ABSTRACT

Title of Dissertation:	Change Detection in Stochastic Shape Dynamical Models with Applications in Activity Modeling and Abnormality Detection
	Namrata Vaswani, Doctor of Philosophy, 2004
Dissertation directed by:	Professor Rama Chellappa Department of Electrical and Computer Engineering

The goal of this research is to model an "activity" performed by a group of moving and interacting objects (which can be people or cars or robots or different rigid components of the human body) and use these models for abnormal activity detection, tracking and segmentation. Previous approaches to modeling group activity include co-occurrence statistics (individual and joint histograms) and Dynamic Bayesian Networks, neither of which is applicable when the number of interacting objects is large. We treat the objects as point objects (referred to as "landmarks") and propose to model their changing configuration as a moving and deforming "shape" using ideas from Kendall's shape theory for discrete landmarks. A continuous state HMM is defined for landmark shape dynamics in an "activity". The configuration of landmarks at a given time forms the observation vector and the corresponding shape and scaled Euclidean motion parameters form the hidden state vector. The dynamical model for shape is a linear Gauss-Markov model on shape "velocity". The "shape velocity" at a point on the shape manifold is defined in the tangent space to the manifold at that point. Particle filters are used to track the HMM, i.e. estimate the hidden state given observations.

An abnormal activity is defined as a change in the shape activity model, which could be slow or drastic and whose parameters are unknown. Drastic changes can be easily detected using the increase in tracking error or the negative log of the likelihood of current observation given past (OL). But slow changes usually get missed. We have proposed a statistic for slow change detection called ELL (which is the Expectation of negative Log Likelihood of state given past observations) and shown analytically and experimentally the complementary behavior of ELL and OL for slow and drastic changes. We have established the stability (monotonic decrease) of the errors in approximating the ELL for changed observations using a particle filter that is optimal for the unchanged system. Asymptotic stability is shown under stronger assumptions. Finally, it is shown that the upper bound on ELL error is an increasing function of the "rate of change" with increasing derivatives of all orders, and its implications are discussed.

Another contribution of the thesis is a linear subspace algorithm for pattern classification, which we call Principal Components' Null Space Analysis (PCNSA). PCNSA was motivated by Principal Components' Analysis (PCA) and it approximates the optimal Bayes classifier for Gaussian distributions with unequal covariance matrices. We have derived classification error probability expressions for PCNSA and compared its performance with that of subspace Linear Discriminant Analysis (LDA) both analytically and experimentally. Applications to abnormal activity detection, human action retrieval, object/face recognition are discussed.

Change Detection in Stochastic Shape Dynamical Models with Applications in Activity Modeling and Abnormality Detection

by

Namrata Vaswani

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2004

Advisory Committee:

Professor Rama Chellappa, Chairman
Professor P.S. Krishnaprasad
Professor Adrian Papamarcou
Professor Prakash Narayan
Professor Larry Davis

DEDICATION

To my mother

ACKNOWLEDGEMENTS

I would like express my gratitude towards my advisor, Prof. Rama Chellappa for his help and support and encouragement. Also, I would like to thank Prof. Adrian Papamarcou, Prof. Andre Tits and Prof. Eric Slud for their suggestions and comments on my research.

I would like to thank all my ECE department and CFAR colleagues, in particular, Dr. Amit Roy Chowdhury, Fumin Zhang and Dr. Vinay Chande for their suggestions and comments on my work. Also I would like to thank Dr Amit Kale, Dr Kevin Zhou, Naresh Cuntoor, Sudha Dhoorjaty, Gunjan Dang, Vikrant Lal, Sowmya Deverajan, and all members of the Association for India's Development for their friendship and support.

I would like to thank my husband, Aditya, who listened to a lot of my ideas and also read and critiqued all the papers I wrote. Besides the academic help, I would like to thank Aditya also for being a continuous source of emotional support all through my Ph.D.

Words are not enough to express my gratitude towards my mother. She has been my inspiration always. Her Ph.D. thesis and long list of publications first motivated me many years ago to want to get a Ph.D.

TABLE OF CONTENTS

List of Tables **vii**

List of Figures **viii**

1 Introduction **1**

1.1 Organization of the Thesis . . . 3

1.2 Main Ideas . . . 4

1.2.1 Change Detection in Continuous State HMMs . . . 4

1.2.2 Landmark Shape Dynamics . . . 6

1.2.3 Application to Group Activity Modeling, Abnormal Activity Detection and Segmentation . . . 7

1.2.4 Principal Component Null Space Analysis (PCNSA) . . . 9

1.3 Related Work . . . 11

1.3.1 Change Detection . . . 11

1.3.2 Landmark Shape Dynamics . . . 13

1.3.3 Activity Recognition and Tracking . . . 15

1.3.4 Principal Component Null Space Analysis (PCNSA) . . . 17

2 Change Detection in General HMMs **19**

2.1 Problem Formulation . . . 19

2.1.1 The General HMM Model . . . 19

2.1.2 Problem Definition . . . 20

2.1.3 The Approach . . . 20

2.2 Preliminaries and Notation . . . 22

2.2.1 Notation and Definitions . . . 22

2.2.2 Approximate Non-linear Filtering Using a Particle Filter . . . 25

2.3 Change Detection Statistics . . . 26

2.3.1 The ELL statistic . . . 26

2.3.2 When does ELL work: A Kerridge Inaccuracy perspective . . . 28

2.3.3 When ELL fails: The OL Statistic . . . 29

2.3.4 Practical Issues . . . 30

2.4 Errors in ELL Approximation . . . 32

2.5 Effect of Increasing Rate of Change on Approximation Errors . . . 39

2.6 Complementary behavior of ELL and OL 46
2.7 Discussion 48
2.7.1 An Example 48
2.7.2 Sufficient Conditions for Mixing Unnormalized Filter Kernels 53
2.7.3 Generalizations 54
2.7.4 Implications of “rate of change” bound on error 55
2.7.5 OL and Tracking Error 58
2.8 Simulation Results 59

3 Landmark Shape Dynamics 63
3.1 Problem Formulation 63
3.2 Preliminaries and Notation 64
3.3 Stationary Shape Activity 68
3.3.1 Shape Deformation Model in Tangent Space 68
3.3.2 Partially Observed (Hidden) Shape Dynamics 71
3.4 Non-Stationary Shape Dynamics 74
3.5 Piecewise Stationary Shape Dynamics 77
3.6 Particle Filtering and Extended Kalman Filtering 79
3.7 Time-Varying Number of Landmarks 80

4 Applications to Abnormal Activity Detection, Tracking and Segmentation 84
4.1 Abnormal Activity Detection 84
4.1.1 Stationary Shape Activity: Fully Observed Case 85
4.1.2 Stationary Shape Activity: Partially Observed Case 86
4.1.3 Nonstationary Shape Activity: Abnormality Detection 87
4.2 Tracking to Obtain Observations 89
4.3 Activity Sequence Identification and Tracking 91
4.4 Shape Activity Sequence Segmentation 92
4.5 Experimental and Simulation Results 92
4.5.1 Dataset and Experiments 93
4.5.2 Group Activity: Abnormality Detection 95
4.5.3 Group Activity: Tracks 98
4.5.4 Group Activity: Nonstationary Shape Activity (NSSA) Model 99
4.5.5 Simulated Shape Sequence: Comparing NSSA and SSA 99
4.5.6 Human Actions: Comparing NSSA and SSA 100

5 Principal Component Null Space Analysis (PCNSA) 108
5.1 Introduction 108
5.1.1 Problem Statement 110
5.2 Principal Component Null Space Analysis 111
5.2.1 Assumptions 111
5.2.2 Algorithm 112
5.3 Classification Error Probability 113

5.3.1 One-dimensional ANS per class 113
5.3.2 $\mathbf{M_i}$-dimensional ANS per class 115
5.3.3 Extension to Non-Gaussian Distributions 117
5.4 Comparison with Subspace Linear Discriminant Analysis (SLDA) 118
5.4.1 Subspace Linear Discriminant Analysis (SLDA) 118
5.4.2 Classification Performance Comparison 118
5.4.3 Comparing Size of Training Data 123
5.4.4 Comparing 'New' (Untrained) Class Detection Ability 124
5.5 Relation to Multispace KL 124
5.6 Experiments and Results 125
5.6.1 Synthetic Data Verification 126
5.6.2 New Class Detection and Progressive-PCNSA 127
5.6.3 Image Classification Experiments 129
5.6.4 Video Classification/Retrieval Experiments 132

6 Summary and Future Directions 148
6.1 Summary 148
6.2 Future Directions 151
6.3 Contributions 152

7 Appendix: Proofs of Chapter 2 154

LIST OF TABLES

5.1 Object Recognition results: (a) shows results for training and testing on data from 20 trained classes (no new). (b) shows results for training 3 object classes and using data from all 20 for testing. Correct new class detection probability is $1 - P(\text{Miss})$. For (a), a 50 dimensional PCA space was used while for (b) a 20 dimensional PCA space was used. In (a), 15 LDA directions were used and ANS dimension was varied between 4 and 9 for progressive-PCNSA while in (b) 2 LDA dimensions were used and ANS dimension varied between 3 and 6. 143

5.2 Facial Feature Matching results: (a) shows results for training and testing on data from the 7 trained classes (no new). For (a) a 50 dimensional PCA space was used while for (b) a 20 dimensional PCA space was used. In (a), 6 LDA directions were used and ANS dimension was varied between 3 and 9 for progressive-PCNSA while in (b) 2 LDA dimensions were used and ANS dimension varied between 3 and 6. 144

5.3 Face Recognition Results (UMIST database): (a) shows results for training and testing on data from 15 trained classes (no new). (b) shows results for training 3 classes and using data from all 15 for testing. The false alarm and miss probabilities of LDA in (b) are skewed because new class detection thresholds were kept constant at the same value for all the three applications (not optimized for each application separately). PCNSA new class detection is not sensitive to the type of application (numerical values of class variances) and hence performs equally good new class detection for all applications. 145

5.4 Face Recognition Results (Yale database): (a) shows results for training and testing on data from 15 trained classes (no new). (b) shows results for training 3 classes and using data from all 15 for testing. 146

5.5 Retrieving actions using PCNSA in tangent to shape space: The distances of the query sequences (top row) in ANS space of tangent to mean shape (equation (5.38)) of each of the 3 database sequences (in leftmost column) are shown. The underlined distance in each column corresponds to the closest match to query. 147

LIST OF FIGURES

2.1 Simulated example: In (a) and (b), we show ELL and OL (negative log of observation likelihood) plots for the no change case (blue -o), and for changes with $\Delta a = r\sigma_{noise}$ for r=1 (red-*), r=2 (green -△) and r=5 (black -*square*). In all cases change was introduced at time $t_c = 5$ and lasted till $t_f = 15$. The plots are averaged over 20 realizations of observation sequences. For the case $r = 5$ (drastic change) and $r = 2$, the OL plot goes to infinity at or after $t = 5$ (computer overflow) and hence the change is detected immediately using OL while ELL completely fails for it. In (c), we plot $0.5a_t^2/\sigma_t^2$, its simulation average calculated using 20 realizations of the observation sequence and its spread (average plus and minus the standard deviation) for a change with $\Delta a = \sigma_{noise}$. We also plot the theoretical bound on the standard deviation obtained in The example given in Section 2.7.1. 61

2.2 ROC curves for comparing performance of ELL and OL for slow and drastic changes 61

2.3 Effect of increasing the system noise variance of the particle filter on performance of ELL 62

3.1 SSA & NSSA on the shape manifold which is depicted using a circle ($\mathcal{M}$), instead of a complex $\mathcal{C}^{k-1}$ sphere. In (a), we show a sequence of shapes from a SSA; at all times the shapes are close to the mean shape and hence the dynamics can be approximated in T_μ (tangent space at μ). In (b), we show a sequence of shapes from an NSSA, the shapes move on the shape manifold and hence we need to define a new tangent space at every time instant. 83

4.1 Airport example: Passengers deplaning 101

4.2 ELL, Tracking error (TE) and Observation Likelihood (OL) plots: Simulated Observation noise, $\sigma_{obs}^2 = 9$ (3-pixel noise). TE and OL plots look alike because of the reasons discussed in section 2.7.5 In the rest of the experiments, we show only the TE plots. 101

4.3 ELL and Tracking error plots: Real Observations. Abnormality was introduced at $t = 5$. The ELL is able to detect slow changes better while the tracking error works better for drastic changes. The plots are discussed in Section 4.5.2. 102

4.4 ROCs for Change detection using ELL. Blue circles, red stars, majenta triangles and cyan diamonds plots are for $\sigma_{obs}^2 = 3, 9, 27, 81$ respectively. Note that the two plots have different y axis ranges. The ELL completely fails for drastic changes. Detection delays in (b) are very large (60 time units) while for the slow change maximum detection delay is only 7 time units. Plots are discussed in Section 4.5.2. 102

4.5 ROCs for Change detection using Tracking error. Blue circles, red stars, majenta triangles and cyan diamonds plots are for $\sigma_{obs}^2 = 3, 9, 27, 81$ respectively. Please note that the two plots have different y axis ranges. Tracking error does not detect slow changes easily. Detection delays in (a) are large (maximum delay is 28 time units) while drastic changes are detected almost immediately with delay ≤ 4 time units. Plots are discussed in Section 4.5.2. 103

4.6 ROCs for Change detection using combined ELL-Tracking error. In this case, for each observation noise variance, there are multiple curves, since one needs to vary thresholds for both ELL and tracking error to get the ROC. A single curve is for the ELL threshold fixed and tracking error threshold varying. We have a set of curves for varying ELL thresholds. The maximum detection delay is 2 and 3 time units for $\sigma_{obs}^2 = 3$ ((a) and (b)), and 7 and 4 time units for $\sigma_{obs}^2 = 81$ ((c) and (d)). Plots are discussed in Section 4.5.2. 104

4.7 ELL plot for Temporal abnormality detection. Abnormality was introduced at $t = 5$. The plot is discussed in Section 4.5.2. 105

4.8 Tracks: Real Observations. Plotting the observed and tracked positions of the landmarks (passengers) on the x-y plane. The plots are discussed in Section 4.5.3. 105

4.9 Tracking and detecting slow & drastic abnormalities (introduced at $t = 5$): Comparing NSSA and SSA 105

4.10 Simulated shape: Tracking normal and abnormal behavior (introduced at $t = 40$) using SSA and NSSA. NSSA tracks abnormality as well. 106

4.11 Simulated Shape Statistics: Abnormality introduced at $t = 40$. NSSA detects abnormality using ELL (since it is able to track) while SSA detects using tracking error (loses track). 106

4.12 Tracking the figure skater: The green triangles line is the observed (noisy) data, the cyan -+ line is the ground truth, the blue circles and red stars are filtered shape using NSSA and SSA respectively. Abnormality introduced at $t = 20$. 107

4.13 Tracking the figure skater: Abnormality introduced at $t = 20$. NSSA detects abnormality using ELL while SSA detects using the tracking error. 107

5.1 θ is the angle between the line AB and the Y-axis in (a) and between AB and the X-axis in (b). **Case 1** with ANS directions (Y-axis) of both classes coinciding is shown in (a). **Case 2** is shown in (b). Y axis is ANS for class 1 & maximum variance direction for class 2, vice versa for X axis. 137

5.2 Average probability of error as a function of θ for different values of condition number R for (a) Case 1 (b) Case 2. As can be seen the LDA error probability does not vary much with R in either case (curves for all R values are coincident) and also does not degrade much as $\theta \rightarrow 90^o$. 138
5.3 Error probability variation with reduced training data sizes per class 139
5.4 (a) Object recognition classes (b) Facial Feature Matching classes 140
5.5 Face recognition databases 141
5.6 Detecting Abnormal Activities: The blue solid and dotted plots in both figures are the activity metric for a normal activity as a function of time. The red -o plot in (a) is for a temporal abnormality introduced at $t = 5$ and the green -* plot in (b) is for a spatial abnormality introduced at $t = 5$. 142

Chapter 1

Introduction

The goal of this research is to model an “activity” performed by a group of moving and interacting objects (which can be people or cars or robots or different rigid components of the human body) and use these models for abnormal activity detection, tracking and segmentation. We treat the objects as point objects (referred to as “landmarks” in shape theory literature). We model the changing configuration of objects as a moving and deforming “shape”. A stochastic shape dynamical model is defined to represent a “normal activity”. Abnormal activity is defined as a change in the shape dynamics learnt for the normal activity.

Previous approaches to modeling activity performed by groups of point objects include co-occurrence statistics (e.g. [1]) and discrete state Dynamic Bayesian Networks (DBNs) (e.g. [2]). Co-occurrence statistics involves learning individual and joint histograms of the objects. Joint histograms for modeling interactions is feasible only when the number of interacting objects is small. Stochastic shape dynamics on the other hand implicitly models interactions and independent motion of a group of objects of any size i.e. it is scalable in the number of interacting objects. DBNs define high level relations between different events and typically use heuristics for event detection. Our algorithms can be used to provide a more principled strategy for event detection. Another advantage of our framework is that using shape and its dynamics makes the representation invariant to translation, in-plane rotation or sensor zoom. The idea of using "shape" to model activities performed by groups of moving objects is similar to recent work in literature on controlling formations of groups of robots using shape (e.g. [3]).

We define a continuous state hidden Markov model (HMM)[1] for modeling the dynamics of the changing configuration of landmarks in an activity. The observed configuration of landmarks at a given time forms the observation vector and the corresponding shape and motion parameters form the hidden state vector. The HMM is nonlinear and hence we use a particle filter [4] to track the state (estimate the hidden state given the noisy observations). An abnormal activity is defined as a change in the shape dynamics, which could be slow or drastic and whose parameters are unknown. The problem of abnormal activity detection motivated our research on slow and drastic change detection in continuous state HMMs.

[1] A continuous state HMM is a partially observed state space model with a continuous state vector $\{X_t\}$ and a continuous observation vector $\{Y_t\}$ (In our work, the time instants t are discrete (but they can be continuous in general). Y_t is a noisy linear or nonlinear function of X_t and $\{X_t\}$ is a Markov process

1.1 Organization of the Thesis

The thesis is organized into four parts, each forming one chapter. In the first part (chapter 2), we propose statistics for slow and drastic change detection in continuous state HMMs and study the effect of approximation errors in estimating the statistics using a particle filter optimal for the unchanged system. In the second part (chapter 3), we propose shape dynamical models for landmarks shapes (stationary, non-stationary and piecewise stationary shape models). The third part of the thesis (chapter 4) is an application of the first two parts (shape dynamical models and change detection statistics) to represent activities performed by groups of moving and interacting objects and detecting abnormal activities. We also discuss extensions to activity segmentation and tracking. In the last part of the thesis (chapter 5), we discuss Principal Component Null Space Analysis (PCNSA) which is a pattern classification algorithm motivated by PCA, evaluate its classification error probability and compare its performance with LDA. We present applications of this algorithm to abnormal activity detection, action retrieval and to face and object recognition. Finally in chapter 6, we summarize the entire thesis, discuss future directions and list the contributions of the thesis. We now introduce each of the four parts and the main ideas and then discuss related work.

1.2 Main Ideas

1.2.1 Change Detection in Continuous State HMMs

The problem of abnormal activity detection explained above motivated our research on slow and drastic change detection in continuous state HMMs when change parameters are unknown. Drastic changes can be detected easily using the increase in tracking error or the negative log of observation likelihood (OL). But slow changes usually get missed. We use a particle filter (PF) to estimate the posterior probability distribution of the state at time t (X_t) given observations up to t $(Y_{1:t})$, $Pr(X_t \in dx|Y_{1:t}) \overset{\triangle}{=} \pi_t(dx)$. We propose here a statistic called ELL (which stands for Expected Log-Likelihood) which is able to detect slow changes. ELL is the conditional expectation of the negative log-likelihood of the state at time t $([-\log p_t(X_t)])$, given past observations, $Y_{1:t}$. It is evaluated as the expectation under π_t of $[-\log p_t(X_t)]$.

Now, the PF is optimal for the unchanged system and hence when estimating π_t for the changed system, there is modeling error. Also the particle filtering error (error due to finite number of Monte Carlo samples or particles) is much larger. But using stability results from [5], we are able to show that the approximation errors are eventually monotonically decreasing (and hence stable) with time for large enough number of particles (in section 2.4). We also show asymptotic stability under stronger assumptions. We show in section 2.5, that the bound on the error is proportional to the rate of change. Thus for slow changes, the estimation error in π_t is small i.e. ELL is approximated correctly for such changes. Hence the approximate value of ELL detects the slow change as soon as it becomes “detectable” (defined in Definition 5 of section 2.3.2). ELL fails to detect drastic changes because of large estimation error in evaluating π_t. But large estimation error in evaluating π_t also corresponds to a large value of OL (or tracking error) which can be used for detecting such changes. We discuss this in Section 2.6.

It is easy to see that ELL is equivalent to the Kerridge Inaccuracy [6] between the posterior and prior state distributions. Averaging the log likelihood over a time sequence of i.i.d. observations is often used in hypothesis testing and in [7] it is shown to be equivalent to the Kerridge Inaccuracy between the empirical distribution of the i.i.d. observations and their actual pdf. But to the best of our knowledge, ELL defined as the expectation of log likelihood of state given past observations, in the context of Hidden Markov Models (and its estimation using a PF) has not been used before.

ELL detects a slow change before the PF loses track. This is useful in any target(s) tracking problem where the target(s)’ dynamics might change over time. If one can detect the change, one can learn its parameters on the fly and use the changed system model (or atleast increase the system noise variance to track the change), without losing track of the target(s). We have used ELL to detect changes in landmark shape dynamical models (defined in chapter 3) and this has applications in abnormal activity detection, medical image processing (detecting motion disorders by tracking patients’ body parts) and activity segmentation (segmenting a long activity sequence into piecewise stationary elementary activities). This is discussed in chapter 4. Other applications of ELL are in neural signal processing (detecting changes in response of animals' brains to changes in stimuli provided to them) and medical signal processing (detecting slow changes in disease progression). ELL can also potentially be used for network congestion detection since congestion quite often starts as a slow change.

1.2.2 Landmark Shape Dynamics

We develop models for the configuration (shape+scaled Euclidean motion) dynamics of a group of moving landmarks (here point objects) in shape space. The shape of a group of discrete points (known as 'landmarks') is defined by Kendall [8] as all the geometric information that remains when location, scale and rotational effects (we refer to these as *motion* parameters) are filtered out. The original vector of landmark locations is known as the "configuration" vector. The book by Dryden and Mardia on statistical shape analysis [9] provides a good overview of the literature in this field. Statistical shape theory began in the late 1970s and has evolved into viable statistical approaches for modeling the shape of a single object with applications in object recognition and matching. In this work, we extend the static approaches to defining dynamical models for landmark shape. Also, we use these models for the dynamics of shape formed by a group of objects.

For a dataset of similar shapes, the shape variability can be modeled in the tangent hyperplane to the shape space at the mean. The tangent hyperplane is a linearized version of the shape space linearized at a particular point known as the pole of tangent projection. Typically one uses the Procrustes mean [9] of the dataset as the pole. The tangent plane is a vector space and hence techniques from linear multivariate statistics can be used to model shape variability in tangent space.

We use the term *"shape activity"* to denote a continuous state HMM for shape deformation and scaled Euclidean motion in the activity. A "stationary shape activity" is defined as one for which the shape vector is stationary i.e. the expected value of shape (mean shape) remains constant with time and the shape deformation model is stationary, while for a "non-stationary shape activity", the mean shape is time-varying (see figures 3.1(a) and (b)). For a stationary shape activity, the dynamics on the shape manifold is approximated by linear Gauss-Markov dynamics in a single tangent space at the mean shape. On the other hand, for a non-stationary shape activity, the shape moves on the shape manifold[2]. Dynamics is defined by a linear Gauss-Markov model on the shape "velocity" (time derivative of shape). The "velocity" at a point on a manifold is defined in the tangent space to the manifold at that point.

1.2.3 Application to Group Activity Modeling, Abnormal Activity Detection and Segmentation

The "shape activity" is a generic framework that can be used to model the dynamics of moving configurations in many applications depending on what is treated as the landmark. The "landmark" can be a person or a vehicle (in general any moving object) and one can learn a shape dynamical model for an activity performed by a group of moving people or

[2] Note here this motion of the shape vector on the shape manifold should not be confused with scaled Euclidean motion of the shape to obtain a configuration

model the traffic flow and use it to detect abnormal (suspicious) behavior. We have modeled the normal activity of passengers deplaning and moving towards the airport terminal (see figure 4.1) [10]. We use a stationary shape activity (SSA) model for this case. SSA is good for accurately modeling normal behavior and detecting abnormality when the mean shape does not change much. It is very specific to the learnt activity and hence less robust to model error and unable to track abnormality (except very slow ones). Abnormal activity can be a slow or drastic change in the shape dynamics and hence we use a combination of ELL and tracking error to detect it.

The landmarks could be the various parts of a human body (see figure 4.12(a)). Our framework (nonstationary shape activity models required in this case) can be used to learn models for the actions and detect and track abnormality in the action. This ability can be useful to medical professionals trying to analyze motion disorders in their patients. It would be useful, if software can detect the disorder and also provide its tracks to the medical professional. We use a nonstationary shape activity (NSSA) model for this application since it can track unmodeled shape changes. It is thus able to track a large class of abnormal activities and yet detect them using ELL.

Also for slowly varying shapes, we define a piecewise stationary shape activity model for which the mean shape is assumed to be piecewise constant. PSSA can be used in conjunction with ELL for activity segmentation (segmenting a long activity sequence into a sequence of stationary elementary activities). We discuss this in Section 4.4. Our approach is sensor independent; the landmark observations could be obtained by tracking moving objects in low resolution video or using radar sensors for vehicles or acoustic or infra-red sensors; and only the observation model changes.

1.2.4 Principal Component Null Space Analysis (PCNSA)

Another contribution of this thesis is a classification algorithm, PCNSA, which approximates the optimal Bayes classifier for Gaussian class conditional distributions with unequal covariance matrices. The abnormal activity detection problem(described above) can be viewed as a sequential hypothesis testing problem. For abnormality detection in the fully observed case, i.e. when the observation noise is negligible, we use the log-likelihood of the state to detect abnormality (discussed in 4.1.1). PCNSA, which approximates the optimal LRT for Gaussian distributions can also be used in this case and as discussed in section 5.6.4 (more detailed discussion in [11]), it has certain advantages. PCNSA can also be used for data retrieval. We show its application to human action retrieval in section 5.6.4.

The PCNSA algorithm was originally proposed by us for "apples from oranges" type classification problems like object recognition or face recognition under large pose variations. During the last several years much progress has been made towards recognizing faces under small variations in lighting and pose, for a detailed survey see [12], [13]. But reliable techniques for more extreme variations and for the more difficult image classification problems like object recognition have proved elusive. Problems like face recognition under small pose variations that involve discriminating similar objects can be categorized as "apples from apples" type classification problems. More precisely, "apples from apples" type problems are those in which different classes have similar class covariance matrices (in particular similar directions of low and high intra-class variance) while for “apples from oranges” type problems, different classes can have very different class covariance matrix structures. As an extreme case, the minimum variance direction of one class could be a maximum variance direction for another. We propose a linear classifier for this situation of unequal covariance matrices, which actually approximates the optimal Bayesian solution.

We have evaluated bounds on PCNSA's classification error probability (in Section 5.3) and discussed conditions under which it would outperform Linear Discriminant Analysis (LDA) and when it would fail (in Section 5.4). Applications of PCNSA to object recognition (figure 5.4(a)), feature matching (see figure 5.4(b)), face recognition under large pose/expression variations (see figure 5.5), abnormal group activity detection [11] (see figure 4.1) and video retrieval are discussed in Section 5.6. Feature matching is required for image registration [14] which is a first step for any baseline stereo or structure from motion (SfM) algorithm. Occlusion and new feature detection is an important issue in feature matching for SfM algorithms and as has been demonstrated in Section 5.6, the PCNSA algorithm has a very good ‘new’ (untrained) class detection ability. Also, since PCNSA defines a class specific metric, it is suitable for abnormality detection problems where only the “normal” class is characterized. In fact abnormality detection is the most extreme example of an “apples from oranges” type problem, since in this case the abnormal activity is not characterized at all. Finally, we show the application of PCNSA for retrieving human action videos from a database. Some other problems to which PCNSA can be applied are image retrieval; character recognition and distinguishing moving vehicles from people in low resolution images.

