{W}$ spans the principal sub-space of the data. However as mentioned earlier, linear models like PCA are unable to unfold complex non-linear geometries in the ambient space. Hence it is more appropriate to learn the mapping (6.3) using a non-linear regression.

One of the desirable features of the low-dimensional embeddings is that it they should respect the topology of the points in the original space. As likelihood function has no specific provision to enforce this, it is not sufficient to optimize the likelihood function 6.2 in order to estimate embeddings that preserve the geometric structure in the ambient space. This is primarily due to the following:

- The objective function in its current form has no provision to constrain the optimization search in order to preserve the topological structure of the ambient data points in the latent space.

- The objective function may have complex response surface and optimizing it may cause the framework to get stuck in local optima, yielding sub-optimal results.

Non-linear latent variable models like mixture of factor analyzers or PPCA[179] can although model complicated non-linear structures, but do not provide global latent co-ordinate systems

or latent spaces that preserve local or global geometric properties of the observed data. Whereas regular grid based frameworks, like Generative Topographic Map(GTM) fail to unfold many convoluted manifolds as the embedding grid is oblivious of the topological structure of the ambient data. The optimization algorithm has strong tendency for getting stuck at the local optima during the training. More recent methods like Gaussian Process Latent Variable Model (GPLVM)[100] also face similar challenges. GPLVM is a probabilistic non-linear latent variable model that uses non-parametric regression (Gaussian process) to map latent points to ambient space. The objective function with zero mean unit Gaussian regularizer in latent space is data independent and does not put constraint to explicitly preserve the geometric properties of ambient data. The lack of appropriate latent space prior in GPLVM makes it somewhat more prone to local minima and therefore yields sub-optimal embeddings (ref. fig. 6.1(d)). In the SLVM framework, we overcome the above limitations by using strong latent space prior obtained from non-linear, spectral embedding methods like Isomap, LLE, Hessian Eigenmaps or Laplacian Eigenmaps. We use the latent prior to learn a probabilistic generative model for estimating the conditional map from the latent space to observed space. The backward map from the ambient space to the latent space is obtained by probabilistic inversion using Bayes' rule. In the next section, we provide the details of Sparse SLVM formulation.

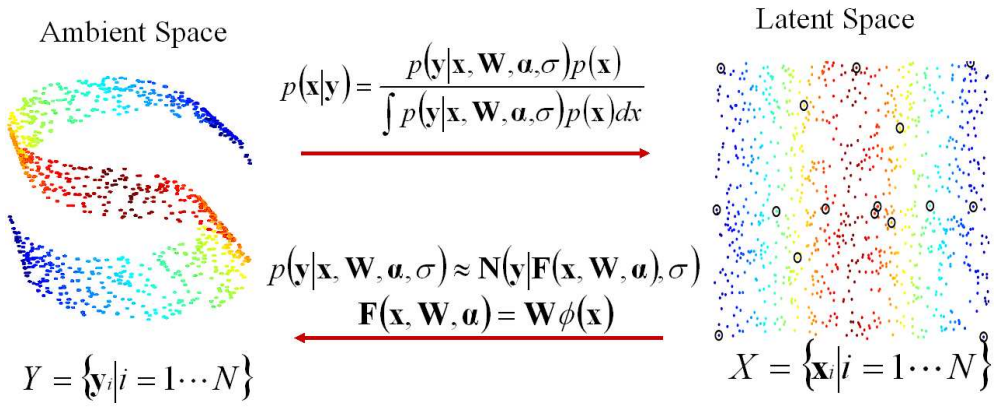


Figure 6.2: Overview of the sparse Spectral Latent Variable Model illustrated on an original dataset consisting of 1000 points sampled from a planar S-shaped 3D manifold (referred to as S-curve). The forward mapping from the latent space to ambient space is a non-linear regression mapping while the backward mapping is obtained by inversion of the forward conditional map using Bayes' rule. Details are discussed in the section §6.2.1

6.2.1 Sparse Spectral Latent Variable Model

Fig. 6.2 illustrates the Sparse Spectral Latent Variable Model(SLVM). We work with two sets of vectors, $\mathcal{X}$ and $\mathcal{Y}$, of equal size N, in the two spaces referred to as latent(or perceptual) and ambient(observation space surrounding the latent space) respectively. The sets of vector are unordered but in correspondence. We model the joint distribution over latent and ambient variables using a constructive form: $p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x})p(\mathbf{y}|\mathbf{x})$. We select the latent space prior $p(\mathbf{x})$ to be a non-parametric kernel density estimate, with kernel K and covariance θ, centered at the embedded points $\mathbf{x}_i$, obtained from the spectral methods like Isomap, LLE and Laplacian Eigenmaps.

$$p(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N}\mathbf{K}_\theta(\mathbf{x}, \mathbf{x}_i) \tag{6.4}$$

In order to avoid complex integral for estimating the marginal((6.1)), the prior may as well be formulated as sum of delta functions centered at the embedded points $\mathbf{x}_i$.

$$p(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N}\delta(\mathbf{x} - \mathbf{x}_i) \tag{6.5}$$

Spectral methods use graph-based representations of the observed data, with nodes that represent observations and links that stand for neighborhood relations. The connected graph can be viewed as a discrete approximation of the sub-manifold directly sampled from the observed data. Different methods derive different matrices from the graph. Their spectral decompositions (the top or bottom eigenvectors) reveal the low-dimensional, latent structure of the data.

We use the marginal distribution in the latent space and a mapping function from latent space to the ambient space to construct joint probability distribution over the latent and ambient variables $p(\mathbf{x}, \mathbf{y})$. The vectors in the ambient space are related to the latent space via a non-linear map:

$$\mathbf{F}(\mathbf{y}, \mathbf{W}, \alpha) = \sum_{i=1}^{M}\mathbf{w}_i^T\mathbf{K}_\gamma(\mathbf{y}, \mathbf{y}_i) = \mathbf{W}^T\mathbf{K}_\gamma(\mathbf{y}) \tag{6.6}$$

In general, the kernels for the latent prior and the mapping function may be different. The weight matrix $\mathbf{W}$ has size $M \times D$, with the column vectors $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_N\}$ for each of the independent variate of the D-dimensional output vector.

$$p(\mathbf{y}|\mathbf{x}, \mathbf{W}, \boldsymbol{\alpha}, \sigma) \sim \mathcal{N}(\mathbf{y}|\mathbf{F}(\mathbf{x}, \mathbf{W}, \boldsymbol{\alpha}), \sigma) \tag{6.7}$$

We assume a radially symmetric Gaussian noise model for the multi-variate response variables $\mathbf{y}$. Notice, that nonlinear mapping uses M basis vectors and can be sampled directly from the training data points.

We train the mapping function in sparse bayesian learning (SBL) framework. SBL uses hierarchical priors on the parameters $\mathbf{W}$ of the mapping $\mathbf{F}$, that are governed by a set of hyperparameters. A hyperparameter is associated to each weight $\mathbf{w}_i$ column vector and is iteratively estimated using an optimization framework called *automatic relevance determination*[113, 124]. The learning essentially computes the MAP (maximum a posterior) estimates of the parameters that govern the distribution of weight priors used for regularization of the regression function (6.6).

$$p(\mathbf{W}|\boldsymbol{\alpha}) \sim \prod_{j=1}^{D}\prod_{k=1}^{N}\mathcal{N}(w_{jk}|0,\frac{1}{\alpha_k}) \tag{6.8}$$

$$p(\boldsymbol{\alpha}) = \prod_{i=1}^{N}\text{Gamma}(\alpha_i|a,b) \tag{6.9}$$

$$\text{Gamma}(\alpha|a,b) = \Gamma(a)^{-1}b^a\alpha^{a-1}e^{-b\alpha} \tag{6.10}$$

and $a = 10^{-2}$, $b = 10^{-4}$ chosen to give broad hyperpriors. Using broad hierarchical priors cause the posterior probability of the hyperparameters α to concentrate at very large values. The consequence is that the posterior probability of the associated weights are concentrated at zero. These weights are iteratively pruned and yields a sparse subset for relevant weights[180] that correspond to the basis vectors of the mapping. As discussed in earlier chapters, Sparse Bayesian learning generates compact regression models that have better generalization ability and are robust to overfitting even in the absence of sufficient training data.

The marginal distribution of the ambient data is obtained by integrating out the latent variables:

$$p(\mathbf{y}|\mathbf{W},\boldsymbol{\alpha},\sigma) = \int p(\mathbf{y}|\mathbf{x},\mathbf{W},\boldsymbol{\alpha},\sigma)p(\mathbf{x})\mathbf{dx} \tag{6.11}$$

For the prior based on kernel density, this integral is analytically intractable and can be approximately computed using Monte Carlo(MC) estimates of the prior distribution $p(\mathbf{y})$. The integral is straightforward for the prior based on delta functions 6.5 and is computed at the latent points $\mathbf{y}_i$ as:

$$p(\mathbf{y}|\mathbf{W},\boldsymbol{\alpha},\sigma) \sim \frac{1}{K}\sum_{i=1}^{K}p(\mathbf{y}|\mathbf{x}_i,\mathbf{W},\boldsymbol{\alpha},\sigma) \tag{6.12}$$

The conditional distribution representing the backward mapping of the observed data points to the latent space are obtained using the inverse Bayes' rule:

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{\int p(\mathbf{y}|\mathbf{x})p(\mathbf{x})} = \frac{p(\mathbf{y}|\mathbf{x}, \mathbf{W}, \boldsymbol{\alpha}, \sigma) \sum_{i=1}^{K} \mathbf{K}_\theta(\mathbf{x}, \mathbf{x}_i)}{\sum_{i=1}^{K} p(\mathbf{y}|\mathbf{x}_i, \mathbf{W}, \boldsymbol{\alpha}, \sigma)} \tag{6.13}$$

The conditional map can be used to project any out-of-sample ambient data point to the latent space. Using the learned conditional from the latent space to the ambient space, we can estimate the responsibilty of the latent data point i for the ambient data point j as $p(\mathbf{x}_i|\mathbf{y}_j, \mathbf{W}, \boldsymbol{\alpha}, \sigma)$. The latent point corresponding to any new ambient variable $\mathbf{y}$ can then be obtained either as the weighted mean over the responsibilities of all the Monte Carlo samples or the mode (better for multimodal distributions) in the latent space:

$$\mathrm{E}\{\mathbf{x}|\mathbf{y}_n, \mathbf{W}, \boldsymbol{\alpha}, \sigma\} = \int p(\mathbf{x}|\mathbf{y}_n, \mathbf{W}, \boldsymbol{\alpha}, \sigma)\mathbf{x}\mathbf{d}\mathbf{x} \tag{6.14}$$

$$= \sum_{i=1}^{K} p_{(i,n)}\mathbf{x}_i \tag{6.15}$$

$$i_{max} = \arg\max_i p_{(i,n)} \tag{6.16}$$

The model contains all the ingredients for efficient computation in both latent and ambient space: (6.4) gives the prior in the latent space, (6.12) the ambient marginal, (6.7) provides the conditional distribution (or mapping) from latent to ambient space, and (6.14) and (6.16) give the mean or mode of the mapping from the ambient to latent space (a more accurate but also more expensive mode-finding approximation than (6.16) can be obtained by direct gradient ascent on (6.13)).

Note that even when the vector $\mathbf{y}$ is partially observed, the conditional distributions in the latent space can still be computed using (6.13). This can be done by marginalizing out the unobserved part of the vector $\mathbf{y}$. This has important applications in estimating missing values (or data cleaning) by estimating the latent data point using the incomplete vector and re-projecting the latent point back to the ambient space.

Learning: The learning is done using expectation maximization algorithm, by maximizing the penalized log likelihood of the model at each EM iteration. The log likelihood has the form:

$$\mathcal{L} = \log \prod_{n=1}^{N} p(\mathbf{y}_n|\mathbf{W}, \boldsymbol{\alpha}, \sigma) = \sum_{n=1}^{N} \log\{\frac{1}{K} \sum_{i=1}^{K} p(\mathbf{y}_n|\mathbf{x}_i, \mathbf{W}, \boldsymbol{\alpha}, \sigma)\} \tag{6.17}$$

The log likelihood in its current form is difficult to optimize. An application of EM generally begins with the observation that the optimization of the likelihood function $\mathcal{L}$ can be simplified if only a set of additional variables, called missing variables, are known. We define the missing variables as the indicator functions z that denote the associations between the observed space points and the latent space points. The indicator functions z_{ij} is one if the latent data $\mathbf{x}_i$ is mapped to the ambient data $\mathbf{y}_i$. In the EM-framework, we maximize the complete data likelihood $p(\mathbf{y}, \mathbf{z}|\mathbf{W}, \sigma)$, the optimization of which gaurantees the optimization of incomplete likelihood. Denoting $\mathbf{z}_n = \{z_{n1}, z_{n2}, \cdots, z_{nK}\}$ as a vector that encodes the association between the latent points and ambient points, the complete likelihood is defined as :

$$\mathcal{L}_c(\mathbf{W}, \boldsymbol{\alpha}, \sigma) = \log \prod_{n=1}^{N} p(\mathbf{y}_n, \mathbf{z}_n|\mathbf{W}, \boldsymbol{\alpha}, \sigma) = \log \prod_{n=1}^{N} \prod_{i=1}^{K} p(\mathbf{y}_n|\mathbf{x}_i, \mathbf{W}, \boldsymbol{\alpha}, \sigma)^{z_{ni}} \tag{6.18}$$

The use of indicator variables simplifies the complete likelihood function as:

$$\mathcal{L}_c(\mathbf{W}, \boldsymbol{\alpha}, \sigma) = \sum_{n=1}^{N} \{ \sum_{i=1}^{K} z_{ni} \log p(\mathbf{y}_n|\mathbf{x}_i, \mathbf{W}, \boldsymbol{\alpha}, \sigma) \} \tag{6.19}$$

During the E-Step, we estimate the expected value of the complete likelihood, where the expected value of the indicator variables are computed as:

$$E[z_{ni}|\mathcal{X}, \mathcal{Y}] = p(\mathbf{x}_i|\mathbf{y}_n, \mathbf{W}, \boldsymbol{\alpha}, \sigma) \tag{6.20}$$

$$= \frac{p(\mathbf{y}_n|\mathbf{x}_i, \mathbf{W}, \boldsymbol{\alpha}, \sigma) \sum_{k=1}^{K} \mathbf{K}_\theta(\mathbf{x}, \mathbf{x}_k)}{\sum_{k=1}^{K} p(\mathbf{y}_n|\mathbf{x}_k, \mathbf{W}, \boldsymbol{\alpha}, \sigma)} \tag{6.21}$$

The M-step involves estimation of the weight parameters $\mathbf{W}$ of the conditional map (6.6). However, instead of maximizing the likelihood, we optimize the penalized likelihood under the bayesian learning framework. This involves use of hierarchical priors on the parameters $\mathbf{W}$ of the mapping function $\mathbf{F}$. The hyper-parameters $\boldsymbol{\alpha}, \sigma$ are estimated by maximizing the marginal likelihood and used to prune off weights at each EM iteration in the (6.18):

$$p(\mathbf{y}_n|\mathbf{x}_i, \boldsymbol{\alpha}, \sigma) = \int p(\mathbf{y}_n|\mathbf{x}_i, \mathbf{W}, \boldsymbol{\alpha}, \sigma) p(\mathbf{W}|\boldsymbol{\alpha}) d\mathbf{W} \tag{6.22}$$

Maximization of the marginal likelihood with respect to hyper-parameters $(\boldsymbol{\alpha}, \sigma)$ yields the estimates for the variance from the prediction error as:

$$\sigma = \frac{1}{ND - \sum_{i=1}^{W} \gamma_i} \sum_{n=1}^{N} \sum_{k=1}^{K} p_{(kn)} ||\mathbf{W}^* \boldsymbol{\phi}(\mathbf{x}) - \mathbf{y}_n||^2 \tag{6.23}$$

where a "*" superscript identifies an updated variable estimate of the weight parameters for the current EM-iteration. Here, N denotes the number of training points, D denotes the dimensionality of the ambient space vectors and K denotes the number of basis vectors. γ is a variable denoting effective number of $\mathbf{W}$ parameters in the model and is estimated as

$$\gamma_i = 1 - \alpha_i \Sigma^*_{ii} \tag{6.24}$$

The hyperparameters $\boldsymbol{\alpha}$ are estimated using the automatic relevance determination equations [113]:

$$\alpha^*_i = \frac{\gamma_i}{||\mu_i||^2}\ , \tag{6.25}$$

where μ_i is the i^{th} column of $\mathbf{W}$. Σ_{ii} denotes the diagonal values of the covariance matrix for the mapping distribution from the latent space to the ambient space(6.6).

The estimated hyperparameters are used to select a sparse set of weights for which the α_i variables does not exceed a threshold. The weights $\mathbf{W}$ are estimated by maximizing the penalized log-likelihood:

$$\mathcal{L}_{cp}(\alpha, \sigma) = \log \left[\prod_{n=1}^{N} \prod_{i=1}^{K} p(\mathbf{y}_n | \mathbf{x}_i, \mathbf{W}, \boldsymbol{\alpha}, \sigma)^{z_{ni}} \right] p(\mathbf{W}|\boldsymbol{\alpha}) \tag{6.26}$$

In maximization step we make an approximation of ignoring the terms involving derivatives of the responsibilities $p_{(n,i)} = E[z_{ni}]$ with respect to $\mathbf{W}$. This yields a simplified form of the updates as denoted in the (6.27):

$$\boldsymbol{\Sigma} = (\sigma \boldsymbol{\phi}^\top \mathbf{G} \boldsymbol{\phi} + \mathbf{S})^{-1} \tag{6.27}$$

$$\mathbf{W}^\top = \sigma \boldsymbol{\Sigma} \boldsymbol{\phi}^\top \mathbf{R} \mathbf{Y} \tag{6.28}$$

where $\mathbf{S} = \text{diag}(\alpha_1, \ldots, \alpha_M)$ with α corresponding only to the active set, $\mathbf{G} = \text{diag}(G_1, \ldots, G_K)$ with $G_i = \sum_{j=1}^{N} p_{(i,j)}$, $\mathbf{R}$ is a KxN matrix with elements $p_{(i,j)}$, and $\mathbf{Y}$ is an NxD matrix that stores the output vectors $\mathbf{y}_i, i = 1 \ldots N$ row-wise, and $\boldsymbol{\phi}$ is a KxM matrix with elements $\mathcal{G}(\mathbf{x}_i|\mathbf{x}_j, \boldsymbol{\theta})$. The Expectation Maximization algorithm effectively searches in the parameter space of $\{\mathbf{W}\}$ by reweighing the associations between the latent data points and the ambient data points $\mathbf{z}$. At each M-step it updates the weights parameters of the mapping function using the expected value of the associations (indicator variables). The sparse SLVM algorithm is summarized in fig. 6.5. In the next two sections, we discuss some of the applications of the spectral latent variable model to real world vision problems.

6.2.2 Feed-forward 3D Human Pose Prediction from Monocular Image Sequence

Due to highly articulated structure of the human body, the human pose (represented as 3D joint angles) has high degree of freedom (≈ 60 DOF). In order to predict 3D human pose directly from the 2D images, we learn multi-valued mappings that can predict multiple plausible poses for an observation(typically an image descriptor computed over a bounding box obtained from a human detector). Multi-valued mappings from the input feature space to high dimensional joint angles space can be learned by separate regression on independent joint angles. This approach is computationally expensive and sub-optimal as it involves learning large number of regressors and does not take into account the dependencies between the joint angles. Learning a joint distribution in the human pose angle space requires a large set of training exemplars that are difficult to acquire. A large number of human activities have much lower intrinsic dimensionality compared to the joint angle state space due to strong correlation across various joints. For example, activities like running and walking will always have the two leg/arm joint angles moving coherently with respect to each other.

We, therefore use low-dimensional representations of 3D human pose to efficiently learn these mappings to predict latent space points directly from the 2D image descriptors. Prior models on the human 3D pose have played central role in 3D monocular people tracking by alleviating problems due to 2D-3D ambiguities and noisy or partially observed poses. A key advantage of these models is their ability to effectively learn high dimensional variability of the poses and non-linearity of the human dynamics using only a few latent parameters. However, the learned representations should be able to preserve the topology of the human pose states during the motion and generalize well to motions outside the training set.

We employ sparse Spectral Latent Variable Model(SLVM) to learn low dimensional representations of human 3D pose. We train discriminative models to predict latent points using the image descriptors as inputs. In a discriminative framework, these image descriptors are extracted from the image bounding box obtained by running a person detector on the image. To predict these latent points from image features, we use a conditional Bayesian Mixture of Expert predictors, where each expert is a sparse Bayesian linear regressors. Each one is paired with an observation dependent gate (a softmax function with sparse linear regressor exponent)

that scores its competence when presented with different images. For different inputs, different experts may be active and their rankings (relative probabilities) may change. The model is:

$$p(\mathbf{x}|\mathbf{r}) = \sum_{i=1}^{M} g_i(\mathbf{r}) p_i(\mathbf{x}|\mathbf{r}) \tag{6.29}$$

$$g_i(\mathbf{r}) = \frac{\exp(\boldsymbol{\lambda}_i^\top \mathbf{r})}{\sum_k \exp(\boldsymbol{\lambda}_k^\top \mathbf{r})} \tag{6.30}$$

$$p_i(\mathbf{x}|\mathbf{r}) = \mathcal{G}(\mathbf{x}|\mathbf{E}_i\mathbf{r}, \boldsymbol{\Omega}_i) \tag{6.31}$$

with $\mathbf{r}$ image descriptors, $\mathbf{x}$ state outputs, g_i input dependent gates, computed using linear regressors *c.f.* (6.30), with weights $\boldsymbol{\lambda}_i$. g are normalized to sum to 1 for any given input $\mathbf{r}$ and p_i are Gaussian functions (6.31) with covariance $\boldsymbol{\Omega}_i$, centered at the expert predictions, sparse linear regressors with weights $\mathbf{E}_i$. The model is trained using a double-loop EM algorithms [92, 164].

The SLVM framework allows us to map the latent points to original 3D joint angle space using the bidirectional mapping. We map latent states (estimated using BME) to 3d human joint angles (using (6.6)) in order to recover body configurations for visualization and error reporting.

6.2.3 Discriminative Density Propagation in Low-dimensional Embedded Space

Tracking a human pose is performed to enforce a temporal smoothness constraint and allows principled resolution of ambiguities by assigning higher prior probabilities to poses similar to those hypothesized by the learned dynamical model. However, tracking in original state space is not only computationally wasteful in terms of resources, but also sub-optimal. For many visual tracking tasks in high dimension state space, it is more appropriate to de-correlate the state and project them into lower dimensional subspace that preserve the intuitive geometric properties of the original space. For example, in many human activities with repetitive structure such as walking or running, the components of the joint angle state and the image observation vector are strongly correlated.

We applied the Sparse SLVM framework to track the 3D human pose in the low-dimensional latent space. Visual inference in the low dimensional latent space is done in the same fashion

as in the original state space, as described in more detail in chapter 4. We learn the low dimensional representations of the pose vector $\mathbf{x}_t$ using Sparse SLVM as $\mathbf{y}_t$ and learn the conditional distributions $p(\mathbf{y}_t|\mathbf{r}_t)$ and $p(\mathbf{y}_t|\mathbf{y}_{t-1}, \mathbf{r}_t)$ using the labeled observation vector $\mathbf{r}_t$ over a sequence of time frames t. The 3D pose inference at each time step is done using the filtered density $p(\mathbf{y}_t|\mathbf{R}_t = \{\mathbf{r}_1, \mathbf{r}_2, \cdots, \mathbf{r}_t\})$ which is propagated by marginalizing out the state in the dynamical conditional $p(\mathbf{y}_t|\mathbf{y}_{t-1}, \mathbf{r}_t)$:

$$p(\mathbf{y}_t|\mathbf{R}_t) = \int_{\mathbf{y}_{t-1}} p(\mathbf{y}_t|\mathbf{y}_{t-1}, \mathbf{r}_t) p(\mathbf{y}_{t-1}|\mathbf{R}_{t-1}) d\mathbf{y} \tag{6.32}$$

Both the conditionals $p(\mathbf{y}_t|\mathbf{r}_t)$ and $p(\mathbf{y}_t|\mathbf{r}_t, \mathbf{y}_{t-1})$ are learned using M Gaussian components using Bayesian Mixture of Experts. We integrate M^2 pairwise products of Gaussians analytically. The means of the expanded posterior are clustered and the low weighted Gaussian components in the mixture are pruned off.

6.2.4 Fitting Non-Linear Shape Models to Human Face

In this section, we illustrate Sparse Spectral Latent Variable Models on an application to localize facial features in the images. Our methodology builds upon conventional active shape models(ASM)[177, 52] and extends the framework to handle large scale variations in shapes of face contours, across large and uneven head movement.

Landmark based deformable models like Active Shape Models(ASM) have proved effective for identification and localization of object shapes in the 2D images, and have lead to advanced tools for statistical shape analysis. ASM detects features in the image by using a combined prior shape information with the observed image likelihood. A shape $\mathbf{S}$ is typically represented as a $\mathbf{N} \times 2$ dimensional vector, consisting of 2D location co-ordinates of $\mathbf{N}$ landmark points $\mathbf{S} = [x_1, y_1, \cdots, x_N, y_N]$. The shape of an object is a geometrical characteristic that is invariant to location, scale and rotation of the object. In order to define a shape, we first define the similarity measure between the shapes using the Procrustes distance. The distance metric between two shapes $\mathbf{Y}_1$ and $\mathbf{Y}_2$(with one-to-one correspondence) is defined as:

$$\mathcal{D}_p(\mathbf{Y}_1, \mathbf{Y}_2) = \|\mathbf{Y}_1 - \mathcal{T}_{x_0,y_0,s,\theta}\{\mathbf{Y}_2\}\|^2 \tag{6.33}$$

where the transformation $\mathcal{T}_{x_0,y_0,s,\theta}$ aligns the two shapes using translation, rotation and scaling:

$$\mathcal{T}_{x_0,y_0,s,\theta}\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + s\begin{pmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{pmatrix}\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} \tag{6.34}$$

In order to estimate the intrinsic geometrical shape, all the normalized shape vectors are projected into a common reference frame using (generalized) procrustes analysis. This procedure iteratively estimates the mean shape and aligns all the shapes to it by rotation, scaling and translation. The aligned set of shapes form a Kendall shape space that is a highly non-linear manifold. The similarity between any two shapes lying on this manifold can be computed using the geodesic distance between them. For two nearby shapes, the Euclidean distance between the shape vectors are an accurate similarity measure. However, regular statistical learning methods cannot still be applied. A well known solution to this problem is to project the shapes to the tangent plane of the manifold at the mean shape. In the tangent space, Euclidean distance is a valid similarity metric. The projections on the tangent space lie on a hyperplane as opposed to the curved manifold surface and therefore various statistical methods can be used to perform shape analysis. Standard Active Shape Model(ASM) uses linear PCA to learn principal modes of variations of the shapes. This is based on the assumption that any aligned shape $\mathbf{X}$ can be expressed as different coefficients on these principal modes of variations:

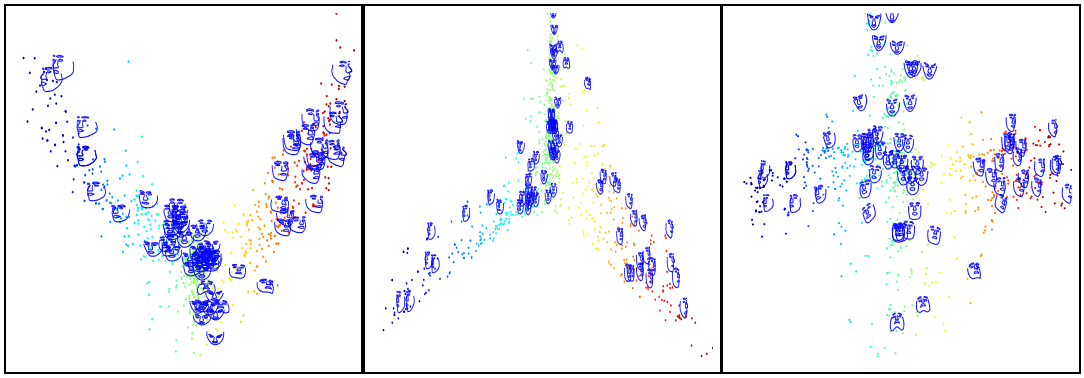
$$\mathbf{X} = \overline{\mathbf{X}} + \mathbf{P}\mathbf{b} \tag{6.35}$$

where the $\mathbf{P} = [\mathbf{p}_1, \cdots, \mathbf{p}_K]$ are the principal components(modes of variations) and $\mathbf{b}$ are the coefficients. Constraints can be applied to coefficients for each mode of variation to ensure that shape lies in a plausible shape subspace. The technique is efficiently used to iteratively search the shape in the learned subspace, by deforming the shape by matching the image likelihood of landmark points in a local image neighborhood, followed by applying global constraints to confine the deformed shape to the learned shape space.

A major limitation of the conventional ASM is that it ignores the non-linear geometry of the shape manifold. Aspect changes of 3D objects(e.g. human head) causes the shapes to vary non-linearly on a hyper-spherical manifold. The tangent space is an accurate approximation for only shapes lying in the vicinity of the reference shape (usually the mean shape). ASM, in its current form therefore cannot handle large variability in the shape. Recent research in shape analysis

and registration have proposed improved methodologies for searching for the globally optimal shape in a highly nonlinear Riemannian manifold. We propose to use low-dimensional, perceptual representations learned from non-linear Sparse Spectral Latent Variable Model(SLVM) to model the large variability in the shapes. Such variations in the shapes are typically observed for the facial features undergoing significant aspect changes due to head movement.

Fig. 6.3 illustrates the learned 2D representations obtained using different shape analysis methods. Fig. 6.3(a) shows the 2D representations learned using PCA on the shapes projected on the tangent space. Representations in the fig. 6.3(b) were obtained using Isomap on the tangent space projection of the shapes. This used Euclidean distance to construct the connectivity graph. Fig. 6.3(c) shows the embeddings obtained using isomap with connectivity graph constructed using Procrustes distances between each pair of shapes.



(a) (b) (c)

Figure 6.3: 2 dimensional representations learned using different methods: (a) Principal Component Analysis on the shapes aligned using generalized procrustes analysis and projected to tangent space at the mean shape(frontal face). (b) Isomap embeddings on the shapes aligned using generalized Procrustes analysis and projected to the tangent space at the mean shape(frontal face). Here, we use Euclidean distance in the tangent space to construct the pairwise connectivity graph. (c) Low dimensional embeddings obtained using Isomap with pairwise connectivity graph constructed using Procrustes distance between each pair of shapes. Notice that tangent space approximation distorts the similarity metric for shapes far from mean shape. The shapes due to face looking down and up are thus mapped to nearby regions in the latent space computed using tangent space approximation, as shown in (b).

Notice, that the 2D representations learned from the Isomap based on Procrustes distances are more accurate in preserving the geometric relationship of the manifold. The facial feature shapes due to head facing down and up are mapped close to each other in the embeddings learned in the tangent space, whereas in the observed space they are perceptually very different

shapes. The connected graph formed using Procrustes distance between every pair of shapes, provides a more accurate, discrete approximation of the non-linear shape manifold and preserves the topology in the latent space. The distribution of the latent points learned from the

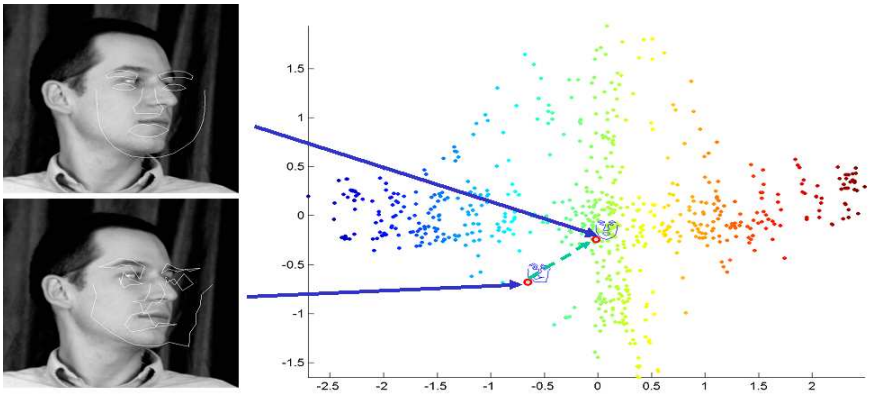


Figure 6.4: Iterative shape search using shape prior learned using non-linear Sparse SLVM. Each landmark point is matched using the learned appearance model in a local image neighborhood. This deforms the shape and may generate a shape away from the shape manifold *(bottom, left image)*. The shape is constrained using a gradient descent optimization on the latent variables to generate a nearest shape lying on the learned manifold.

Isomap on the connected graph, are used as a prior for our SLVM model. The aligned shape in the ambient space is related to the latent space using a non-linear vector valued function (6.6). The search for the most optimal shape that fits the observed face in the image is done by alternating optimization of the shape in the ambient space by first deforming the average shape by matching appearance of the landmark points, followed by constraining its corresponding latent point to lie in the learned embedded space. This is illustrated in the fig. 6.4. The shape obtained by matching the local appearance models of the landmark points $\mathbf{Y}'$ may not lie on the plausible shape space. This shape is first aligned to the common reference frame using inverse transformation $\mathcal{T}^{-1}_{x_0,y_0,s,\theta}$. The optimal latent space point $\mathbf{x}$ corresponding to the aligned shape is obtained by first finding the mode $\mathbf{x}'$ of the conditional distribution in the latent space and optimizing it using direct gradient descent with the cost function as:

$$\mathbf{x} = \arg\, min_{\mathbf{x}'} \|\mathcal{T}^{-1}_{x_0,y_0,s,\theta}(\mathbf{Y}') - \mathbf{F}(\mathbf{x}', \mathbf{W}, \boldsymbol{\alpha})\| \tag{6.36}$$

where the $\mathbf{F}$ is the non-linear mapping learned using sparse Bayesian regression(6.6). As the optimization in the (6.36) is analytically intractable, we use numerical methods based on BFGS to minimize the cost. This iterative procedure is repeated to obtain a shape that globally fits the

Inputs: Points in high-dimensional, observed(ambient) space: $\mathcal{Y} = \{\mathbf{y}_i\}_{i=1\ldots N}$.

Latent Variable Model: Parameters $(\mathbf{W}, \boldsymbol{\alpha}, \sigma)$ of Sparse Spectral Latent Variable Model(SLVM), marginal distribution in latent space $\mathcal{X}$ and ambient space $\mathcal{Y}$, conditional distribution from the ambient space to the latent space and back.