1.3 Related Work

1.3.1 Change Detection

Online detection of changes for partially observed *linear* dynamical systems has been studied extensively. For *known changed system parameters*, the CUSUM (cumulative sum) [15] algorithm can be used directly. The CUSUM algorithm uses as change detection statistic, the maximum (taken over all previous time instants, j) of the likelihood ratio assuming that the change occurred at time j, i.e. $CUSUM_t \overset{\triangle}{=} \max_{1 \leq j \leq t} LR(j), \ LR(j) = \frac{p_{\theta_1}(y_j, y_{j+1} \ldots y_t)}{p_{\theta_0}(y_j, y_{j+1} \ldots y_t)}$. For *unknown changed system parameters*, the Generalized Likelihood Ratio Test can be used whose solution for *linear* systems in well known [15]. When a nonlinear system experiences a change, linearization techniques like extended Kalman filters and change detection methods for linear systems are used [15]. Linearization techniques are computationally efficient but are not always applicable (require a good initial guess at each time step and hence are not robust to noise spikes).

Approaches for sudden change detection using PFs are discussed in a recent survey article [16]. [17] is an attempt to use a particle filtering approach for *sudden* change detection in nonlinear systems without linearization. It assumes that *the parameters of the changed system are known* and defines a modification of the CUSUM change detection statistic that can be efficiently evaluated using PFs. It runs a sequence of PFs to evaluate $LR(j)$ for

$1 \leq j \leq t$. Both CUSUM and the statistic of [17] assume known change parameters and are based on the likelihood ratio of the current $(t - j)$ observations $(LR(j))$.

An entirely different class of approaches (e.g. see [18]) used extensively with PFs uses a *discrete state variable* to denote the mode that the system is operating in. A change is detected by looking at the expected or most probable value of the mode variable. This is typically used when the system can operate in multiple modes each associated with a different and *known* system model. The mode variable's transition between states is governed by the mode that maximizes the likelihood of the observations.

When *changed system parameters are not known*, *sudden changes* can be detected using tracking error [19] which is the distance (usually Euclidean distance) between the current observation and its prediction based on past observations.

We have also studied the stability of errors in approximating the ELL for changed observations using a PF that is optimal for the unchanged system. There has been a lot of recent research on studying the stability of the optimal nonlinear filter. Asymptotic stability results w.r.t. initial condition were first proved in [20]. The Hilbert projective metric has been used to prove stability w.r.t. the initial condition and also w.r.t. the model [21, 22]. New approaches have been proposed recently for noncompact state spaces [23, 24]. The results for stability w.r.t. the model have been used to prove convergence of the PF estimate of the posterior with number of particles, $N \to \infty$ [5, 25]. We use results from [5] in which the authors have replaced the mixing transition kernel assumption required for proving stability with a much weaker mixing unnormalized filter kernel assumption.

1.3.2 Landmark Shape Dynamics

Some of the commonly used representations for shape are Fourier descriptors [26], splines [27] and deformable snakes all of which model the shape of continuous curves. But in our work we are attempting to model the dynamics of a group of discrete landmarks (which could be moving point objects or moving parts of an articulated object like the human body). Since the data is inherently finite dimensional, using infinite dimensional representations of a continuous curve is not necessary and hence we look only at the representation of shape in $\Re^n$ (modulo Euclidean similarity transformations) which was first defined by Kendall in 1977. There has been a lot of work on defining shape coordinates for Kendall's shape, some commonly used shape coordinates are due to Bookstein, Jacobi, Kendall, and tangent shape representations by Dryden and Mardia (all of these are discussed in [9]). These shape coordinates have been used frequently in control literature for satellite or aircraft or robot formation control (e.g. see [28] and references therein). Probability distributions for shape and preshape space and for the tangent to shape space at the mean are discussed in chapter 6, 7, 11 of [9], and in [29], [30], [31]. In [32], Cootes and Taylor have proposed 'Point Distribution Models' which are principal component models for shape variation using Procrustes residuals.

Active Shape Models [33] also considers the configuration of points in $\Re^n$ but they define affine deformation models in configuration space. These are good for modeling deformation of approximately rigid objects where the main source of nonrigidity is camera motion. But we are trying to define models for the changing configuration of a group of objects and hence our approach which provides a number of degrees of freedom proportional to the number of landmarks is useful. Other models for shape deformation of one shape into another have been proposed which include thin plate splines, and principal and partial warps (discussed in chapter 10 of [9]). These are good for studying shape change between two objects but are computationally infeasible to define dynamical models on the shape manifold. Our idea of defining "shape activity" models by separating scaled Euclidean motion and shape dynamics is motivated by [34], where the authors split the deformation of a deforming and moving shape into scaled translation, scale, rotation of a shape plus its non-rigid deformations.

We propose partially observed dynamical models (that also satisfy the Hidden Markov Model property and hence we also refer to them as HMMs) for stationary and non-stationary shape activities. Our model for non-stationary shape activities is similar in spirit to those in [35] and [36] where the authors define dynamical models for motion on Lie groups and Grassmann manifolds (for time-varying subspace estimation), respectively, using piecewise geodesic priors and track them using particle filters. Also, in [37], the authors have discussed how to define geodesic paths on shape spaces parametrized by direction functions. Another work [38] defines principal geodesic analysis on Lie groups which is motivated by PCA in Euclidean spaces and uses it for developing representations of geometry based on the medial axis description.

1.3.3 Activity Recognition and Tracking

There is a large body of work in computer vision on modeling and recognition of activities, human actions and events. The work can be classified (based on the formalisms used) as Bayesian networks (BNs) and Dynamic Bayesian networks (DBNs) [39, 2]; finite state Hidden Markov models for representing an activity [40, 41] ; stochastic grammars [42]; and factorization method based approaches [43, 44]. In [1], the authors perform clustering to learn the co-occurrence statistics of individual objects and their interactions with other objects. In [45], events are treated as long spatio-temporal objects and clustered based on their behavioral content. [46] uses projective invariants to represent different human actions like running, walking and climbing. In [47], action "objects" are represented using generalized cylinders with time forming the cylinder axis.

Now, [1, 43, 44, 45, 47] present non-parametric approaches to activity/event recognition, while HMMs, stochastic grammars, BNs and DBNs [39, 2] are model based approaches. Our work also defines a parametric model for an activity performed by a group of objects, but it is a continuous state HMM and there are some other differences: First, we treat objects as point objects and hence we can get our observations from low resolution video or even from other sensors like radar, acoustic or infra-red. Second, we provide a single global framework for modeling the interactions and independent motion of multiple moving objects by treating them as a deforming shape. The approach is scalable in the number of interacting objects. This is in contrast to [1], where joint histograms are needed to model interactions between objects, thus making the approach infeasible for large number of interacting objects. Also, using a deformable shape to represent a group activity or human actions makes the approach invariant to motion of a camera or any other sensor (under scaled orthography assumption). Another work which also models human motion using a dynamical model is [48]. They learn a linear dynamical model for the gait of different subjects and define a "distance" between dynamical models as a metric for gait recognition.

Particle Filters for Tracking Multiple Moving Objects: Particle filters [4] have been used extensively for tracking [3] a single moving object in conjunction with a measurement algorithm to obtain observations [49, 50, 18, 51]. In [52], PFs are used to track multiple moving objects in 3D by simultaneously estimating their structure (3D location) and motion (velocity) information. Joint Probability Data Association Filtering (JPDAF) is used along with a PF in [53] to track multiple moving objects. It uses separate state vectors for each object and defines data association events to associate the state and observation vectors. In this framework, defining interactions between a large number of objects can become very complicated. In our work, we represent the combined state of all the moving objects using the shape of its configuration and define a dynamic model for it (the shape dynamical model implicitly models the interactions). We track using a PF to filter out the shape from noisy observations of the object locations and use the filtered shape distribution for abnormal activity detection.

[3] "tracking" here refers to "tracking to obtain observations"

1.3.4 Principal Component Null Space Analysis (PCNSA)

Existing linear classification algorithms like principal component analysis (PCA), linear discriminant analysis (LDA) and subspace LDA are optimal for 'apples from apples' type of problems. PCA [54] yields projection directions that maximize the total scatter but do not minimize the within class variance of each class and also sometimes retains directions with unwanted large variations due to variation in lighting etc. LDA [55] encodes discriminatory information by finding directions that maximize the ratio of between class scatter to within-class (or intra-class) scatter. In [56] , PCA and LDA are combined to yield a subspace LDA (SLDA) based classification algorithm for face recognition which uses PCA first for dimensionality reduction and then LDA. Subspace LDA is also used in [57] for view based image retrieval from a database of real world objects. Also PCNSA is similar in spirit to an algorithm called Multispace KL (MKL) which appeared in [58] around the same time as our conference paper [59] on PCNSA. We discuss in Section 5.5 the connection between MKL and our algorithm and how our error probability analysis can be extended to analyze MKL.

Several non-linear techniques exist in the literature that could be applied to the 'apples from oranges' type classification problems. In [60], Murase and Nayar present such an algorithm for object recognition. They propose a representation of object appearance in the PCA space parameterized by pose and illumination. Each object class is represented in the PCA space using a B-spline manifold. A query image is recognized based on the manifold that it is closest to in the PCA space. The computational complexity of their algorithm is much higher than any of the linear algorithms including ours. Kernel PCA [61] and Kernel Discriminant Analysis (KDA) [62] attempt to transform non-linearly separable data into a higher dimensional space in which it becomes linearly separable. But projecting a query along one dimension in KDA requires computation of n inner products (where n is the number of training samples) which is n times that of any linear method. Mao and Jain [63] describe neural network algorithms, Sammon's nonlinear projection and nonlinear discriminant analysis (NDA) for PCA and LDA. Neural network algorithms are suitable for easy hardware implementation or for applications where distribution of patterns in feature space is changing with time (non-stationary data) or for non-linearly separable data. [63] also compares performance of linear (PCA,LDA) and nonlinear algorithms (NDA) for different kinds of data. Two interesting examples of linearly non-separable data are shown, one in which clusters of the two classes are non-convex and the other in which the two clusters are spheres with almost coinciding centres but different radii and hence are not disjoint. For both these data distributions, linear methods are shown to fail.

Chapter 2

Change Detection in General HMMs

2.1 Problem Formulation

2.1.1 The General HMM Model

We assume a general HMM [5] with an $\Re^{n_x}$ valued state process $X = \{X_t\}$ and an $\Re^{n_y}$ valued observation process $Y = \{Y_t\}$[1]. The system (or state) process $\{X_t\}$ is a Markov process with state transition kernel $Q_t(x_t, dx_{t+1})$ and the observation process is a memoryless function of the state given by $Y_t = h_t(X_t) + w_t$ where w_t is an i.i.d. noise process and h_t is, in general, a nonlinear function. We denote the conditional distribution of observation given state by $G_t(dy_t, x_t)$. It is assumed to be absolutely continuous [64] and its pdf is given by $g_t(Y_t, x) \overset{\triangle}{=} \psi_t(x)$. The prior initial state distribution (denoted by $p_0(x)$), the conditional

[1] We use the subscript 't' (e.g. X_t, Y_t) instead of 'n' for (discrete) time instants, to avoid confusion with N used for the number of particles in Particle Filtering

distribution of observation given state and the state transition kernel are known and assumed to be absolutely continuous[2]. With this assumption, the prior distribution of the state at any time t is also absolutely continuous and admits a density which we denote by $p_t(x)$.

2.1.2 Problem Definition

We study the problem of detecting slow and drastic changes in the system model of a general HMM (described above) when the change parameters are unknown. We assume that the normal (original/unchanged) system has state transition kernel Q_t^0. A change in the system model begins to occur at some time t_c and lasts till a final time t_f (change duration finite). In the time interval, $[t_c, t_f]$, the state transition kernel is Q_t^c and after t_f it again becomes Q_t^0. Both Q_t^c and the change start and end times t_c, t_f are assumed to be unknown. The goal is to detect the change, with minimum delay. Note that although the change in system model is assumed to last for a finite time, $[t_c, t_f]$, its effect on the prior state pdf $p_t(x)$ is either permanent or it lasts for a much longer time (Q_t^0 is either not mixing or very slowly mixing).

2.1.3 The Approach

We repeat from Section 1.2.1 a summary of our approach. Drastic changes can be detected easily using the increase in tracking error or the negative log of observation likelihood (OL).

[2] Note that for ease of notation, we denote the pdf either by the same symbol or by the lowercase of the probability distribution symbol

But slow changes usually get missed. We use a PF to estimate the posterior probability distribution of the state at time t (X_t) given observations up to t ($Y_{1:t}$), $Pr(X_t \in dx|Y_{1:t}) \triangleq \pi_t(dx)$. We propose a statistic called ELL (which stands for Expected Log-Likelihood) which is able to detect slow changes. ELL is the conditional Expectation of the negative Log-Likelihood of the state at time t ($[-\log p_t(X_t)]$), given past observations, $Y_{1:t}$. It is evaluated as the expectation under π_t of $[-\log p_t(X_t)]$.

The PF is optimal for the unchanged system and hence when estimating π_t for the changed system, an unspecified amount of modeling error exists. Also the particle filtering error (error due to a finite number of Monte Carlo samples or particles) is much larger. But using the stability results from [5], we are able to show that the approximation errors are eventually monotonically decreasing (and hence stable) with time for large enough number of particles (in section 2.4). We also show asymptotic stability under stronger assumptions. We show in section 2.5, that the bound on the error is proportional to the rate of change. Thus for slow changes, the estimation error in π_t is small i.e. ELL is approximated correctly for such changes. Hence the approximate value of ELL detects the slow change as soon as it becomes "detectable" (defined in Definition 5 of section 2.3.2). ELL fails to detect drastic changes because of large estimation error in evaluating π_t. But a large estimation error in evaluating π_t also corresponds to a large value of OL (or tracking error) which can be used for detecting such changes. We discuss this in Section 2.6.

2.2 Preliminaries and Notation

We present below some notation and definitions of terms used in the rest of the chapter. We then present in Section 2.2.2, the optimal nonlinear filter and its approximation using a particle filter.

2.2.1 Notation and Definitions

We use H_0 to denote the original or unchanged system hypothesis and H_c to denote the changed system hypothesis. Also, the superscript ^{c} is used to denote any parameter related to the changed system, ^{0} for the original system and ^{c,0} for the case when the observations of the changed system are filtered using a filter optimal for the original system[3]. Thus the posteriors, $\pi_t^{0,0}(dx) = Pr(X_t \in dx|Y_{1:t}^0, H_0)$ (also denoted by π_t^0), $\pi_t^{c,c}(dx) = Pr(X_t \in dx|Y_{1:t}^c, H_c)$ (also denoted by π_t^c) and $\pi_t^{c,0}(dx) = Pr(X_t \in dx|Y_{1:t}^c, H_0)$ where

$$\begin{aligned} Y_{1:t}^c &= (Y_{1:t_c-1}^0, Y_{t_c:t}^c), \ \forall t \leq t_f \\ &= (Y_{1:t_c-1}^0, Y_{t_c:t_f}^c, Y_{t_f+1:t}^0), \ \forall t > t_f. \end{aligned} \tag{2.1}$$

Also, for PF estimates of these distributions, we add the superscript ^{N} to denote the number of particles, for e.g. $\pi_t^{0,N}$, $\pi_t^{c,N}$ or $\pi_t^{c,0,N}$. The (possibly unnormalized) conditional pdf of Y_t given state x is $\psi_t^0(x) \stackrel{\triangle}{=} g_t(Y_t^0, x)$ and $\psi_t^c(x) \stackrel{\triangle}{=} g_t(Y_t^c, x)$.

With any nonnegative kernel, J, defined on the state space E, is associated a nonnegative linear operator denoted by J and defined by $J\mu(dx') \stackrel{\triangle}{=} \int_E \mu(dx)J(x, dx')$ for any nonnegative

[3] At most places ^{0,0} is replaced by ^{0} and ^{c,c} by ^{c}

measure μ [5]. For any finite measure, μ, the normalized measure is denoted by $\bar{\mu} \overset{\triangle}{=} \mu/\mu(E)$. The normalized nonnegative nonlinear operator $\bar{J}$ is defined by $\bar{J}(\mu) \overset{\triangle}{=} \frac{J\mu}{(J\mu)(E)}$. Also, $(.,.)$ is the inner product notation.

The prior state distribution at time t, $(Q_t^0, ...Q_1^0\pi_0)(dx)$ has pdf $p_t(x)$ while the changed system's prior state distribution, $(Q_t^0, ...Q_{t_f}^c, ..Q_{t_c}^c...Q_1^0\pi_0)(dx)$ has pdf $p_t^c(x)$. In a lot of cases (for example if the system model is linear Gaussian with Gaussian initial state pdf) it is possible to define the pdfs $p_t(x)$ and $p_t^c(x)$ in a closed form. In cases where it cannot be defined in a closed form, it can be approximated by a single or a mixture of Gaussians or by any other parametric family of distributions (we discuss this in section 2.3.4).

Note that throughout the chapter, **"event occurs a.s."** refers to the event occurring almost surely w.r.t. the measure corresponding to the probability distribution of $Y_{1:t}$. Also, E_μ denotes expectation under the measure μ, for example E_{π_t} is expectation under the posterior state distribution. E_Y denotes expectation under the distribution of the random variable Y, for example $E_{Y_{1:t}}$ denotes expectation under the distribution of the observation sequences. Finally, Ξ_{pf} denotes averaging over different realizations of the PF each of which produces a different realization of the random measure π_t^N [4].

Now we would like to clarify here the difference between the terms "system model error" and "modeling error" or "model error" as used in this work. We use "system model error" to denote the error in the system model at a given time because of the change, i.e. it is the

[4] expectation under the probability distribution of the random measure π_t^N or equivalently of the random particles, $\{x_t^{(i)}\}_{i=1}^N$.

"distance" between the changed and the unchanged model. We use "model error" to denote the error in the posterior state distribution (and the error in ELL estimation because of this) introduced because of the system model error. Model error is explained in section 2.4.

We now present some definitions of terms used in the chapter:

Definition 1 *The* **unnormalized filter kernel** *[5] for a system with state transition kernel Q_t and probability of observation given state ψ_t, is given by $R_t(x, dx') = Q_t(x, dx')\psi_t(x')$. So $R_t^0 = Q_t^0\psi_t^0$ is the unnormalized filter kernel for the original system observations estimated using the original system model, Q_t^0; $R_t^c = Q_t^c\psi_t^c$ is the unnormalized filter kernel for the changed system observations using the changed system model, Q_t^c; while $R_t^{c,0} = Q_t^0\psi_t^c$ is the unnormalized filter kernel for the changed system observations using the original system transition kernel, Q_t^0 (this is what is used in practice since Q_t^c is unknown).*

Definition 2 *[5] A nonnegative kernel J defined on E is* **mixing** *if there exists a constant, $0 < \epsilon \leq 1$ and a nonnegative measure λ s.t. $\epsilon\lambda(A) \leq J(x, A) \leq \frac{1}{\epsilon}\lambda(A)$ $\forall x \in E$ and for any Borel subset $A \subset E$. A (time) sequence of mixing kernels $\{J_t\}$ is said to be* **uniformly mixing** *if $\epsilon = \sup_t \epsilon_t > 0$.*

Definition 3 *[5] The* **Birkhoff's contraction coefficient** *of any kernel J is, $\tau(J) = \sup_{0 \leq h(\mu,\mu') < \infty} \frac{h(J\mu, J\mu')}{h(\mu,\mu')} = tanh[\frac{1}{4}\sup_{\mu,\mu'} h(J\mu, J\mu')]$. h here denotes the Hilbert metric which is defined and explained in [5]. $\tau(J) \leq 1$ always and if J is mixing, $\tau(J) \leq \tilde{\tau}(J) < 1$ where $\tilde{\tau}(J) \stackrel{\triangle}{=} \frac{1-\epsilon^2}{1+\epsilon^2} < 1$.*

We denote $\tau(R_t)$ by τ_t and $\epsilon(R_t)$ by ϵ_t. Note that R_t depends on Y_t and hence τ_t and ϵ_t

are, in general, random variables. So a correct statement would be that R_t is a.s. mixing ($\epsilon_t > 0, a.s.$ and $\tau_t < 1, a.s.$).

2.2.2 Approximate Non-linear Filtering Using a Particle Filter

The problem of nonlinear filtering is to compute at each time t, the conditional probability distribution, of the state X_t given the observation sequence $Y_{1:t} = (Y_1, Y_2, ...Y_t)$, $\pi_t(dx) = Pr(X_t \in dx|Y_{1:t})$. The transition from π_{t-1} to π_t is defined using the Bayes recursion as follows:

$$\pi_{t-1} \text{ —-> } \pi_{t|t-1} = Q_t \pi_{t-1} \text{ —-> } \pi_t = \frac{\psi_t \pi_{t|t-1}}{(\pi_{t|t-1}, \psi_t)}$$

Now if the system and observation models are linear Gaussian, the posteriors would also be Gaussian and can be evaluated in closed form using a Kalman filter. For nonlinear or nonGaussian system or observation model, except in very special cases, the filter is infinite dimensional. Particle Filtering [25] is a sequential monte carlo technique for approximate nonlinear filtering which was first introduced in [4] as Bayesian Bootstrap Filtering.

A **particle filter** is a recursive algorithm which produces at each time t, a cloud of N particles $\{x_t^{(i)}\}$ whose empirical measure, π_t^N (which is a random measure), closely “follows” π_t. It starts with sampling N times from π_0 to approximate it by $\pi_0^N(dx) \triangleq \frac{1}{N}\sum_{i=1}^N \delta_{x_0^{(i)}}(dx)$. Then for each time step it runs the Bayes recursion which can be summarized as follows:

$$\pi_{t-1}^N \triangleq \frac{1}{N}\sum_{i=1}^N \delta_{x_{t-1}^{(i)}}(dx) \text{ —–> } \pi_{t|t-1}^N \triangleq \frac{1}{N}\sum_{i=1}^N \delta_{\bar{x}_t^{(i)}}(dx)$$
$$\text{—–> } \bar{\pi}_t^N \triangleq \frac{1}{N}\sum_{i=1}^N w_t^{(i)} \delta_{\bar{x}_t^{(i)}}(dx) \text{ —–> } \pi_t^N \triangleq \sum_{i=1}^N \delta_{x_t^{(i)}}(dx)$$

$$
\begin{aligned}
\text{where} \quad \bar{x}_t^{(i)} &\sim Q_t(x_{t-1}^{(i)}, dx), \\
x_t^{(i)} &\sim \text{Multinomial}(\{\bar{x}_t^{(i)}, w_t^{(i)}\}_{i=1}^N) \\
w_t^{(i)} &\triangleq \frac{\psi_t(\bar{x}_t^{(i)})}{(\pi_{t|t-1}^N, \psi_t(\bar{x}_t^{(i)}))}
\end{aligned}
\tag{2.2}
$$

Note that both $\bar{\pi}_t^N$ and π_t^N approximate π_t but the last step is aimed at reducing the degeneracy of the particles. The samples $\{\bar{x}_t^{(i)}\}$ are sampled according to a multinomial distribution proportional to their weights, $w_t^{(i)}$, so that particles with very low weights get eliminated while those with higher weights get repeated in proportion to their weights.

2.3 Change Detection Statistics

2.3.1 The ELL statistic

"Expected (negative) Log Likelihood" or ELL at time t, is the *conditional expectation of the negative log of the prior likelihood of the state at time t, under the no change hypothesis (H_0), given observations till time t*, i.e.

$$ELL(Y_{1:t}) \triangleq E[-\log p_t^0(x)|Y_{1:t}] = E_{\pi_t}[-\log p_t^0(x)]. \tag{2.3}$$

The second equality follows from the definition of π_t, $\pi_t(dx) = Pr(X_t \in dx|Y_{1:t})$. For systems where exact filters do not exist and a PF is used to estimate π_t, the estimate of ELL using the empirical distribution π_t^N becomes

$$ELL^N = \frac{1}{N}\sum_{i=1}^{N}[-\log p_t^0(x_t^{(i)})]. \tag{2.4}$$

It is interesting to note that ELL as defined above is equal to the Kerridge Inaccuracy [6] between the posterior and prior state pdf.

Definition 4 *The* **Kerridge Inaccuracy** *between two pdfs* p, q *is defined as* $K(p : q) = \int p(x)[-\log q(x)]dx$. *It is used in statistics as a measure of inaccuracy between distributions and was first defined by Kerridge in [6].*

We have $ELL(Y_{1:t}) \overset{\triangle}{=} E_{\pi_t}[-\log p_t^0(x)] = K(\pi_t : p_t^0)$[5]. Henceforth, we denote

$$ELL(Y_{1:t}^0) = K(\pi_t^0 : p_t^0) \overset{\triangle}{=} K_t^0 \text{ and } ELL(Y_{1:t}^c) = K(\pi_t^c : p_t^0) \overset{\triangle}{=} K_t^c. \tag{2.5}$$

Motivation for ELL

The use of ELL (or equivalently Kerridge Inaccuracy) for partially observed systems is motivated by the use of log likelihood for hypothesis testing in the fully observed case. For a fully observed system (h_t invertible and zero observation noise), one can evaluate $X_t = h_t^{-1}(Y_t)$ from the observation Y_t and then $\log p_t(X_t)$ would be the log likelihood of state taking value X_t under H_0 (this is proportional to likelihood of Y_t under H_0). Thus if $Y_t = Y_t^0$, then its likelihood, (and also the likelihood of the state X_t) under H_0 will be larger than if $Y_t = Y_t^c$ [6]. But for partially observed systems, X_t is not a deterministic function of $Y_{1:t}$. It is a random variable with distribution π_t. Hence we propose to replace the log likelihood of the state by its expectation under π_t which is the ELL. Note that ELL can also be interpreted as the

[5] it is actually $K(\frac{d\pi_t}{dx} : p_t^0)$ but as mentioned earlier, we denote the density $\frac{d\pi_t}{dx}$ by the same symbol as the distribution

[6] Note that here observation likelihood and state likelihood (=ELL) differ only by a constant.

MMSE estimate of log likelihood of state obtained from the noisy observations.

2.3.2 When does ELL work: A Kerridge Inaccuracy perspective

Taking expectation of $ELL(Y_{1:t}^0) = K(\pi_t^0 : p_t^0)$ over normal observation sequences, we get

$$\begin{aligned} E_{Y_{1:t}^0}[ELL(Y_{1:t}^0)] &= E_{Y_{1:t}^0} E_{\pi_t^0}[-\log p_t^0(x)] \\ &= E_{p_t^0}[-\log p_t^0(x)] = H(p_t^0) = K(p_t^0 : p_t^0) \overset{\triangle}{=} EK_t^0 \end{aligned}$$

where $H(.)$ denotes entropy. Similarly, for the changed system observations, $E_{Y_{1:t}^c}[ELL(Y_{1:t}^c)] = K(p_t^c : p_t^0) \overset{\triangle}{=} EK_t^c$, i.e. the expectation of ELL of changed system observations is actually the Kerridge Inaccuracy between the changed system prior, p_t^c, and the original system prior, p_t^0, which will be larger than the Kerridge Inaccuracy between p_t^0 and p_t^0 (entropy of p_t^0) [7].

Now, ELL will detect the change when EK_t^c is "significantly" larger than EK_t^0. Setting the change threshold to

$$\kappa_t \overset{\triangle}{=} EK_t^0 + 3\sqrt{VK_t^0}, \text{ where } VK_t^0 = Var_{Y_{1:t}}(K_t^0), \tag{2.6}$$

will ensure a false alarm probability less than 0.11 (0.05 if unimodal)[7]. By the same logic, if $EK_t^c - 3\sqrt{VK_t^c} > \kappa_t$ then the miss probability [65] (probability of missing the change) will also be less than 0.11 (0.05 if unimodal). Now evaluating VK_t^0 or VK_t^c analytically is not possible without having an analytical expression for π_t^0 or π_t^c. But we can use Jensen's inequality [66] to bound VK_t^0 (and similarly VK_t^c) as follows (by applying Jensen's inequality

[7] 0.11 follows from the Chebyshev inequality [65]. But if the pdf of $K_t^0(Y_{1:t})$ is unimodal, Gauss's inequality [65] can be applied to show that the probability is less than 0.05

on z^2, which is a convex function, with $z = [-\log p_t(x)]$):

$$\begin{aligned}
{K_t^0}^2 &= (E_{\pi_t}[-\log p_t(x)])^2 \leq E_{\pi_t}[[-\log p_t(x)]^2] \\
\text{So,} \quad VK_t^0 &= Var_{Y^0_{1:t}}(K_t^0) = E_{Y_{1:t}}[{K_t^0}^2] - (EK_t^0)^2 \\
&\leq E_{Y_{1:t}}[E_{\pi_t}[[-\log p_t(x)]^2]] - (EK_t^0)^2 \\
&= E_{p_t^0}[[-\log p_t^0(x)]^2] - (EK_t^0)^2 \qquad (2.7)
\end{aligned}$$

Definition 5 *We define a change to be* **"detectable"** *by ELL (with false alarm and miss probabilities less than 0.11) if*

$$EK_t^c - 3\sqrt{VK_t^c} > \kappa_t, \quad where \ \ \kappa_t \triangleq EK_t^0 + 3\sqrt{VK_t^0} \qquad (2.8)$$

2.3.3 When ELL fails: The OL Statistic

The above analysis assumed no estimation errors in evaluating ELL. But, the PF is optimal for the unchanged system. Hence when estimating π_t (required to evaluate ELL) for the changed system, there is modeling error. Also the particle filtering error is much larger in this case. The approximation error in estimating the ELL is proportional to the "rate of change" (discussed in section 2.5). Hence the ELL is approximated accurately for a slow change and thus detects such a change when it becomes "detectable" (see definition 5 above in Section 2.3.2). But ELL fails to detect drastic changes because of large estimation error in evaluating π_t.