Step 1. Obtain initial, non-linear embeddings in the latent space as $\mathcal{X} = \{\mathbf{x}_i\}_{i=1\ldots N}$ from the points in the ambient space $\mathcal{Y}$. Use standard spectral methods like ISOMAP, LLE, HE, LE

Step 2. Construct latent space prior distribution either as non-parametric kernel density estimate or summation of delta functions at the latent points *c.f*. (6.4) obtained in **step 1**

Step 3. Learn Spectral Latent Variable Model using Expectation Maximization(EM) Algorithm

- **Initialize** the parameters $(\mathbf{W}_0, \boldsymbol{\alpha}_0, \sigma_0)$ of the conditional distribution $p(\mathbf{y}_n|\mathbf{x}_i, \mathbf{W}, \boldsymbol{\alpha}, \sigma)$ using the correspondences $(\mathbf{x}_i, \mathbf{y}_i)_{i=1\cdots N}$ obtained in **step 1**
- **E-step:** Apply Bayes rule(6.13) to compute the posterior probabilities $p(\mathbf{x}_i|\mathbf{y}_n, \mathbf{W}, \boldsymbol{\alpha}, \sigma)$ for re-estimating the soft assignment of the points in latent space $\mathbf{x}_i$ to the points $\mathbf{y}_n$ in ambient space. The latent points $\mathbf{x}_i$ can be obtained either as Monte-Carlo samples from the latent prior $p(\mathbf{x})$ or the original embeddings obtained from the Spectral methods (under the assumption that the latent prior is represented as summation of delta functions).
- **M-step:** Re-estimate the parameters, $(\mathbf{W}, \boldsymbol{\alpha}, \sigma)$ of the conditional map $p(\mathbf{y}_n|\mathbf{x}_i, \mathbf{W}, \boldsymbol{\alpha}, \sigma)$ according to (6.27). This is done by solving a weighted non-linear Bayesian regression problem. We employ Relevance Vector Machine [124, 113, 180, 102], that uses *Automatic Relevance Determination(ARD)* mechanism to select a sparse set of weight parameters of the regression function.

Figure 6.5: **The SLVM Learning Algorithm**

observed face in the image.

6.3 Experiments

We illustrate the SLVM algorithm both on synthetic datasets, and on real computer vision problems of estimating 3D human pose from monocular images and 3D head pose estimation by tracking facial features. Both the real world problems require search in high dimensional state space and therefore necessitates the use of low-dimensional models for learning perceptual representations that can concisely capture non-linear variations of the data.

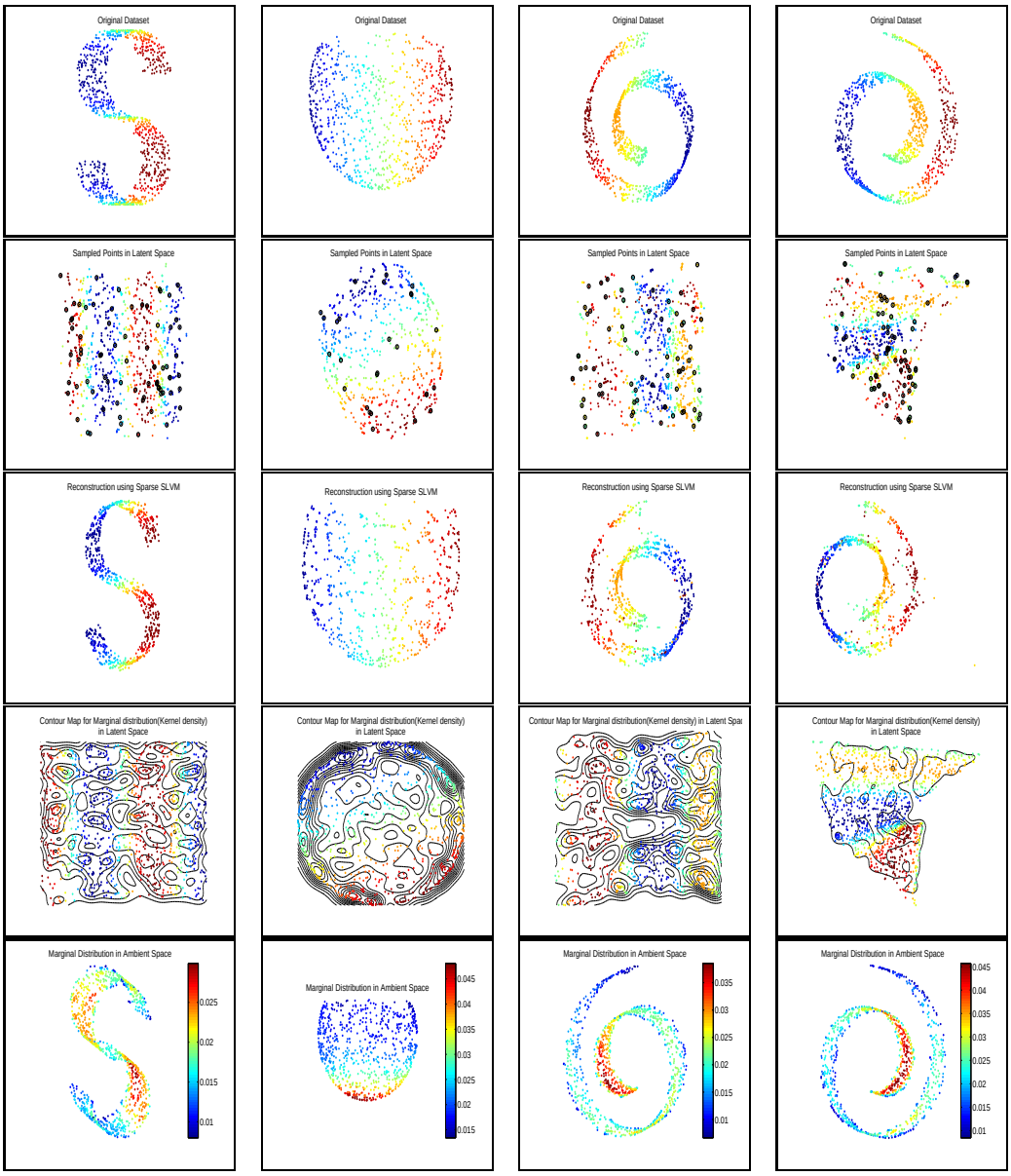
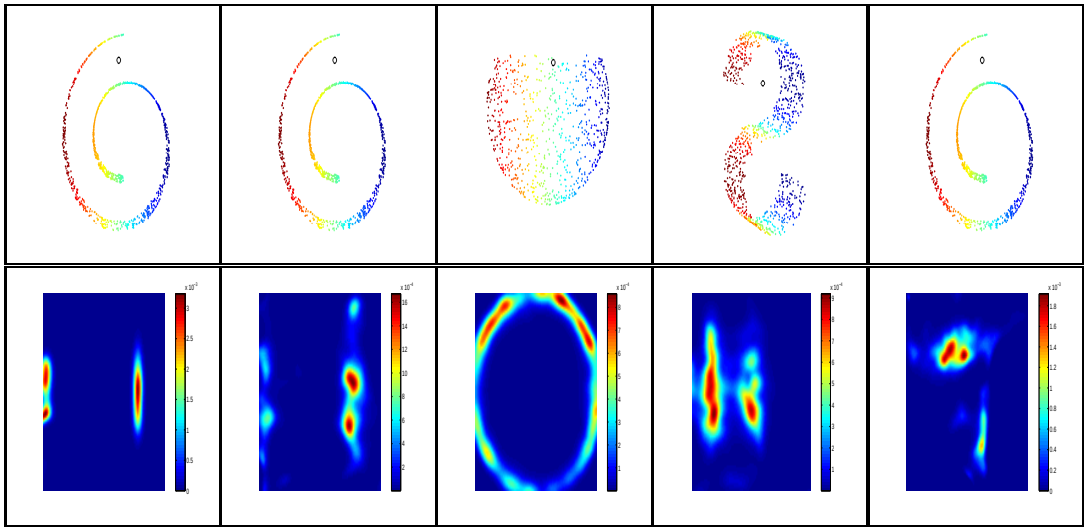


Figure 6.6: Results of Sparse SLVM on the toy dataset sampled from 3D manifolds S-Sheet, Swiss roll and Punctured sphere *(First Row)* Original dataset of 1000 points from the 3 different datasets. Color code depict the geometric ordering. *(Second Row)* 2D embeddings obtained using Sparse SLVM where the latent prior in (a-c) are obtained using ISOMAP and (d) is obtained using LLE. The circled points are active basis set used in the mapping and are automatically selected using sparse Bayesian learning.*(Third Row)* Reconstruction using Sparse SLVM. Notice that the geometric ordering is preserved as depicted by the color coding of the reconstruction. *(Fourth Row)* Latent space prior shown as contour plots for the 3 different datasets. *(Fifth Row)* Ambient marginal of SLVM computed using (6.12) on the samples obtained from the prior using Monte Carlo estimate. Unlike all the other plots shown in the figure, the color of the points represents probability, and not geometric ordering.



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

Figure 6.7: Conditional distribution (6.13) in the latent space for an out-of-sample point in ambient space. We illustrate this on the toy datasets of S-curve, Swiss roll and Punctured sphere. (a) conditional distribution for an SLVM model with latent prior as sum of delta functions centered at latent points obtained from Isomap.(b-d) shows the multimodal conditional distribution for Sparse SLVM with latent prior as kernel density centered at latent points obtained from ISOMAP. (e) shows the conditional distribution for Sparse SLVM with prior latent points obtained from LLE (fig. 6.6(d))

6.3.1 Synthetic Datasets

This set of experiments are illustrated in fig. 6.6 where we apply Sparse SLVM algorithm on 3 different toy datasets. The original data set fig. 6.6*(top row)* consists of 1000 points sampled regularly from the highly non-linear surfaces(S-curve, swissroll and punctured sphere) in 3D and color coded along one of the dimensions to highlight their relative spatial ordering. Results in the first 3 columns use Isomap and the last column used Locally linear embedding to obtain the latent prior distribution as kernel density. Fig. 6.6*(second row)* shows the learned embedding in 2D latent space and the selected relevant basis vectors (obtained using *Automatic Relevance Determination* mechanism) used in the mapping from the latent space to ambient space. Typically, the sparsity level (i.e. the number of relevant vectors) achieved using ARD for these datasets is $\approx 8\%$. Plots in the fig. 6.6*(third row)* show the reconstruction of the 3D manifold in the ambient space using the Sparse spectral latent variable model. Fig. 6.6*(fourth row)* shows the contour map for the latent prior (learned as kernel density estimation) in the

latent space. Fig. 6.6*(last row)* shows the marginal distribution in the ambient space learned using Sparse SLVM. The color of the points shows the probability and not the geometric ordering as depicted in the other plots. Notice, that the marginal probability peaks at the regions where density of data is high.

One of the key strengths of SLVM is that it enables computation of multimodal conditional distribution in the latent space unlike GPLVM that allows unimodal approximation of one of the modes. For any data point, the latent point can be chosen as weighted combination of the latent points or as latent space sample having highest responsibility (if the conditional distribution is multimodal). Fig. 6.7 plots the conditional latent space distribution (6.13) for an out-of-sample data point, computed using the Sparse SLVM. We show on 3 different datasets of S shaped curve, Swiss roll and Punctured sphere. The first column shows the results using SLVM with priors as sum of delta functions. Rest of the columns used latent prior based on Kernel density estimation. Last column shows the results for SLVM initialized using Locally Linear Embedding(LLE) whereas the rest of the results were obtained using SLVM based on Isomap. The test point in the experiments was carefully chosen to be in proximity to multiple distant points in the ambient space.

The bi-directional mapping can be effectively integrated in any probabilistic framework to learn low-dimensional, perceptual representations that are less noisy and more suitable for machine learning applications. Moreover the data projected to the latent space can be recovered using the backward mapping. In the following, we illustrate this in the 2 real world applications involving optimization and learning in high dimensional ambient space.

6.3.2 3D Human Pose Reconstruction

In this section, we report results on one of the key application of low-dimensional models. We use Sparse SLVM to learn latent perceptual representations of high-dimensional 3D human pose state space. We use motion capture data from CMU MoCap repository[1] in our experiments and apply the learned models on both quasi-real sequences obtained by rendering a synthetic human model on realistic backgrounds, and real sequences from the movie 'Run Lola Run' and static images from INRIA human detection dataset. We use 41d joint angle representation for the 3D human pose configuration. As the angles lie in cyclic space, we map

each joint angle to a pair (sin, cos) that varies continuously, thus totaling to an 82d vector in the observed pose state space. We use this 82d vector as inputs to learn the Sparse SLVM.

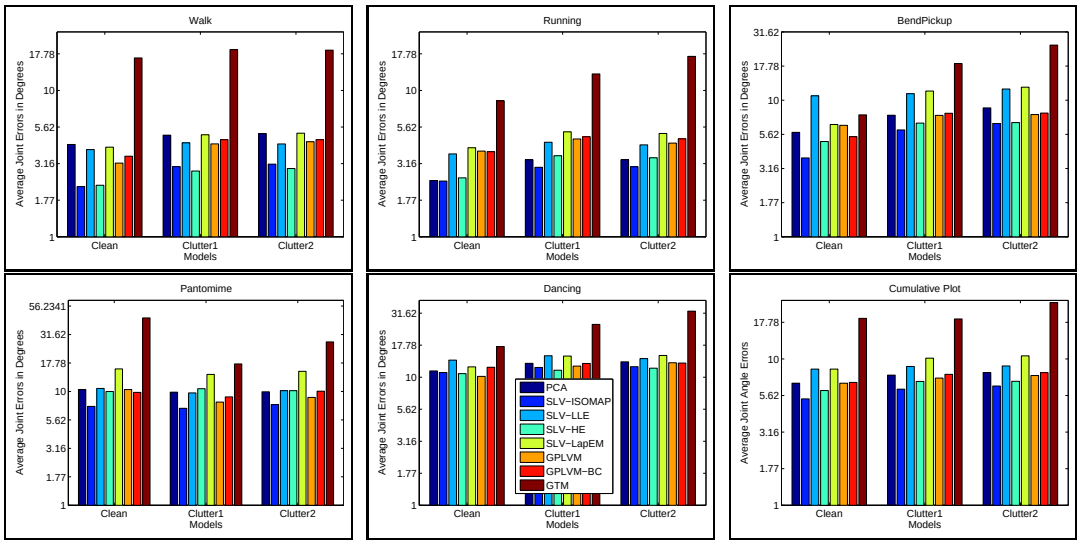
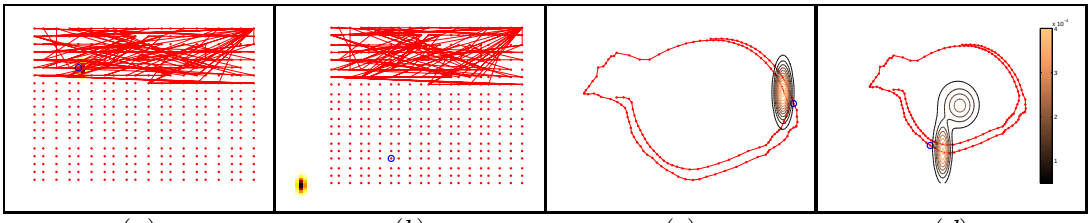


Figure 6.8: Joint angles prediction error in degrees for 5 different image sequences (Walking, Running, Bending-Pickup, Pantomime and Dancing) and 3 different imaging conditions(*Clean, Clutter1* and *Clutter2*). The bottom right plot is the cumulative prediction error averaged over all the 5 sequences. We compare several methods for learning low-dimensional representations - SLVM with different spectral embeddings (Isomap, LLE, HE, LE), GPLVM (with and without back-constraints)[100], GTM [31] and PCA. In all the models we project the joint angles to 2d latent space. We learn Bayesian Mixture of Experts(BME) model to predict the low-dimensional embeddings using image descriptors based on Multilevel Spatial Block(MSB)[16] as the inputs. In our experiments we trained both the Latent Variable Models and the corresponding BME model separately for each motion type. The error plots shown here, is obtained from the prediction of the most probable expert (as predicted by the gate distribution 6.2.2 for test input).

Image Descriptor: Image descriptors form a key component in any discriminative framework and their choice is task dependent. A desirable characteristic of the descriptors is to be sufficiently discriminative in order to distinguish between different poses of the humans in the image. However, image descriptor should be, at the same time invariant to perturbations and variabilities in the observations due to changing illumination, anthropometry and the appearance of the targets. We use Multilevel Spatial Blocks (MSB) encoding, as discussed in previous chapter, to compute a description of the image at multiple levels of semantics. These multi-level encodings are obtained by computing the dense grid based local SIFT descriptors over progressively larger neighborhood[16]. SIFT descriptors, proposed by [112], has been widely applied for the task of feature matching and object recognition. The SIFT descriptor is constructed as a



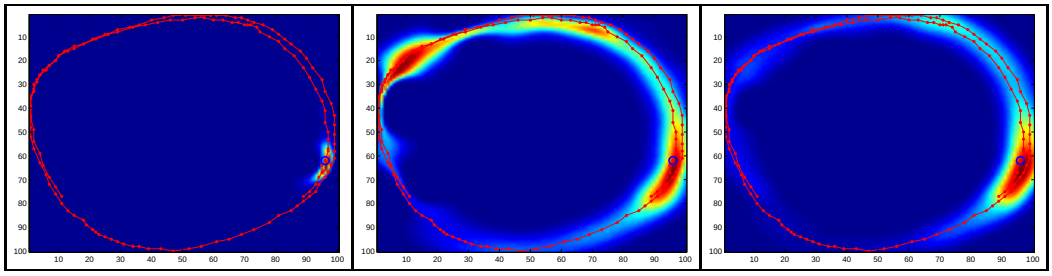
(a) (b) (c) (d)

Figure 6.9: We plot the learned conditional distribution $p(\mathbf{x}|\mathbf{r})$ for a quasi-real walking sequence, in the latent space learned using GTM(*a and b*) and SLVM-Isomap(*c and d*), given an observed image descriptor $\mathbf{r}$. The plots show the probability map (color-coded contours) of the predicted pose in the latent space. *(a)* and *(b)* show the training walking sequence in the 2d grid based latent space learned using GTM, with consecutive latent points of the temporal sequence linked to each other. The circle denote the ground truth pose of the test input in the latent space. *(c)* and *(d)* show the same for the SLVM-Isomap for the same input test data $\mathbf{r}$. Notice that the geometric ordering of the latent points associated to the poses of a walking sequence is not preserved by GTM, and multiple latent points get associated to the points in the ambient, 3d joint angles space.

concatenation of gradient orientation histograms computed over the block of regularly placed cells. The cells accumulate the gradient orientation information over a small region of the image pixels and uses bilinear interpolation to soft vote to the quantized orientation bins of the histogram. This is the key step for enhancing the invariance to affine transformation and misalignment. The multilevel spatial block in our framework was of size 1344 and was obtained by encoding the image at 3 levels of varying cell sizes. The levels had 16, 4 and 1 SIFT block, with 4x4 cells per block. The cell size for the 3 levels were 12x12, 24x24 and 48x48.

Database: For quantitative experiments we use our own database consisting of $3 \times 3247 = 9741$ clean and quasi-real images, generated using a synthetic human model. The human model was animated by importing motion capture data [1] to it using the Maya graphics package, and randomly rendering on real image backgrounds. Our dataset consisted of 3247 different 3d poses from common human motion sequences - different walking, either viewed frontally or from one side, dancing, conversation, bending and picking, running and pantomime. As discussed in the previous chapter, the data consisted of 3 categories of data depending on whether the computer graphics model is rendered on clean background(*Clean*), on real background images already seen in the training set(*Clutter1*) and on real background not seen in the training set(*Clutter2*).

For testing, we collect three sets of data from five different motion sequences, with 150



(a) (b) (c)

Figure 6.10: This figure plots the conditional distribution $p(\mathbf{x}|\mathbf{y})$ of a running sequence in the latent space for the Sparse SLVM model, given partially observed joint angles $\mathbf{y}^o$. The latent point corresponding to the complete joint angle vector $\mathbf{y}$ is shown as a circle, **(a)** plots the conditional distribution in the latent space using the entire joint angle vector $p(\mathbf{x}|\mathbf{y})$, **(b)** shows the plot of the conditional $p(\mathbf{x}|\mathbf{y}^{o_1})$ obtained using only the joints corresponding to the left arm (shoulder and elbow joints that constitute 5 dof out of a total of 41 dof in the articulated human skeleton). Notice that the 3 modes in the conditional map in the latent space arise due to the pose ambiguity caused by missing data. **(c)** plots the conditional $p(\mathbf{x}|\mathbf{y}^{o_2})$ using only the joints corresponding to the right leg (5 dof out of a total of 41 dof in the articulated human skeleton). The conditional is mostly unimodal, which suggests that the leg joint angles are more informative than the arm joint angles in a typical human walking sequence.

poses for each of the sequence. The test motions are executed by different subjects, not in the training set.

For computing the image descriptors, we obtained the bounding box around the centroid of the rendered silhouette. The bounding boxes had fixed aspect ratio and were rescaled to 320×240. The background clutter exhibited significant variability due to random background and varying aspect ratios of the human 3D poses (e.g. a person standing with arms closed compared to person with arms wide open).

We train Bayesian Mixture of Experts model on each activity in the dataset and evaluate various latent variable models in terms of 3D joint angle prediction accuracy. The latent variable models were learned separately on each of the activities. We used 5 experts with radial basis function(rbf) kernels and linear softmax gate distribution. We compare several latent variable models - SLVM-Isomap, SLVM-HE(Hessian Eigenmaps), LE(Laplacian Eigenmap), LLE(Locally Linear Embedding), GPLVM with and without back-constraints, GTM and PCA. We project the 82d (41x2=82d where we map each angle to a pair (sin, cos)) human pose vector to 2d latent space using various latent variable models. We then use Bayesian Mixture of Expert(BME) model with exactly same parameters to learn the mapping from the image descriptor

space to 2d latent space. In fig. 6.8, we show quantitative comparisons (prediction error, per joint angle, in degrees) for 5 different motions (+ a cumulative plot) and 3 sets of data: *Clean* backgrounds, *Clutter1* backgrounds and *Clutter2* backgrounds. The error was computed with respect to the prediction from the most probable expert of the BME. For visualization and error reporting we use the mapping from the 2d latent space to the joint ambient space to retrieve the joint angles. In our experiments, SLVM based on Isomap was performed best with minimum average joint angle prediction error. Performance of SLVM-HE and GPLVM(with and without back-constraints) were next best, followed by rest of the Sparse SLVM implementation based on LLE and LE. GTM on the other hand, gives significantly higher prediction error compared to rest of the latent variable models and has difficulty unfolding the high-dimensional human joint angle trajectories on its regular 2d grid. Among the 5 different sequences, the joint angle prediction errors are highest for the dancing sequence. This is primarily due to significant semantic variability between the training and testing sequences. Computationally, GPLVM is the most expensive model and PCA the cheapest to train, whereas in testing all models are about the same. In comparison, Sparse SLVMs have competitive training times.

Fig. 6.9 compares the posterior plots of predictions using GTM and SLVM-Isomap on a quasi-real walking sequence with the clean background. The circle shows ground truth test data point and the predicted posterior is shown as a color coded contour. The prediction from the SLVM-Isomap is shown on the right, with relative spatial ordering of different walking poses accurately preserved in the latent space.

Fig. 6.10 shows the conditional distribution $p(\mathbf{x}|\mathbf{y})$ in the latent space for the running sequence of partially observed joints angles for the SLVM based on Laplacian Eigenmap. The distribution is highly multi-modal when only left arm is observed. The right leg joints however are more informative than the arm and has distribution that is effectively unimodal. The modes of the distribution can be used to reconstruct the entire pose using incomplete data only.

In another set of experiments, we show in fig. 6.17 results based on real images from the movie 'Run Lola Run'. These are automatic 3d reconstructions obtained with our SLVM-Isomap. The human target in this sequence is fast moving and is filmed in non-instrumented environments. We use a model trained with 2000 walking and running poses only (quasi-real data of our model placed on real backgrounds). The walking training data used Computer

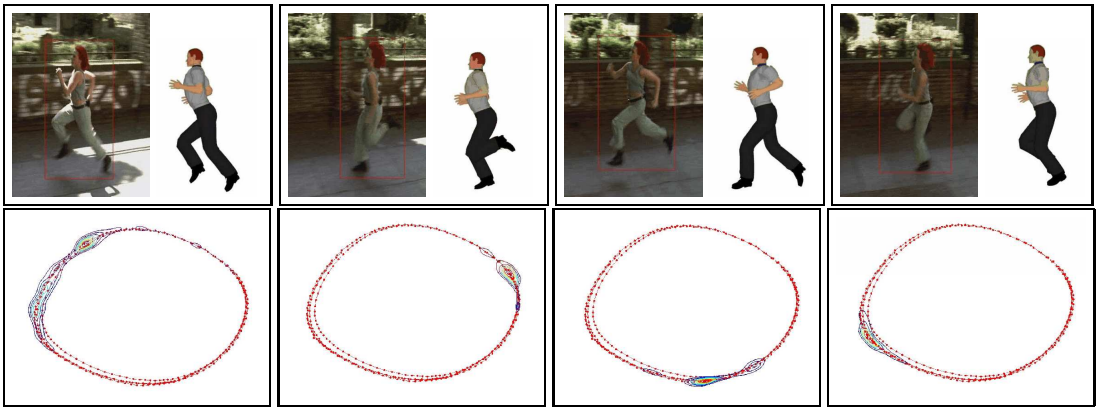


Figure 6.11: Plot of the conditional map in the latent space for a test running sequence from the movie 'Run Lola Run'. We used MSB to predict the 3D pose using Bayesian Mixture of Experts (BME). The BME model was trained on quasi-real running sequences observed from the side. Test data included frame of motion of the target parallel to the camera plane. The target is localized in the image using the human detector[53] based on SVM classifier. The bottom row shows the conditional distribution $p(\mathbf{x}|\mathbf{r})$ in the latent space where $\mathbf{r}$ denotes the image descriptor

Graphics(CG) model rendered from 8 different viewpoints. While the running sequence used the model rendered from side, frontal and back views only. Fig. 6.12 and fig. 6.11 shows the conditional distribution $p(\mathbf{x}|\mathbf{r})$ in the latent space for the 3D human pose reconstruction. The multimodal distribution $p(\mathbf{x}|\mathbf{r})$ are obtained as mixture of Gaussians, as predicted by the multi-valued mixture of experts model. In this figure, we show the distribution on the embeddings for the running sequence used in the training data, obtained using Sparse SLVM based on Laplacian Eigenmaps.

Fig. 6.13 shows the human pose reconstruction results from the real images taken from the movie 'Run Lola Run' and the INRIA pedestrian dataset. As typical with many discriminative methods, the solutions are not always entirely accurate in a classical alignment sense (this is largely due to lack of typical training data) these are nevertheless fully automatic reconstructions of a fast moving person(Lola), filmed with significant viewpoint and scale variability. Notice that the phase of the run and the synchronicity between arms and legs varies significantly across frames.

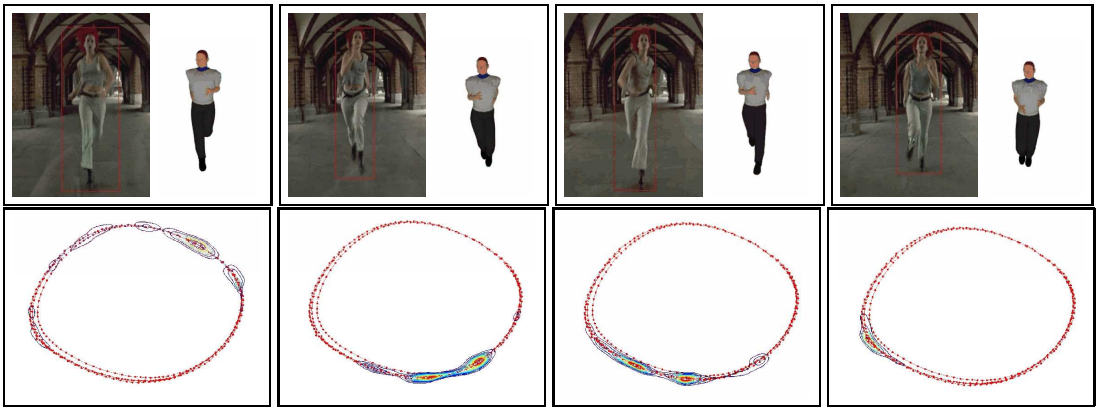


Figure 6.12: Plot of the conditional map in the latent space for a test running sequence from the movie 'Run Lola Run'. We used MSB to predict the 3D pose using Bayesian Mixture of Experts (BME). The BME model was trained on quasi-real running sequences observed from the side. Test data included frame of motion of the target parallel to the camera plane. The target is localized in the image using the human detector[53] based on SVM classifier. The bottom row shows the conditional distribution $p(\mathbf{x}|\mathbf{r})$ in the latent space where $\mathbf{r}$ denotes the image descriptor

Figure 6.13: 3d Pose reconstruction results on images from the movie 'Run Lola Run' (leftmost 4 images) and the INRIA pedestrian dataset (rightmost 4 images) [53]. *(Top row)* shows the observed 2d images, *(Bottom row)* shows 3d reconstructions from the same viewpoint as the test images

6.3.3 Visual tracking in Low dimensional Embedded Space

In another set of experiments, we used our discriminative framework based on Bayesian Mixture of Experts model and Sparse SLVM to track low dimensional representations of 3D human pose in the learned latent space. We track walking sequences from the CMU motion capture repository[1] to train the BME model, which is composed of 4 non-linear experts and a kernelized gate distribution (softmax with kernel inputs). We train the BME model on a data set containing ≈ 2000 labeled walking examples, performed in a similar setting as the test dataset. In these settings, the camera is static and background model can be learned, we used descriptors based on Shape Context Histogram[121], computed for the randomly sampled edge pixels of the foreground image. We used SLVM-Isomap to reduce the dimensionality of the 82d joint angle ambient space to a mere 3d latent space. Fig. 6.14 shows the 3d pose inference results of the walking sequence.

6.3.4 Tracking Facial Features

The prior shape model is learned using 1029 labeled images (79 landmark points) in various head poses. Each of the image is manually labeled by accurately placing the landmark points on the face on an edge contour of the facial feature(eyes, nose, mouth, eye brows and face contour). The location on the contour should be invariant across subjects and face deformation and is visually estimated by taking into account the scale changes. The training of the non-linear Active Shape Model involves learning of both the appearance and the shape model. The local appearance model for each of the landmarks is learned as image gradient profile, normal to the edge boundary of the facial features. Since this does not exhibit much variability, we employ linear PCA based model to learn local appearance models. The global shape model is more challenging to learn as it exhibits much larger variability that cannot be captured using linear models. We use Procrustes distance between every pair of images to construct a connectivity graph and use Isomap to obtain the latent prior distribution of the shapes in the latent space. Fig. 6.15 show some of the shapes used for training SLVM based active shape model. We extended active shape model to handle nonlinearities in shapes by learning perceptual representations using sparse SLVM. Sparse SLVM allows us to capture larger variability in shapes due to

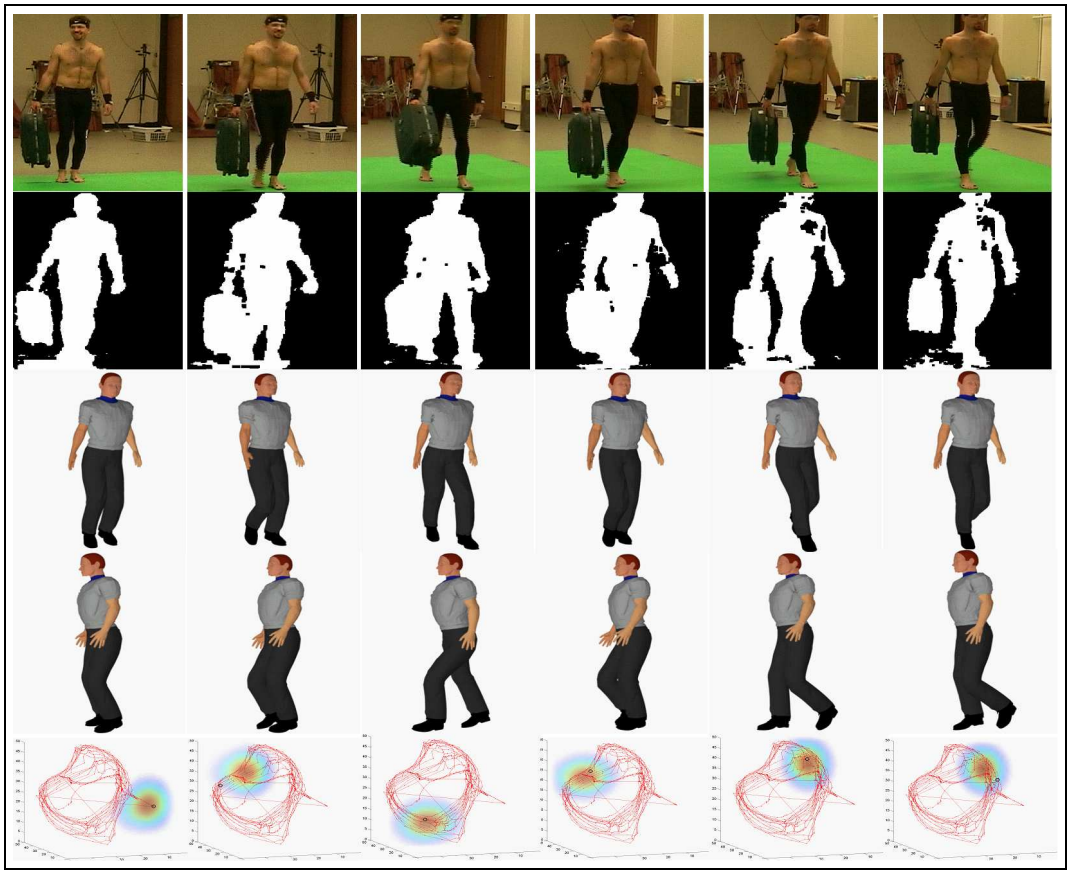


Figure 6.14: 3d Pose reconstruction results on image sequence of a person walking with a suitcase, from the CMU MoCap data repository[1] *(First row)* shows observation image sequence, *(Second row)* shows the silhouettes obtained using background subtraction[64],*(Third row)* 3d reconstructions from the same viewpoint as the test images, *(Fourth row)* 3d reconstructions from a novel viewpoint and *(Last row)* filtered conditional $p(\mathbf{y}_t|\mathbf{R}_t)$ in the 3d latent space obtained using SLVM-Isomap. Here the circled point denotes the latent point of the ground truth pose

relative positioning of facial features (eyes, mouth and nose), and learn a plausible shape space that can, in principle, be used to fit feature shapes to any novel face. We therefore used 6D embeddings to learn the non-linear SLVM based shape model. We fit the shape using alternating optimization of likelihood by sampling along the normals at the landmark points of the shape, followed by shape regularization using the non-linear SLVM based shape model. The shape regularization is done by non-linear optimization of the cost function (6.36). As this cannot be done analytically, we use numerical methods based on BFGS to optimize the cost function and estimate the nearest shape in the plausible shape space as fixed number of iterations of the gradient based search. For efficient and accurate search over scales, rotations, translations and the

Figure 6.15: Facial feature shapes of various human subjects from the training dataset. Here we show only 4 (out of total 7) different class of poses used for learning the shape space. 7 different class of poses include - frontal, left-frontal, right-frontal, right-profile, left-profile, upwards and downwards. Each of the pose class had shapes with varying degrees of head rotation for 100 subjects. Shapes were manually labeled by precisely placing the landmark points on the 2D image of the face along the contour edges of the face.

shape parameter space, we used coarse-to-fine search over 4 levels of Gaussian scale pyramid. Fig. 6.16 shows the trajectory of the shape search on the low dimensional embedded space(here we show only the first 2 dimensions of the 6D latent space). The facial features is tracked across a sequence of images and is used to estimate the head pose of the subject. We track the features using Kanade-Lucas-Tomasi (KLT) tracker which is an image registration method and computes feature points correspondences on the two images by minimizing the *Sum of Squared Intensity Difference* computed over a fixed sized window and across consecutive frames. KLT tracker assumes that for small displacements of intensity surface of the image window around a feature point, the horizontal and vertical displacement of the feature point is a function of gradient vector at that image pixel.