But large estimation error in evaluating π_t also corresponds to a large value of OL (or tracking error) which can be used for detecting such changes (discussed in section 2.6). OL is the negative log likelihood of current observation conditioned on past observations under the no change hypothesis, i.e. $OL = -\log Pr(Y_t|Y_{1:t-1}, H_0)$. A change is declared if OL exceeds a threshold. OL is evaluated as $OL_t^N = -\log(Q_t^0 \pi_{t-1}^N, \psi_t)$. Thus for changed observations, $OL_t^{c,0,N} = -\log(Q_t^0 \pi_{t-1}^{c,0,N}, \psi_t^c)$ (notation defined in section 2.2.1).

On the other hand, OL takes longer to detect a slow change (or does not detect it at all) because of the following reason: Assuming that $\pi_{t-1}^{c,0,N}$ correctly approximates π_{t-1}^c (which is true for a slow change), OL uses only the change magnitude at the current time step, $D_{Q,t}$ (defined in Definition 6 of section 2.5), to detect the change. For a slow change, $D_{Q,t}$ is also small. This intuitive idea becomes clearer in Theorem 3 of section 2.5. OL starts detecting the slow change only when the approximation error in $\pi_{t-1}^{c,0,N}$ becomes large enough, but ELL detects it faster (see figure 2.2).

2.3.4 Practical Issues

Defining $p_t(x)$

The ELL is given by $E_{\pi_t}[-\log p_t(X)]$ for which we need to know the state prior $p_t(x)$ (note we denote $p_t^0(x)$ by $p_t(x)$ in the rest of this chapter) at each time instant. If the state dynamics (or the part of the state dynamics used for detecting change) is linear with Gaussian system noise and Gaussian initial state distribution and for some other cases, this can be easily defined in closed form.

If the prior probability (likelihood) of the part of the state vector used to detect the change cannot be defined in closed form for each t, then one solution is to use prior knowledge to define $p_t(x)$ as coming from a certain parametric family such as a mixture of Gaussians. Its parameters can be learnt using training data sequences. If $p_t(x)$ is assumed to be piecewise constant in time, one can use a single observation noise-free training sequence to learn its parameters. Both these ideas are demonstrated in section 4.1.3 of chapter 4 when defining the abnormal activity detection problem for nonstationary shape activities.

Time Averaging

A second practical issue is that single time instant estimates of ELL or OL may be noisy. Hence in practice, we average the statistic over a set of past time frames. Averaging OL over past p frames gives $aOL(p) = \frac{1}{p}[-\log Pr(Y_{t-p+1:t}|Y_{1:t-p})]$. Averaging ELL over past frames is given by $aELL(p) = \frac{1}{p}\sum_{k=t-p+1}^{t} ELL(Y_{1:k})$ but this cannot be justified unless we can show that $ELL(Y_{1:t})$ is ergodic. But one can evaluate joint ELL as $jELL(p) = \frac{1}{p}E[-\log p_{t-p+1:t}(X_{t-p+1:t})|Y_{1:t}]$ which is the Kerridge Inaccuracy between the joint posterior distribution of $X_{t-p+1:t}$ given $Y_{1:t}$ and their joint prior. If using $aELL(p,t)$, the threshold $Th(p,t)$ will depend on the sum of individual entropies of $X_{t-p+1:t}$. If using $jELL(p)$, the threshold, $Th(p,t)$, will depend on the joint entropy of $X_{t-p+1:t}$.

Now the value of p can either be set heuristically or one can modify the CUSUM algorithm [15] to deal with unknown change parameters: Declare a change if

$$\max_{1\leq p\leq t}[Statistic(p) - Th(p,t)]) > \lambda. \tag{2.9}$$

The change time is estimated as $t-p^*+1$ where p^* is the argument maximizing $[Statistic(p)-Th(p,t)]$.

2.4 Errors in ELL Approximation

Now the above analysis for ELL assumes that there are no errors in estimating $ELL(Y_{1:t}^0) = K(\pi_t^0 : p_t) \triangleq K_t^0$ and $ELL(Y_{1:t}^c \triangleq K_t^c$ which is true only if exact finite dimensional filters exist for a problem and correct models for the transition kernel and conditional probability of observation given state are used. The estimation of K_t^0 in the linear Gaussian case using a Kalman filter is an example of this condition. But in all other cases there are three kinds of errors: When we are trying to estimate K_t^c using the transition kernel for the original system, what we are really evaluating is $K_t^{c,0} \triangleq E_{\pi_t^{c,0}}[-\log p_t^0(x)]$ instead of K_t^c (**model error**). Note that $\pi_t^{c,0}$ is the posterior state distribution for the changed observations estimated using a PF optimal for the unchanged system. We can use stability results from [5] to show under certain assumptions, that the model error goes to zero for large time instants, for posterior expectations of bounded functions of the state. Under weaker assumptions, we can show that the error is eventually monotonically decreasing and hence stable. But $K_t^{c,0} = E_{\pi_t^{c,0}}[-\log p_t^0(x)]$ and $[-\log p_t^0(x)]$ is an unbounded function. Considering its bounded approximation introduces **bounding errors** which go to zero as the bound goes to infinity. We need a bounded approximation because the stability results hold only for bounded functions of the state. Also, when we use a PF with a finite number of particles to approximate the optimal filter, the **PF approximation error** is introduced. This error goes to zero as the number of particles goes to infinity. For a given finite number of particles, the PF approximation error increases with increasing system model error (We show this in Section 2.5).

Now, we quantify our claims. Our aim is to *either* show a result of the type $\lim_{M\to\infty}(\lim_{N\to\infty}\Xi_{pf}[|K(\pi_t^0 : p_t) - K(\pi_t^{0,N} : p_t^M)|]) = 0$ and $\lim_{M\to\infty}(\lim_{t\to\infty}(\lim_{N\to\infty}\Xi_{pf}[|K(\pi_t^c : p_t) - K(\pi_t^{c,0,N} : p_t^M)|])) = 0$, *a.s.*, where $p_t^M(x) \triangleq \max\{p_t(x), e^{-M}\}$[8] or show that $[-\log p_t(x)]$ is uniformly bounded for all t, so that the outermost convergence with M trivially follows. Under weaker assumptions, we show that even though the error does not converge to zero with time, it is eventually monotonically decreasing with time and hence stable. We use the following two results from [5]:

Lemma 1 (Model Error bound, Theorem 4.8 of [5]): *If for all k, the kernel R_k is a.s. mixing ($\Rightarrow \epsilon_k > 0, a.s.$ & Birkhoff's contraction coefficient $\tau_k \le \tilde{\tau}_k(\epsilon_k) < 1, a.s.$), then the weak norm between the correct optimal filter density μ_t and the incorrect one μ'_t is upper bounded as follows:*

$$\begin{aligned} \sup_{\phi:||\phi||_\infty \le 1} |(\mu_t - \mu'_t, \phi)| &\le \delta_t + \frac{2\delta_{t-1}}{\epsilon_t^2} + \frac{4}{\log 3}\sum_{k=1}^{t-2}\tilde{\tau}_{t:k+3}\frac{\delta_k}{\epsilon_{k+1}^2\epsilon_{k+2}^2} && (2.10)\\ &\triangleq \theta_t(\delta_k, \epsilon_k, 0 \le k \le n), a.s. && (2.11)\end{aligned}$$

$$\textit{where}\ \ \delta_k \triangleq \sup_{\phi:||\phi||_\infty \le 1} |(\mu'_k - \bar{R}_k\mu'_{k-1}, \phi)| \le 2 \qquad (2.12)$$

Lemma 2 (PF error bound) :

[8] Note p_t^M is not a pdf.

1. **(Theorem 5.7 of [5])** *If for all k, the kernel R_k is a.s. mixing ($\epsilon_k > 0, a.s.$ & $\tau_k \leq \tilde{\tau}_k(\epsilon_k) < 1, a.s.$), and $\sup_{x \in E_{x,y}} \psi_k(x) < \infty, a.s.$, then the weak norm between the correct optimal filter density μ_t and the approximation μ_t^N (evaluated using the PF) is upper bounded as follows:*

$$\sup_{\phi:||\phi||_\infty \leq 1} \Xi_{pf}[|(\mu_t - \mu_t^N, \phi)|] \leq \frac{2(\rho_t + \frac{2\rho_{t-1}}{\epsilon_t^2} + \frac{4}{\log 3}\sum_{k=1}^{t-2} \tilde{\tau}_{t:k+3} \frac{\rho_k}{\epsilon_{k+1}^2 \epsilon_{k+2}^2})}{\sqrt{N}} \tag{2.13}$$

$$\triangleq \frac{\beta_t(\rho_k, \epsilon_k, 0 \leq k \leq n)}{\sqrt{N}}, a.s. \tag{2.14}$$

$$\textit{where} \;\; \rho_k \triangleq \frac{\sup_{x \in E} \psi_k(x)}{\inf_{\mu \in \mathcal{P}(E)} (Q_k \mu, \psi_k)} < \infty, a.s. \tag{2.15}$$

2. **(Corollary 5.11 of [5])** *If the sequence of kernels R_t is uniformly a.s. mixing with t i.e. $\epsilon_k > \epsilon > 0$, then convergence averaged over observations sequences holds uniformly in t, i.e. there exists a $\beta^* < \infty$ s.t. $\sup_{\phi:||\phi||_\infty \leq 1} E_{Y_{1:t}}[\Xi_{pf}[|(\mu_t - \mu_t^N, \phi)|]] < \frac{\beta^*}{\sqrt{N}}$.*

Now we can claim the following results under progressively weaker assumptions. The proofs are given in the Appendix

Theorem 1 : Asymptotic Stability Results

1. *Assuming (i) Change occurs for only a finite time period $[t_c : t_f]$ and starting time $t_c \leq T^* < \infty$; (ii) $\sup_{x \in E_{x,y}} \psi_k(x) < \infty, a.s., \; \forall k$; (iii) R_k^c, R_k^0 and $R_k^{c,0} \triangleq Q_k^0(x, dx')\psi_k^c(x')$ are a.s. uniformly mixing with time (i.e. there exists an $\epsilon > 0$ s.t. the mixing parameter*

$\epsilon_t > \epsilon \; \forall t$, a.s.) and (iv) The posterior state space, $E_{x,Y_t} \triangleq \{x \in E_t : \psi_{t,Y_t}(x) > 0\}$, is a uniformly compact and proper subset of $E_t \triangleq \{x : p_t(x) > 0\}$, then the following result holds:

$$\begin{aligned} \lim_{N\to\infty} E_{Y_{1:t}}[\Xi_{pf}[|K(\pi_t^0 : p_t) - K(\pi_t^{0,N} : p_t)|]] &= 0, a.s., \; \textit{uniformly in } t \\ \lim_{t,N\to\infty} E_{Y_{1:t}}[\Xi_{pf}[|K(\pi_t^c : p_t) - K(\pi_t^{c,0,N} : p_t)|]] &= 0, a.s. \end{aligned} \tag{2.16}$$

i.e. $error^c(t,N) \triangleq |K(\pi_t^{c,c} : p_t) - K(\pi_t^{c,0,N} : p_t)|$ averaged over PF realizations and observation sequences is asymptotically stable with t for large N [9].

2. *Assuming (i), (ii), (iii) as above, and a weaker assumption (iv)′: Convergence of the error $E_{Y_{1:t}}[|K(\pi_t^c : p_t^M) - K(\pi_t^c : p_t)|]$ to zero as $M \to \infty$ is uniform in t, then we have*

$$\begin{aligned} \lim_{M\to\infty}(\lim_{N\to\infty} E_{Y_{1:t}}[\Xi_{pf}[|K(\pi_t^0 : p_t) - K(\pi_t^{0,N} : p_t^M)|]]) &= 0, a.s., \; \textit{uniformly in } t \\ \lim_{M\to\infty}(\lim_{t,N\to\infty} E_{Y_{1:t}}[\Xi_{pf}[|K(\pi_t^c : p_t) - K(\pi_t^{c,0,N} : p_t^M)|]]) &= 0 \end{aligned} \tag{2.17}$$

It is easy to show that this implies that the $error^c(t,N,M) \triangleq |K(\pi_t^{c,c} : p_t) - K(\pi_t^{c,0,N} : p_t^M)|$ averaged over PF realizations and observation sequences is asymptotically stable with t for large N, M.

3. *Assuming (i), (ii), (iii) and a weaker assumption (iv)″: The posterior state space, $E_{x,Y_t} \triangleq \{x \in E_t : \psi_{t,Y_t}(x) > 0\}$, is a compact and proper subset of $E_t \triangleq \{x : p_t(x) > 0\}$,*

[9] This means the following: For every $\epsilon > 0$, there exists an N^* and a T^* (N^* does not depend on T^*) s.t. $\forall N > N^*$ and $\forall t > T^*$, $E_{Y_{1:t}}[\Xi_{pf}[error^c(t,N)]] < \epsilon$. Also note that for normal observations, the model error is itself zero (hence asymptotic stability with t is meaningless)

and (v) increase of $M_t \overset{\triangle}{=} \max_{x \in E_{x,Y_t}}[-\log p_t(x)]$ *with* t *is atmost polynomial, then* [10] *we have*

$$\lim_{t\to\infty}(\lim_{N\to\infty} \Xi_{pf}[|K(\pi_t^c : p_t) - K(\pi_t^{c,0,N} : p_t)|]) \quad = \quad 0, a.s. \qquad (2.18)$$

i.e. $\lim_{N\to\infty}(error^c(t,N))$ *averaged over PF realizations is asymptotically stable with* t [11]*.*

Proof: See Appendix

The assumption (iv) in Theorem 1.1 implies that $[-\log p_t(x)]$ is uniformly bounded $\forall x$ in the support set of π_t, π_t^c, $\forall t$, so that lemmas 1 and 2 can be directly applied to prove the result. But one can relax this assumption (in Theorem 1.2) by defining a sequence of functions $\{[-\log p_t^M(x)]\}$ with $p_t^M(x) = \max\{p_t(x), e^{-M}\}$, s.t. $\lim_{M\to\infty}[-\log p_t^M(x)] = [-\log p_t(x)]$. Then by a simple extension of Monotone Convergence Theorem ([64], page 87) to functions which could be negative but are bounded from below, we have $\lim_{M\to\infty} K(\pi_t^c : p_t^M) = K(\pi_t^c : p_t)$. We then get Theorem 1.2 which requires the assumption that the above convergence is uniform in t. It is difficult to show the convergence with M holding uniformly for all t, almost surely over all observation sequences since π_t is not known in closed form. But it is easy to find examples of nonlinear systems where one can show that the assumption is

[10] Result for normal observations is same as in (2.17)

[11] This means the following: For every $\epsilon > 0$, there exists a T^* s.t. $\forall t > T^*$, $\lim_{N\to\infty}(\Xi_{pf}[error(t,N)] < \epsilon$ (or that for every $t > T^*$, there exists an N^* which depends on t and ϵ, s.t. for all $N > N^*$, $error(t,N) < 2\epsilon$)

satisfied in mean over observation sequences (see the example of Section 2.7.1). Using this assumption, the convergence result in Theorem 1.2 is also a 'convergence in the mean' result.

One can also relax the assumption (iv) of Theorem 1.1 in a different way, as in Theorem 1.3. Here we assume that the posterior state space is compact for each t and assume that the increase of M_t (the bound on $[-\log p_t(x)]$) is atmost polynomial. Under this assumption, one can show asymptotic stability of the errors, but in this case a different N is required for each t (convergence with N is not uniform in t).

If the unnormalized filter kernels, R_k^c, R_k^0 and $R_k^{c,0}$, are mixing (but not uniformly mixing), convergence of the error to zero (asymptotic stability with time) will not hold. But we can still claim eventual monotonic decrease (and hence stability) of the error with time. We have the following results for changed observations (Note that even under this weaker assumption, the results for normal observations remain the same as in Theorem 1, except that the convergence with N is not uniform for all t):

Theorem 2 : Stability Results

1. *Assuming (i), (ii), a weaker assumption (iii)′: R_k^c, R_k^0 and $R_k^{c,0}$ are mixing and (iv)′: Convergence of the error $E_{Y_{1:t}}[|K(\pi_t^c : p_t^M) - K(\pi_t^c : p_t)|]$ to zero as $M \to \infty$ is uniform in t (as in Theorem 1.2), we have the following result: Given any $\Delta > 0$, there exists an M_Δ s.t.*

$$\lim_{N\to\infty} E_{Y_{1:t}}[\Xi_{pf}[|K(\pi_t^c : p_t) - K(\pi_t^{c,0,N} : p_t^{M_\Delta})|]] \quad \leq \quad \Delta + M_\Delta E_{Y_{1:t}}[\theta_t^{c,0}] \qquad (2.19)$$

where $\theta_t^{c,0} \overset{\triangle}{=} \theta_t(\delta_k^{c,0}, \epsilon_k^c, t_c \leq k \leq t)$ *(*θ_t *defined in (2.11)).* $\theta_t^{c,0}$ *and hence also* $E_{Y_{1:t}}[\theta_t^{c,0}]$ *is eventually monotonically decreasing with time. It is easy to see that this implies that* $\lim_{N\to\infty} E_{Y_{1:t}}[\Xi_{pf}[error]]$ *is eventually monotonically decreasing with* t *and hence stable.*

2. *Assuming (i), (ii), (iii)′ and (iv)″: The posterior state space,* $E_{x,Y_t} \overset{\triangle}{=} \{x \in E_t : \psi_{t,Y_t}(x) > 0\}$ *, is a compact and proper subset of* $E_t \overset{\triangle}{=} \{x : p_t(x) > 0\}$ *(as in Theorem 1.3), we have*

$$\lim_{N\to\infty} \Xi_{pf}[|K(\pi_t^c : p_t) - K(\pi_t^{c,0,N} : p_t)|] \ \leq \ M_t \theta_t^{c,0} \tag{2.20}$$

where $\theta_t^{c,0}$ *eventually monotonically decreases with time. It is easy to see that this implies that* $\frac{\lim_{N\to\infty} \Xi_{pf}[error]}{M_t}$ *is eventually monotonically decreasing with* t *and hence stable.*

3. *If only (i), (ii) and (iii)′ hold, then we have the following result: Given any* $\Delta > 0$*, there exists an* $M_{t,\Delta}$ *s.t.*

$$\lim_{N\to\infty} \Xi_{pf}[|K(\pi_t^c : p_t) - K(\pi_t^{c,0,N} : p_t^{M_{t,\Delta}})|] \ < \Delta + M_{t,\Delta}\theta_t^{c,0}, a.s. \tag{2.21}$$

where $\theta_t^{c,0}$ *monotonically decreases with time. But we cannot claim eventual monotonic decrease of the error in this case.*

Proof: See Appendix

Note that the above analysis generalizes to evaluation of posterior expectation of function of the state under the changed system model, when evaluated using a PF optimal for the unchanged system model.

2.5 Effect of Increasing Rate of Change on Approximation Errors

The aim is to detect a change as soon as possible and with a given finite number of particles. Hence we need to study the finite time, finite number of particles behavior of the bounds obtained in the previous section. ELL will detect the change, if its approximation exceeds the detection threshold (inspite of the errors). Applying theorem 2.2 (we show an example in section 2.7.1 for which the assumptions of this theorem hold), we have

$$\begin{aligned}\Xi_{pf}[|K_t^0 - K_t^{0,M,N}|] &< \frac{M_t \beta_t^0}{\sqrt{N}} \\ \Xi_{pf}[|K_t^c - K_t^{c,0,M,N}|] &< \frac{M_t \beta_t^{c,0}}{\sqrt{N}} + M_t \theta_t^{c,0} \triangleq e_t^{c,0,N} \end{aligned} \tag{2.22}$$

where $\beta_t^0 = \beta_t(\rho_k^0, \epsilon_k^0, 0 \leq k \leq t)$, $\theta_t^{c,0} = \theta_t(\delta_k^{c,0}, \epsilon_k^c, t_c \leq k \leq t)$, and $\beta_t^{c,0} = \beta_t(\rho_k^0, \epsilon_k^0, 0 \leq k \leq t_c, \rho_k^{c,0}, \epsilon_k^{c,0}, t_c \leq k \leq t)$ and θ_t, β_t defined in (2.11), (2.14) respectively. Thus for ELL to detect a change, we need to show that $K_t^c - M_t\theta_t^{c,0} - \frac{M_t\beta_t^{c,0}}{\sqrt{N}}$ exceeds the detection threshold.

We show in this section that the model error bound, $\theta_t^{c,0}$, and the PF error bound coefficient, $\beta_t^{c,0}$ (and hence also the total error, $e_t^{c,0,N}$) are upper bounded by increasing functions of the "distance" between the changed and unchanged transition kernels (which is a metric for "rate of change") with increasing derivatives of all orders. We also show that the observation likelihood, OL, is upper bounded by an increasing function of the "rate of change" metric.

Note that although we prove the above result for the change detection problem, it can directly generalize to bounding the error between the true posterior expectation of any function of the state and its posterior expectation estimated by a PF with incorrect system model assumptions. Also, the distance between the changed and unchanged transition kernels (defined below as a "rate of change" metric) can be generalized to a metric for system model error per time step. We first give below some definitions and then state a sequence of lemmas required to prove the main result. All lemmas requiring long proofs are proved in the Appendix.

Definition 6 *We define a* **distance metric between state transition kernels** Q_t^c **and** Q_t^0 *(a metric for the* **rate of change***), for a given observation* Y_t, $D_{Q,Y_t}(Q_t^c, Q_t^0)$, *as the following distance between* R_{t,Y_t}^c, R_{t,Y_t}^0*:*

$$\begin{aligned} D_{Q,Y_t}(Q_t^c, Q_t^0) &\triangleq D_R(R_{t,Y_t}^c, R_{t,Y_t}^0) \\ &\triangleq \sup_x \int_E |R_{t,Y_t}^c(x,x') - R_{t,Y_t}^0(x,x')| dx' \\ &= \sup_x \int_E \psi_{t,Y_t}(x') |Q_t^c(x,x') - Q_t^0(x,x')| dx' \end{aligned}$$

It is easy to show that, for a given observation Y_t, D_R *and hence* D_Q *satisfy the properties of a metric over the space of transition kernels. We use* $D_{Q,t}$ *in the rest of the chapter, to denote* $D_{Q,Y_t^c}(Q_t^c, Q_t^0)$ *for ease of notation.*

Definition 7 *We define the* **vector of rates of change**, $\underline{D_Q}$ *as*

$$\underline{D_Q} \triangleq [D_{Q,t_c}, ...D_{Q,k}, ...D_{Q,t_f}] \tag{2.23}$$

Definition 8 *The* **total model error in the posterior** *is defined as the total variation norm of the difference between the posteriors evaluated using the correct and the incorrect model, scaled by $\lambda^c_{k,Y^c_k}(E)$ where λ^c_{k,Y^c_k} is the invariant measure [5] corresponding to R^c_{k,Y^c_k}* [12]*:*

$$\tilde{D}_{t,Y_{0:t}} \triangleq \lambda^c_{k,Y^c_k}(E)||\pi^{c,0}_t - \pi^{c,c}_t|| \tag{2.24}$$

$\tilde{D}_{t,Y_{0:t}}$ is a temporary variable used to write the lemmas more clearly. We show that $\tilde{D}_t \triangleq \tilde{D}_{t,Y_{0:t}}$ is also upper bounded by an "Alpha function" (defined below) of $\underline{D_Q}$ and use this to prove that the total error $e^{c,0,N}_t$ is upper bounded by an Alpha function of $\underline{D_Q}$.

Definition 9 *We say that a function $\alpha(z)$ belongs to the* **"Alpha functions' class"** *if it is an increasing function of z and its derivatives w.r.t. z of all orders are also increasing functions. Note z can be a scalar or a vector but $\alpha(z)$ is a scalar.*

We state here a lemma for the Alpha functions' class[13] which we use to prove later lemmas

Lemma 3 (Composition Lemma): *The composition of two Alpha functions is also an Alpha function, i.e. if $\alpha_1(x, z), \alpha_2(y)$ are Alpha functions of their arguments, then their composition function $\alpha_1(x, \alpha_2(y))$ is also an Alpha function of $[x, y]$.*

[12] We scale by $\lambda^c_{k,Y^c_k}(E)$ only for ease of notation in stating theorems

[13] We are not sure if this class of functions the composition lemma given below already exist in literature

Proof: See Appendix

Now, we need to show that θ_t, β_t are upper bounded by Alpha functions of $\underline{D_Q}$. This will follow if we can show a similar result for $\delta_k,\ \rho_k,\ \forall k \geq t_c$. We first show in lemma 4 that $\delta_k,\ \rho_k$ are upper bounded by Alpha functions of $[D_{Q,k}, \tilde{D}_{k-1}]$. Then in lemma 5, we use a mathematical induction argument and the composition lemma (lemma 3) to show that $\tilde{D}_{k-1}$ and δ_k are upper bounded by an Alpha function of $\underline{D_Q}$ for all k. The Alpha function bound on ρ_k (in lemma 5) follows from the Alpha function bound on $\tilde{D}_{k-1}$ and the composition lemma.

Lemma 4 *Defining*

$$\begin{aligned} A_k &\triangleq R^c_{k,Y^c_k}(\pi^{c,0}_{k-1})(E), \quad \textit{and} \\ C &\triangleq R^c_{k,Y^c_k}(\pi^{c,c}_{k-1})(E) \end{aligned} \tag{2.25}$$

and assuming

$$C > \frac{(\tilde{D}_{k-1})}{\epsilon^c_k} + D_{Q,k},\ \forall k \tag{2.26}$$

the following hold:

$$\delta_k \leq \frac{2D_{Q,k}}{A_k} \leq \frac{2D_{Q,k}}{C - \frac{\tilde{D}_{k-1}}{\epsilon^c_k}} \triangleq \tilde{\alpha}_{\delta,k}([D_{Q,k}, \tilde{D}_{k-1}]) \tag{2.27}$$

$$\begin{aligned} \rho_k \leq \frac{\sup_x \psi_{k,Y_k}(x)}{\epsilon^{c,0^2}_k (A_k - D_{Q,k})} &\leq \frac{\sup_x \psi_{k,Y_k}(x)}{\epsilon^{c,0^2}_k (C - \frac{\tilde{D}_{k-1}}{\epsilon^c_k} - D_{Q,k})} \\ &\triangleq \tilde{\alpha}_{\rho,k}([D_{Q,k}, \tilde{D}_{k-1}, \frac{1}{\epsilon^{c,0}_k}]),\ \textit{a.s.} \end{aligned} \tag{2.28}$$

i.e. δ_k and ρ_k are upper bounded by Alpha functions of $[D_{Q,k}, \tilde{D}_{k-1}, \frac{1}{\epsilon_k^{c,0}}]$[14].

Proof: See Appendix

Lemma 5 *Assuming the inequality in (2.26), the following hold:*

$$\tilde{D}_t \leq \alpha_{\tilde{D},t}(\underline{D_Q}), \ \forall t \geq t_c \tag{2.29}$$

$$\delta_t \leq \alpha_{\delta,t}(\underline{D_Q}), \ \forall t \geq t_c \tag{2.30}$$

$$\rho_t \leq \alpha_{\rho,t}(\underline{D_Q}, \frac{1}{\epsilon_t^{c,0}}), \ \forall t \geq t_c \tag{2.31}$$

i.e. $\tilde{D}_t$ and also δ_t, ρ_t are upper bounded by Alpha functions of $\underline{D_Q}$.