We track individual landmarks independently using the KLT tracker. At every frame we ensure that the shape obtained from tracking individual landmark points is a plausible shape by constraining the shape to lie on the shape manifold learned from sparse SLVM. This is again done using LM-BFGS optimization of the reconstruction cost function(6.36) at every tracking step. We learn regression functions to predict pitch and yaw of the head from the latent space variables of the shape manifold, using the manually labeled head poses for all the training images. The tilt of the head on the image plane is estimated during the shape search. Fig. 6.17 shows the tracking results across large head movement. On the left we show the predicted head pose angles in degrees and the simulated movement of a rigid computer graphic head model using the estimated head pose.

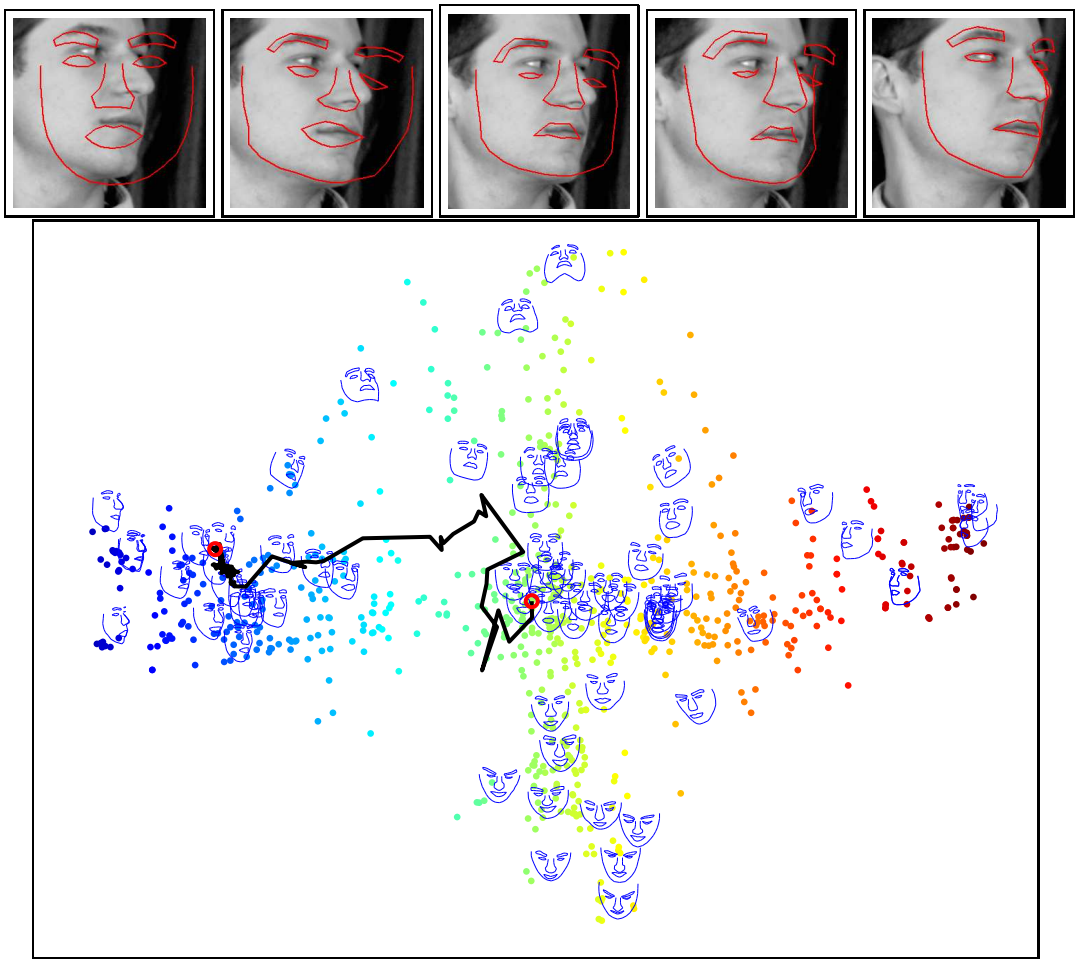


Figure 6.16: Searching for the optimal shape in the latent space obtained using Sparse SLVM. Top row shows the deformation of the mean shape initialized at the center of the face bounding box detected using Viola Jones face detector. Final fitted shape is shown in the image on the right.

6.4 Conclusion

In this chapter, we have presented a framework for learning low-dimensional embeddings using Spectral Latent Variable Models (SLVM) and showed their potential for visual inference in applications requiring search for optimal parameters in high dimensional space. We have argued in support of low-dimensional models that: (1) preserve intuitive geometric properties of the ambient distribution, e.g. locality, as required for visual tracking applications; (2) provide mappings, or more generally multimodal conditional distributions between latent and ambient spaces, and (3) are probabilistically consistent, efficient to learn and estimate and applicable with any spectral non-linear embedding method like Isomap, LLE or LE. To make

Figure 6.17: Tracking of face across large head rotations. We track the landmark points independently using KLT tracker and constrain the global shape to lie in the shape space learned using SLVM. The tracked shape is used to predict the pitch and yaw of the head. The tilt angle is obtained by aligning the 2D shape to the mean shape of the frontal face.

(1)-(3) possible, we propose models that combine the geometric and computational properties of spectral embeddings with the probabilistic formulation and the mappings offered by latent variable models. We demonstrate quantitatively that SLVMs compare favorably with existing linear and non-linear techniques, and show empirically that (in conjunction with discriminative pose prediction methods and multilevel image encodings), SLVMs are effective for the automatic 3d reconstruction of low-dimensional human poses from non-instrumented monocular images. We also demonstrated the practical applicability of the Sparse SLVM framework in learning the low-dimensional representations of highly non-linear shape manifold and using it to localize facial features in an image. The proposed framework can be easily extended to track the shapes of facial features across any head movement and predict head pose using the tracked shapes.

Chapter 7

Conclusion and Future Work

In this thesis we have examined various aspects of the problem of human 3D pose estimation from monocular image sequences. Specifically, we have examined the strengths and limitations of discriminative methods, challenges faced in implementing core components of the framework and approaches to resolve the key issues encountered. We have proposed a framework that can be used as a stand alone system and has capability to automatically self-initialize and recover from failures. By virtue of its generality, we hope that the proposed methodology will be useful in other 3d visual inference and pose estimation problems. The proposed framework can also be used to initialize generative models. We hope that our research will provide a benchmark framework for comparing the performance of discriminative methods and generative methods in general, and for devising novel techniques to combine the two approaches.

- **Multi-valued prediction** - We have developed algorithms for learning multi-valued mappings that are frequently confronted in 3D perceptions. The learning of multi-valued mappings is based on mixture of experts model and used sparse Bayesian learning to avoid overfitting to the training dataset. We demonstrate substantial improvements in human 3d pose prediction accuracy of our multi-valued predictors over Nearest Neighbor and single ridge regressions, on both synthetic and real datasets.

- **Discriminative density propagation** - We have introduced a framework for discriminative density propagation using continuous temporal chain models. We showed that modeling of multimodal conditionals and its propagation are both essential for successful human pose reconstruction. We also demonstrate the flexibility of our framework by tracking 3D pose in low dimensional kernel space and achieve sizeable gains in computational cost without any marked degradation in the pose prediction accuracy.

- **Hierarchical image descriptors** - We have argued the use of improved hierarchical image descriptors, that encode the image at multiple degrees of selectivity and invariance to perturbations due to geometric transformations, misalignment and illumination changes. We showed that hierarchical image descriptors can lead to substantial performance gains. These descriptors are further complemented with noise suppression and metric learning algorithms based on Canonical Correlation Analysis and Relevant Component Analysis. These algorithm further refine and align the image descriptors within individual pose invariance classes, in order to improve tolerance to noise in the visual observation. We also observed that in our tests, use of both - hierarchical descriptors and metric learning lead to improvement in performance.

- **Semi-supervised multi-valued learning** - We showed how unlabeled data can be incorporated into learning framework to train improved multi-valued predictive models. We extend Semi-supervised learning based on Manifold regularization to multi-valued functions. This involved adding constraints to the objective cost function to ensure that the functions are smooth along the intrinsic geometry of the inputs. We back our arguments by strong performance evaluation of the framework, learned using manifold regularization on both synthetic, *quasi real*[1] datasets and real HumanEva datasets. In our experiments, we were able to achieve substantial performance gains by training the mixture of experts model using both labeled and unlabeled examples.

- **Learning low dimensional perceptual representations of correlated covariates** - Finally we have proposed a generic framework of non-linear generative models that combine the advantages of spectral embeddings and parametric latent variable models. These models, referred to as Spectral Latent Variable Models, provide bidirectional mappings between latent and ambient spaces and are probabilistically consistent. We have showed that SLVMs compare favorably with the competing methods based on PCA, GPLVM or GTM for the reconstruction of typical human motions on a benchmark dataset. We empirically observe that SLVMs are effective for the automatic 3d reconstruction of low-dimensional representations of human 3d motion in movies. We also demonstrate the

[1] Synthetic computer graphics model placed on realistic backgrounds

applicability of these models to facial features detection and tracking in 2D images, that involved high dimensional shape search on a non-linear shape manifold.

7.1 Future work

The research work presented in this thesis attempted to develop improved techniques for discriminative human 3D pose estimation from moncular sequences. However, the problem is a challenging one and is far from being solved. While we have tried to address most of the issues in our research work, a number of these could not be investigated in depth in this thesis. The proposed framework can be easily extended to support these and we plan to study these techniques in future:

- **Combining with generative pose estimation framework** - The goal of bottom-up approach is to estimate the global orientation and 3D pose of human targets directly from the image cues. Bottom-up approaches provide fast and efficient methods for 3D pose reconstruction. However their performance depends on how representative is the training dataset. It is difficult to include all possible human poses in the training set. 3d poses characteristic to a subject may not be represented in the training set and hence may not be accurately predicted by discriminative models. For example it would be difficult to identify pose and shape abnormalities using a learning based approach, unless it has already seen the abnormal pose in the training set. In such scenarios, a generative approach is more suitable as they extrapolate better to unseen observation in the absence of sufficient training exemplars. Hence, there is a need to integrate the bottom-up models with more data driven approaches like top-down (generative) models, in order to flexibly fit the 3D pose to any given observation.

- **Partial occlusion** - In the thesis work we have omitted detailed sensitivity analysis of pose estimation in the presence of partial occlusions. To a large extent, this depends on the robustness of the image descriptors and also on the labeled exemplars used for learning the models. The local occlusion should not drastically alter the image descriptor. Rather the descriptors and the trained model should degrade gracefully with occlusions.

In order to train models that are robust to partial occlusion, we may as well use sufficiently representative training set containing examples containing humans with different occluded parts.

- **Improving scalability** - The potential success of discriminative learning approaches for 3D pose reconstruction critically depends on the ability to train models on sufficiently rich set of exemplars. We are looking into various learning techniques to make the existing framework scalable and support multiple activities. The extension entails developing fast and memory efficient algorithms for learning Bayesian Mixture of Experts, which is currently slow due to double loop optimization used for Sparse Bayesian Learning of the gating function(expert ranking function). One possible technique is to use online learning algorithms based on forward basis selection (as opposed to backward elimination) for Sparse Bayesian Learning.

- **Improving motion dynamics** - In the current discriminative tracking framework, dynamics is strongly biased by the training data set and the prior motion model. The tracking is prone to occasional failures as it is difficult to guarantee that the input vector obtained by concatenating the current observation and previous pose estimate will be close to typical input exemplars seen during the training stage. The tracking should be adequately parametrized to balance the weights assigned to the pose estimates from the previous frame and the observation in the current frame, for the 3d pose prediction.

- **Adaptive models** - Another possible extension to the current framework is to incorporate prediction results in the current time step to improve the learned models with time. Discriminative models tend to be strongly biased by the exemplar set used for training. Techniques to train the models online and adapt over time in an unsupervised fashion (or with little supervision) will be a useful step towards developing more robust discriminative frameworks. A key issue in these algorithms is to appropriately adjust the learning rate and avoid learning the model from the incorrectly predicted examples.

Appendix A

Bayesian Multi-Category Classification

In this appendix we provide detailed formulation of the multi-category classification used for training the gate distribution for Bayesian Mixture of Experts model. The framework is an extension of binary classification to polychotomous classification where the likelihood distribution has a multinomial form (generalization of binomial distribution to multiple category classification).

For M classes and N observed data pairs $\mathcal{Z} = (z^{(n)}, \mathbf{r}^{(n)})$ we adopt standard classification framework with the target variables obtained as outputs of a softmax function:

$$\sigma\{f_j(\mathbf{r}^{(n)})\} = e^{-f_j(\mathbf{r}^{(n)})} / \sum_{i}^{M} e^{-f_i(\mathbf{r}^{(n)})} \tag{A.1}$$

where the intermediate function $f_i(x)$ is typically a linear function for Generalized Linear Models. Advantage of using this form of the classification function is to implicitly normalize the output values so that they lie in the range $(0, 1)$. In most learning algorithms, a linear function f_i is preferred, in order to minimize the error due to variance, although it may also have linearly weighted form with non-linear kernel basis functions:

$$f_i(x) = \sum_{n}^{N} \lambda_{n,i} \Phi(x, x_n) = \lambda_{\mathbf{i}}^{\mathbf{T}} \mathbf{\Phi}(\mathbf{x}) \tag{A.2}$$

The above form is highly parametrized and has tendency to overfit the training data. We therefore employ sparse Bayesian learning framework to train these models to accurately learn non-linear boundaries and also avoid overfitting.

We represent the weight parameters for M-category classifier as $\mathbf{\Lambda} = \{\lambda_{\mathbf{1}}, \lambda_{\mathbf{2}}, \cdots, \lambda_{\mathbf{M}}\}$ with hyperparameters for weight parameters of each class as $\mathbf{A} = \{\mathbf{A}_1, \mathbf{A}_2, \cdots, \mathbf{A}_M\}$ and $\mathbf{A}_i = diag(\boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_N)$ for N basis functions(corresponding to each of the N training data

pair). We formulate the likelihood for M-category classification as a multinomial distribution:

$$p(\mathcal{Z}|\mathbf{\Lambda}) = \prod_{k=1}^{M}\prod_{n=1}^{N} \sigma_k \left\{\mathbf{f}(\mathbf{r^{(n)}})\right\}^{z_{nk}} \tag{A.3}$$

The weight parameters have zero centered Gaussian priors, with variance controlled by a second level of Gamma hyperpriors. This avoids overfitting and provides an automatic relevance determination mechanism, encouraging compact models with fewer non-zero expert and gate weights, for efficient prediction [113, 124, 180, 30]. We assume independent weight priors for each of the M classes:

$$p(\mathbf{\Lambda}|\mathbf{A}) = \prod_{k=1}^{M} p(\lambda_k|\mathbf{A}_k) \tag{A.4}$$

where the individual weight priors are Gaussian distributions.

$$p(\boldsymbol{\lambda}_i|\boldsymbol{\beta}_i) = \prod_{n=1}^{N} \mathcal{N}(\lambda_i^k|0, \frac{1}{\beta_i^k}) \tag{A.5}$$

A.1 Posterior Distribution

Having defined the likelihood and the prior, we seek the posterior distribution over the weights parameters $p(\Lambda|\mathcal{Z}, \mathbf{A})$. Note here that for prediction we do not need to estimate the full posterior $p(\Lambda, \mathbf{A}|\mathcal{Z})$ and it is sufficient to compute the ML(*Maximum Likelihood*) estimate of the hyper-parameters. This is typically done by optimizing the marginal evidence (*type II maximum likelihood*) for the hyperparameters. The log posterior over the weights Λ can be conveniently formulated as:

$$\log\{p(\mathbf{\Lambda}|\mathcal{Z}, \mathbf{A})\} = \sum_{k=1}^{M}\sum_{n=1}^{N} z_{nk} \log\{\sigma_k\{f(\mathbf{r}^{(n)})\}\} - (\sum_{k=1}^{M} \lambda_{\mathbf{k}}^{\mathbf{T}} \mathbf{A_k} \lambda_{\mathbf{k}}) \tag{A.6}$$

The posterior has a complex non-gaussian form and cannot be optimized in usual way. However notice that the Hessian of the log posterior is negative definite everywhere:

$$\nabla_{\lambda_{\mathbf{k}}} \nabla_{\lambda_{\mathbf{k}}} (\mathbf{log}\{\mathbf{p}(\mathbf{\Lambda}|\mathcal{Z}, \mathbf{A})\}) = -((\mathbf{\Phi_k^T B_k \Phi_k}) + \mathbf{A_k}) \tag{A.7}$$

where $\mathbf{B_k} = diag(b_k^{(1)}, b_k^{(2)}, \cdots, b_k^{(N)})$ and $b_k^{(n)} = \sigma_k\{f(\mathbf{r}^{(n)})\}[1 - \sigma_k\{f(\mathbf{r}^{(n)})\}]$.