Proof: We use mathematical induction to prove (2.29) and (2.30). (2.31) then follows from (2.28), (2.29) and Lemma 3. First note that $\tilde{D}_t = 0 = \delta_t, \ \forall t < t_c$. The base case, $t = t_c$, is true since

$$\delta_{t_c} \leq \frac{2D_{Q,t_c}}{C} \triangleq \alpha_{\delta,t_c}(\underline{D_Q}) \tag{2.32}$$

$$\tilde{D}_{t_c} = ||\pi_{t_c}^{c,0} - \bar{R}_{t_c}^c \pi_{t_c-1}^0|| = ||\pi_{t_c}^{c,0} - \bar{R}_{t_c}^c \pi_{t_c-1}^{c,0}|| \leq \frac{2D_{Q,t_c}}{C} \triangleq \alpha_{\tilde{D},t_c}(\underline{D_Q}) \tag{2.33}$$

Inequality (2.32) follows from (2.27) by putting $\tilde{D}_{t_c-1} = 0$. The last inequality of (2.33) follows by applying (7.22) from Appendix with $\tilde{D}_{t_c-1} = 0$

Now, assume that (2.29) and (2.30) hold for $t_c \leq k \leq (t-1)$, i.e. assume that

$$\tilde{D}_{t-1} \leq \alpha_{\tilde{D},t-1}(\underline{D_Q}) \tag{2.34}$$

$$\delta_k \leq \alpha_{\delta,k}(\underline{D_Q}), \ \forall t_c \leq k \leq (t-1). \tag{2.35}$$

[14] Note that ϵ_k^c is not a function of the rate of change and hence we treat it as a constant in this entire analysis

By (2.27) of lemma 4, this implies that

$$\delta_t \leq \frac{2D_{Q,t}}{C - \frac{\alpha_{\tilde{D},t-1}(\underline{D_Q})}{\epsilon_t}} \triangleq f(\underline{D_Q}) \tag{2.36}$$

Now it is easy to see that $f(\underline{D_Q}) = \alpha_1(D_{Q,t}, \alpha_2(\underline{D_Q}^{t-1}))$ is a composition of two Alpha functions, $\alpha_1(D_{Q,t}, z) = \frac{2D_{Q,t}}{C-z}$ [15] and $\alpha_2(\underline{D_Q}^{t-1}) = \alpha_{\tilde{D},t-1}(\underline{D_Q}^{t-1})$. Using Lemma 3 (Composition lemma), the composition of two Alpha functions is also an Alpha function. Thus, $f(\underline{D_Q}) = \alpha_{\delta,t}(\underline{D_Q})$. Now, by Theorem 4.6 of [5], we have that

$$\tilde{D}_t \leq \delta_t + \frac{\delta_{t-1}}{\epsilon_t^{c2}} + \sum_{k=t_c}^{t-2} \tilde{\tau}_{t:k+2} \frac{\delta_k}{{\epsilon_{k+1}^{c}}^{2}} \tag{2.37}$$

Also, we have from (2.35) and (2.36) that each of the $\delta_k, k = t_c, ..t$ is upper bounded by an Alpha function. Hence it is easy to see that $\tilde{D}_t$ is also upper bounded by an Alpha function, $\alpha_{\tilde{D},t} \triangleq \alpha_{\delta,t} + \frac{\alpha_{\delta,t-1}}{\epsilon_t^{c2}} + \sum_{k=t_c}^{t-2} \tilde{\tau}_{t:k+2} \frac{\alpha_{\delta,k}}{{\epsilon_{k+1}^{c}}^{2}}$. Thus we have proved that (2.29) and (2.30) hold for t given that they hold for all $t_c \leq k \leq (t-1)$. We showed the base case, $t = t_c$, in (2.32) and (2.33). Hence by Mathematical Induction, (2.29) and (2.30) hold for all $t \geq t_c$.

The third equation, (2.31), follows directly by combining (2.28), (2.29) and the composition lemma (lemma 3).

The main result of this section given below follows as a corollary of the above lemmas.

Theorem 3 ("Rate of Change" bound)

Assuming the inequality (2.26), the following results hold:

1. *Both the modeling error, $\theta_t(\delta_k, \epsilon_k^c, t_c \leq k \leq t)$, and the PF approximation error, $\beta_t(\rho_k, \epsilon_k^{c,0}, 0 \leq k \leq t)$, are upper bounded by Alpha functions of the vector of rates*

[15] It is easy to see that $\frac{2}{(C-z)}$ is an Alpha function

of change, $\underline{D_Q}$, and consequently the total error $e_t^{c,0,N} = M_t\theta_t + \frac{M_t\beta_t}{\sqrt{N}}$ is also upper bounded by an Alpha function of $\underline{D_Q}$. Also $e_t^{c,0,N}$ increases with t as long as the change persists.

2. *The observation likelihood is upper bounded by an* **increasing function** *(note, it is not an Alpha function of $\underline{D_Q}$) of the vector of rates of change, $\underline{D_Q}$, i.e.*

$$OL_t^{c,0} \leq -\log(A_t - D_{Q,t}) \leq -\log(C - \frac{\tilde{D}_{t-1}}{\epsilon_t^c} - D_{Q,t}) \leq -\log(C - \alpha_{\tilde{D},t-1}(\underline{D_Q}) - D_{Q,t}) \tag{2.38}$$

Proof: Part 1 follows from the definitions of θ_t, β_t (equations (2.11) and (2.14)), lemma 5 and the following two facts: (a) ϵ_k^c is independent of $D_{Q,k}$ and (b) $\epsilon_k^{c,0}$ is a decreasing function of the rate of change (We do not have a proof for this in the general case). The intuition is that with increasing rate of change, the overlap between Y_k^c and the spread of Q_k^0 decreases and so the kernel $R_k^{c,0}$ becomes less mixing ($\epsilon_k^{c,0}$ decreases)

For part 2, the first inequality of (2.38) follows by applying (7.20) (in Appendix), the second one follows by (7.21) (in Appendix) and the third inequality follows from (2.29).

Thus we have shown that a small rate of change implies that $OL^{c,0}$ is small (hence does not detect the change). But it also implies that ELL estimation error, $e_t^{c,0,N}$, is small, which implies that ELL will detect the change as soon as it becomes “detectable” (defined in Definition 5).

The Alpha function nature of the bound on ELL approximation error implies that ELL *is approximated accurately for slow changes, and for some time (until total change magnitude is small) but the error blows up quickly to infinity with increasing rate of change ($D_{Q,k}$) or increasing total change magnitude $\tilde{D}_{k-1}$*. We discuss the implications of this fact in section 2.7.4.

2.6 Complementary behavior of ELL and OL

We quantify the complementary behavior of ELL and OL by bounding the ELL approximation error by an increasing function of OL. First consider the PF error coefficient, $\beta_t^{c,0}$. It depends on past values of $\rho_k^{c,0}$ and on $\epsilon_k^{c,0}$. Using Remark 5.10 of [5], we have the following upper and lower bounds on ρ_k which can be expressed in terms of $OL_k^{c,0}$:

$$\begin{aligned} \frac{\sup_{x \in E_{x,Y_t}} \psi_k^c(x)}{(Q_k^0 \pi_{k-1}^{c,0}, \psi_k^c)} &\leq \rho_k^{c,0} \leq \frac{\sup_{x \in E_{x,Y_t}} \psi_k^c(x)}{(\epsilon_k^{c,0})^2 (Q_k^0 \pi_{k-1}^{c,0}, \psi_k^c)} \\ \Rightarrow \frac{\sup_{x \in E_{x,Y_t}} \psi_k^c(x)}{e^{-OL_k^{c,0}}} &\leq \rho_k^{c,0} \leq \frac{\sup_{x \in E_{x,Y_t}} \psi_k^c(x)}{(\epsilon_k^{c,0})^2 e^{-OL_k^{c,0}}} \end{aligned} \qquad (2.39)$$

Now consider the model error, $\theta_t^{c,0}$. It depends on past values of $\delta_k^{c,0}$ and ϵ_k^c. We use inequality (6) of [5] which states that

$$|\bar{\mu} - \bar{\mu}'| \leq \frac{||\mu - \mu'||}{\mu(E)} + \frac{|\mu(E) - \mu'(E)|}{\mu(E)}. \qquad (2.40)$$

Taking $\bar{\mu} = \bar{R}_k^{c,0}(\pi_{k-1}^{c,0})$ and $\bar{\mu}' = \bar{R}_k^{c,c}(\pi_{k-1}^{c,0})$ and using inequalities (7.19) and (7.20) from the Appendix, we can bound $\delta_k^{c,0}$ in terms of $OL_k^{c,0}$:

$$\delta_k^{c,0} \leq \frac{2D_{Q,k}}{e^{-OL_k^{c,0}}} \qquad (2.41)$$

where $D_{Q,k}$ is defined in the previous section (Definition 6) as a metric for the rate of change. Thus we have the following theorem:

Theorem 4 (ELL-OL Complementariness)

1. *The ELL approximation error at time t, $e_t^{c,0,N} \triangleq M_T \theta_t^{c,0} + \frac{M_t \beta_t^{c,0}}{\sqrt{N}}$ is upper bounded by an increasing function of past values of $OL_k^{c,0}$ and past values of $D_{Q,k}$, $\frac{1}{\epsilon_k^{c,0}}$, i.e.*

$$e_t^{c,0,N} \quad \leq \quad \sum_{k=t_c}^{t} e^{OL_k^{c,0}} \omega(\frac{1}{\epsilon_k^{c,0}}, D_{Q,k}) \tag{2.42}$$

   where ω is an increasing function of its arguments and is defined by upper bounding $\theta_t^{c,0}$ and $\beta_t^{c,0}$ using the bounds given in (2.39) and (2.41) respectively.

2. *The PF error in ELL approximation is lower bounded by an increasing function of $OL_k^{c,0}$, i.e.*

$$\beta_t^{c,0} \geq \sum_{k=t_c}^{t} e^{OL_k^{c,0}} (\sup_{x \in E_{x,Y_k}} \psi_k^c(x)) \tilde{\omega}(\frac{1}{\epsilon_k^{c,0}}) \tag{2.43}$$

Proof: The proof of part 1 follows directly by combining the definitions of $\theta_t^{c,0}$ and $\beta_t^{c,0}$ given in (2.11) and (2.14) with (2.39) and (2.41). Proof of part 2 follows directly from (2.39).

Now, if a certain change is not detected by OL until time t, it means that all values of OL, $OL_{t_c}^{c,0}, ...OL_k^{c,0}, ...OL_t^{c,0}$ are small (below threshold). This implies, by the above theorem, that the bound on the ELL approximation error is also small or that ELL is approximated accurately. Thus the change will get detected by ELL once its magnitude becomes large enough to satisfy the "detectability" condition (definition 5 in Section 2.3). Conversely, if ELL does not detect a change that is "detectable", it means that the ELL approximation error is large. By the above theorem this implies that at least one of $OL^{c,0}_{t_c}, ...OL^{c,0}_k, ...OL^{c,0}_t$ is large and hence OL will detect the change. Thus, we *propose to use a combination of ELL and OL to detect a change when the rate of change can be slow or fast and change parameters are unknown.* A change should be declared when either ELL or OL exceed their respective threshold.

Theorem 4.2 implies that if any of $OL^{c,0}_k$ is large, the PF error in ELL approximation (and hence also the total error) is large. Thus large values of $OL^{c,0}_k$ indicate that the ELL estimates obtained are unreliable. As an example, when OL values become very large (infinity due to computer overflow), ELL completely fails to detect the change (see the r=5 plot in figure 2.1(a) and (b)).

2.7 Discussion

2.7.1 An Example

We first discuss a simple example of a nonlinear HMM which illustrates all the points made in this chapter. Consider the case where Q^0_t, Q^c_t and π_0 are linear Gaussian, so that p^0_t and p^c_t are also Gaussian. Assume scalar state and observation and let π_0 be zero mean with zero variance. Let the pdf of $Q_t(x, dx')$ is $\mathcal{N}(x, \sigma^2_{sys})$ and pdf of $Q^c_t(x, dx')$ is $\mathcal{N}(x + \Delta a, {\sigma^c_{sys}}^2)$ with $\sigma^c_{sys} = 0.25\sigma_{sys}$. Also assume that the changed system model lasts for a finite time

$[t_c, t_f]$. Thus $p_t^0(x)$ is $\mathcal{N}(0, \sigma_t^2)$ with $\sigma_t^2 = t\sigma_{sys}^2$ and $p_t^c(x)$ is $\mathcal{N}(a_t, \sigma_t^{c2})$ with $a_t = 0$, $\sigma_t^{c2} = t\sigma_{sys}^2$, $\forall t < t_c$, $a_t = (t - t_c + 1)\Delta a$, $\sigma_t^{c2} = t_c\sigma_{sys}^2 + (t - t_c + 1){\sigma_{sys}^c}^2$, $\forall t_c \le t \le t_f$ and $a_t = a_{t_f}$, $\sigma_t^{c2} = t_c\sigma_{sys}^2 + (t_f - t_c + 1){\sigma_{sys}^c}^2 + (t - t_f)\sigma_{sys}^2$ $\forall t > t_f$. Thus even though the change lasts for a finite time, its effect on $p_t(x)$ is permanent ($p_t^c(x)$ has mean a_{t_f} $\forall t > t_f$).

We consider a simple observation model $Y_t = h(X_t) + w_t$ with $h(x) = x^3$. We let w_t be truncated Gaussian observation noise with variance σ_{obs}^2 and truncation parameter, $B < \infty$. A truncated Gaussian observation noise, and the fact that h^{-1} is continuous, makes the support set of $\psi_k(x)$ compact (a continuous function maps a compact set into another compact set [64]). By the argument given in Example 3.10 of [5] (explained in Section 2.7.2), this along with the fact that π_0 has finite (zero) support makes the unnormalized filter kernels, $R_t^0, R_t^{c,0}, R_t^c$, mixing, even though the state transition kernels Q_t^0, Q_t^c are not mixing. Also, $\sup_{x \in E_{x,Y_t}} \psi_k(x) = \frac{1}{\sqrt{2\pi}\sigma_{obs}} < \infty$ and change lasts for a finite time. For this example, we have $M_t = \sup_{x \in E_{x,Y_t}} [-\log p_t(x)] = \sup_{x \in h^{-1}([Y_t - B, Y_t + B])} [-\log p_t(x)] = -\log p_t((|Y_t| + B)^{1/3})$. Thus we satisfy all assumptions for Theorem 2.2.

Also, we can show that this example satisfies assumption (iv)$'$ and hence Theorem 2.1 also holds. This is shown as follows: Consider $E_{Y_{1:t}}[|K(\pi_t^c : p_t) - K(\pi_t^c : p_t^M)|]$. By definition of p_t^M, $K(\pi_t^c : p_t) > K(\pi_t^c : p_t^M)$ $\forall Y_{1:t}$ and so

$$\begin{aligned} E_{Y_{1:t}}[|K(\pi_t^c : p_t) - K(\pi_t^c : p_t^M)|] &= E_{Y_{1:t}}[K(\pi_t^c : p_t) - K(\pi_t^c : p_t^M)] \\ &= E_{Y_{1:t}}[K(\pi_t^c : p_t)] - E_{Y_{1:t}}[K(\pi_t^c : p_t^M)] \\ &= K(p_t^c : p_t) - K(p_t^c : p_t^M) \triangleq err(M, t) \qquad (2.44) \end{aligned}$$

Now for this example, p_t^c and p_t are both Gaussian and hence $err(M, t)$ simplifies to (w.l.o.g.

assume $a_t > 0$)

$$err(M,t) = K(p_t^c : p_t) - K(p_t^c : p_t^M) = 2\int_{\sqrt{M}\sigma_t^c}^{\infty} \frac{x^2}{2\sigma_t^2} \frac{1}{\sqrt{2\pi}\sigma_t^c} e^{-\frac{(x-a_t)}{2\sigma_t^{c2}}} dx \tag{2.45}$$

Set $y = \frac{x-a_t}{\sigma_t^c}$. Now in this example, $\sigma_t \geq \sigma_t^c$ $\forall t$ and hence we have

$$\begin{aligned} err(M,t) &\leq \int_{\sqrt{M}-\frac{a_t}{\sigma_t^c}}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-y^2/2} (y + \frac{a_t}{\sigma_t^c})^2 dy \\ &\leq \int_{\sqrt{M}}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-y^2/2} (y + \frac{a_t}{\sigma_t^c})^2 dy \\ &= \int_{\sqrt{M}}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-y^2/2} (y^2 + \frac{a_t^2}{\sigma_t^{c2}} + 2\frac{a_t y}{\sigma_t^c}) dy \end{aligned} \tag{2.46}$$

Now the above is an increasing function of a_t and a decreasing function of σ_t^c. Also, we know that $a_t = a_{t_f}$ $\forall t > t_f$. Thus $a_t \leq a_{t_f}$ $\forall t$. Also, $\sigma_t^c \geq \sigma_1^c = \sigma_{sys}$ $\forall t$. Thus we have

$$err(M,t) \leq 2\int_{\sqrt{M}}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-y^2/2} (y^2 + \frac{a_{t_f}^2}{\sigma_{sys}{}^2} + 2\frac{a_{t_f} y}{\sigma_{sys}}) dy \triangleq err^*(M) \tag{2.47}$$

and $err^*(M)$ is independent of t. Also $\lim_{M\to\infty} err^*(M) = 0$ i.e. for any error Δ, we can find an M_Δ s.t. $err^*(M) < \Delta$ $\forall M \geq M_\Delta$ *and* $\forall t$. This proves that assumption (iv)$'$ holds.

Now to analyze the performance, first assume that no errors are present and do the analysis of Section 2.3.2. Assume $t_c \approx 0$ to simplify expressions. Then we get

$$\begin{aligned} EK_t^0 = K(p_t^0 : p_t^0) &= 0.5\log 2\pi\sigma_t^2 + 0.5 \\ EK_t^c = K(p_t^c : p_t^0) &= 0.5\log 2\pi\sigma_t^2 + 0.5\frac{\sigma_t^{c2} + a_t^2}{\sigma_t^2} \\ &\approx 0.5\log 2\pi\sigma_t^2 + 0.031 + 0.5\frac{a_t^2}{t\sigma_{sys}^2}, \ \forall t \leq t_f \\ &\approx 0.5\log 2\pi\sigma_t^2 + 0.5[0.062\frac{t_f+1}{t} + \frac{t-t_f-1}{t}] + 0.5\frac{a_{t_f}^2}{t\sigma_{sys}^2} \ \forall t > t_f \end{aligned}$$

$$VK_t^0 \leq E_{p_t^0}[[-\log p_t^0(x)]^2] - (EK_t^0)^2 = 0.5$$

$$\begin{aligned}
VK_t^c &\leq E_{p_t^c}[[-\log p_t^0(x)]^2] - (EK_t^c)^2 \\
&= 0.5\frac{\sigma_t^{c4}}{\sigma_t^4} + \frac{\sigma_t^{c2} a_t^2}{\sigma_t^4} \\
&\approx 0.002 + \frac{0.062 a_t^2}{t\sigma_{sys}^2} \ \forall t \leq t_f \\
&\approx 0.5[0.062\frac{t_f+1}{t} + \frac{t-t_f-1}{t}]^2 + [0.062\frac{t_f+1}{t} + \frac{t-t_f-1}{t}]\frac{a_{t_f}^2}{t\sigma_{sys}^2}
\end{aligned}$$

The threshold $\kappa_t = EK_t^0 + 3\sqrt{VK_t^0} \leq 2.62$. We set $\kappa_t = 2$. The mean distance of K_t^c from the threshold is then:

$$\gamma_t \triangleq EK_t^c - \kappa_t = 0.5\frac{a_t^2}{\sigma_t^2} + 0.031 - 2 \approx 0.5\frac{a_t^2}{\sigma_t^2} - 2 \tag{2.48}$$

Now consider $t \leq t_f$. We can then apply definition 5 (first assuming no approximating errors) to infer the following: The miss probability at time t will surely be less than 0.11 (0.05 if unimodal) if $\gamma_t > 3\sqrt{VK_t^c}$ which simplifies to $0.5r^2 - 2 > .75r$ with $r = a_t/\sigma_t$. It is easy to see that this equation is satisfied for $r \geq 3$. Now $t_c \approx 0$, and so $r \approx \frac{\sqrt{t}\Delta a}{\sigma_{sys}}$. This implies that if the rate of change is of the order of system noise, $\Delta a \approx \sigma_{sys}$, then with probability greater than 0.89, the change will get detected in $(3)^2 = 9$ time units or more. This of course is obtained using loose bounds (loose variance bound and the loose Chebyshev or Gauss's inequality bound) and in practice changes can get detected much faster if there are no approximation errors. Infact even with approximation errors, we see in simulations that the change gets detected faster than this (see figure 2.2(a)). Approximation errors tend to reduce the value of ELL[16].

[16] In the extreme case (for drastic changes) the PF completely loses track, i.e. the unnormalized filter kernel starts following the system model, $R_t^{c,0} \approx Q_t^0$ causing ELL to not increase above the normal value (see figure 2.1(a),r=5).

Now we analyze the effect of approximation errors. Applying definition 5 while taking into account the approximation errors, we get: A change will get detected w.p. greater than $(1 - 0.11) = 0.89$, if $\gamma_t - M_t\theta_t^{c,0} > 3\sqrt{VK_t^c}$ (assuming $\frac{M_t\beta_t}{\sqrt{N}}$ can be made small enough by taking N large enough). Now, assuming as before that $t_c \approx 0$, we have

$$\begin{aligned} \gamma_t &= 0.5t\frac{(\Delta a)^2}{{\sigma_{sys}}^2} + 0.0031 - 2 \quad \forall\, t_c \le t \le t_f \\ \gamma_t &> \frac{0.5(a_{t_f})^2}{t{\sigma_{sys}}^2} - 2 \quad \forall\, t > t_f \end{aligned} \tag{2.49}$$

Also, $\delta_k = 0, \forall\ k < t_c,\ k > t_f$. For simplicity, assume $\delta_k = \delta, \forall\ t_c \le k \le t_f$, then we have

$$\theta_t = \delta + \frac{\delta}{(\epsilon_t)^2} + \sum_{k=t_c}^{t-2} (\tau_k)^{(t-k-2)}\frac{\delta}{(\epsilon_k)^4} \tag{2.50}$$

From the above two equations, we see that both γ_t and θ_t increase till t_f. γ_t has an approximately linear increase (for small t_c), $\Delta\gamma_t \approx \frac{0.5(\Delta a)^2}{{\sigma_{sys}}^2} = 0.5$, while θ_t increases at decreasing rates of increase[17], $\Delta\theta_t = \tau_t\Delta\theta_{t-1}$. Now, if the change is slow enough so that $\frac{M_t\delta}{\epsilon^4} < 0.5$, then $\gamma_t - M_t\theta_t$ will increase with time until t_f and the change will get detected when $\gamma_t - M_t\theta_t$ exceeds zero.

After $t_f + 1$, both start decreasing but γ_t decreases as $\Delta\gamma_t \approx -\frac{0.5(a_{t_f})^2}{t^2\sigma_{sys}^2}$ (See blue solid line in figure 2.1 (c)) while θ_t decreases as $\theta_t = \tau_t\theta_{t-1}$ so that $\Delta\theta_t = -(1-\tau_t)\theta_{t-1}$ (large decreases for large current value). The initial decrease in θ_t is usually faster than the decrease in γ_t in which case $\gamma_t - M_t\theta_t$ increases with time even after $t_f + 1$ and in some cases the change can get detected even after t_f.

In practice, the assumption of PF error being negligible may not hold when tracking

[17] θ_t goes as $\delta, \delta + \delta/\epsilon^2, \delta + \delta/\epsilon^2 + \delta/\epsilon^4, \delta + \delta/\epsilon^2 + \delta/\epsilon^4 + \tau\delta/\epsilon^4, \delta + \delta/\epsilon^2 + \delta/\epsilon^4 + \tau\delta/\epsilon^4 + \tau^2\delta/\epsilon^4 ...$

changed system observations, using a PF optimal for the original system, since N has been fixed for the original system's observations and with increasing rate of change or increasing total change, the PF error coefficient blows up very quickly (shown in Section 2.5).

2.7.2 Sufficient Conditions for Mixing Unnormalized Filter Kernels

From Example 3.10 of [5], we can get the following sufficient conditions for R_t to be mixing:

1. π_0 has compact support

2. *and* $\psi_t(x)$ has compact support. A sufficient condition for this to hold is that w_t has finite support, say $[-B, B]$ (e.g. truncated Gaussian noise) and $E_{x,Y_t} \triangleq h_t^{-1}([Y_t - B, Y_t + B])$ is compact. A sufficient condition for this is that h_t is invertible and h_t^{-1} is continuous (h_t is a homeomorphism) [64].

3. *and* given that the state transition kernel has the form $X_t = f_t(X_{t-1}) + n_t$, $f_t^{-1}(E_{x,Y_t})$ is a compact set. A sufficient condition for this is that f_t is a homeomorphism [64].

Now condition 2 is equivalent to assumption (iv)$''$ in Theorem 2 (posterior state space is compact). Thus if the above three conditions hold, the change lasts for a finite time and E_{x,Y_t} has a nonzero measure (implies assumption (ii) holds) then Theorem 2.2 holds.

Sufficient conditions for Theorem 2.1 are all the three conditions above and the fact that p_t^c and p_t are Gaussian, $\frac{\sigma_t^c}{\sigma_t^0}$ is bounded away from zero and the change is an additive bias lasting for a finite time (a_{t_f} is finite). These sufficient conditions follow directly by generalizing the example in Section 2.7.1.

2.7.3 Generalizations

The results proved in this chapter for ELL approximation errors can be generalized at two levels. First, all results of Sections 2.4 and 2.5 and 2.6 are true for any function of the state, i.e. $[-\log p_t(x)]$ can be replaced by any other function $f(x)$. Second, $D_{Q,t}$ which measures the "rate of change" here can in general be a metric for system model error per time step (the error being introduced due to any reason). As long as the system model error lasts for a finite time, the results of this chapter will apply directly.

Thus Theorems 1 and 2 can be applied to errors in approximating the posterior estimate of any function of state given past observations, when using a PF with system model error. Note that the posterior estimate of a function of state conditioned on past observations is an MMSE (Minimum Mean Square Error) estimate of the function evaluated based on past observations.

Also, Theorem 3 can be generalized to prove that the ELL approximation error (or approximation error in MMSE estimate of any function f of the state) is upper bounded by an Alpha function of the vector of system model errors per time step, $\underline{D_Q}$. The implication of this is that in situations where slow changes might occur in the system model, using a more "general" system model introduces less total error. More "general" means that instead of using $Q_k^{pf} = Q_k^0$ (making the filter optimal for the original system model), one can use a Q_k^{pf} with a much larger system noise variance than Q_k^0. The effect of doing this is that the distance of Q_k^{pf} from Q_k^c decreases, even though its distance from Q_k^0 is no longer 0. We discuss below in Section 2.7.4 that doing this results in less total error.

The results of Theorem 4.2 can be used as follows: The error in MMSE estimate of any function of the state cannot be measured but because of the result in Theorem 4.2, we can use an estimate of OL to decide when the errors are large.

2.7.4 Implications of "rate of change" bound on error

Theorem 3 given in Section 2.5 has many interesting implications. The most obvious one is that a small rate of change implies that $OL^{c,0}$ is small (hence does not detect the change). But it also implies that the ELL estimation error, $e_t^{c,0,N}$, is small, which implies that ELL will detect the change as soon as it becomes "detectable". Also, the Alpha function nature of the bound on the ELL approximation error implies that *ELL is approximated accurately for slow changes, and for some time (until total change magnitude is small) but the error blows up quickly to infinity with increasing rate of change ($D_{Q,k}$) or increasing total change magnitude $\tilde{D}_{k-1}$*. This has the following implications:

1. For change detection to work best (detect change with minimum delay and minimum false alarms), the error in approximating ELL should be small for approximating both $ELL(Y_{0:t}^c)$ and $ELL(Y_{0:t}^0)$. Now in the current framework, error in $ELL(Y_{0:t}^0)$, $e_t^{0,N}$ is quite small (for N chosen large enough) since there is no system model error ($Q_k^{pf} = Q_k^0$)

which implies that $\theta_t^0 = \delta_t^0 = 0$ and the bound on ρ_t is also small, i.e.

$$\theta_t^0 = \delta_t^0 = 0 \;\forall t, \quad \text{and} \quad \rho_t^0 \leq \frac{\sup_x \psi_{t,Y_t}(x)}{\epsilon_t^2 C}. \tag{2.51}$$

But the error in $ELL(Y_{0:t}^c)$, $e_t^{c,0,N}$, depends on the smallness of the system model error, $\underline{D_Q}$. The nonlinearity of the error bounds as a function of $\underline{D_Q}$, suggests that *if the system model error $D_{Q,t}$ was divided equally between the unchanged and changed systems, (i.e. the Q_t used by the particle filter was not equal to Q_t^0, but chosen so that its distance from changed and unchanged system was equal), the bound on the total error $e_t^{c,0,N} + e_t^{0,N}$ would be smaller.* For example, consider the first change time instant, $t = t_c$, in the current framework. For the changed system, $\delta_{t_c}^{c,0} \leq \frac{2D_{Q,t_c}}{C}$ and $\rho_{t_c}^{c,0} \leq \frac{\sup_x \psi_{k,Y_k}(x)}{\epsilon_{t_c}^2 (C - D_{Q,t_c})}$. Thus

$$\delta_{t_c}^{c,0} + \delta_{t_c}^0 \;\; \leq \;\; \frac{2D_{Q,t_c}}{C}, \quad \rho_{t_c}^{c,0} + \rho_{t_c}^0 \leq \frac{\sup_x \psi_{t_c,Y_{t_c}}(x)}{\epsilon_{t_c}^2 (C - D_{Q,t_c})} + \frac{\sup_x \psi_{t_c,Y_{t_c}}(x)}{\epsilon_{t_c}^2 C}. \tag{2.52}$$

On the other hand, if the system model error D_{Q,t_c} was divided equally between the unchanged and changed systems (i.e. $D_{Q,t_c}^{0,pf} = D_{Q,t_c}^{c,pf} = \frac{D_{Q,t_c}}{2}$), the bound on $\delta_{t_c}^{c,0} + \delta_{t_c}^0$ would be the same (bound is linear in D_Q) but that on $\rho_{t_c}^{c,0} + \rho_{t_c}^0$ would be smaller. In this case we have[18],

$$\delta_{t_c}^{c,0} + \delta_{t_c}^0 \;\; \leq \;\; \frac{2D_{Q,t_c}}{C}, \quad \rho_{t_c}^{c,0} + \rho_{t_c}^0 \leq \frac{2 \sup_x \psi_{k,Y_k}(x)}{\epsilon_{t_c}^2 (C - \frac{D_{Q,t_c}}{2})} \tag{2.53}$$

The above example indicates that *instead of using Q_k^0 as the transition kernel in particle filtering ($Q_k^{pf} = Q_k^0$), using a Q_k^{pf} that is closer to Q_k^c (even if its distance from Q_k^0*

[18] It is easy to see that the RHS of (2.53) is smaller than that of (2.52)

is not zero) will be a better idea (Note here though that this inference is based on comparing upper bounds with upper bounds). If Q_k^c is known, one could attempt to use a mixture of Q_k^0 and Q_k^c as Q_k^{pf}. For unknown Q_k^c, one could use Q_k^0 with a larger system noise variance as Q_k^{pf}. Both these ideas have been used in past works on tracking using a particle filter [67, 4]; we have in this chapter provided a justification for using them.

2. Another implication of Theorem 3 is that *a sequence of small changes would introduce less total error than one drastic change of the same magnitude.* The most general case of this statement is difficult to prove because of the many free variables involved. But we demonstrate here a simpler case: At $t = t_c$, consider the case where $D_{Q,t_c} = C$. In this case, the bound on ρ_{t_c} is $\frac{S}{\epsilon_{t_c}^{c,0^2}(C-C)} = \infty$. But if the same change was spread over two time instants, i.e. $D_{Q,t_c} = D_{Q,t_c+1} = \frac{C}{2}$, then (assuming $C > \frac{2}{\epsilon_{t_c}^{c,0}}$)

$$\rho_{t_c} + \rho_{t_c+1} \leq \frac{S}{\epsilon_{t_c}^{c,0^2}(C - D_{Q,t_c})} + \frac{S}{{\epsilon_{t_c+1}^{c,0}}^2(C - \frac{\tilde{D}_{t_c}}{\epsilon_{t_c}^{c,0}} - D_{Q,t_c+1})} \tag{2.54}$$

$$\leq \frac{S}{\epsilon_{t_c}^{c,0^2}\frac{C}{2}} + \frac{S}{{\epsilon_{t_c+1}^{c,0}}^2(\frac{C}{2} - \frac{1}{\epsilon_{t_c}^{c,0}})} < \infty \tag{2.55}$$

(2.55) follows by applying (2.33) and the fact that $\frac{2D_{Q,t_c}}{C} = 1$ in this case. This is observed in simulations also. See figure 2.1(a). The change is introduced at $t_c = 5$ in all cases. For $r = 5$ (drastic change), the PF loses track immediately (bound goes to infinity) and the posterior starts following the prior, causing ELL to be very close to that of the normal system (ELL cannot detect this change). For $r = 2$ (slow change), within 3 time instants the total change magnitude is larger than that of $r = 5$ for one time unit. But still the PF does not completely lose track ever and ELL is able to detect this change.

2.7.5 OL and Tracking Error

The OL is approximately equal to tracking error (TE), when the observation noise is white Gaussian and hence either can be used to detect a sudden change. The TE is the square of the Euclidean distance between the current observation Y_t and its prediction based on past observations, $\hat{Y}_t = E_{\pi_{t|t-1}}[h(X_t)]$, i.e.

$$TE = ||Y_t - \hat{Y}_t||^2 \approx E_{\pi_{t|t-1}}[||Y_t - h(X_t)||^2] \tag{2.56}$$

The approximation follows from the first order Taylor series expansion of $E_{\pi_{t|t-1}}[||Y_t - h(X_t)||^2]$ about $X_t = h^{-1}[\hat{Y}_t]$. Now for white Gaussian observation noise with covariance $\Sigma_{obs} = \sigma_{obs}^2 I$, OL can be written as

$$OL(Y_{1:t}) \quad = \quad -\log E_{\pi_{t|t-1}}[\psi_t(X_t)] = -\log E_{\pi_{t|t-1}}[e^{-\frac{||Y_t - h(X_t)||^2}{2\sigma_{obs}^2}}] + K \tag{2.57}$$

Writing the series expansion of the exponential,

$$\begin{aligned} OL(Y_{1:t}) &= -\log E_{\pi_{t|t-1}}[1 - \frac{||Y_t - h(X_t)||^2}{2\sigma_{obs}^2} + O(||Y_t - h(X_t)||^4) + K & (2.58) \\ &\approx -\log(1 - \frac{TE}{2\sigma_{obs}^2} + E_{\pi_{t|t-1}}[O(||Y_t - h(X_t)||^4)]) & (2.59) \\ &= \frac{TE}{2\sigma_{obs}^2} + O(E_{\pi_{t|t-1}}[O(||Y_t - h(X_t)||^4)]^2) & (2.60) \end{aligned}$$

(2.59) follows from (2.56) and (2.60) follows from the series expansion of $\log(1 + z)$. We show results for drastic abnormality detection using the tracking error in chapter 4. As can be seen from figure 4.2, the plots of OL and TE look very similar.

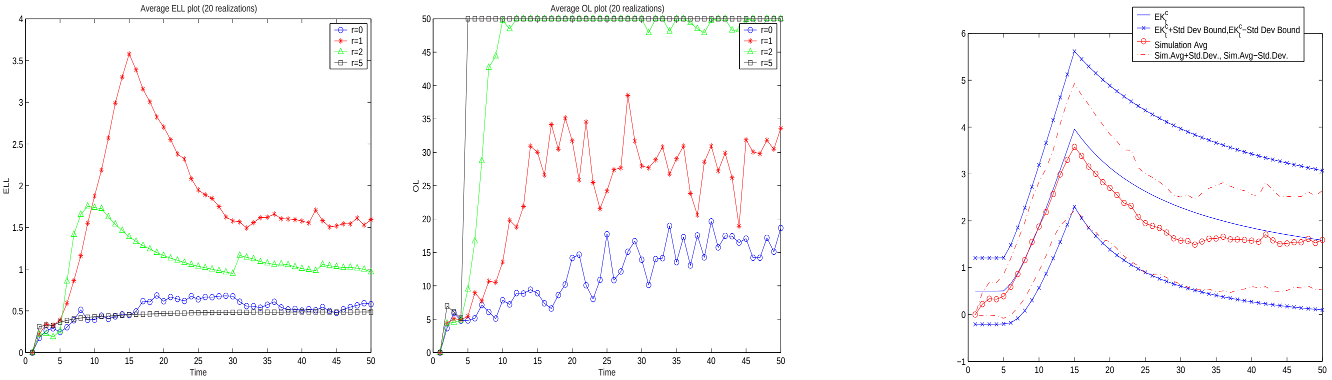
2.8 Simulation Results

We simulated the example given in Section 2.7.1 with truncated Gaussian observation noise with truncation parameter, B=10. We tested for increasing magnitudes of Δa, $\Delta a = r\sigma_{noise}$ with $r = 0$ (no change) and $r = 1, 2, 5$. We show in figure 2.1, plots for detecting the changes using ELL and OL averaged over 20 realizations of The example given in Section 2.7.1. Change was introduced in the system model at $t = 5$ and lasted till $t = 15$ (but as discussed earlier, its effect on state prior was permanent). As can be seen from these graphs all changes are detected by either OL or ELL. The slow change r=1 gets detected by ELL at t=7 but the OL detects it only at t=14. The r=2 ("faster change") gets detected at t=6 using ELL and at t=8 using OL. Also note that when OL takes the value infinity (shown as OL=50 in the plot), due to computer overflow, ELL starts to fail. The $r = 5$ ELL plot in figure 2.1(a) almost coincides with that of $r = 0$ (normal system). This is because when the PF loses track, the posterior starts following the normal system model, i.e. $R_t^c \approx Q_t^0$. But as discussed in earlier sections, the OL detects such a change immediately.

We also show the ROC (Receiver Operating Characteristic) plots in figure 2.2 for the slow, faster and drastic changes which quantify the above discussion. The ROC for a change detection problem [15] plots the average detection delay against the mean time between false alarms by varying the detection threshold. As can be seen from the figure, the ELL works better for $r = 0.5$ and $r = 1$, OL and ELL have comparable performance for $r = 2$ and ELL completely fails but OL works best for $r = 5$.

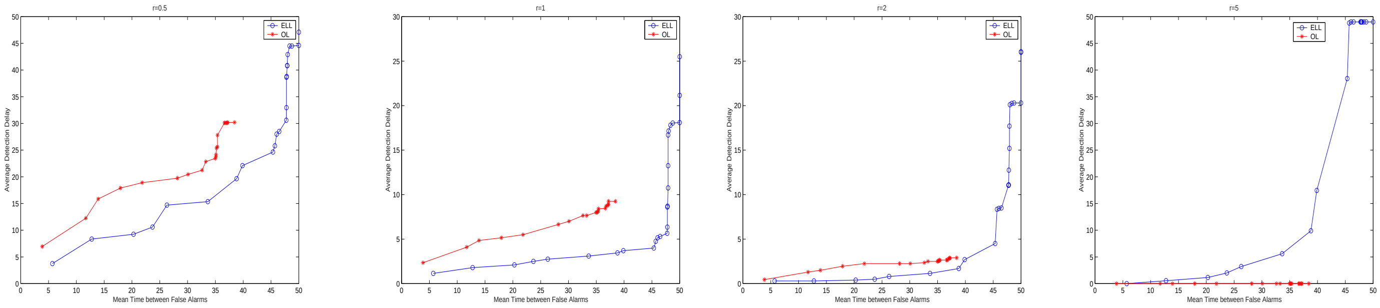
Now as discussed in section 2.7.4, setting the system noise variance in particle filtering to a larger value than that for the unchanged system helps reduce the ELL approximation error and hence improve its detection performance. We experimented with this idea and show results in figure 2.3. We compare the performance of ELL estimated using PFs with increasing σ_{pf} by plotting the ROC curves. As can be seen from the figure, $\sigma_{pf} = 10\sigma_{sys}$ works best for detecting the “drastic change” and also for the “faster change” but it is too large for the slow change, which is intuitive. $\sigma_{pf} = 3\sigma_{sys}$ has best average performance for all the three rates of change.

The application of ELL and OL (or equivalently Tracking Error) to the real problem of abnormal activity detection is discussed in chapter 4. We discuss examples of both the strategies for defining $p_t^0(x)$ (discussed in section 2.3.4) in section 4.1.3.



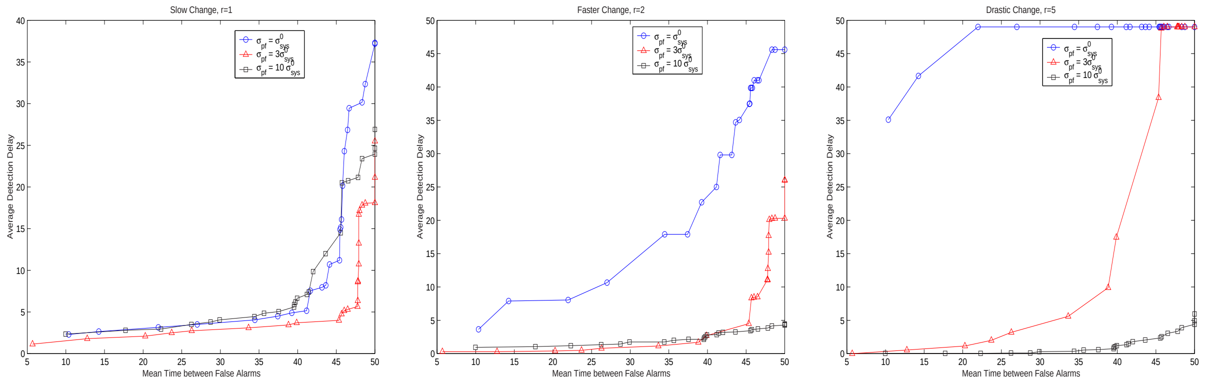
(a)ELL plot (b)OL plot (c)Analytical & Simulation average and spread

Figure 2.1: Simulated example: In (a) and (b), we show ELL and OL (negative log of observation likelihood) plots for the no change case (blue -o), and for changes with $\Delta a = r\sigma_{noise}$ for r=1 (red-*), r=2 (green -△) and r=5 (black -*square*). In all cases change was introduced at time $t_c = 5$ and lasted till $t_f = 15$. The plots are averaged over 20 realizations of observation sequences. For the case $r = 5$ (drastic change) and $r = 2$, the OL plot goes to infinity at or after $t = 5$ (computer overflow) and hence the change is detected immediately using OL while ELL completely fails for it. In (c), we plot $0.5a_t^2/\sigma_t^2$, its simulation average calculated using 20 realizations of the observation sequence and its spread (average plus and minus the standard deviation) for a change with $\Delta a = \sigma_{noise}$. We also plot the theoretical bound on the standard deviation obtained in The example given in Section 2.7.1.



(a) Very Slow Change, r=0.5 (b) Slow Change, r=1 (c) Faster Change, r=2 (d) Drastic Change,r=5

Figure 2.2: ROC curves for comparing performance of ELL and OL for slow and drastic changes



(a) Slow Change

(b) Faster Change

(c) Drastic Change

Figure 2.3: Effect of increasing the system noise variance of the particle filter on performance of ELL

Chapter 3

Landmark Shape Dynamics

3.1 Problem Formulation

We define here dynamical models for representing the changing configurations of landmarks. The distinction between motion and deformation of a deforming and moving configuration is not clear. We separate the dynamics of a deforming configuration into scaled Euclidean motion (translation, rotation, uniform scaling) and non-rigid shape deformations. This idea was inspired by [34]. We define a continuous state HMM for the changing configuration of a group of moving landmarks (point objects) with the shape and motion being the hidden state variables and the noisy configuration vector forming the observation. We refer to it as "shape activity". A *"stationary shape activity"* is defined as one for which the shape vector is stationary i.e. the mean (expected value of) shape remains constant with time and the deformation model is stationary while in a *"non-stationary shape activity"*, the mean shape changes with time (see figure 3.1(a) and (b)).

We discuss in this chapter the stationary, nonstationary and piecewise stationary shape activity models. The entire discussion assumes a fixed number of landmarks. But in certain applications like the airport example with people deplaning (figure 4.1), the number of landmarks varies with time. We currently deal with this by resampling the curve formed by joining the landmarks to a fixed number of points. This is discussed in section 3.7. Also, note that in this representation of shape, the correspondence between landmarks is assumed to be known across frames. Since the number of landmarks is usually small ($k = 8$ in this case), this is easy to ensure. We begin the chapter with a brief review of the definitions and tools for statistical shape analysis and clarifying some notation.

3.2 Preliminaries and Notation

We would first like to clarify that the terms partially observed dynamical model and HMM are used interchangeably for "shape activity" models since the partially observed dynamic model that we define is also an HMM. We use "arg" to denote the angle of a complex scalar as well as in "arg min" for the argument minimizing a function, but the meaning is clear from the context. * is used to denote conjugate transpose. $||.||$ is used for the Euclidean norm of a complex or real vector and $|.|$ for the absolute value of a complex scalar. I_k denotes the $k \times k$ identity matrix and 1_k denotes a k dimensional vector of ones. Also note that to simplify notation we do not distinguish between a random process and its realization. We review below the tools for statistical shape analysis as described in [9].

Definition 10 *[9]* **Configuration** *is a k-tuple (ordered set) of landmarks (which in our case is the k-tuple of point object locations). The* **configuration matrix** *is the $k \times m$ matrix of Cartesian coordinates of the k landmarks in m dimensions. For 2D data ($m = 2$), a more compact representation is a k dimensional complex vector with x and y coordinates forming the real and imaginary parts. The* **configuration space** *is the space of all k-tuples of landmarks i.e. $\Re^{km}$.*

Translation Normalization: The complex vector of the configuration (Y_{raw}) can be centered by subtracting out the centroid of the vector yielding a **centered configuration**, i.e.

$$Y = CY_{raw} \quad where \quad C = I_k - \frac{1_k {1_k}^T}{k}. \tag{3.1}$$

Definition 11 *[9] The* **pre-shape** *of a configuration matrix (or complex vector), Y_{raw}, is all the geometric information about Y_{raw} that is invariant under location and isotropic scaling. The* **pre-shape space***, S_m^k, is the space of all possible pre-shapes. S_m^k is a hyper-sphere of unit radius in $\Re^{(k-1)m}$ and hence its dimension is $(k-1)m-1$ (a unit hyper-sphere in $\Re^P$ has dimension $P-1$).*

Scale Normalization: The pre-shape is obtained by normalizing the centered configuration, Y, by its Euclidean norm, $s(Y) = ||Y||$ (known as **scale or size** of the configuration), i.e. $w(Y) = Y/s(Y)$.

Definition 12 *[9] The* **shape** *of a configuration matrix (or complex vector), Y_{raw}, is all the geometric information about Y_{raw} that is invariant under location, isotropic scaling and* *rotation (Euclidean similarity transformations) i.e.* $[z] = \{sY_{raw}R + 1_k\alpha^T : s \in \Re^+, R \in SO(m), \alpha \in \Re^m\}$. *The* **shape space** *is the set of all possible shapes. Formally, the shape space,* Σ_m^k, *is the orbit space of the non-coincident* k *point set configurations in* $\Re^m$ *under the action of Euclidean similarity transformations. The dimension of shape space is* $M = (k-1)m - 1 - m(m-1)/2$. *It is easy to see that* $\Sigma_m^k = S_m^k/SO(m)$, *i.e.* Σ_m^k *is the quotient space of* S_m^k *under the action of the special orthogonal group of rotations,* $SO(m)$.

Rotation Normalization: Shape, z, is obtained from a pre-shape, w, by rotating it in order to align it to a reference pre-shape γ. The optimal rotation angle is given by $\theta(Y,\gamma) = \arg(w^*\gamma) = \arg(Y^*\gamma)$, and the shape, $z(Y,\gamma) = we^{j\theta(Y,\gamma)} = \frac{Y}{s(Y)}e^{j\theta(Y,\gamma)}$.

In this work we deal with $m = 2$ dimensional shapes and hence the configuration vector is represented as a k dimensional complex vector and the shape space dimension is $(2k-4)$.

Distance between shapes: A concept of distance between shapes is required to fully define the non-Euclidean shape metric space. We use the Procrustes distance which is defined below.

Definition 13 *[9] The* **full Procrustes fit** *of* w *onto* y *is*

$$\begin{aligned} w^P(y) &= \hat{\beta}e^{j\hat{\theta}}w + \hat{a} + j\hat{b} \quad \textit{where} \\ \hat{\beta}, \hat{\theta}, \hat{a}, \hat{b} &= \arg\min_{(\beta,\theta,a,b)} D(y,w), \;\; D(y,w) = ||y - (\beta e^{i\theta}w + a + jb)||. \end{aligned}$$

If y and w are preshapes, it is easy to see that the matching parameters are (result 3.1 of [9])

$$\hat{a} + j\hat{b} = 0, \;\; \hat{\theta} = \arg(w^*y), \;\; \hat{\beta} = |w^*y| = (y^*ww^*y)^{1/2}$$

Definition 14 *[9] The* **full Procrustes distance** *between preshapes w and y is the Euclidean distance between the Procrustes fit of w onto y, i.e.*

$$\begin{aligned} D_F(w,y) &= \inf_{\beta,\theta,a,b} D(y,w) = ||y - w^P(y)|| \\ &= \sqrt{1 - y^* w w^* y} \end{aligned} \tag{3.2}$$

Definition 15 *[9] The* **full Procrustes estimate of mean shape** *(commonly referred to as* **full Procrustes mean***), of a set of preshapes $\{w_i\}$ is the minimizer of the sum of squares of full Procrustes distances from each w_i to an unknown unit size mean configuration μ, i.e.*

$$\begin{aligned} [\hat{\mu}] &= \arg \min_{\mu:||\mu||=1} \sum_{i=1}^{n} \min_{\beta_i,\theta_i,a_i,b_i} D^2(w_i,\mu) \\ &= \arg \min_{\mu:||\mu||=1} \sum_{i=1}^{n} D_F^2(w_i,\mu) \\ &= \arg \min_{\mu:||\mu||=1} \sum_{i=1}^{n} (1 - \mu^* w_i w_i^* \mu) \\ &= \arg \max_{\mu:||\mu||=1} \mu^* [\sum_{i=1}^{n} w_i w_i^*] \mu \end{aligned} \tag{3.3}$$

i.e. $[\hat{\mu}]$ is given by set of complex eigenvectors corresponding to the largest eigenvalue of $S \triangleq \sum_{i=1}^{n} w_i w_i^*$ (Result 3.2 of [9]).

Shape Variability in Tangent to Shape Space: The structure of shape variability of a dataset of similar shapes can be studied in the tangent hyperplane to shape space at the Procrustes mean of the dataset. The tangent space is a linearized local approximation of shape space at a particular point in shape space which is called the **pole** of tangent projection. We shall consider the tangent projections to the preshape sphere after normalizing for rotation (w.r.t. the pole), which form a suitable tangent coordinate system for shape. The tangent space to shape space is a vector space and the Euclidean distance in tangent space is a good approximation to Procrustes distance in the vicinity of the pole. (See chapter 4 of [9] for more details).

Definition 16 *[9] The* **Procrustes tangent coordinates** *of a centered configuration, Y, taking μ as the pole, are obtained by projecting $z(Y,\mu)$ (the shape of Y aligned to μ) into the tangent space at μ, i.e.*

$$v(Y,\mu) = [I_k - \mu\mu^*]z(Y,\mu) = [I_k - \mu\mu^*]\frac{Y}{s(Y)}e^{j\theta(Y,\mu)}. \tag{3.4}$$

The inverse of the above mapping (tangent space to centered configuration space) is

$$Y(v,\theta,s,\mu) = [(1 - v^*v)^{1/2}\mu + v]se^{-j\theta}. \tag{3.5}$$

The shape space is a non-linear manifold in $\mathcal{C}^{k-1}$ and hence its dimension is $k-2$. Thus the tangent plane at any point of the shape space is a $k-2$ dimensional hyperplane in $\mathcal{C}^k$ (or equivalently, a $(2k-4)$-dim hyperplane in $\Re^{2k}$) [9].

3.3 Stationary Shape Activity

3.3.1 Shape Deformation Model in Tangent Space

A sequence of point configurations from a stationary shape activity (SSA), with small system noise variance, would lie close to each other and to their mean shape (see figure 3.1(a)). Hence a single tangent space at the mean is a good approximate linear space to learn the shape deformation dynamics for a SSA. We represent a configuration of landmarks by a complex vector with the x and y coordinates of a landmark forming the real and imaginary parts[1]. We first discuss the training algorithm i.e. how to learn the shape dynamics given a single training sequence of configurations. Given a sequence of configurations with negligible observation noise, $\{Y_{raw,t}\}$, we learn its Procrustes mean and evaluate the tangent coordinates of shape (using the Procrustes mean as the pole), as

$$\begin{aligned}
Y_t &= CY_{raw,t}, \\
s_t &\triangleq s(Y_t) = ||Y_t||, \qquad w_t = Y_t/s_t, \\
\mu &= \arg \max_{\mu:||\mu||=1} \mu^*[\sum_{t=1}^{T} w_t w_t^*]\mu \\
\theta_t &\triangleq \theta(Y_t, \mu) = \arg(w_t^* \mu), \qquad z_t = w_t e^{j\theta_t} \qquad (3.6)
\end{aligned}$$

$$v_t \triangleq v(Y_t, \mu) = [I_k - \mu\mu^*]z_t = [I_k - \mu\mu^*]\frac{Y_t e^{j\theta_t}}{s_t} \qquad (3.7)$$

Since the tangent coordinates are evaluated w.r.t. the mean shape of the data, assuming that they have zero mean is a valid assumption. We string the complex tangent vector as a $2k$ dimensional real vector and then define a linear Gauss Markov model in the tangent space to model the shape deformation dynamics. Note that since we are assuming small variations about a mean shape, a first order Gauss Markov model is sufficient to model the shape dynamics in this case, i.e.

$$v_t = A_t v_{t-1} + n_t$$

[1] Note that all transformations between the configuration space to shape space and tangent to shape space are defined in $\mathcal{C}^k$ (k-dim complex space) but the dynamical model on tangent coordinates is defined in $\Re^{2k}$ by vectorizing the complex vector. This is done only for compactness of representation. The entire analysis could instead have been done in $\Re^{2k}$.

$$v_0 \sim \mathcal{N}(0, \Sigma_{v,0}), \quad n_t \sim \mathcal{N}(0, \Sigma_{n,t}) \tag{3.8}$$

The deformation process is assumed to be stationary and ergodic. Under this assumption the above is a first order autoregressive model. Thus, $\Sigma_{v,0} = \Sigma_{v,t} = \Sigma_v$, $\Sigma_{n,t} = \Sigma_n$ and $A_t = A$ is the autoregression matrix with $A < I$. $\{n_t\}$ is i.i.d. Gaussian system noise. Thus all the three parameters can be *learnt using a single training sequence* of tangent coordinates, $\{v_t\}$, as follows [68]

$$\begin{aligned}
A &= R_v(1)\Sigma_v^{-1} \quad \text{where} \\
\Sigma_v &= \frac{1}{T}\sum_{t=1}^{T} v_t v_t^T \quad \text{and} \quad R_v(1) = \frac{1}{T-1}\sum_{t=2}^{T} v_t v_{t-1}^T \\
\Sigma_n &= \frac{1}{T}\sum_{t=1}^{T} (v_t - Av_{t-1})(v_t - Av_{t-1})^T
\end{aligned} \tag{3.9}$$

and the joint pdf of v_t is given by

$$\begin{aligned}
p(v_t) &= \mathcal{N}(0, \Sigma_v), \quad \forall t \\
p(v_t|v_{t-1}) &= \mathcal{N}(Av_{t-1}, \Sigma_n), \quad \forall t.
\end{aligned} \tag{3.10}$$

Note that the asymptotically stationary case where $A < I$ but $\Sigma_{v,0} \neq \Sigma_v$ so that $\Sigma_{v,t} \rightarrow \Sigma_v$ only for large time instants ($t \rightarrow \infty$), can also be dealt with in the above framework. In that case $\Sigma_{v,0}$ is defined using a-priori knowledge, Σ_n can be learnt exactly as in (3.9), and $\Sigma_v, R_v(1)$ can also be learnt as in (3.9) but by excluding the summation over the initial (transient) time instants.

Now, the tangent coordinates obtained in (3.7) lie on a $(k-2)$-dim hyperplane of $\mathcal{C}^k$ and hence they have only $(k-2)$ degrees of freedom in complex coordinates (or equivalently

$(2k-4)$ degrees of freedom in real coordinates). We can evaluate an orthogonal basis, $U(\mu)$, for the tangent space at μ by evaluating the Singular Value Decomposition [68] of the tangent projection matrix, $[I_k - \mu\mu^*]C$ and retaining the $(k-2)$ directions with nonzero singular values. The $(k-2)$ independent coefficients of the tangent coordinate are then given by $c_t = U(\mu)^* v_t$. Now since this is a linear transformation, (3.8) implies a linear Gauss-Markov model on c_t as well. We use this tangent coefficient representation while defining nonstationary and piecewise stationary models.