This indicates log-concavity of the posterior function that has single mode. Under such circumstances we may conveniently make a local Gaussian approximation to the posterior in the neighborhood of its mode(Laplace approximation). If we define:

$$\mathcal{H}(\Lambda, \mathcal{Z}, \mathbf{A}) = -\ln\{p(\mathcal{Z}|\Lambda, \mathbf{A})p(\Lambda|\mathbf{A})\} \tag{A.8}$$

then we can re-write the posterior in the form:

$$p(\mathbf{\Lambda}|\mathcal{Z},\mathbf{A}) \propto \exp\{-\mathcal{H}(\Lambda,\mathcal{Z},\mathbf{A})\} \tag{A.9}$$

$$\simeq \exp\{-\mathcal{H}(\Lambda^{MP},\mathcal{Z},\mathbf{A})\}\exp\{-\frac{1}{2}(\Lambda-\Lambda^{MP})^T\mathbf{C}^{-1}(\Lambda-\Lambda^{MP})\} \tag{A.10}$$

where $\mathbf{C}$ is the curvature of the posterior and is computed as the Hessian in the (A.7). Note that the approximation is nothing but expanding the logarithm of the integrand using taylor series and retaining terms upto second order. Also note that the first order term vanishes at the modes $\Lambda^{MP} = \{\Lambda_1^{MP},\cdots,\Lambda_M^{MP}\}$.

The size of the covariance matrix $\mathbf{C}$ scales with the number of classes M and its inversion may become computationally expensive. We therefore assume block diagonal form for the covariance matrix $\mathbf{C}$ and independently optimize the posterior for the gate parameters for multiple classes. This simplification enables the multivariate gaussian to be factorized into individual posteriors for each class so that the covariance matrices can be inverted independently. Assuming block diagonal covariance matrix $\mathbf{C} = diag(\mathbf{C_1},\mathbf{C_2},\cdots,\mathbf{C_M})$ for M classes, we can factorize (A.9) as:

$$p(\mathbf{\Lambda}|\mathcal{Z},\mathbf{A}) \simeq \left\{\prod_{k=1}^{M} p(\lambda_{\mathbf{k}}^{\mathbf{MP}}|\mathcal{Z},\mathbf{A}_k)\right\}\exp\left\{\sum_{k=1}^{M}-\frac{1}{2}(\lambda_{\mathbf{k}}-\lambda_{\mathbf{k}}^{\mathbf{MP}})^{\mathbf{T}}\mathbf{C}_{\mathbf{k}}^{-\mathbf{1}}(\lambda_{\mathbf{k}}-\lambda_{\mathbf{k}}^{\mathbf{MP}})\right\} \tag{A.11}$$

where we have represented the joint weight parameters $\mathbf{\Lambda^{MP}}$ as a vector $\begin{bmatrix}\lambda_1^{MP} & \lambda_2^{MP}\cdots\lambda_M^{MP}\end{bmatrix}$. The most probable parameters are obtained by optimizing the joint posterior distribution using numerical optimization methods. The above assumption allows us to simplify the Hessian matrix and to a large extent improves the computational cost of gradient based search (discussed in next section).

A.2 Posterior Optimization

The joint log posterior is optimized using *Iterative Reweighted Least Square(IRLS)* by maximizing it with respect to the weight parameters λ_k, independently for each class k. IRLS is a special form of iterative Newton-Raphson method applied to generalized linear models. The iterative steps for gradient based updates for the weight parameter of k^{th} class:

$$\lambda_k^{(t+1)} = \lambda_k^{(t)} - \frac{\partial \ln\{p(\mathbf{\Lambda}|\mathcal{Z},\mathbf{A})\}}{\partial\lambda_{\mathbf{k}}}\left[\frac{\partial^2\log\{p(\mathbf{\Lambda}|\mathcal{Z},\mathbf{A})\}}{\partial\lambda_{\mathbf{k}}^{\mathbf{2}}}\right]^{-1} \tag{A.12}$$

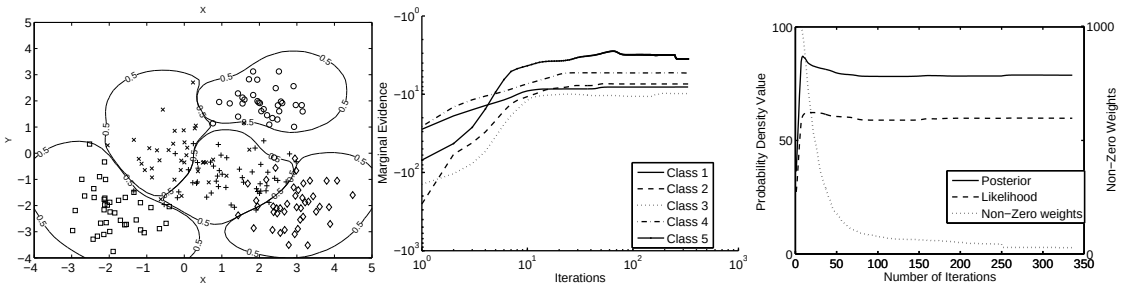


Figure A.1: *(Left plot)* Classification boundaries (0.5 probability contours) of the Bayesian multi-category classifier on a synthetic toy dataset*(Middle plot)* Marginal Evidence maximization conditioned on the hyper-parameters for each of class $\alpha_{\mathbf{k}}$ on log-scale. Notice that the marginal evidence increases with iterations for every class simultaneously. *(Right plot, Left scale)* Corresponding change of Posterior and Likelihood values with iterations. *(Right plot, Right Scale)* Corresponding non-zero weights(model complexity) with no. of iterations. Note here that most of the change occurs in first few iterations.

The gradient is computed as:

$$\nabla_{\lambda_k}(\log\{p(\Lambda|\mathcal{Z},\mathbf{A})\}) = -\sum_{n=1}^{N}[\Phi_k(\mathbf{r}^{(n)})(z_k^{(n)} - z_k^{(n)}\sigma_k\{f(\mathbf{r}^{(n)})\}) - \sum_{i\neq k}^{M}\Phi_i(\mathbf{r}^{(n)})z_i^{(n)}\sigma_k\{f(\mathbf{r}^{(n)})\}] - \lambda_k\mathbf{A}_k \tag{A.13}$$

$$= -\sum_{n=1}^{N}\Phi_k(\mathbf{r}^{(n)})(z_k^{(n)} - \sigma_k\{f(\mathbf{r}^{(n)})\}) - \lambda_k\mathbf{A}_k \tag{A.14}$$

Here we use the fact that $\sum_{i=1}^{M} z_i^{(n)} = 1$. The Hessian is approximated as a block diagonal matrix with non-diagonal blocks $\nabla_{\lambda_{\mathbf{k1}}}\nabla_{\lambda_{\mathbf{k2}}}$ set to $\mathbf{0}$. The inverse of Hessian computed for each class k in (A.7) is used for the block diagonal covariance matrix $\mathbf{C} = diag(\mathbf{C}_1, \cdots, \mathbf{C}_M)$ of the multivariate Gaussian posterior (A.11).

A.3 Optimizing the Hyperparameters

Sparse Bayesian inference proceeds by estimating the most probable estimate of the hyperparameters $\mathbf{A}_k$ for each class and using it to prune off the weights λ_k. The most probable parameters are obtained by maximizing the posterior of hyperparameters $p(\mathbf{A}_k|\mathcal{Z}) \propto$

$p(\mathcal{Z}|\mathbf{A}_k)p(\mathbf{A}_k)$. The term $p(\mathcal{Z}|\mathbf{A}_k)$ is called marginal likelihood and its optimization is referred to as *Type II Maximum Likelihood*.

$$p(\mathcal{Z}_k|\mathbf{A}_k) = \int p(\mathcal{Z}_k|\lambda_k)p(\lambda_k|\mathbf{A}_k)d\lambda_k \tag{A.15}$$

$$\approx (2\pi)^{-N/2}|\mathbf{B}_k + \Phi\mathbf{A}_k^{-1}\Phi^T|^{-1/2}\exp\{-\frac{1}{2}\hat{\mathbf{z}}_k^T(\mathbf{B}_k + \Phi\mathbf{A}_k^{-1}\Phi^T)^{-1}\hat{\mathbf{z}}_k\} \tag{A.16}$$

Where we locally linearize the non-linear likelihood function using the first order approximation around the mode Λ_{MP}:

$$\hat{\mathbf{z}}_k = \Phi\lambda_{k,MP} + \mathbf{B}_k^{-1}(\mathbf{z}_k - \sigma_k\{\Phi\lambda_{k,MP})\} \tag{A.17}$$

The hyperparameter for each class k is obtained by maximizing the log of marginal evidence with respect to log of $\mathbf{A}_k$. Maximization is done in log space for analytical tractability. This cannot be done in closed form and is iteratively estimated by differentiating the log of marginal (A.15) with respect to log of $\mathbf{A}_k = diag(\boldsymbol{\beta}_{k,1}, \boldsymbol{\beta}_{k,2}, \cdots, \boldsymbol{\beta}_{k,N})$. For initial value of $\boldsymbol{\beta}_{k,i}$ and $\lambda_{\mathbf{k}}$, the equations for the iterative updates for the i^{th} weight hyper-parameter:

$$\boldsymbol{\beta}_{\mathbf{k,i}}^{\mathbf{new}} = \frac{1 - \boldsymbol{\beta}_{\mathbf{k,i}}\{\mathbf{C}_{\mathbf{k}}^{(\mathbf{i,i})}\}}{\lambda_{\mathbf{k}}^{\mathbf{2}}} \tag{A.18}$$

where $\mathbf{C}_{\mathbf{k}}^{(\mathbf{i,i})}$ is the i^{th} diagonal element of the covariance matrix of the weights posterior, expressed as a multivariate gaussian using Laplace approximation. The $\boldsymbol{\beta}_{\mathbf{k,i}}$ updated at each iteration step is used to selectively remove irrelevant basis functions. The relevance of the the weight parameters $\lambda_{k,i}$ is determined using the ARD (automatic relevance determination) mechanism. The weight parameters λ_k in the (A.18) are the most probable estimates obtained using IRLS as discussed in the previous section. The iterative updates for the hyperparameters and subsequent pruning off the weights $\lambda_{\mathbf{k}}$ of each class ensures simultaneous increase of marginal evidence at each iteration as shown in fig. A.1*(middle plot)*. On the right we show the change of overall posterior and likelihood with iteration (on left y-axis) and weights pruning (on right y-axis). Fig. A.1*(left plot)* shows the classification results on a synthetic dataset with 5 classes. We used Radial Basis Function as the bases function in these experiments.

Appendix B

Semi-Supervised Sparse Bayesian Classification

For a two-class classifier with the inputs $\mathbf{r}_i$ and the class labels $\mathbf{x}_i$, the non-linear classification boundary is represented as $f(\mathbf{r} : \mathbf{W}) = \mathbf{W}^T\mathbf{K}(\mathbf{r})$. For linear boundaries the function takes the following form $f(\mathbf{r} : \mathbf{W}) = \mathbf{W}^T\mathbf{r}$ with the kernel mapping $\mathbf{K}$ replaced by the original input vectors $\mathbf{r}$. The binary classification function is learned as the logistic sigmoid function $\sigma(f) = 1/(1 + e^{-f})$, with the likelihood as a binomial distribution:

$$p(\mathbf{x}|\mathbf{r}, \mathbf{W}) = \prod_{n=1}^{N} \sigma\{\mathbf{K}(\mathbf{r}_n), \mathbf{W}\}^{x_n} \left[1 - \sigma\{\mathbf{K}(\mathbf{r}_n), \mathbf{W}\}\right]^{1-x_n} \tag{B.1}$$

Here the target labels $\mathbf{x}$ lie in the set $0, 1$.

In sparse Bayesian learning, we estimate the hyperparameters by maximizing the marginal likelihood obtained by integrating out the weight parameters:

$$p(\mathbf{x}|\boldsymbol{\alpha}, \gamma_I) = \int p(\mathbf{x}|\mathbf{W}, \boldsymbol{\alpha}, \gamma_I) p(\mathbf{W}|\boldsymbol{\alpha}, \gamma_I) d\mathbf{W} \tag{B.2}$$

Hyperparameters in the (B.2) are the parameters $\boldsymbol{\alpha}$ and γ_I that are the inverse variance of the ambient prior and the intrinsic geometry regularization prior respectively. The integral is analytically intractable for Bayesian classification as the likelihood $p(\mathbf{x}|\mathbf{W}, \boldsymbol{\alpha}, \gamma_I)$ is a binomial distribution [180, 33]. We therefore use Laplace approximation that estimates the integral as a local gaussian approximation of the integrand over the neighborhood of the mode. The integrand in this case is the weights $\mathbf{W}$ posterior:

$$p(\mathbf{W}|\mathbf{x}, \boldsymbol{\alpha}, \gamma_I) \propto p(\mathbf{x}|\mathbf{W}, \boldsymbol{\alpha}, \gamma_I) p(\mathbf{W}|\boldsymbol{\alpha}, \gamma_I) \tag{B.3}$$

$$\approx p(\mathbf{W}_{MP}|\mathbf{x}, \boldsymbol{\alpha}, \gamma_I) \exp(-\frac{1}{2}(\mathbf{W} - \mathbf{W}_{MP})^T \boldsymbol{\Sigma}^{-1} (\mathbf{W} - \mathbf{W}_{MP}) \tag{B.4}$$

where $\mathbf{W}_{MP}$ is the mode of the non-gaussian posterior distribution. The above equation is a second order Taylor series approximation of the posterior distribution. The covariance is obtainted as the curvature of the posterior computed (Hessian), $\boldsymbol{\Sigma}^{-1} = -\bigtriangledown^2_{\mathbf{W}} p(\mathbf{W}|\mathbf{x}, \boldsymbol{\alpha}, \gamma_I)$.

The modes of the non-gaussian posterior distribution can be obtained by conveniently optimizing its logarithm using gradient based methods. We use *Iterative Reweighted Least Square(IRLS)*:

$$\log\left[p(\mathbf{x}|\mathbf{W},\boldsymbol{\alpha},\gamma_I)p(\mathbf{W}|\boldsymbol{\alpha},\gamma_I)\right] = \sum_{n=1}^{N}\left[x_n \log(\sigma\{\mathbf{K}_L,\mathbf{W}\}) + (1-x_n)\log(1-\sigma\{\mathbf{K}_L,\mathbf{W}\})\right] \tag{B.5}$$

$$-\frac{1}{2}\mathbf{W}^T\mathbf{A}\mathbf{W} - \gamma_I\mathbf{W}^T\mathbf{K}^T\mathbf{L}\mathbf{K}\mathbf{W} \tag{B.6}$$

Here $\mathbf{K}_L$ is the kernel matrix for only labeled examples whereas $\mathbf{K}$ is the kernel matrix for both labeled and unlabeled examples. $\mathbf{L}$ is the graph laplacian matrix using the both labeled and unlabeled exemplars. The covariance matrix of the posterior are obtained as:

$$\boldsymbol{\Sigma} = (\mathbf{K}_L^T\mathbf{B}\mathbf{K}_L + \mathbf{A} + \gamma_I\mathbf{K}^T\mathbf{L}\mathbf{K})^{-1} \tag{B.7}$$

where $\mathbf{B} = diag\{b_1, b_2, \ldots, b_N\}$ and $b_i = \left[\sigma\{\mathbf{W}^2\mathbf{K}_L(\mathbf{r}_i)\}(1-\sigma\{\mathbf{W}\mathbf{K}_L(\mathbf{r}_i)\})\right]$. The logarithm of the marginal likelihood is obtained as:

$$\mathcal{M}_{\mathcal{L}} = \log(p(\mathbf{x}|\boldsymbol{\alpha},\gamma_I)) = -\tfrac{N}{2}\log(2\pi) - \tfrac{1}{2}\log|\mathbf{B}^{-1} + \mathbf{K}_L^T(\mathbf{A}+\gamma_I\mathbf{K}^T\mathcal{L}\mathbf{K})^{-1}\mathbf{K}_L|$$

$$-\frac{1}{2}\mathbf{x}^T(\mathbf{B}^{-1} + \mathbf{K}_L^T(\mathbf{A}+\gamma_I\mathbf{K}^T\mathcal{L}\mathbf{K})^{-1}\mathbf{K}_L)^{-1}\mathbf{x} \tag{B.8}$$

We use standard determinant identity

$|\mathbf{A}+\gamma_I\mathbf{K}^T\mathcal{L}\mathbf{K}||\mathbf{B}^{-1}+\mathbf{K}_L^T(\mathbf{A}+\gamma_I\mathbf{K}^T\mathcal{L}\mathbf{K})^{-1}\mathbf{K}_L| = |\mathbf{B}^{-1}||(\mathbf{A}+\gamma_I\mathbf{K}^T\mathcal{L}\mathbf{K})+\mathbf{K}_L^T\mathbf{B}\mathbf{K}_L|$.