3.3.2 Partially Observed (Hidden) Shape Dynamics

In the previous subsection we defined a dynamic model on the shape of a configuration of moving points. We assumed that the observation sequence used for learning the shape dynamics has zero (negligible) observation noise associated with it (e.g. if it were hand-picked). But a test sequence of point configurations, $\{Y_{raw,t}\}$, will usually be obtained automatically using a measurement algorithm (e.g. a motion detection algorithm [69]). It will thus have large observation noise associated with it, i.e. $Y_{raw,t} = Y_{raw,t}^{actual} + \zeta_{raw,t}$ where $\zeta_{raw,t}$ is zero mean Gaussian noise, $\zeta_{raw,t} \sim \mathcal{N}(0, \Sigma_{obs,raw,t})$. If the different landmarks are far apart, the noise can be assumed to be i.i.d. over the different landmarks as well (i.e. white $\Sigma_{obs,raw,t}$). Now translation normalization is a linear process and hence $Y_t = CY_{raw,t}$ is also

Gaussian[2] with observation noise, ζ_t, given by

$$\Sigma_{obs,t} = C\Sigma_{obs,raw,t}C^T \tag{3.11}$$

(C is the centering matrix defined in (3.1)). But the mapping from centered configuration space to the tangent space is nonlinear (scaling by $||Y_t||$ followed by rotation to align with mean) and hence it is not possible to obtain a closed form expression for the pdf of noise in the tangent coordinates due to observation noise in the configuration vector. To deal with this, one has to define a partially observed dynamical model (which is a continuous state HMM), which can then be tracked using a PF to estimate the shape from the noisy observations. The observed centered configuration, Y_t, forms the observation vector and the shape, scale and rotation form the hidden state vector.

We discuss the advantage of a PF over an Extended Kalman Filter in section 3.6. Now, we have the following *observation model* for a "stationary shape activity" with the observation vector Y_t being the centered configuration vector and the state vector $X_t = [v_t, s_t, \theta_t]$:

$$\begin{aligned} Y_t &= h(X_t) + \zeta_t, \qquad \zeta_t \sim \mathcal{N}(0, \Sigma_{obs,t}) \\ h(X_t) &= z_t s_t e^{-j\theta_t}, \quad \text{where} \quad z_t = (1 - v_t * v_t)^{1/2}]\mu + v_t \end{aligned} \tag{3.12}$$

[2] Note that here we have assumed Gaussian observation noise, $\zeta_{raw,t}$, but in general a PF can track with any kind of noise. But for non-Gaussian $\zeta_{raw,t}$, it is in general not possible to define a distribution for ζ_t and one would have to treat the translation as part of the state vector.

Defining scale and rotation (motion parameters) as part of the state vector implies that we need to define prior dynamic models for them (motion model). The *motion model* can be defined based on either the motion of the shape if it is a moving configuration or based on motion of the measurement sensor if the sensor is moving (for e.g. a moving camera or just an unstable camera undergoing a slight random motion) or a combined effect of both. A camera on an unstable platform, like an unmanned air vehicle (UAV), will have small random x-y motion (translation), motion in z direction (scale change) and rotation about the z axis (rotation angle change). The translation gets removed when centering $Y_{raw,t}$. The scale and rotation can be modeled in this case by using a linear Gauss-Markov model (AR model) both for log of scale and for the unwrapped rotation angle[3], i.e.,

$$\begin{aligned}
\log s_t &= \alpha_s \log s_{t-1} + (1-\alpha_s)\mu_s + n_{s,t} \\
\log s_0 &\sim \mathcal{N}(\mu_s, \sigma_s^2), \quad n_{s,t} \sim \mathcal{N}(0, \sigma_r^2) \\
\theta_t &= \alpha_\theta \theta_{t-1} + (1-\alpha_\theta)\mu_\theta + n_{\theta,t} \\
\theta_0 &\sim \mathcal{N}(0, \sigma_\theta^2), \quad n_{\theta,t} \sim \mathcal{N}(\mu_\theta, \sigma_\theta^2)
\end{aligned} \tag{3.13}$$

The motion model parameters can be learnt using the training sequence values of $\{s_t\}_{t=1}^T$ and $\{\theta_t\}_{t=1}^T$ given by (3.6). $\{\theta_t\}_{t=1}^T$ will have to be the unwrapped value of the rotation angle to learn a Gaussian model. Also, one can either assume wide sense stationarity, in which case $\mu_s, \sigma_s^2, \sigma_r^2, \alpha_s$ and $\mu_\theta, \sigma_\theta^2, \sigma_u^2, \alpha_\theta$ can be learnt using Yule-Walker equations [68], or assume a random walk motion model(set $\alpha_s = 1$ and $\alpha_\theta = 1$), depending on the application.

[3] Since we are modeling only random motion of a camera, a first order linear Markov model for log of scale and rotation is sufficient in this case

The *shape deformation dynamics* (equation (3.8) in section 3.3.1) and the *motion model* defined above (equation (3.13)) form the *system model* while equation (3.12) defines the *observation model.* Thus we have defined a *a continuous state HMM (partially observed dynamic model)* for a “stationary shape activity”. The model is non-linear since the mapping $h(X_t)$ is nonlinear.

3.4 Non-Stationary Shape Dynamics

For a “non-stationary shape activity” model, the mean shape is time-varying and hence modeling the shape dynamics requires a time-varying tangent space (see figure 3.1(b)) defined with the current shape as the pole. Note that, modulo reflections, there is a one to one mapping between the tangent space at any point on the shape manifold and the shape manifold. But the distance between two points on a tangent plane is a good approximation to the distance on the shape manifold only for points close to the pole of the tangent plane. Hence the assumption of i.i.d. system noise to go from shape at t to shape at $t+1$ is valid only for shapes in the vicinity of the pole. Thus when the shape variation is large (for NSSA), there is a need to define a tangent space with the current shape being the pole.

The state space now consists of the mean shape at time t, z_t, the “shape velocity coefficients” vector, c_t, and the motion parameters (scale s_t, rotation θ_t) i.e. state $X_t = [z_t, c_t, s_t, \theta_t]$. Denote the tangent space at z_t by T_{z_t}. We then have the following dynamics: The tangent coordinate of z_t in $T_{z_{t-1}}$ (denoted by $v_t(z_t, z_{t-1})$) defines a *“shape velocity”* (time derivative of shape) vector. We perform a Singular Value Decomposition [68] of the tangent projection matrix, $[I_k - z_{t-1}z_{t-1}^*]C$, to obtain an orthogonal basis for the $(k-2)$-dim tangent hyperplane $T_{z_{t-1}}$. Denote the orthogonal basis matrix for $T_{z_{t-1}}$ by $U(z_{t-1})$ [4]. The $(k-2)$-dim vector of coefficients along these basis directions, denoted by $c_t(z_t, z_{t-1})$, is a coefficients vector for the "shape velocity", v_t, i.e. $v_t = U(z_{t-1})c_t$. The shape at t, z_t is obtained by "moving" z_{t-1} on the shape manifold as follows: Move an amount v_t (from origin) in $T_{z_{t-1}}$ and then project back onto shape space. Thus z_t is evaluated as $z_t = (1 - v_t^* v_t)^{1/2} z_{t-1} + v_t$.

We assume a linear Gauss-Markov model on shape velocity v_t which corresponds to a linear Gauss Markov model for c_t. We can then summarize the shape dynamics as follows:

$$\begin{aligned} c_t &= A_{c,2,t}c_{t-1} + n_t, \quad n_t \sim \mathcal{N}(0, \Sigma_{n,c,2,t}) \\ v_t &= U(z_{t-1})c_t, \quad U(z_{t-1}) = \text{orthogonal basis}(T_{z_{t-1}}) \\ z_t &= (1 - v_t^* v_t)^{1/2} z_{t-1} + v_t. \end{aligned} \tag{3.14}$$

If we assume a time invariant AR model on $\{v_t\}$, i.e. $v_t = A_{v,2}v_{t-1} + n_{v,t}$ then we have have a time varying Gauss-Markov model on c_t with

$$A_{c,2,t} = U(z_{t-1})^* A_{v,2} U(z_{t-2}), \quad \text{and} \quad \Sigma_{n,c,2,t} = U(z_{t-1})^* \Sigma_{n,v,2} U(z_{t-2}). \tag{3.15}$$

Note that a Markov model on the shape velocity corresponds to a second order Markov model on shape, z_t (hence the subscript '2' on the parameters). Some special cases are $A_{v,2} = 0$ or i.i.d. velocity (first order Markov model on shape); $A_{v,2} = I$ which corresponds to i.i.d. shape acceleration and $A_{v,2} = A_{AR}$ or stationary shape velocity.

[4] The basis vectors, $\{\underline{u}_{t,i}\}_{i=1}^{k-2}$, are arranged as column vectors of a matrix, $U(z_{t-1})$, i.e. $U_t^{k\times(k-2)} = [\underline{u}_{t,1}, \underline{u}_{t,2} \cdots \underline{u}_{t,k-2}]$. $U_t^{k\times(k-2)} = \text{orthogonal basis}(T_{z_{t-1}})$ is evaluated as : $U_t = U_{full,t}Q$ where $U_{full,t}SU_{full,t}^* = [I_k - z_{t-1}z_{t-1}^*]C$, and $Q = [I_{(k-2)\times(k-2)}, 0_{(k-2)\times 2}]^T$

The *motion model* (model on s_t, θ_t) can be defined exactly as in equation (3.13) but now θ_t is the rotation angle of current configuration w.r.t. the current mean shape $\mu_t = z_{t-1}$ and hence is a measure of rotation speed. As before, one can assume the motion model to be stationary or non-stationary. The shape and motion model, (3.14) and (3.13)), form the *system model.* The *observation model* is as follows:

$$Y_t = \tilde{h}(X_t) + \zeta_t, \quad \text{where} \quad \tilde{h}(X_t) = z_t s_t e^{-j\theta_t}. \tag{3.16}$$

Training

Given a training sequence of centered (translation normalized) configurations, $\{Y_t\}_{t=1}^T$, we first evaluate $\{c_t, v_t, s_t, \theta_t\}_{t=1}^T$ as follows [5] :

$$\begin{aligned}
s_t &= ||Y_t||, \quad w_t = Y_t/s_t, \\
\theta_t(Y_t, z_{t-1}) &= arg(w_t^* z_{t-1}), \quad z_t(Y_t, z_{t-1}) = w_t e^{j\theta_t}, \\
v_t(Y_t, z_{t-1}) &= [I_k - z_{t-1} z_{t-1}^*] z_t, \\
c_t(Y_t, z_{t-1}) &= U(z_{t-1})^* z_t.
\end{aligned} \tag{3.17}$$

Assuming a time invariant AR model on shape velocity, v_t, one can learn its parameters $(A_{v,2}, \Sigma_{n,v,2})$ as in (3.9) and then define the time-varying Markov model for c_t using (3.15).

[5] Note, the last equation, $c_t = {U_t}^* z_t$, holds because $c_t = {U_t}^* v_t = {U_t}^*[I - z_{t-1} z_{t-1}^*] z_t = {U_t}^*[I - z_{t-1} z_{t-1}^*] C z_t = {U_t}^* U_t {U_t}^* z_t = {U_t}^* z_t$.

3.5 Piecewise Stationary Shape Dynamics

When the shape is not stationary but is slowly varying, one could model the mean shape as being piecewise constant. Now in SSA, the mean shape is constant i.e. $\mu_t = \mu$ for all t and hence all the dynamics can be described in a single tangent space while in NSSA, the tangent space changes at each time instant: $\mu_t = z_{t-1}$ is the pole of the tangent space at time t. But for PSSA we let the mean μ_t (and hence also the tangent space) be piecewise constant.

Let the mean shape change times be $t_1, t_2, t_3, ...$ and the corresponding means be $\mu_1, \mu_2, \mu_3, ...$ Then we have the following dynamics: Between $t_{j-1} < t < t_j$, $\mu_t = \mu_{t-1}$ and so $c_{t-1}(z_{t-1}, \mu_t) = c_{t-1}(z_{t-1}, \mu_{t-1})$. Hence in this interval, the dynamics is similar to that for an SSA[6], i.e.

$$\begin{aligned}
c_t(z_t, \mu_t) &= A_{c,1,t} c_{t-1}(z_{t-1}, \mu_t) + n_t, \\
v_t &= U(\mu_t) c_t, \\
z_t &= (1 - v_t^* v_t)^{1/2} \mu_t + v_t. && (3.18)
\end{aligned}$$

At the change time instant, $t = t_j$, $\mu_t = \mu_j$ and so the tangent coefficient c_{t-1} needs to be recalculated in the new tangent space w.r.t. $\mu_t = \mu_j$. This is achieved as follows:

$$\begin{aligned}
c_{t-1}(z_{t-1}, \mu_t) &= U(\mu_t)^* z_{t-1} e^{j\theta(z_{t-1}, \mu_t)} \\
c_t(z_t, \mu_t) &= A_{c,1,t} c_{t-1}(z_{t-1}, \mu_t) + n_t, \\
v_t &= U(\mu_t) c_t,
\end{aligned}$$

[6] Now we have defined SSA dynamics on v_t, but we can equivalently define it on c_t there as well, with a constant transformation, $c_t = U(\mu)^* v_t$

$$z_t = (1 - v_t^* v_t)^{1/2} \mu_t + v_t. \qquad (3.19)$$

Note that in NSSA, v_t is a tangent coordinate w.r.t. $\mu_t = z_{t-1}$ and hence it measures shape velocity while in this case, v_t (and hence also c_t) is a tangent shape coordinate w.r.t. the current mean shape μ_t. Hence like in SSA, here also we have a first order Markov model on shape. Hence the subscript '1' on $A_{c,1,t}$.

Now the times at which the changes occur and the changed means could both be unknown or known or one of them could be unknown. When both change times and the corresponding means are known, PSSA can be used for tracking a sequence of stationary shape activities (each with its known shape mean and known transition times) and detecting abnormality. Abnormality can be defined as ELL w.r.t. the current mean shape exceeding a threshold.

When times at which the changes occur are unknown, one can use ELL (discussed in chapter 2) w.r.t. the current mean shape to detect a change. This is useful for activity sequence identification (figuring out when one activity ends and the next one starts) and tracking.

When both change times and changed system means are not known, one can detect the change using ELL. The "best" estimate of the shape at the t based on observations $Y_{1:t}$ can be used as the new shape mean. Now since the shape space is nonlinear, the expected value of shape given observations, $E_{\pi_t^N}[z_t]$ (the MMSE estimate), may not lie in the shape space at all. But we can instead estimate a Procrustes mean [9] of the shape which is the minimum mean Procrustes distance square estimator. This is discussed in section 3.2, definition 15. As explained there, the Procrustes mean can be evaluated as the largest eigenvector of the matrix $S \triangleq E_{\pi_t^N}[z_t z_t^*] = \frac{1}{N}\sum_{i=1}^N z_t^i z_t^{i*}$. Note that the Procrustes mean is an intrinsic mean for the shape manifold. One can also evaluate the extrinsic mean [37] which is the projection of the data mean of tangent coordinates, $E_{\pi_t^N}[v_t]$, onto the shape space, i.e. $\mu_t^{extrinsic} = (1 - E_{\pi_t^N}[v_t]^* E_{\pi_t^N}[v_t])^{1/2}\mu_{t-1} + E_{\pi_t^N}[v_t]$.

Now setting the mean this way will be valid as long as the tracking error (or equivalently the observation likelihood, OL, discussed in chapter 2) is still below the tracking error threshold (the posterior π_t^N is estimated correctly). This follows from theorem 4 of chapter 2. Now this form of PSSA can be used for activity sequence segmentation and tracking by using the change times detected using ELL as segmentation boundaries to split a long sequence into piecewise stationary pieces. We discuss the algorithm for segmentation in more detail in section 4.4.

3.6 Particle Filtering and Extended Kalman Filtering

We have discussed stationary, nonstationary and piecewise stationary shape models in the above three subsections all of which are tracked using a particle filter. We discuss here the need for a PFr and why it is better than an extended Kalman filter.

An Extended Kalman Filter (EKF) [70] linearizes the non-linear system at each time instant using Taylor series and runs a Kalman filter for the linearized system. For the Taylor series approximation to be accurate, one requires the initial guess (point about which you linearize) to be close to the actual value at every time instant. Typically linearization is done about the predicted state. This means that one poorly estimated state will cause more error in the linearization matrices for the next prediction and this error will propagate (thus an EKF cannot recover once it loses track). Loss of track can occur due to an outlier observation, modeling error, large system noise or large linearization error. A PF on the other hand is stable under mild assumptions [71, 5] and hence it gets back in track more easily after losing track.

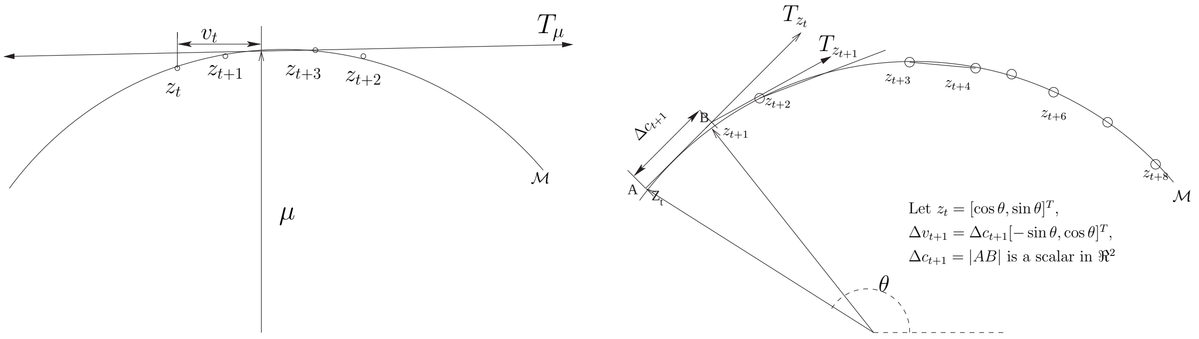
An EKF is unable to track non-Gaussian systems, in particular systems with multi-modal priors or posteriors, while a PF can. Multi-modal system models are required to model a sequence of activities or multiple simultaneous activities. Also in particle filtering, the number of particles, N, required to achieve a certain performance guarantee on estimation error, does not increase with increasing dimension of the state space [49], it depends only on the total randomness in the system. So for a system which is more random (larger system noise or observation noise), the PF performance can be improved by increasing N.

3.7 Time-Varying Number of Landmarks

All the analysis until now assumes that a configuration of points is represented as an element of $\Re^{2k}$ where k is a fixed number of landmarks. Now we consider what happens when the number of landmarks (here the point objects) is time-varying even though the curve formed by joining their locations remains similar. For example, a group of people (or also a group of vehicles) moving on a certain path with fixed initial and final points but number of people on the path decreases by one when a person leaves and increases by one when someone enters. In such a case, we linearly interpolate the curve by joining the landmark points in a predefined order and then re-sample the interpolated curve to get a fixed number of landmarks. The interpolation depends on the parametrization of the curve, which is an ill-posed problem when the data is inherently discrete. We have attempted to use two different schemes which exist in the literature - "arc-length re-sampling" (also known as "equidistant sampling") and "uniform re-sampling" which use two different parameterizations.

In **"arc-length resampling"**, one looks at the curve formed by joining the landmarks in a predefined order, and parameterizes the x and y coordinates by the length, l, of the curve, upto that landmark. Let $[x_t(l), y_t(l)]$ be one-dimensional functions of the curve length and seen this way the discrete landmarks $x_{t,j} = x_t(l_j), y_{t,j} = y_t(l_j), j = 0, 1, ..k_t - 1$ are non-uniformly sampled points from the function $[x_t(l), y_t(l)]$ with $l_0 = 0, l_j^2 = l_{j-1}^2 + (x_{t,j} - x_{t,j-1})^2 + (y_{t,j} - y_{t,j-1})^2$. We linearly interpolate using these discrete points to estimate the function $[\hat{x}_t(l), \hat{y}_t(l)]$ and then re-sample it uniformly at points $\tilde{l}_j = (j-1)L/k, j = 0, 1, ..k-1$ (L is the total length, $L^2 = \sum_j l_j^2$) to get a fixed number, k, of uniformly spaced landmarks. Thus, for every configuration of k_t landmarks, we get a new configuration of uniformly sampled (and hence uniformly spaced) k landmarks. The linear interpolation and resampling stages can be approximated as a linear transformation, B_t (a $k_t \times k$ matrix), applied to the original points. The covariance of observation noise in the re-sampled points becomes $\Sigma_{obs,t}^{k} = B_t \Sigma_{obs,t}^{k_t} B_t^T = B_t C^{k_t} \Sigma_{obs,raw,t}^{k_t} {C^{k_t}}^T B_t^T$.

"Uniform resampling", on the other hand, assumes that the observed points are uniformly sampled from some process, $[x_t(s), y_t(s)]$, i.e. it assumes that the observed points are parameterized as $x_{t,j} = x_t(s_j), y_{t,j} = y_t(s_j)$ with $s_j = (j-1)/k_t$. We linearly interpolate to estimate $[\hat{x}_t(s), \hat{y}_t(s)]$ and re-sample it uniformly at points $\tilde{s}_j = (j-1)/k$, to get a fixed number of landmarks, k. Assuming the observed points to be uniformly sampled makes this scheme very sensitive to the changing number of landmarks. Whenever the number of landmarks changes, there is a large change in the re-sampled points' configuration. This leads to more false alarms while performing abnormal activity detection. But unlike "arc-length resampling", this scheme gives equal importance to all observed points irrespective of the distance between consecutive points and so is more quick to detect abnormalities in shape caused even by two closely spaced points. We discuss an example in section 4.5.2.



Let $z_t = [\cos\theta, \sin\theta]^T$,
$\Delta v_{t+1} = \Delta c_{t+1}[-\sin\theta, \cos\theta]^T$,
$\Delta c_{t+1} = |AB|$ is a scalar in $\Re^2$

(a) Stationary Shape Activity (SSA)

(b) Nonstationary Shape Activity (NSSA)

Figure 3.1: SSA & NSSA on the shape manifold which is depicted using a circle ($\mathcal{M}$), instead of a complex $\mathcal{C}^{k-1}$ sphere. In (a), we show a sequence of shapes from a SSA; at all times the shapes are close to the mean shape and hence the dynamics can be approximated in T_μ (tangent space at μ). In (b), we show a sequence of shapes from an NSSA, the shapes move on the shape manifold and hence we need to define a new tangent space at every time instant.

Chapter 4

Applications to Abnormal Activity Detection, Tracking and Segmentation

4.1 Abnormal Activity Detection

An abnormal activity (suspicious behavior in our case) is defined as a change in the system model, which could be slow or drastic, and whose parameters are unknown. Given a test sequence of observations and a "shape activity" model, we use the change detection statistics defined in chapter 2 to detect a change (i.e. detect when observations stop following the given shape activity model). We first consider stationary shape activities. The cases of negligible observation noise (Fully Observed) and non-negligible observation noise (Partially observed) are discussed separately. In section 4.1.3, we formulate the abnormality detection problem for nonstationary shape activities.

4.1.1 Stationary Shape Activity: Fully Observed Case

The system is said to be fully observed when the function $h(.)$ is invertible and the observation noise is zero (negligible compared to the system noise, n_t). For such a test sequence, the shape dynamics of section 3.3.1 fully defines the "shape activity model". We can evaluate the tangent coordinates of shape (v_t) directly from the observations using (3.7). We use log-likelihood to test for abnormality. A given test sequence is said to be generated by a *normal activity iff* the probability of occurrence of its tangent coordinates using the pdf defined by (5.2) is large (greater than a certain threshold). Thus the distance to activity statistic for an '$L+1$' length observation sequence ending at time t, $d_{L+1}(t)$, is the negative log likelihood of the sequence of tangent coordinates of the shape of the observations (first used by us in [11]). We can test for abnormality at any time t by evaluating $d_{L+1}(t)$ for the past $L+1$ frames. $d_{L+1}(t)$ is defined as follows: (K is a constant defined in equation (4.3))

$$\begin{aligned} d_{L+1}(t) &= -2\log p(v_{t-L}, v_{t-L+1}, ...v_t) \\ &= v_{t-L}^T \Sigma_v^{-1} v_{t-L} \\ &+ \sum_{\tau=t-L+1}^{t} (v_\tau - Av_{\tau-1})^T \Sigma_n^{-1} (v_\tau - Av_{\tau-1}) + K \end{aligned} \tag{4.1}$$

Note here that, Σ_v is always rank deficient since $\{v_t\}$ lie in a $(2k-4)$-dim hyperplane of $\Re^{2k}$ and hence the inverse defined above actually represents the pseudo-inverse.

Some results using this statistic combined with our PCNSA classification algorithm are shown in section 5.6.4 of chapter 5.

4.1.2 Stationary Shape Activity: Partially Observed Case

In a partially observed system, the observation noise in the configuration landmarks' measurements is non-negligible and it is defined by the observation model discussed in section 3.3.2. The PF is used to estimate the posterior distribution of shape at time t given observations upto $t-1$ (prediction) and upto t (filtering). We use the change detection strategy described in chapter 2.

1. If the abnormality is a drastic one it will cause the PF, with N large enough to accurately track only normal activities, to lose track. This is because under the normal activity model (equations (3.8) and (3.13)), the abnormal activity observations (which do not follow this model) would appear to have a very large observation noise. Thus the tracking error will increase for an abnormal activity (very quickly for a drastic one) and this can be used to detect it. This intuitive idea is discussed in more detail in chapter 2. The *tracking error* or prediction error is the distance between the current observation and its prediction based on past observations, i.e.

$$\begin{aligned}\text{Tracking error} \triangleq ||Y_t - \hat{Y}_t||^2 &= ||Y_t - E[Y_t|Y_{0:t-1}]||^2 \\ &= ||Y_t - E_{\pi_{t|t-1}}[h(X_t)]||^2\end{aligned}$$

   Also, instead of tracking error, OL can also be used and as discussed in chapter 2, $OL \approx TE$ for white Gaussian noise.

2. For the case when the abnormality is a slow change (say a person walking away slowly in a wrong direction), the PF does not lose track very quickly (the tracking error

increases slowly) or if it is a short duration change it may not lose track at all. The tracking error will thus take longer to detect the change or it may not detect it at all. For such a case, we use the *expected (negative) log likelihood (ELL)* [10, 72].

$$ELL = E_{\pi_{t|t}}[-logf(v_t)] \tag{4.2}$$

Note that the ELL is a posterior expectation of the right hand side of (4.1) with $L = 0$. In general, one could use a sequence of past shapes ($L > 0$) in this case as well. The expression for ELL is approximated by ELL^N as follows

$$\begin{aligned} ELL^N \triangleq E_{\pi_t^N}[-logp(v_t)] &= \frac{1}{N}\sum_{i=1}^{N} v_t^{(i)T} \Sigma_v^{-1} v_t^{(i)} + K, \\ \text{where} \;\; K &\triangleq -\log\sqrt{(2\pi)^{2k-4}|\Sigma_v|}. \end{aligned} \tag{4.3}$$

Now since the PF loses track slowly, the estimated posterior $\pi_t^{c,0,N}$ remains a good approximation of $\pi_t^{c,c}$ until the PF has lost track. But a slowly changing shape introduces a systematically increasing bias in the tangent coordinates of shape (they no longer remain zero mean) and hence ELL would increase.

Thus to *detect any kind of abnormality (slow or drastic) without knowing its rate of change, we use a combination of ELL and tracking error. We declare a sequence of observations to be abnormal when either ELL or tracking error exceeds its corresponding threshold.*

4.1.3 Nonstationary Shape Activity: Abnormality Detection

A change being drastic or slow depends on the system model used in particle filtering. A more general system model can track a lot more changes and hence the nonstationary shape activity model does a better job of tracking abnormal observations than the stationary one. Whenever changed observations get tracked correctly, the ELL detects the change while if the PF loses track, the tracking error detects the change.

Now for abnormality detection, the normal activity needs to be characterized first. We can either use shape velocity or shape or both to represent normalcy depending on the practical problem being dealt with. To use shape to detect abnormality, we represent a normal activity by a stationary shape activity model or by a PSSA model (whichever is appropriate for a given problem). First, assume an SSA normal activity. Then the normal prior is a time invariant Gaussian distribution of the tangent coordinates w.r.t. the normal activity mean (μ_0), $\mathcal{N}(0, \Sigma_{v,0})$. Now for a Gaussian prior, the discriminating term of ELL reduces to expectation, under the posterior, of the Mahalonobis distance from the prior's mean. We evaluate it as follows: We project the filtered shape of the observations at time t into T_{μ_0} to obtain $v(z_t, \mu_0)$ and evaluate $E_{\pi_t}[v(z_t, \mu_0)^T \Sigma_{v,0}^{-1} v(z_t, \mu_0)]$. Thus given the particle filtered shape distribution $\pi_t^N(dz_t) \triangleq \sum_{i=1}^N \frac{1}{N} \delta_{z_t^{(i)}}(dz_t)$ (which approximates $\pi_t(dz_t)$), we evaluate

$$\begin{aligned} \pi_t^N(dv_{t,\mu_0}) &\triangleq \sum_{i=1}^N \frac{1}{N} \delta_{v_{t,\mu_0}^{(i)}}(dv_{t,\mu_0}), \quad \text{where} \\ v_{t,\mu_0}^{(i)} \triangleq v(z_t^{(i)}, \mu_0) &= [I_k - \mu_0 \mu_0^*] z_t^{(i)} e^{j\theta(z_t^{(i)}, \mu_0)}. \end{aligned} \tag{4.4}$$

The ELL, henceforth referred to as ELL (Shape) is then approximated by

$$ELL^N(Shape) = \frac{1}{N} \sum_{i=1}^N {v_{t,\mu_0}^{(i)}}^T \Sigma_{v,0}^{-1} v_{t,\mu_0}^{(i)} \tag{4.5}$$

If PSSA is used to define a normal activity, the prior is a Gaussian distribution on the tangent coordinates in the tangent space of the current mean μ_t.

Depending on the practical problem, one might want to use shape velocity (directions and magnitude of rate of change of shape) to define normalcy. In this case, assume that a stationary Gauss Markov model has been defined for the shape velocity, v_t, with parameters $\Sigma_{v,2}, A_{v,2}, \Sigma_{n,v,2}$. The change detection statistic in this case will simplify to $E_{\pi_t^N}[v_t^T \Sigma_{v,2}^{-1} v_t]$ where $v_t = v(z_t, z_{t-1})$ [1]. We refer to this statistic as "ELL (Shape Velocity)".

Now, the above two cases correspond to the two ideas proposed in section 2.3.4 to define the normal prior. Using ELL (Shape Velocity) as described above is an example of using a part of the state vector which has linear Gaussian dynamics for change detection. Using ELL (Shape) by first defining an SSA or PSSA model for the normal activity and learning its parameters, is an example of the second idea discussed in section 2.3.4 with the parametric model for normal activity being defined by SSA or PSSA respectively.

4.2 Tracking to Obtain Observations

In the entire discussion till now, we used a PF in the filtering mode to estimate the probability distribution of shape from noisy observations and used this distribution for abnormality detection. But the PF also provides at each time instant the prediction distribution, $\pi_t(X_t|Y_{1:t-1})$, which can be used to predict the expected configuration at the next time instant using past observations, i.e. $\hat{Y}_t \triangleq E[Y_t|Y_{0:t-1}] = E_{\pi_{t|t-1}}[h(X_t)]$. We can use this information to improve the measurement algorithm used for obtaining the observations (a

[1] Note that $v(z_t, \mu_0)$ denotes the tangent shape coordinate of z_t w.r.t. μ_0 while $v_t = v(z_t, z_{t-1})$ denotes the shape velocity

motion detector [69] in our case). Its computational complexity can be reduced and its ability to ignore outliers can be improved by using the predicted configuration and searching only locally around it for the current observation[2]. As we show in section 4.5.3, the observed configuration is close to its prediction when there is no abnormality or change and hence the prediction can be used to obtain the observation. An SSA model can track a normal activity while the SSA is able to track abnormality as well.

If used in this "tracking observations and filtering" framework, a lot of drastic abnormalities can be detected at the measurement stage itself because no observations will be found in the "vicinity" (region of search defined using observation noise variance) of the predicted position. But an outlier might get confused with a drastic abnormality since even for an outlier we will not find any observation in the "vicinity". The difference is that outliers would be temporary (one or two time instants and then the PF comes back in track), while a drastic abnormality will appear to be an outlier for a sequence of frames. Thus by averaging the number of detects over a sequence of past time instants, we can separate outliers from real abnormalities.

Also, if the configuration is a moving one, then the predicted motion information can be used to translate, zoom or rotate the camera (or any other sensor) to better capture the scene but in this case, one would have to alter the motion model to include a control input.

[2] One thing to note here is that in certain cases (for example, if the posterior of any state variable is multimodal), evaluating the posterior expectation as a prediction of the current observation is not the correct thing to do. In such a case, one can track the observations using the CONDENSATION algorithm [50] which searches for the current observation around each of the possible $h(\bar{x}_t^i), i = 1, 2...N$.

4.3 Activity Sequence Identification and Tracking

Consider two possible situations for tracking a sequence of activities. Assume each activity is represented by an SSA so that the sequence of activities is characterized by a PSSA. The mean shape of each SSA component is known but the transition times are unknown.

1. First consider the simple case when there are just two possible activities and their order of occurrence is known, only the change time is unknown. In this case, one can detect the change using ELL (before the particle filter loses track) and then start tracking it with the second activity's transition model.

2. Now consider the general case when a sequence of activities occur, and we do not know the order in which they occur. In this case, we can use a discrete mode variable as part of the state vector to denote each activity type. We make the state transition model a mixture distribution and keep the mode variable as a state. Whenever a change occurs, it takes the mode variable a few time instants to stabilize to the correct mode. One could replace the multimodal dynamics with that of the detected mode once the mode variable has stabilized. Also, in this case we can declare an activity to be abnormal (i.e. neither of the known activity types) if the ELL w.r.t all known models exceeds a threshold.

4.4 Shape Activity Sequence Segmentation

The PSSA model with unknown mean shapes and unknown change times can be used along with ELL for activity sequence segmentation as follows:

- Track observations using PSSA, until the ELL of tangent coordinates w.r.t. the current μ_t, $ELL(\mu_t) = E_{\pi_t}[v_t^T \Sigma_{v,t}^{-1} v_t]$ exceeds the change detection threshold.

- Use time instants when $ELL(\mu_t)$ exceeds its threshold, as segmentation boundaries.

- If at time t, $ELL(\mu_t)$ has exceeded its threshold but the tracking error is still below its threshold (PF is still in track, i.e. π_t^N approximates π_t^c correctly), then set μ_{t+1} as the posterior Procrustes mean of the shape at t, given past observations, $Y_{1:t}$. This is explained in the last two paragraphs of section 3.5.

- Recalculate v_t and c_t in the new tangent space at μ_{t+1} (as discussed in section 3.5).

4.5 Experimental and Simulation Results

We now present experimental results for abnormal activity detection and tracking using SSA and NSSA models. We have used the airport video (group of passengers deplaning and moving towards the terminal), a simulated configuration sequence and a human action video to test our algorithms.

4.5.1 Dataset and Experiments

We have used a video sequence of passengers deplaning and walking towards the airport terminal as an example of a "stationary shape activity". The number of people in the scene varies with time. We have resampled the curve formed by joining their locations using "arc-length resampling" (described in section 3.7) in all experiments except the temporal abnormality [1] detection where we use "uniform resampling". As we needed observation noise-free data to learn the system model, we used hand-marked passenger locations for training. The mean shape, μ, and the tangent space Gauss Markov model parameters, A, Σ_v, Σ_n, were learnt using this data (as discussed in section 3.3.1). Also the motion model parameters (which in this case model random motion of the camera) were estimated with this data. Simulated test sequences were produced by adding observation noise to the hand-marked data. We did this to study robustness of the method to increasing observation noise. We also tested with real observations obtained using a motion detection algorithm [69]. Both real and simulated observation sequences were tracked using the PF described in section 2.2.2 with the number of particles, $N = 1000$.

This video was provided to us by the Transport Security Administration (TSA) and did not have any instances of abnormal behavior. Abnormal behavior was simulated by making one of the persons walk away in an abnormal direction (in the results shown one person was made to walk away at an angle of 45^o to the X-axis, see figure 4.1(b); 4.1(a) shows a normal activity frame). Now, the person could be moving away at any speed which will make the abnormality a slow or a drastic change. We have simulated this by testing for walk away speeds of $1, 2, 4, 16, 32$ pixels per time step in both x and y directions. The average speed of any person in the normal sequence is about 1 pixel per time step. Thus walk-away velocity of 1 pixel per time step, denoted as $vel. = 1$, corresponds to a slow change which does not go out of track for a long time while $vel. = 32$ is a drastic change that causes the PF to lose track immediately.

We show change detection results and tracks using real observations of the passengers' locations in each frame obtained using a motion detection algorithm described in [69]. The ability of our algorithm to deal with temporal abnormalities [1] is demonstrated as well. We also plot the ROC curves for change detection using the ELL, the tracking error (TE) and a combination of both.

Now all of the above was done using a SSA model. As will be discussed in later subsections, the SSA model is able to track normal behavior and detect abnormality but is not very good for tracking abnormal behaviors. Thus we also experimented with using an NSSA model to track the observation sequences better and yet detect abnormality using ELL. We also generated a simulated shape sequence of normal and abnormal behavior to compare performance of SSA and NSSA. Finally, we have also applied the NSSA model to tracking normal human actions and tracking and detecting abnormal actions.

4.5.2 Group Activity: Abnormality Detection

ELL versus Tracking Error: Slow and Drastic Changes

Figure 4.2 shows ELL, tracking error and OL plots for simulated observation noise. As can seen from Figure 4.2(b) and (c), the tracking error and OL plots look very similar (reason discussed in section 2.7.5).

Next, we discuss the results for observations obtained using a motion detector [69] which have observation noise because of the sensor noise and motion detection error. Figure 4.8(b) shows a slow abnormality ($vel. = 1$) introduced at $t = 5$ which is tracked correctly for a long time (figure 4.3(b) plots the tracking error) and hence we need to use ELL to detect it (ELL plots shown in figure 4.3(a)). Figure 4.8(c) shows a drastic abnormality ($vel. = 32$) which was also introduced at $t = 5$ but loses track immediately. In this case the abnormal observations are ignored and the PF continues to follow the system model. As a result, the ELL (figure 4.3(a)) confuses it for a normal sequence and fails completely, while tracking error (figure 4.3(b)) detects it immediately. In figure 4.3(a), we show the ELL plot for increasing rates of change. With $vel. = 1$, the abnormality (introduced at $t = 5$) gets detected at $t = 27$ and with $vel. = 4$ it gets detected at $t = 12$. For $vel. = 32$, the ELL is unable to detect the abnormality. The tracking error (figure 4.3(b)) detects this abnormality immediately (at $t = 6$) while it misses detecting the slow abnormality ($vel. = 1$). Also, note the OL plots are very similar to the tracking error plots and hence are not shown here.

This demonstrates the need to use a combination of ELL and tracking error to detect both slow and drastic changes (since the aim is to be able to detect any kind of abnormality with rate of change not known). As explained earlier, we declare an abnormality if either the ELL or the tracking error exceeds its corresponding thresholds. The ROC curves for this combined ELL/TE strategy are shown in Figure 4.6. As is discussed below, by combining ELL and TE we are able to detect all slow and drastic changes with detection delay less than 7 time units.

ROC curves and Performance Degradation with increasing Observation Noise

The intuition discussed above is captured numerically in the ROC (Receiver Operating Characteristic) curves [68, 15] for change detection using ELL (figure 4.4(a) and (b) for slow and drastic changes respectively), using tracking error (figure 4.5(a) and (b)) and using a combination of both (figure 4.6(a),(b),(c),(d)). Please note that every figure in the ROC plot has a different y axis range. The blue circles, red stars, magenta triangles and cyan diamonds are the ROC plots for simulated observation noise with increasing variances of $3, 9, 27, 81$ square pixels. The ROC for a change detection problem [15] plots the average detection delay against the mean time between false alarms by varying the detection threshold. The aim of an ROC plot is to choose an *operating point* threshold which minimizes detection delay for a given value of mean time between false alarms.

For the slow change ($vel. = 1$), the detection delay is much lesser using ELL than using the tracking error while the opposite is true for the drastic change ($vel. = 32$). The detection performance degradation of ELL for slow change and of tracking error for drastic change with increasing observation noise is slow. In figure 4.4(a) (ELL for slow change), detection delay is less than or equal to 2 time units for $\sigma^2_{obs} = 3$ and 7 time units for $\sigma^2_{obs} = 81$. In figure 4.5(b) (tracking error for drastic change), the detection delay is less than or equal to 3 time units for $\sigma^2_{obs} = 3$ and 4 time units for $\sigma^2_{obs} = 81$. Since the aim is to be able to detect all kinds of abnormalities (abnormality parameters are assumed not known), we propose to use a combination of the ELL and the tracking error and declare a change when either exceeds its threshold. In figure 4.6, we plot the ROC curves for slow and drastic change detection using a combination of ELL and tracking error. In this case, for each observation noise variance, there are multiple curves, since one needs to vary thresholds for both the ELL and the tracking error to get the ROC. A single curve is for the ELL threshold fixed and tracking error threshold varying. We have a set of curves for varying ELL thresholds. We plot the low and high observation noise cases in two separate plots. As can be seen, the combined strategy has better performance than either ELL or tracking error for all rates of change and for all observation noises (detection delay less than 7 time units in all cases).

Temporal abnormality [1] detection

We also tested our method for detecting what is referred to in [1] as a temporal abnormality (one person stopped in his or her normal path). It gets detected in this framework because there is a change in shape when the person behind the stopped person goes ahead of him (curve becomes concave). We used "uniform resampling" (discussed in section 3.7) which detected temporal abnormality easily using ELL (figure 4.7). "Arc-length resampling" does not work too well in this case. This is because it tends to average out the locations of two closely spaced points, thus smoothing out the concavity which needs to be detected. “Uniform resampling”, on the other hand, assumes the observed points are uniformly sampled and hence gives equal weight to all the observed points irrespective of the distances between them. Thus it is able to detect concavity caused even by two closely spaced points. Another way to detect temporal abnormality would be to use a NSSA model and look at deviations from the expected value of shape velocity.

4.5.3 Group Activity: Tracks

Figure 4.8(a) shows a normal observation frame (circles) and the corresponding tracked configuration (stars), for real observations obtained using a motion detector [69] on the image sequences. The observation noise was modeled to be Gaussian (although the PF can filter non-Gaussian noise as well) and its covariance was learnt from a training sequence of observations obtained using the motion detector. This shows the ability of our model to potentially be used for “tracking to obtain observations”. Figure 4.8(b),(c) show tracking of a slow ($vel = 1$) and drastic ($vel = 32$) abnormality both introduced at $t = 5$. As can be seen, the drastic abnormality has lost track at $t = 7$ while the slow one is not totally out of track even at $t = 13$. The NSSA model tracks abnormality better as is shown below. Note that since we use only a point object abstraction for moving objects (here persons), we show observed and tracked point object locations only without showing the actual images.

4.5.4 Group Activity: Nonstationary Shape Activity (NSSA) Model

We compare here the performance of SSA and NSSA for tracking and detecting abnormal behavior. Very noisy observations were obtained by using the motion detection algorithm of [69]. In figure 4.9(a), we show the tracking error for a "faster" abnormality. NSSA is able to track much better than SSA. In 4.9(b), we show the ELL plots. Thus NSSA detects the "faster" abnormality using ELL, while the SSA detects it using tracking error (loses track). Note here that we use the NSSA model for tracking but the normal system is assumed to be a stationary shape activity (SSA) and so the ELL is evaluated w.r.t. the SSA model only.

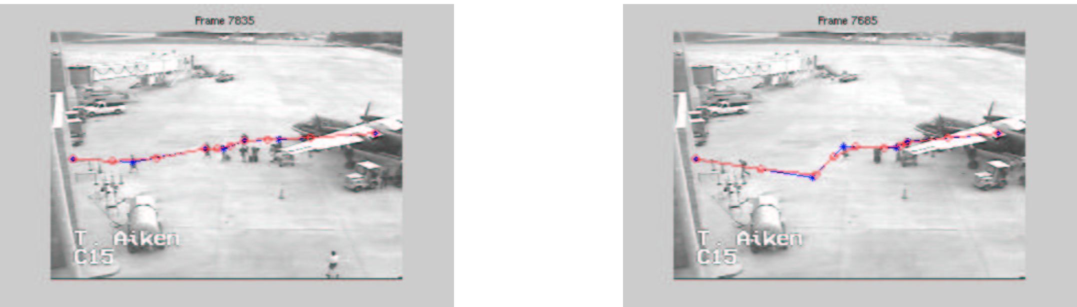
4.5.5 Simulated Shape Sequence: Comparing NSSA and SSA

We first simulated a shape activity sequence, starting with a regular hexagon as the mean. The sequence was stationary for the first 40 frames (around the regular hexagon) and for the next 40 frames, a bias was added to the tangent coordinate at every frame, which resulted in unmodeled non-stationary deformations of the shape (abnormality). We also scaled and rotated each frame according to Markov log-Gaussian and Gaussian models. We used only the stationary part of one such sequence (first 40 frames) as training data to learn both SSA and NSSA parameters. Another such sequence with 40 stationary and 40 nonstationary (abnormal frames) was generated. Four pixel and nine pixel i.i.d. white Gaussian observation noise was added to each frame to produce the noisy observation data.

We attempted to track the noisy observation sequence using both SSA and NSSA models. Both SSA and NSSA track the normal observations equally well (figure 4.10(a)). But within a few frames of introducing the abnormality SSA loses track, while NSSA is able to remain in track till the very end (figure 4.10(b)). Even in 9-pixel noise (very large noise), NSSA is able to track the abnormality. We also plot the tracking error and the ELL(Shape) for 4-pixel observation noise in figure 4.11(a) and (b) and the ELL(Shape) for 9-pixel noise in (c). SSA and NSSA are able to detect abnormality using tracking error and ELL respectively.

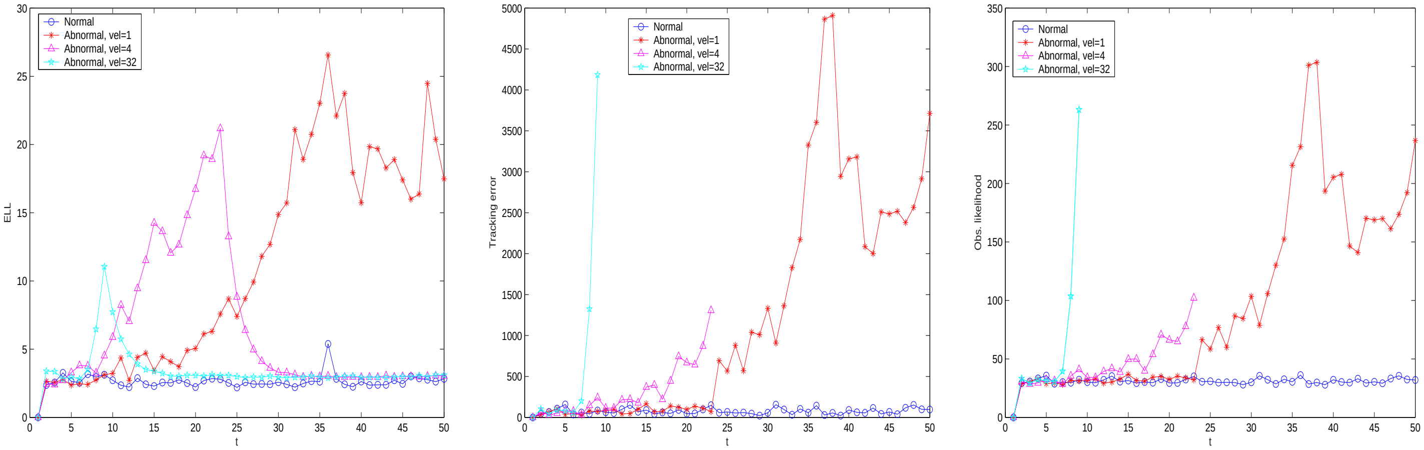
4.5.6 Human Actions: Comparing NSSA and SSA

Next we attempted to track human actions and track as well as detect abnormality in the action. We show here results on tracking a figure skater (shown in figure 4.12(a)). We had observation noise-free locations of landmarks in the normal skater sequence as well as the abnormal one. The 10 landmarks used were head, torso, both elbows, hands, knees and feet. The abnormality was the knee deviating too far away. As before, we used the normal sequence for training SSA and NSSA models; added observation noise to the abnormal one and attempted to track it. We show the tracks (of the landmark locations) along with the ground truth in figure 4.12(b) and (c). The SSA is able to track the normal sequence better than NSSA while it completely fails to track the abnormality. But NSSA is able to track both. In figure 4.13, we show tracking error and ELL (Shape). Here as well, NSSA is able to detect using the ELL while SSA can detect this change using the tracking error (since it has lost track).



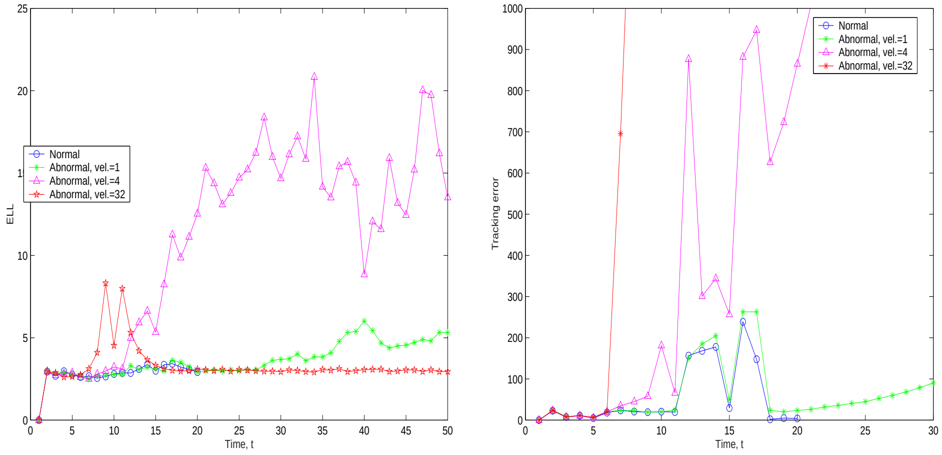
(a) A ‘normal activity’ frame with 4 people (b)Abnormality introduced by making one person walk-away in an abnormal direction

Figure 4.1: Airport example: Passengers deplaning



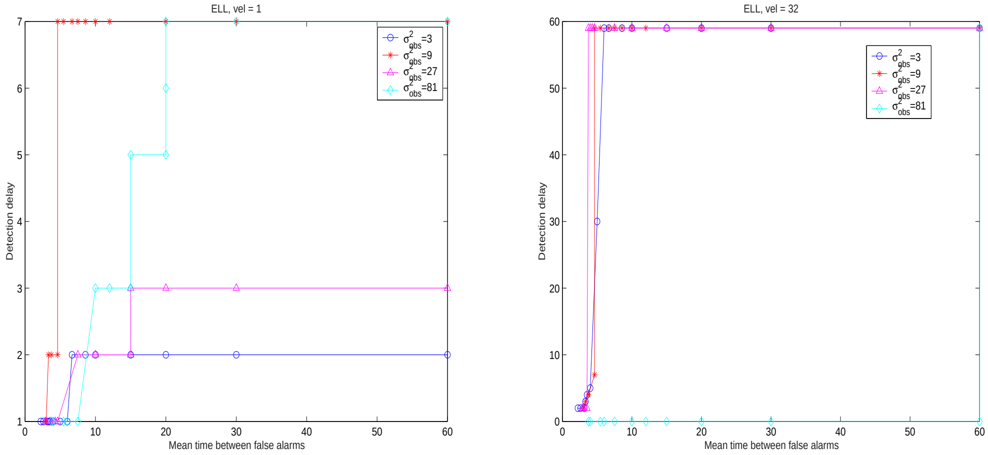
(a) ELL (b) Tracking error (c) OL

Figure 4.2: ELL, Tracking error (TE) and Observation Likelihood (OL) plots: Simulated Observation noise, $\sigma^2_{obs} = 9$ (3-pixel noise). TE and OL plots look alike because of the reasons discussed in section 2.7.5 In the rest of the experiments, we show only the TE plots.



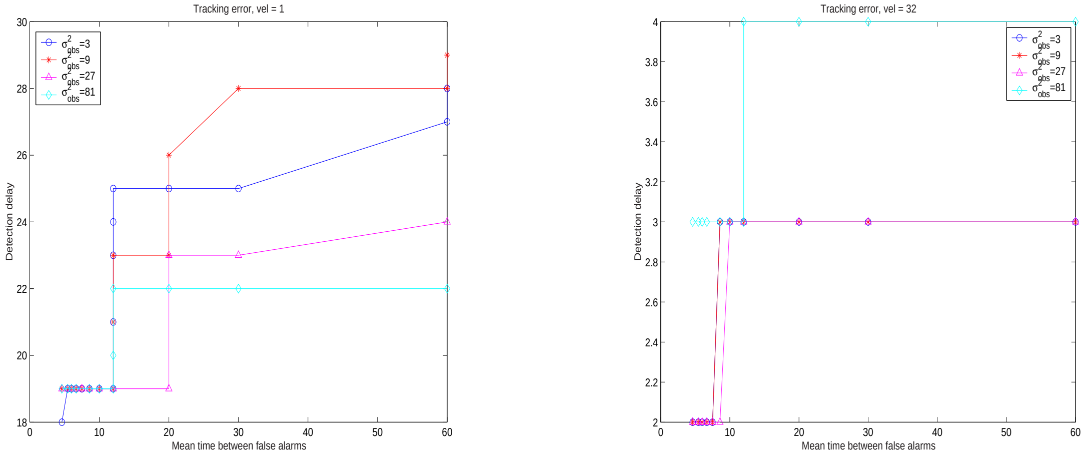
(a) ELL (b) Tracking error

Figure 4.3: ELL and Tracking error plots: Real Observations. Abnormality was introduced at $t = 5$. The ELL is able to detect slow changes better while the tracking error works better for drastic changes. The plots are discussed in Section 4.5.2.



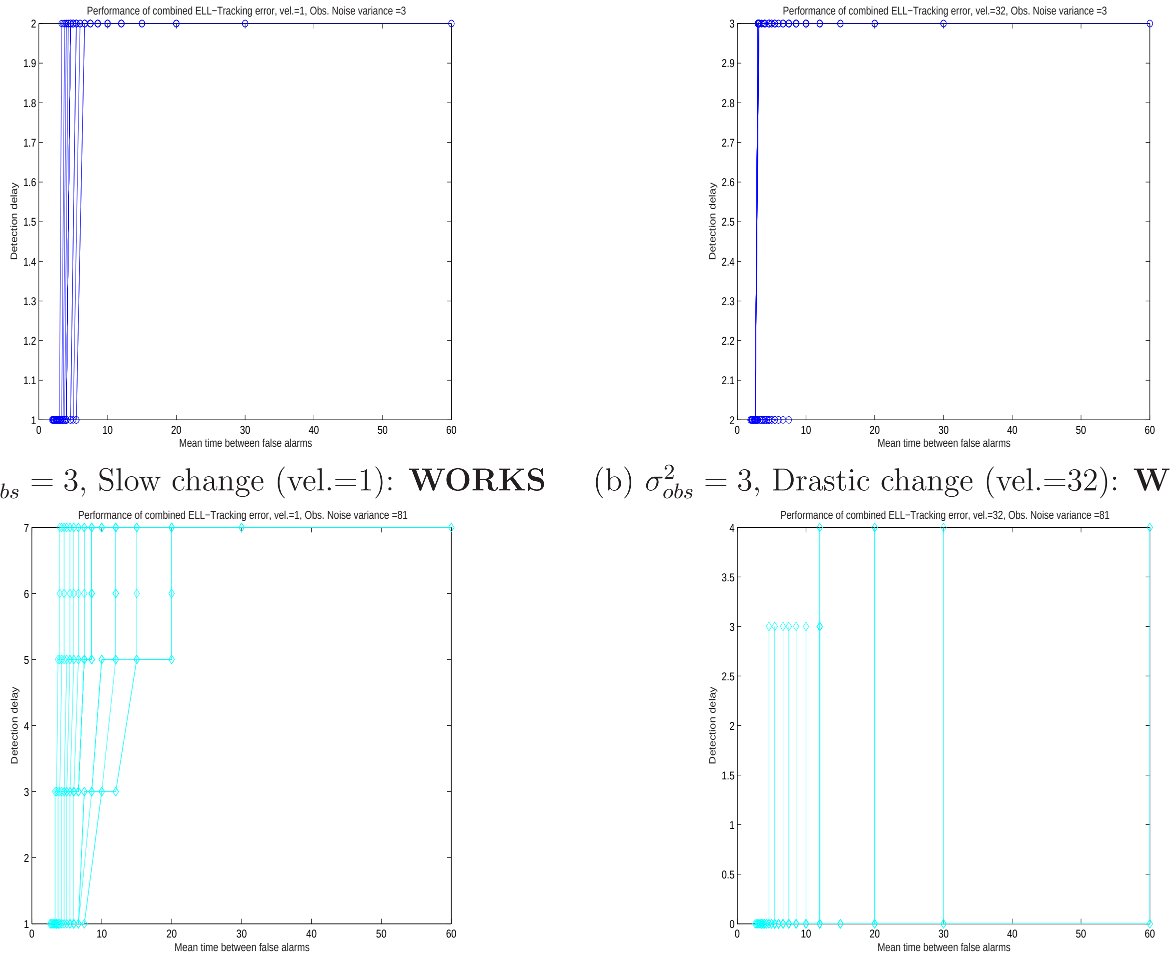
(a) Slow change (vel.=1): **WORKS** (b) Drastic change (vel.=32): FAILS

Figure 4.4: ROCs for Change detection using ELL. Blue circles, red stars, majenta triangles and cyan diamonds plots are for $\sigma^2_{obs} = 3, 9, 27, 81$ respectively. Note that the two plots have different y axis ranges. The ELL completely fails for drastic changes. Detection delays in (b) are very large (60 time units) while for the slow change maximum detection delay is only 7 time units. Plots are discussed in Section 4.5.2.