Also notice that $|\boldsymbol{\Sigma}| = |\mathbf{A}+\gamma_I\mathbf{K}^T\mathcal{L}\mathbf{K}+\mathbf{K}_L^T\mathbf{B}\mathbf{K}_L|$. Therefore

$$\log|\mathbf{B}^{-1}+\mathbf{K}_L^T(\mathbf{A}+\gamma_I\mathbf{K}^T\mathcal{L}\mathbf{K})^{-1}\mathbf{K}_L| = -\log|\boldsymbol{\Sigma}| - \log(|\mathbf{B}|) - \log|(\mathbf{A}+\gamma_I\mathbf{K}^T\mathcal{L}\mathbf{K})| \tag{B.9}$$

Using Woodbury inversion identity and $\mathbf{W} = \boldsymbol{\Sigma}\mathbf{K}_L^T\mathbf{B}\mathbf{x}$ to express the last term in the eqn. (B.9):

$$\mathbf{x}^T(\mathbf{B}^{-1} + \mathbf{K}_L^T(\mathbf{A}+\gamma_I\mathbf{K}^T\mathcal{L}\mathbf{K})^{-1}\mathbf{K}_L)^{-1}\mathbf{x} = \mathbf{x}^T\mathbf{B}\mathbf{x} - \mathbf{x}^T\mathbf{B}\mathbf{K}_L\boldsymbol{\Sigma}\mathbf{K}_L^T\mathbf{B}\mathbf{x} \tag{B.10}$$

$$= (\mathbf{x}-\mathbf{K}_L\mathbf{W})\mathbf{B}(\mathbf{x}-\mathbf{K}_L\mathbf{W})^T + \mathbf{W}^T\mathbf{A}\mathbf{W} + \gamma_I\mathbf{W}^T\mathbf{K}^T\mathcal{L}\mathbf{K} \tag{B.11}$$

We maximize the marginal likelihood $\mathcal{M}_{\mathcal{L}}$ (B.8) in the log space by taking its derivative w.r.t. to $\log\boldsymbol{\alpha}$ and $\log\gamma_I$:

$\frac{\partial\mathcal{M}_{\mathcal{L}}}{\partial\log\alpha_i} = -\mathbf{Tr}[(\mathbf{K}_L^T\mathbf{B}\mathbf{K}_L + \mathbf{A} + \gamma_I\mathbf{K}^T\mathcal{L}\mathbf{K})^{-1}\frac{\partial\mathbf{A}}{\partial\log\alpha_i}]$

$$-\mathbf{Tr}[(\mathbf{A}+\gamma_I\mathbf{K}^T\mathcal{L}\mathbf{K})^{-1}\frac{\partial\mathbf{A}}{\partial\log\alpha_i}]+\mathbf{W}^T\frac{\partial\mathbf{A}}{\partial\log\alpha_i}\mathbf{W} \tag{B.12}$$

Equating eqn. (B.12) to zero gives the following equations for iterative updates:

$$\alpha_i^{(k+1)}=\frac{\alpha_i^{(k)}T_{ii}}{\Sigma_{ii}+\mathbf{W}_i^2} \text{ where } \mathbf{T}=(\mathbf{A}+\gamma_I\mathbf{K}^T\mathcal{L}\mathbf{K})^{-1} \tag{B.13}$$

γ_I is estimated using the following update equations:

$$\frac{\partial\mathcal{M}_\mathcal{L}}{\partial\log\gamma_I}=-\mathbf{Tr}[(\mathbf{K}_L^T\mathbf{B}\mathbf{K}_L+\mathbf{A}+\gamma_I\mathbf{K}^T\mathcal{L}\mathbf{K})^{-1}\gamma_I(\mathbf{K}^T\mathcal{L}\mathbf{K})$$

$$-\mathbf{Tr}[(\mathbf{A}+\gamma_I\mathbf{K}^T\mathcal{L}\mathbf{K})^{-1}\gamma_i(\mathbf{K}^T\mathcal{L}\mathbf{K})]+\gamma_i\mathbf{W}^T(\mathbf{K}^T\mathcal{L}\mathbf{K})\mathbf{W} \tag{B.14}$$

Setting this to zero and solving for γ_I:

$$\gamma_I^{(k+1)}=\frac{\gamma_I^{(k)}\mathbf{Tr}\left[\mathbf{T}(\mathbf{K}^T\mathcal{L}\mathbf{K})\right]}{\mathbf{Tr}\left[\mathbf{\Sigma}(\mathbf{K}^T\mathcal{L}\mathbf{K})\right]+\mathbf{W}^T(\mathbf{K}^T\mathcal{L}\mathbf{K})\mathbf{W}} \tag{B.15}$$

References

[1] CMU Human Motion Capture DataBase. Available online at http://mocap.cs.cmu.edu/search.html, 2003.

[2] Autodesk maya. Available online at http://www.Autodesk.com/Maya, 2007.

[3] H. Haussecker A. Balan, M. J. Black and L. Sigal. Shining a light on human pose: On shadows, shading and the estimation of pose and shape. *ICCV*, 2007.

[4] A. Agarwal and B. Triggs. 3d human pose from silhouettes by Relevance Vector Regression. In *CVPR*, 2004.

[5] A. Agarwal and B. Triggs. Learning to track human motion from silhouettes. In *International Conference on Machine Learning*, 2004.

[6] A. Agarwal and B. Triggs. Tracking articulated motion using a mixture of autoregressive models. In *ECCV*, 2004.

[7] A. Agarwal and B. Triggs. Monocular human motion capture with a mixture of regressors. In *Workshop on Vision for Human Computer Interaction*, 2005.

[8] A. Agarwal and B. Triggs. Hyperfeatures multilevel local coding for visual recognition. In *ECCV*, 2006.

[9] A. Agarwal and B. Triggs. A local basis representation for estimating human pose from cluttered images. In *ACCV*, 2006.

[10] A. Agarwal and B. Triggs. Recovering 3d human pose from monocular images. *PAMI*, 2006.

[11] S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE Pattern Analysis and Machine Intelligence*, 26:1475–1490, 2004.

[12] A. Aggarwal and B. Triggs. Hyperfeatures multilevel local coding for visual recognition. *ECCV*, 14(1), 2006.

[13] F. Aherne, N. Thacker, and P. Rocket. Optimal pairwise geometric histograms. In *British Machine Vision Conference*, 1997.

[14] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: Shape completion and animation of people. In *SIGGRAPH*, 2005.

[15] A. Athistos and S. Sclaroff. Estimating 3d hand pose from a cluttered image. In *ICCV*, 2003.

[16] Cristian Sminchisescu Atul Kanaujia and Dimitris N. Metaxas. Semi-supervised hierarchical models for 3d human pose reconstruction. *CVPR*, 2007.

[17] Rosenhahn B., Schmaltz C., Brox T., Weickert J., Cremers D., and Seidel H.-P. Markerless motion capture of man-machine interaction. *CVPR*, 2008.

[18] G. Bakir, J. Weston, and B. Scholkopf. Learning to find pre-images. In *Advances in Neural Information Processing Systems*, 2004.

[19] A. Balan, L. Sigal, M. J. Black, J. Davis, and H. Haussecker. Detailed human shape and pose from images. *CVPR*, 2007.

[20] A. Bar-hillel, Tomer Hertz, Noam Shental, and Daphna Weinshall. Learning distance functions using equivalence relations. In *International Conference on Machine Learning*, 2003.

[21] M. Belkin and P. Niyogi. Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering. In *Advances in Neural Information Processing Systems*, 2002.

[22] M. Belkin and P. Niyogi. Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering. In *Advances in Neural Information Processing Systems*, 2002.

[23] M. Belkin, P. Niyogi, and V. Sindhwani. On manifold regularization. In *Artificial Intelligence and Statistics*, 2005.

[24] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *PAMI*, 24, 2002.

[25] J. Bengio, J. Paiement, and P. Vincent. Out-of-Sample Extensions for LLE, Isomap, MDS, Eigenmaps and Spectral Clustering. In *Advances in Neural Information Processing Systems*, 2003.

[26] Yoshua Bengio, Jean Paiement, and Pascal Vincent. Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. In *Technical Report 1238,Departement de Informatique et Recherche Operationnelle Universite de Montreal*, 2003.

[27] K. Bennett and A. Demiriz. Semi-supervised support vector machines. *Advances in Neural Information Processing Systems*, (11):368–374, 1999.

[28] J. O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer Series, 1985.

[29] D. P. Bertsekas. Nonlinear programming, belmont. ma: Athena scientific, 2nd edition. In *IEEE International Conference on Face and Gesture Recognition*, 1999.

[30] C. Bishop and M. Svensen. Bayesian hierarchical mixtures of experts. In *Uncertainty in Artificial Intelligence*, 2003.

[31] C. Bishop, M. Svensen, and C. K. I. Williams. Gtm: The generative topographic mapping. *Neural Computation*, (1):215–234, 1998.

[32] C. Bishop, M. Svensen, and C. K. I. Williams. Gtm: The generative topographic mapping. *Neural Computation*, (1):215–234, 1998.

[33] Christopher M. Bishop and Michael E. Tipping. Bayesian regression and classification. *Advances in Learning Theory: Methods, Models and Applications*, 190(1), 2003.

[34] A. Bissacco. Modeling and learning contact dynamics in human motion. *CVPR*, (1):421–428, 2005.

[35] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. *COLT: Proceedings of the Workshop on Computational Learning Theory*, 1998.

[36] Liefeng Bo, Cristian Sminchisescu, Atul Kanaujia, and Dimitris Metaxas. Fast algorithms for large scale conditional 3d prediction. *CVPR*, 2008.

[37] A. bottino and A. Laurentini. A silhouette based technique for the reconstruction of human movement. *CVIU*, 2001.

[38] O. Bousquet, O. Chapelle, and M. Hein. Measure based regularization. *Advances in Neural Information Processing Systems*, 2003.

[39] M. Brand. Shadow Puppetry. In *ICCV*, pages 1237–44, 1999.

[40] C. Bregler and J. Malik. Tracking People with Twists and Exponential Maps. In *CVPR*, 1998.

[41] CHRISTOPHER J.C. BURGES. A tutorial on support vector machines for pattern recognition, 1998. Data Mining and Knowledge Discovery.

[42] A. Burl, M. Weber, and P. Perona. A probabilistic approach to object recognition using local photometry and global geometry. *European Conference on Computer Vision*, 1998.

[43] J. Carranza, C. Theobalt, M. Magnor, and H.-P. Seidel. Freeviewpoint video of human actors. *Proc. ACM Siggraph*, 2003.

[44] O. Chapelle, B. Scholkopf, and A. Smola. *Semi-supervised Learning*. MIT Press, 2006.

[45] O. Chapelle, J. Weston, and B. Schölkopf. Cluster kernels for semi-supervised learning. volume 15 of *NIPS*. 2003.

[46] G. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In *CVPR*, 2003.

[47] G. Cheung, T. Kanade, J.-Y. Bouguet, and M. Holler. A real time system for robust 3d voxel reconstruction of human motions. *CVPR*, 2:714–720, 2000.

[48] K. Choo and D. Fleet. People Tracking Using Hybrid Monte Carlo Filtering. In *ICCV*, 2001.

[49] K. Choo and D. Fleet. People tracking using hybrid monte carlo filtering. *ICCV*, pages 321–328, 2001.

[50] S. Chretien and A. O. Hero. Kullback proximal algorithms for maximum likelihood estimation. 46(5):1800–1810, 2000. IEEE Transactions on Information Theory.

[51] I. Cohen, G. Medioni, and H. Gu. Inference of 3d human body posture from multiple cameras for vision-based user interface. *In 5th World Multi-Conference on Systemics, Cybernetics and Informatics, Orlando*, 2001.

[52] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. In *ECCV*, 1998.

[53] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[54] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection, 2005. International Conference Vision Pattern Recognition.

[55] Quentin Delamarre and Olivier D. Faugeras. 3d articulated models and multi-view tracking with silhouettes. *ICCV*, (1):716–721, 1999.

[56] W. DeSarbo and W. Cron. A maximum likelihood methodology for clusterwise linear regression. *Journal of Classification*, 5:249–282, 1988.

[57] J. Deutscher, A. Blake, and I. Reid. Articulated Body Motion Capture by Annealed Particle Filtering. In *CVPR*, 2000.

[58] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. *CVPR*, (1):2126–2133, 2000.

[59] D. DiFranco, T. Cham, and J. Rehg. Reconstruction of 3-D Figure Motion from 2-D Correspondences. In *CVPR*, 2001.

[60] D. Donoho and C. Grimes. Hessian Eigenmaps: Locally Linear Embedding Techniques for High-dimensional Data. *Proc. Nat. Acad. Arts and Sciences*, 2003.

[61] D. Donoho and C. Grimes. Hessian Eigenmaps: Locally Linear Embedding Techniques for High-dimensional Data. *Proc. Nat. Acad. Arts and Sciences*, 2003.

[62] G. Doretto, A. Chiuso, Y.N. Wu, and S. Soatto. Dynamic textures. *IJCV*, (2):91–109, 2003.

[63] T. Drummond and R. Cipolla. Real-time tracking of highly articulated structures in the presence of noisy measurements. *ICCV*, pages 315–320, 2001.

[64] A. Elgammal, R. Duraiswami, D. Harwood, and L. Davis. Foreground and background modeling using non-parametric kernel density estimation for visual surveillance. *Proc.IEEE*, 2002.

[65] A. Elgammal and C. Lee. Inferring 3d body pose from silhouettes using activity manifold learning. In *CVPR*, 2004.

[66] Anita Faul and M. E. Tipping. Analysis of sparse bayesian learning. In *Advances in NIPS 2002*, 2000.

[67] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1), 2003.

[68] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 1, 2005.

[69] R. Fergus, A. Zisserman, and P. Perona. Object class recognition by unsupervised scale-invariant learning. *CVPR*, 1, 2003.

[70] M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *IEEE. Trans. Computers*, 1(22), 1973.

[71] Glenn Fung and Olvi L. Mangasarian. Proximal support vector machine classifiers. In *Proceedings KDD-2001: Knowledge Discovery and Data Mining*, pages 77–86.

[72] J. Gall, B. Rosenhahn, T. Brox, U. Kersting, and H.-P. Seidel. Learning for multi-view 3d tracking in the context of particle filters. *International Symposium on Visual Computing (ISVC), Springer-Verlag, LNCS 4292*, (1):59–69, 2006.

[73] A. Garg, S. Aggarwal, and P. Perona. Fusion of global and local information for object detection. *International Conference on Pattern Recognition*, 2002.

[74] D. Gavrila and L. Davis. 3-D Model Based Tracking of Humans in Action:A Multiview Approach. In *CVPR*, pages 73–80, 1996.

[75] D. M. Gavrila and L. S. Davis. 3-d model-based tracking of humans in action: a multi-view approach. *CVPR*, (1):73–80, 1996.

[76] K. Grauman, G. Shakhnarovich, and T. Darell. Inferring 3D structure with a statistical image-based shape model. In *ICCV*, 2003.

[77] Kristen Grauman and Trevor Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV*, 2005.

[78] A. Gupta, T. Chen, F. Chen, D. Kimber, and L. S. Davis. Context and observation driven latent variable model for human pose estimation. *CVPR*, 2008.

[79] David R. Hardoon, Sandor Szedmak, and John Shawe Taylor. Canonical correlation analysis an overview with application to learning methods. *Technical Report CSD-TR-03-02- University of London*, 2003.

[80] N. Hasler, B. Rosenhahn, T. Thorhlen, M. Wand, J. Gall, and H. Seidel. Markerless motion capture with unsynchronized moving cameras. *CVPR*, 2009.

[81] H. Hotelling. Relations between two sets of variants. *Biometrika*, 28(1):321–377, 1936.

[82] N. Howe, M. Leventon, and W. Freeman. Bayesian Reconstruction of 3D Human Motion from Single-Camera Video. *Advances in Neural Information Processing Systems*, 1999.

[83] M. Isard and A. Blake. CONDENSATION – Conditional Density Propagation for Visual Tracking. *IJCV*, 1998.

[84] M. Isard and A. Blake. A mixed-state condensation tracker with automatic model-switching. *ICCV*, pages 107–112, 1998.

[85] Gall J., Rosenhahn B., and Seidel H.-P. Drift-free tracking of rigid and articulated objects. *CVPR*, 2008.

[86] Gall J., Rosenhahn B., Brox T., and Seidel H.-P. Optimization and filtering for human motion capture: A multi-layer framework. *IJCV*, 2008.

[87] R. Jacobs, M. Jordan, S. Nowlan, and G. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3:79–87, 1991.

[88] T. Jaeggli, E. Koller-Meier, and L. Van Gool. Monocular tracking with a mixture of view-dependent learned models. In *IV Conference on Articulated Motion and Deformable Objects, AMDO*, pages 494–503, 2006.

[89] A. Jepson, D.Fleet, and T. El-Maraghi. Robust on-line appearance models for visual tracking. In *PAMI*, volume 25(10), pages 1296–1311, 2003.

[90] T. Joachims. Making large-scale svm learning practical, 1999. Advances in Kernel Methods - Support Vector Learning.

[91] Thorsten Joachims. Transductive inference for text classification using support vector machines. *International Conference on Machine Learning (ICML)*, 1999.

[92] M. Jordan and R. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6:181–214, 1994.

[93] I. Kakadiaris and D. Metaxas. Model-Based Estimation of 3D Human Motion with Occlusion Prediction Based on Active Multi-Viewpoint Selection. In *CVPR*, pages 81–87, 1996.

[94] I. Kakadiaris and D. Metaxas. Model-based estimation of 3d human motion. *PAMI*, (22):1453–1459, 2000.

[95] A. Kanaujia, C. Sminchisescu, and D. Metaxas. Semi-supervised hierarchical models for 3d human pose reconstruction. *CVPR*, 2007.

[96] R. Kehl, M. Bray, and L. V. Gool. Full body tracking from multiple views using stochastic sampling. In *CVPR*, 2005.

[97] Teresa Ko, Stefano Soatto, and Deborah Estrin. Background subtraction on distributions. *ECCV*, 2008.

[98] T. Kohonen. Self-organizing maps. 1995.

[99] B. H. Sigelman L. Sigal, M. Isard and M. J. Black. Attractive people: Assembling loose-limbed models using non-parametric belief propagation. *Advances in Neural Information Processing Systems*, 2003.

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

[101] N. Lawrence and J. Candela. Local distance preservation in the gplvm through back constraints. *International Conference on Machine Learning*, 2006.

[102] N. Lawrence, M. Seeger, and R. Herbrich. Fast sparse Gaussian process methods: the informative vector machine. In *Advances in Neural Information Processing Systems*, 2003.

[103] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[104] H. J. Lee and Z. Chen. Determination of 3D Human Body Postures from a Single View. *Computer Vision, Graphics and Image Processing*, 30:148–168, 1985.

[105] J. Lee, J. Chai, P. Reitsma, J. Hodgins, and N. Pollard. Interactive control of avatars animated with human motion data. *Proc. SIGGRAPH*, pages 491–500, 2002.

[106] M. Lee and I. Cohen. Proposal maps driven mcmc for estimating human body pose in static images. In *CVPR*, 2004.

[107] M. Lee and I. Cohen. Proposal maps driven MCMC for estimating human body pose in static images. In *CVPR*, 2004.

[108] B. Leibe, E. Seeman, and B. Schiele. Pedestrian detection in crowded scenes. In *CVPR*, 2005.

[109] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. *IEEE Conference on Computer Vision and Pattern Recognition*, 1:878–885, 2005.

[110] R. Li, M. Yang, S. Sclaroff, and T. Tian. Monocular Tracking of 3D Human Motion with a Coordianted Mixture of Factor Analyzers. In *ECCV*, 2006.

[111] R. Li, M.-H. Yang, S. Sclaroff, and T.-P. Tian. Monocular tracking of 3d human motion with a coordinated mixture of factor analyzers. *ECCV*, (1):137–150, 2006.

[112] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2), 2004.

[113] D. Mackay. Bayesian interpolation. *Neural Computation*, 4(5):720–736, 1992.

[114] R. Memisevic. Kernel Information Embeddings. In *International Conference on Machine Learning*, 2006.

[115] Ivana Mikic, Mohan Trivedi, Edward Hunter, and Pamela Cosman. Human body model acquisition and tracking using voxel data. *IJCV*, (53):199–223, 2003.

[116] K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. *ECCV*, 1, 2004.

[117] Kim Minyoung and Vladimir Pavlovic. Dimensionality reduction using covariance operator inverse regression. *CVPR*, 2008.

[118] A. Mohan, C. Papageogiou, and T. Poggio. Example-based object detection in images by components. *Transaction of PAMI*, 23:349–362, 2001.

[119] A. Monnet, A. Mittal, N. Paragios, and V. Ramesh. Background modeling and subtraction of dynamic scenes. *ICCV*, pages 1305–1312, 2003.

[120] Kooksang Moon and Vladimir Pavlovic. Impact of dynamics on subspace embedding and tracking of sequences. *CVPR*, (1):198–205, 2006.

[121] G. Mori and J. Malik. Estimating human body configurations using shape context matching. In *ECCV*, 2002.

[122] G. Mori and J. Malik. Efficient shape matching using shape contexts. *PAMI*, 27(11):1832–1837, 2005.

[123] R. Navaratnam, A. W. Fitzgibbon, and R. Cipolla. Semi-supervised learning of joint density models for human pose estimation. In *British Machine Vision Conference*, 2006.

[124] R. Neal. *Bayesian learning for neural networks*. Springer-Verlag, 1996.

[125] A. Ng and M. Jordan. On discriminative versus generative classifiers. A comparison of logistic regression and naive Bayes. In *Advances in Neural Information Processing Systems*, 2002.

[126] A. Ng, M. Jordan, and Y. Weiss. On Spectral Clustering: Analysis and an Algorithm. In *Advances in Neural Information Processing Systems*, 2001.

[127] D. Nistér and H. Stévenius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006.

[128] B. North and A. Blake. Learning Dynamical Models by Expectation Maximization. In *ICCV*, 1998.

[129] A. Opelt, M. Fussenegger, A. Pinz, and P. Auer. Weak hypothesis and boosting for generic object detections and recognition. *ECCV*, 1, 2004.

[130] C. Papageorgiou and T. Poggio. A trainable system for object detection. *IJCV*, 1, 2000.

[131] V. Pavlovic and J. Rehg. Imact of dynamic model learning on the classification of human motion. In *CVPR*, 2000.

[132] V. Pavlovic, J. Rehg, T. Cham, and K. Murphy. A Dynamic Bayesian Approach to Figure Tracking using Learned Dynamical Models. In *ICCV*, 2001.

[133] R. Quandt and J. Ramsey. A new approach to estimating switching regressions. *Journal of the American Statistical Society*, 67:306–310, 1972.

[134] A. Rahimi, B. Recht, and T. Darrell. Learning appearance manifolds from video. *CVPR*, pages 868–875, 2005.

[135] R. T. Rockafeller. Monotone operators and the proximal point algorithm. In *SIAM Journal on Control and Optimization*, 1976.

[136] R. Ronfard, C. Schmid, and W. Triggs. Learning to parse pictures of people. *In European Conference on Computer Vision*, 1, 2002.

[137] R. Rosales and S. Sclaroff. Inferring Body Pose without Tracking Body Parts. In *CVPR*, pages 721–727, 2000.

[138] R. Rosales and S. Sclaroff. Learning Body Pose Via Specialized Maps. In *Advances in Neural Information Processing Systems*, 2002.

[139] R. Rosales and S. Sclaroff. Combining generative and discriminative models in a framework for articulated pose estimation. In *IJCV*, 2003.

[140] B. Rosenhahn, T. Brox, U. Kersting, D. Smith, J. Gurney, and R. Klette. A system for marker-less human motion estimation. *Kuenstliche Intelligenz (KI)*, pages 45–51, 2006.

[141] S. Roth, L. Sigal, and M. Black. Gibbs Likelihoods for Bayesian Tracking. In *CVPR*, 2004.

[142] S. Roweis and L. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 2000.

[143] S. Roweis and L. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 2000.

[144] H. Rowley, S. Baluja, and T. Kanade. Neural network based face detection. *Trans. PAMI*, 20:23–38, 1998.

[145] P. Sabzmeydani and G. Mori. Detecting pedestrians by learning shapelet features. *CVPR*, 1, 2007.

[146] H. Schneiderman and T. Kanade. Object detection using statistics of parts. *IJCV*, 1:151–177, 2004.

[147] T. Serre, L Wolf, and T. Poggio. Object recognition with features inspired by visual cortex. In *CVPR*, pages 994–1000, Washington, DC, USA, 2005.

[148] G. Shakhnarovich, P. Viola, and T. Darrell. Fast Pose Estimation with Parameter Sensitive Hashing. In *ICCV*, 2003.

[149] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[150] Y. Sheikh and M. Shah. Bayesian modeling of dynamic scenes for object detection. *PAMI*, (27):1778–1792, 2005.

[151] H. Sidenbladh and M. Black. Learning Image Statistics for Bayesian Tracking. In *ICCV*, 2001.

[152] H. Sidenbladh, M. Black, and D. Fleet. Stochastic Tracking of 3D Human Figures Using 2D Image Motion. In *ECCV*, 2000.

[153] H. Sidenbladh, M. Black, and L. Sigal. Implicit Probabilistic Models of Human Motion for Synthesis and Tracking. In *ECCV*, 2002.

[154] L. Sigal, S. Bhatia, S. Roth, M. Black, and M. Isard. Tracking Loose-limbed People. In *CVPR*, 2004.

[155] L. Sigal and M. Black. Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion. Technical Report CS-06-08, Brown University, 2006.

[156] L. Sigal and M. Black. Predicting 3d people from 2d pictures. In *IV Conference on Articulated Motion and Deformable Objects, AMDO*, 2006.

[157] L. Sigal and M. J. Black. Predicting 3d people from 2d pictures. *IV Conference on Articulated Motion and Deformable Objects (AMDO)*, 2006.

[158] L. Sigal, R. Memisevic, and D. Fleet. Shared kernel information embedding for discriminative inference. *CVPR*, 2009.

[159] Leonid Sigal, A. Balan, and M. J. Black. Combined discriminative and generative articulated pose and non-rigid shape estimation. *Advances in Neural Information Processing Systems*, 2007.

[160] C. Sminchisescu. Consistency and Coupling in Human Model Likelihoods. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 27–32, Washington D.C., 2002.

[161] C. Sminchisescu and A. Jepson. Generative Modeling for Continuous Non-Linearly Embedded Visual Inference. In *International Conference on Machine Learning*, pages 759–766, Banff, 2004.

[162] C. Sminchisescu and A. Jepson. Generative modeling for continuous non-linearly embedded visual inference. *International Conference on Machine Learning*, 2004.

[163] C. Sminchisescu and A. Jepson. Variational Mixture Smoothing for Non-Linear Dynamical Systems. In *CVPR*, volume 2, pages 608–615, Washington D.C., 2004.

[164] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Discriminative Density Propagation for 3D Human Motion Estimation. In *CVPR*, volume 1, pages 390–397, 2005.

[165] C. Sminchisescu, A. Kanaujia, and D. Metaxas. Learning Joint Top-down and Bottom-up Processes for 3D Visual Inference. In *CVPR*, 2006.

[166] C. Sminchisescu, A. Kanaujia, and D. Metaxas. BM3E : Discriminative Density Propagation for Visual Tracking. *PAMI*, 2007.

[167] C. Sminchisescu, D. Metaxas, and S. Dickinson. Incremental Model-Based Estimation using Geometric Consistency Constraints. *PAMI*, 2005.

[168] C. Sminchisescu and B. Triggs. Covariance-Scaled Sampling for Monocular 3D Body Tracking. In *CVPR*, volume 1, pages 447–454, Hawaii, 2001.

[169] C. Sminchisescu and B. Triggs. Estimating Articulated Human Motion with Covariance Scaled Sampling. *International Journal of Robotics Research*, 22(6):371–393, 2003.

[170] C. Sminchisescu and B. Triggs. Kinematic Jump Processes for Monocular 3D Human Tracking. In *CVPR*, volume 1, pages 69–76, Madison, 2003.

[171] C. Sminchisescu and B. Triggs. Mapping Minima and Transitions in Visual Models. *IJCV*, 61(1), 2005.

[172] C. Sminchisescu and M. Welling. Generalized darting Monte Carlo. Technical Report CSRG-543, University of Toronto, October 2006.

[173] E. Sudderth, A. Ihler, W. Freeman, and A.Wilsky. Non-parametric belief propagation. In *CVPR*, 2003.

[174] C. J. Taylor. Reconstruction of Articulated Objects from Point Correspondences in a Single Uncalibrated Image. In *CVPR*, pages 677–684, 2000.

[175] J. Tenenbaum, V. Silva, and J. Langford. A Global Geometric Framewok for Nonlinear Dimensionality Reduction. *Science*, 2000.

[176] J. Tenenbaum, V. Silva, and J. Langford. A Global Geometric Framewok for Nonlinear Dimensionality Reduction. *Science*, 2000.

[177] T.F.Cootes and C.J.Taylor. Locating objects of varying shape using statistical feature detectors. In *Proc. European Conference on Computer Vision*, 1996.

[178] Y.L. Tian, M. Lu, and A. Hampapur. Robust and efficient foreground analysis for real-time video surveillance. *CVPR*, 2005.

[179] M. Tipping. Mixtures of probabilistic principal component analysers. *Neural Computation*, 1998.

[180] M. Tipping. Sparse Bayesian learning and the Relevance Vector Machine. *Journal of Machine Learning Research*, 2001.

[181] M. E. Tipping and Anita Faul. Fast marginal likelihood maximization for sparse bayesian models. In *Advances in NIPS 2002*, 2000.

[182] C. Tomasi, S. Petrov, and A. Sastry. 3d tracking = classification + interpolation. In *ICCV*, 2003.

[183] O. Tuzel, Fatih Porikli, and Peter Meer. Pedestrian detection via classification on riemannian manifold, 2008. Trans. Pattern Analysis and Machine Intelligence.

[184] N. Ueda and Z. Ghahramani. Bayesian model search for mixture models based on optimizing variational bounds. *Neural Networks*, 15:1223–1241, 2002.

[185] R. Urtasun, D. Fleet, A. Hertzmann, and P. Fua. Priors for people tracking in small training sets. In *ICCV*, 2005.

[186] Raquel Urtasun and Trevor Darrell. Sparse probabilistic regression for activity-independent human pose inference. *CVPR*, 2008.

[187] Raquel Urtasun, David J. Fleet, and Pascal Fua. 3d people tracking with gaussian process dynamical models. *CVPR*, (1):238–245, 2006.

[188] V. Vapnik. Statistical learning theory. *Wiley-Interscience*, 1998.

[189] P. Viola and M. Jones. Rapid object detection using boosted cascade of simple features. *IEEE Conference on Computer Vision and Pattern Recognition*, 1:511–518, 2001.

[190] M. Vondrak, L. Sigal, and D. Fleet. Physical simulation for probabilistic motion tracking. *CVPR*, 2008.

[191] S. Wachter and H. Nagel. Tracking Persons in Monocular Image Sequences. *CVIU*, 74(3):174–192, 1999.

[192] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroed. Constrained kmeans clustering with background knowledge. In *ICML*, 2001.

[193] Q. Wang, G. Xu, and H. Ai. Learning Object Intrinsic Structure for Robust Visual Tracking. In *CVPR*, 2003.

[194] S. Waterhouse, D.Mackay, and T.Robinson. Bayesian methods for mixtures of experts. In *Advances in Neural Information Processing Systems*, 1996.

[195] K. Weinberger and L. Saul. Unsupervised Learning of Image Manifolds by Semidefinite Programming. In *CVPR*, 2004.

[196] J. Weston, O. Chapelle, A. Elisseeff, B. Scholkopf, and V. Vapnik. Kernel dependency estimation. In *Advances in Neural Information Processing Systems*, 2002.

[197] J. Weston, B. Schölkopf, O. Bousquet, T. Mann, and W. Noble. Joint kernel maps. Technical Report 131, Max Planck Institute, November 2004.

[198] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems*, 2002.

[199] L. Xu, M. Jordan, and G. Hinton. An alternative model for mixture of experts. In *Advances in Neural Information Processing Systems*, 1995.

[200] J. Zhang, M. Marszaek, S. Lazebnik, and C. Schmid. Local Features and Kernels for Classifcation of Texture and Object Categories: A Comprehensive Study. *Beyond Patches Workshop*, 2006.

[201] Zhenyue Zhang and Hongyuan Zha. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM J. Scientific Computing*, (1):313–338, 2004.

[202] Q. Zhu, S. Avidan, M. C. Yeh, and K. T. Cheng. Fast human detection using a cascade of histograms of oriented gradients. *IEEE Conference on Computer Vision and Pattern Recognition*, 2:1491–1498, 2006.

[203] Xiaojin Zhu Zhuxj, Zoubin Ghahramani, and John Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *In ICML*, pages 912–919, 2003.

Vita

Atul Kanaujia

2000 B. Tech., Computer Science, Indian Institute of Technology, Bombay

2003 M.S., Computer Science, Rutgers - The State University of New Jersey, New Brunswick, NJ, USA

2010 Ph.D., Computer Science, Rutgers - The State University of New Jersey, New Brunswick, NJ, USA

2000-2003 Teaching Assistant, Division of Computer and Information Sciences, Rutgers - The State University of New Jersey, New Brunswick, NJ, USA

2003-2004 Associate Member of Technical Staff, Mentor Graphics R&D, Hyderabad, India.

2004-2008 Research Assistant, Division of Computer and Information Sciences, Rutgers - The State University of New Jersey, New Brunswick, NJ, USA

2008-Current Research Scientist, ObjectVideo Research Labs, Inc. Reston, VA, USA

Publications

Journal Papers:

BM^3E: Discriminative Density Propagation for Visual Tracking, C. Sminchisescu, Atul Kanaujia, D. Metaxas, *IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007*

Conditional models for contextual human motion recognition, C. Sminchisescu, Atul Kanaujia, Dimitris Metaxas, *Computer Vision and Image Understanding, 2006,*

Selected Conference Papers:

Fast algorithms for large scale conditional 3D prediction, Liefeng Bo, Cristian Sminchisescu, Atul Kanaujia, Dimitris N. Metaxas, *In Proc. Computer Vision and Pattern Recognition 2008*

Latent Variable Models for Perceptual Inference, Atul Kanaujia, Cristian Sminchisescu, Dimitris N. Metaxas, *In Proc. International Conference on Computer Vision 2007*

The Best of Both Worlds: Combining 3D Deformable Models with Active Shape Models, C. Vogler, Z. Li, Atul Kanaujia, S. Goldenstein, D. Metaxas, *In Proc. International Conference on Computer Vision 2007*

Semi-supervised Hierarchical Models for 3D Human Pose Reconstruction, Atul Kanaujia, Cristian Sminchisescu, Dimitris N. Metaxas, *In Proc. Computer Vision and Pattern Recognition 2007*

Learning Joint Top-Down and Bottom-up Processes for 3D Visual Inference, Cristian Sminchisescu, Atul Kanaujia, Dimitris N. Metaxas, *In Proc. Computer Vision and Pattern Recognition 2006*

Discriminative Density Propagation for 3D Human Motion Estimation, Cristian Sminchisescu, Atul Kanaujia, Zhiguo Li, Dimitris N. Metaxas, *In Proc. Computer Vision and Pattern Recognition 2005*

Conditional Random Fields for Contextual Human Motion Recognition, Cristian Sminchisescu, Atul Kanaujia, Zhiguo Li, Dimitris N. Metaxas, *In Proc. International Conference on Computer Vision 2005*

Conditional Visual Tracking in Kernel Space, Cristian Sminchisescu, Atul Kanaujia,Zhiguo Li, Dimitris N. Metaxas, *In Proc. Neural Information Processing Systems 2005*

Other Conference and Workshop Papers

Large Scale Learning of Active Shape Models, Atul Kanaujia and Dimitris Metaxas,*In Proc. International Conference on Image Processing 2007*

Emblem Detections by Tracking Facial Features, Atul Kanaujia, Y. Huang, Dimitris Metaxas, *In International Workshop on Semantic Learning Applications in Multimedia (SLAM) in association with CVPR 2006*

Tracking Facial Features Using Mixture of Point Distribution Models, Atul Kanaujia, Yuchi Huang, Dimitris Metaxas, *ICVGIP 2006*

Recognizing Facial Expressions by Tracking Feature Shapes, Atul Kanaujia, Dimitris N. Metaxas, *In Proc. International Conference on Pattern Recognition 2006*

Learning Ambiguities Using Bayesian Mixture of Experts, Atul Kanaujia, Dimitris Metaxas, *In Proc. International Conference on Tools with Artificial Intelligence 2006*

Learning Multi-category Classification in Bayesian Framework, Atul Kanaujia, Dimitris N. Metaxas, *In Proc. Asian Conference on Computer Vision 2006*