(a) Slow change (vel.=1): DOES NOT WORK (b) Drastic change (vel.=32): **WORKS**

Figure 4.5: ROCs for Change detection using Tracking error. Blue circles, red stars, majenta triangles and cyan diamonds plots are for $\sigma^2_{obs} = 3, 9, 27, 81$ respectively. Please note that the two plots have different y axis ranges. Tracking error does not detect slow changes easily. Detection delays in (a) are large (maximum delay is 28 time units) while drastic changes are detected almost immediately with delay ≤ 4 time units. Plots are discussed in Section 4.5.2.



(a) $\sigma^2_{obs} = 3$, Slow change (vel.=1): **WORKS**

(b) $\sigma^2_{obs} = 3$, Drastic change (vel.=32): **WORKS**

(c) $\sigma^2_{obs} = 81$, Slow change (vel.=1): **WORKS**

(d) $\sigma^2_{obs} = 81$, Drastic change (vel.=32): **WORKS**

Figure 4.6: ROCs for Change detection using combined ELL-Tracking error. In this case, for each observation noise variance, there are multiple curves, since one needs to vary thresholds for both ELL and tracking error to get the ROC. A single curve is for the ELL threshold fixed and tracking error threshold varying. We have a set of curves for varying ELL thresholds. The maximum detection delay is 2 and 3 time units for $\sigma^2_{obs} = 3$ ((a) and (b)), and 7 and 4 time units for $\sigma^2_{obs} = 81$ ((c) and (d)). Plots are discussed in Section 4.5.2.

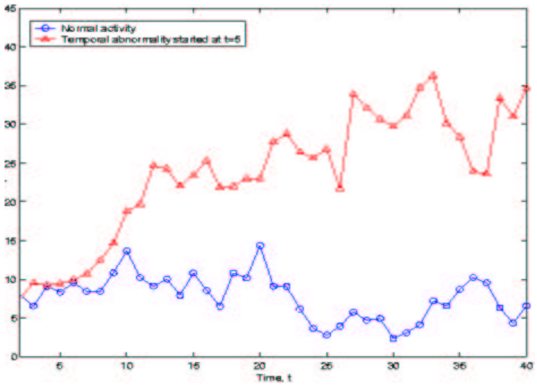
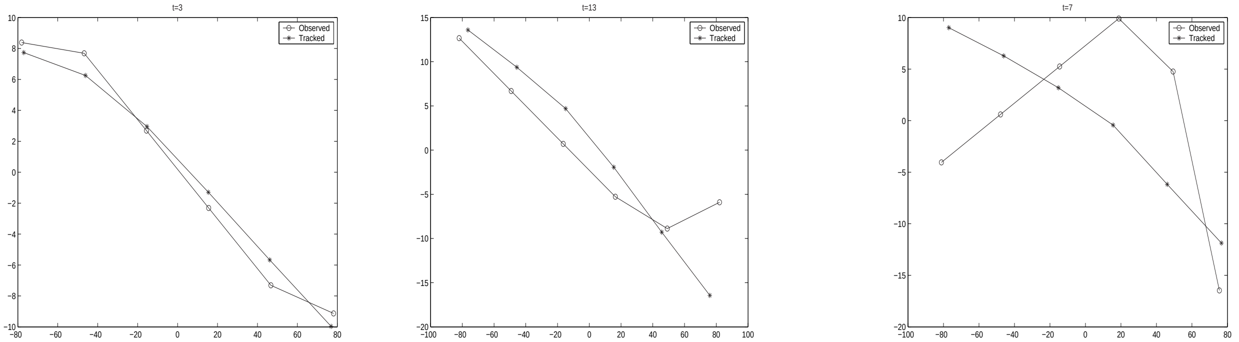


Figure 4.7: ELL plot for Temporal abnormality detection. Abnormality was introduced at $t = 5$. The plot is discussed in Section 4.5.2.

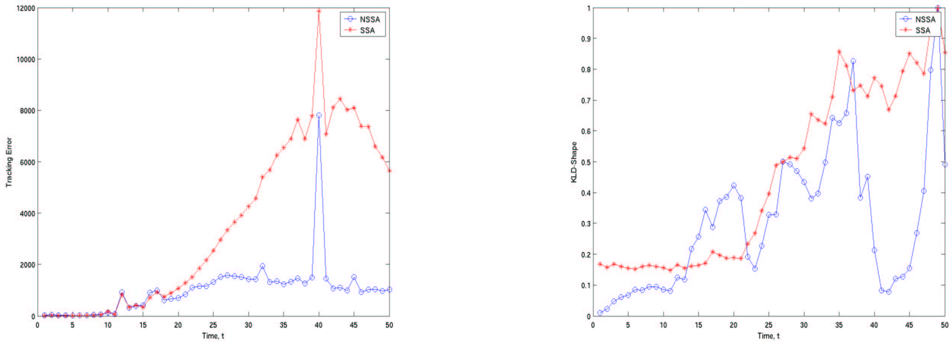


(a)Normal frame: In track

(b)Slow abnormality ($vel = 1$): Still in track

(c)Drastic abnormality ($vel = 32$): Loses track

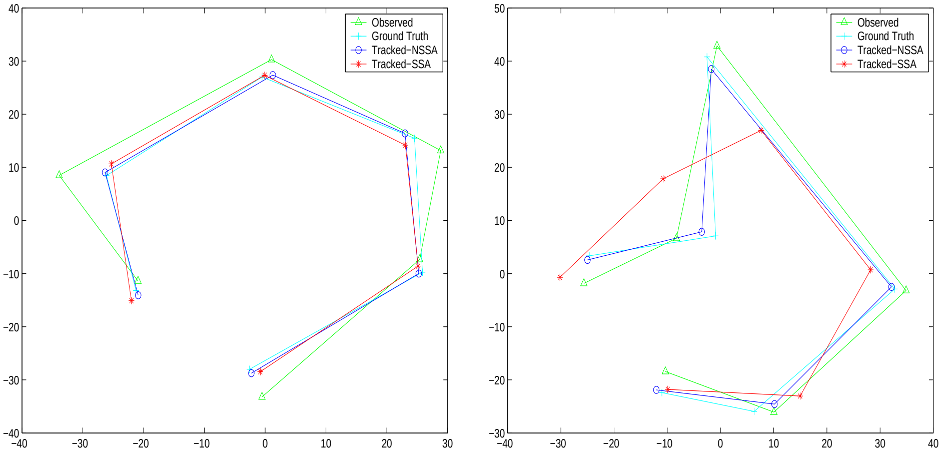
Figure 4.8: Tracks: Real Observations. Plotting the observed and tracked positions of the landmarks (passengers) on the x-y plane. The plots are discussed in Section 4.5.3.



(a) Tracking Error, Fast Change (vel.=4)

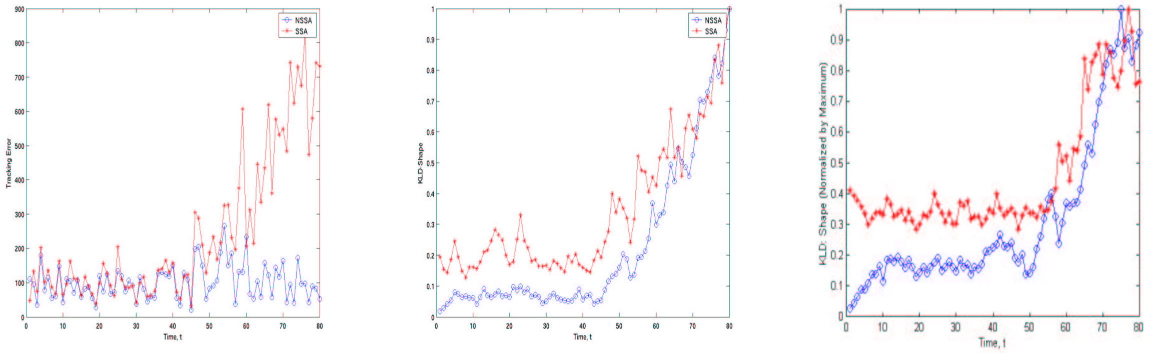
(a) ELL (Shape), Fast Change (vel.=4)

Figure 4.9: Tracking and detecting slow & drastic abnormalities (introduced at $t = 5$): Comparing NSSA and SSA



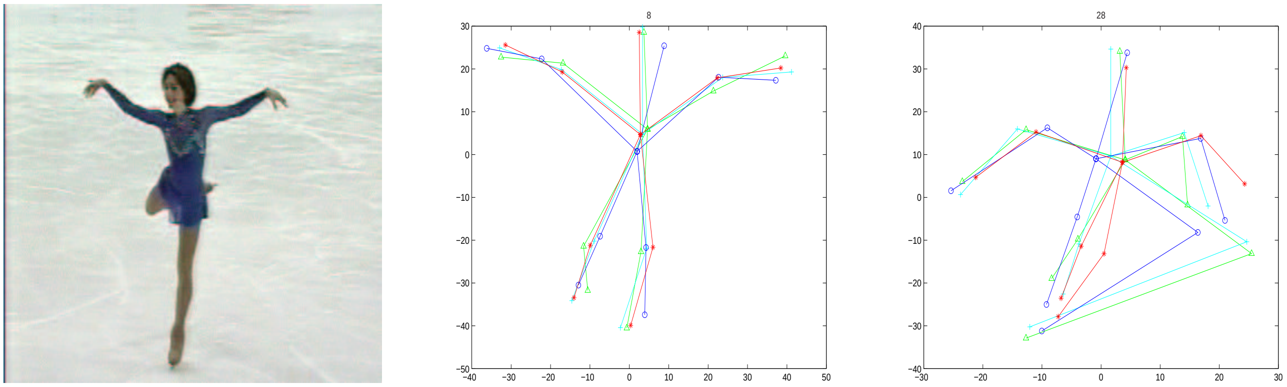
(a) Normal, $\sigma^2_{obs} = 16$ (b) Abnormal, $\sigma^2_{obs} = 16$

Figure 4.10: Simulated shape: Tracking normal and abnormal behavior (introduced at $t = 40$) using SSA and NSSA. NSSA tracks abnormality as well.



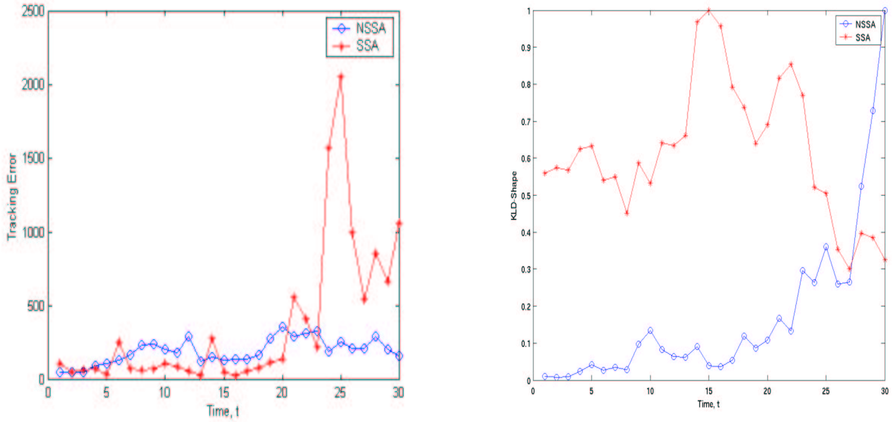
(a) Tracking Error, $\sigma^2_{obs} = 16$ (a) ELL (Shape), $\sigma^2_{obs} = 16$ (b) ELL (Shape), $\sigma^2_{obs} = 81$

Figure 4.11: Simulated Shape Statistics: Abnormality introduced at $t = 40$. NSSA detects abnormality using ELL (since it is able to track) while SSA detects using tracking error (loses track).



(a)The Figure Skater (b) Normal (SSA tracks it better) (c) Abnormal (SSA fails)

Figure 4.12: Tracking the figure skater: The green triangles line is the observed (noisy) data, the cyan -+ line is the ground truth, the blue circles and red stars are filtered shape using NSSA and SSA respectively. Abnormality introduced at $t = 20$.



(a) Tracking Error (b) ELL (Shape)

Figure 4.13: Tracking the figure skater: Abnormality introduced at $t = 20$. NSSA detects abnormality using ELL while SSA detects using the tracking error.

Chapter 5

Principal Component Null Space Analysis (PCNSA)

5.1 Introduction

A fourth contribution of this thesis is Principal Component Null Space Analysis (PCNSA) which is a classification algorithm that approximates the optimal Bayes classifier for Gaussian class conditional distributions with unequal covariance matrices. The abnormal activity detection (described in the previous chapter) is a sequential hypothesis testing problem. For abnormality detection in the fully observed case, i.e. when observation noise is negligible, we use log-likelihood of the state to detect abnormality (discussed in 4.1.1). PCNSA, which approximates the optimal LRT for Gaussian distributions can also be used in this case and as discussed in section 5.6.4 (more detailed discussion in [11]), it has certain advantages.

The PCNSA algorithm can also be used for retrieval problems, we show its application to human action retrieval in section 5.6.4.

The PCNSA algorithm was originally proposed by us for “apples from oranges” type classification problems like object recognition or face recognition under large pose variations. Problems like face recognition under small pose variations that involve discriminating similar objects can be categorized as “apples from apples” type classification problems. “Apples from apples” type problems are those in which different classes have similar class covariance matrices (in particular similar directions of low and high intra-class variance) while for “apples from oranges” type problems classes can have very different class covariance matrix structures. As an extreme case of this situation, the minimum variance direction of one class could be a maximum variance direction for another. We propose a linear classifier (which we call PCNSA) for this situation of unequal covariance matrices, which actually approximates the optimal Bayesian solution.

We have evaluated bounds on PCNSA’s classification error probability (in Section 5.3) and discussed conditions under which it would outperform Linear Discriminant Analysis (LDA) and when it would fail (in Section 5.4). Applications of PCNSA to object recognition (figure 5.4(a)), feature matching (see figure 5.4(b)), face recognition under large pose/expression variations (see figure 5.5), abnormal group activity detection [11] (see figure 4.1) and video retrieval are discussed in Section 5.6.

5.1.1 Problem Statement

Consider a P-dimensional data sample $\mathbf{Y}$ from class i (denote class i by C_i).

$$(\mathbf{Y})_{P\times 1}|\{Y \in C_i\} \sim \mathcal{N}(\mu_{full,i}, \Sigma_{full,i}) \tag{5.1}$$

For high dimensional data like images, the real dimensionality of data (with noise removed) is much smaller than P. Thus we perform Principal Component Analysis (PCA) to remove directions with only noise and retain directions with large between class variance. The PCA takes data from all classes as a single sample and evaluates the common mean, $\bar{\mu}_{full}$, and common covariance matrix, $\bar{\Sigma}_{full}$, and chooses the L leading eigenvectors of $\bar{\Sigma}_{full}$ as the principal components' subspace (PCA space). The data sample of class i projected in the L-dimensional PCA space with projection matrix, $(W^{PCA})_{P\times L}$, is

$$\begin{aligned}
(\mathbf{X})_{L\times 1} &\triangleq W^{PCA^T}(\mathbf{Y} - \bar{\mu}_{full}) \sim \mathcal{N}(\mu_i, \Sigma_i) \text{ where} \\
(\mu_i)_{L\times 1} &\triangleq W^{PCA^T}(\mu_{full,i} - \bar{\mu}_{full}), \\
(\Sigma_i)_{L\times L} &\triangleq W^{PCA^T}\Sigma_{full,i}W^{PCA}.
\end{aligned} \tag{5.2}$$

In this work, we address the classification problem for the most general class covariance matrices (unequal, non-white) with eigenvalue decomposition $\Sigma_i = U_i\Lambda_iU_i^T$. LDA, on the other hand, assumes same eigenvectors for all classes ($U_i = U$) i.e. similar directions of low and high variance while the PCA when used for classification assumes $U_i = I, \Lambda_i = \sigma_i^2 I$ i.e. the class covariance matrices are white in PCA space.

5.2 Principal Component Null Space Analysis

Principal Component Null Space Analysis (PCNSA) first applies the PCA transform on the entire data for dimensionality reduction and for maximizing the between class variance [56]. In the PCA space, it finds for each class i, an M_i dimensional subspace along which the class's intra-class variance is smallest. We call this subspace the *approximate null space (ANS)* of class i since for most applications, the lowest variance(s) are usually "much smaller" than the highest (the class covariance matrix is usually ill-conditioned). A query is classified into class i if its distance from the i^{th} class's mean in the i^{th} class' ANS is a minimum. We first discuss below the assumptions required for PCNSA to work as a classification algorithm and in 5.2.2, provide the stepwise algorithm.

5.2.1 Assumptions

1. For all classes i, Σ_i has a high enough condition number, $R = \lambda_{max}/\lambda_{min}$ so that an approximate null space (ANS) exists. This would happen for most real classification problems especially the "apples from oranges" ones.

2. Distance of the mean of any class, j, from mean of any other class i in ANS of class i (denoted by N_i) is "significant" compared to the total distance $||\mu_j - \mu_i||$, i.e. there exists a $\rho < 1$ "significantly" greater than zero s.t. $||N_i^T(\mu_j - \mu_i)|| > \rho||\mu_j - \mu_i||$ [1]. This assumption is also not very restrictive when the number of classes is small.

[1] This condition is required because if it were not satisfied for two classes i and j, and if their null spaces coincide, i.e. $N_i = N_j$, then we would have $d_i(\mathbf{X}) = d_j(\mathbf{X})$ always, causing the algorithm to fail always

3. We assume approximate linear separability (which is required for any linear classification algorithm). Gaussian distributed classes would be approximately linearly separable if the square of the distance between class means of any two classes is of the order of the maximum eigenvalues (variance) of both classes.

5.2.2 Algorithm

1. **Obtain the PCA Space:** Evaluate the sample mean, $\bar{\mu}_{full}$ and covariance, $\bar{\Sigma}_{full}$ of the training data of all classes taken together as one sample set. Obtain the PCA projection matrix, $(W^{PCA})_{P\times L}$ whose columns are the L leading eigenvectors of $\bar{\Sigma}_{full}$.

2. Project the training data samples of each class into the PCA space. Evaluate for each class i, the class mean, μ_i, and the class covariance, Σ_i, in the PCA space.

3. **Obtain Class ANS:** Evaluate the approximate null space, $(N_i)_{L\times M_i}$, for each class i as the M_i trailing eigenvectors of Σ_i (choose M_i so that the eigenvalues in ANS satisfy, $\lambda \leq 10^{-4}\lambda_{max}$). Assumption 1 ensures that it exists.

4. **Obtain Valid Classification Directions in ANS:** Now $N_i = [e_{i,1}|e_{i,2}|...e_{i,k}...|e_{i,M_i}]$. A null space direction, e, is a valid classification direction only if the distance between class means along that direction is "significantly" greater than zero i.e. $e = e_{i,k}$ satisfies

$$\begin{aligned} |(\mu_i - \mu_j)^T e| &> \rho||\mu_i - \mu_j||, \ \forall j \neq i, \ 0 < \rho < 1 \\ \text{or equivalently, } \ \theta \triangleq \cos^{-1}(\frac{|(\mu_i - \mu_j)^T e|}{||\mu_i - \mu_j||}) &< \theta_0 < \frac{\pi}{2}. \end{aligned} \tag{5.3}$$

The PCNSA projection matrix for class i (${W_i}^{NSA}$) is chosen as those columns of N_i which satisfy this condition. By assumption 2, this is possible to do.

5. **Classification:** Project the query $\mathbf{Y}$ into the PCA space as $\mathbf{X} = W^{PCA^T}(\mathbf{Y} - \bar{\mu}_{full})$. The most likely class, c, is given by $c = \arg\min_i d_i(\mathbf{X})$ where

$$d_i(\mathbf{X}) \triangleq ||{W_i}^{NSA^T}(\mathbf{X} - \mu_i)||. \tag{5.4}$$

5.3 Classification Error Probability

We obtain the error probability bound for classification using PCNSA for a two class problem. We first evaluate error probability assuming a one dimensional ANS per class so that $W_i^{NSA} = (N_i)_{L\times 1}$ and Gaussian distributed classes. We then show how this can be extended to the general case of M_i dimensional ANS per class. We discuss in Section 5.3.3, how the error probability analysis can be extended to non-gaussian but symmetric, unimodal distributions. The two class error probability expressions can be used to obtain a union bound [65] for the multi-class error probability.

5.3.1 One-dimensional ANS per class

Define E_i as the event that error occurs given query $\mathbf{X} \in C_i$ (class i). The average error probability is $P_{e,avg} = \frac{P(E_1)+P(E_2)}{2}$. Using PCNSA's class specific metric defined in (5.4), the error event E_1 is

$$E_1 \triangleq \{{d_2}^2(\mathbf{X}) < {d_1}^2(\mathbf{X}) | \mathbf{X} \in C_1\} \tag{5.5}$$

Now since ANS is one dimensional, ${W_1}^{NSA} = N_1$ and $d_1(\mathbf{X}) = |N_1^T(\mathbf{X} - \mu_1)|$ is a scalar. Using (5.2),

$$N_1^T(\mathbf{X} - \mu_1)|\{\mathbf{X} \in C_1\} \sim \mathcal{N}(0, \lambda_{ANS,1}). \tag{5.6}$$

To upper bound $P(E_1)$, define

$$\Delta = k\sqrt{\lambda_{ANS,1}} \tag{5.7}$$

$$\text{Then,} \quad P({d_1}^2(\mathbf{X}) > \Delta^2|\mathbf{X} \in C_1) = 2(1 - \Phi(k)) \overset{\triangle}{=} g(k) \tag{5.8}$$

where $\Phi(.)$ is the cdf of an $\mathcal{N}(0, 1)$ random variable. We choose k large enough so that $g(k)$ is small. For $k = 10$, $g(k) = 10^{-23}$. Now the error event E_1 (defined in (5.5)) can be split as [2],

$$\begin{aligned} E_1 &= \{{d_2}^2(\mathbf{X}) \leq {d_1}^2(\mathbf{X}), {d_1}^2(\mathbf{X}) \leq \Delta^2\} \cup \{{d_2}^2(\mathbf{X}) \leq {d_1}^2(\mathbf{X}), {d_1}^2(\mathbf{X}) > \Delta^2\} \\ &\subseteq \{{d_2}^2(\mathbf{X}) \leq \Delta^2\} \cup \{{d_1}^2(\mathbf{X}) > \Delta^2\}. \end{aligned} \tag{5.9}$$

Thus, $P(E_1) \leq P({d_2}^2(\mathbf{X}) \leq \Delta^2) + g(k)$ [3]. Now $d_2(\mathbf{X}) = |N_2^T(\mathbf{X} - \mu_2)|$. Using (5.2) we get,

$$\mathbf{Z} \overset{\triangle}{=} \frac{N_2^T(\mathbf{X} - \mu_1)}{\sqrt{N_2^T \Sigma_1 N_2}} \sim \mathcal{N}(0, 1). \tag{5.10}$$

So defining,

$$\alpha \overset{\triangle}{=} |N_2^T(\mu_1 - \mu_2)|, \quad \text{and} \quad \sigma \overset{\triangle}{=} \sqrt{N_2^T \Sigma_1 N_2}, \tag{5.11}$$

we get, $P({d_2}^2(\mathbf{X}) < \Delta^2) = P(\frac{\alpha - \Delta}{\sigma} < \mathbf{Z} < \frac{\alpha + \Delta}{\sigma})$. Thus

$$P(E_1) \leq P(\frac{\alpha - \Delta}{\sigma} < \mathbf{Z} < \frac{\alpha + \Delta}{\sigma}) + g(k)$$

[2] Assume $\mathbf{X} \in C_1$ everywhere

[3] This bound is tight when the right hand side is small or equivalently when λ_{ANS} is small.

$$\begin{aligned} &= \Phi(\frac{\alpha+\Delta}{\sigma}) - \Phi(\frac{\alpha-\Delta}{\sigma}) + g(k) \\ &= \int_{\frac{\alpha}{\sigma}(1-\frac{\Delta}{\alpha})}^{\frac{\alpha}{\sigma}(1+\frac{\Delta}{\alpha})} \mathcal{N}(z;0,1)dz + g(k) \end{aligned} \tag{5.12}$$

Similar expressions can be obtained for $P(E_2)$.

5.3.2 M_i-dimensional ANS per class

In this case N_1 and N_2 are $L \times M_i, i = 1,2$ dimensional matrices. Define Δ as

$$\Delta^2 = k^2(\sum_{j=1}^{M_1} \lambda^2_{ANS,1,j}) \tag{5.13}$$

Error event E_1 is as defined in (5.5) and can be bounded using exactly the same logic as in (5.9). Thus we have

$$P(E_1) \le P({d_2}^2(\mathbf{X}) < \Delta^2 | \mathbf{X} \in C_1) + P({d_1}^2(\mathbf{X}) > \Delta^2 | \mathbf{X} \in C_1) \tag{5.14}$$

First consider $P({d_1}^2(\mathbf{X}) > \Delta^2)$. Define

$$\mathbf{Z_{N_1}} \triangleq N_1^T(\mathbf{X} - \mu_1) \sim \mathcal{N}(0, \Lambda_{ANS,1}), \qquad \Lambda_{ANS,1} \text{ is diagnol} \tag{5.15}$$

then ${d_1}^2(\mathbf{X}) = ||\mathbf{Z_{N_1}}||^2$. It is easy to see that

$$\{{d_1}^2(\mathbf{X}) > \Delta^2\} \subseteq \{\cap_j A_j\}^c, \ A_j = \{Z^2_{N_1,j} < k^2\lambda^2_{ANS,1,j}\} \tag{5.16}$$

By (5.15), the components of the vector $\mathbf{Z_{N_1}}$ are independent and hence the events A_j are independent. Also, $P(A_j) = 1 - g(k)$ where $g(k)$ is defined in (5.8). Thus using (5.16),

$$P({d_1}^2(\mathbf{X}) > \Delta^2) \le P(\{\cap_j A_j\}^c) = 1 - \prod_{j=1}^{M_1} P(A_j) = 1 - (1 - g(k))^{M_1} \triangleq g_{M_1}(k). \tag{5.17}$$

Now consider $P({d_2}^2(\mathbf{X}) \leq \Delta^2)$. Define

$$\begin{aligned} \beta &\triangleq N_2^T(\mu_2 - \mu_1) \quad \text{and} \quad \Sigma \triangleq N_2^T \Sigma_1 N_2 \\ \text{and} \ \ \mathbf{Z_{N_2}} &\triangleq N_2^T(\mathbf{X} - \mu_1) \sim \mathcal{N}(0, \Sigma), \end{aligned} \tag{5.18}$$

then ${d_2}^2(\mathbf{X}) = ||\mathbf{Z_{N_2}} - \beta||^2$. Let $\Sigma = USU^T$ be the eigenvalue decomposition of Σ. U is the $M_2 \times M_2$ matrix of eigenvectors and $S = diag(\sigma_j^2)$ is a diagnol matrix of its eigenvalues. Using U to diagnolize $\mathbf{Z_{N_2}}$, we get

$$\mathbf{Z_{N_2}^{indep}} = U^T \mathbf{Z_{N_2}} \sim \mathcal{N}(0, S), \ S \text{ is diagnol} \tag{5.19}$$

$$\text{Also define, } \ \alpha \triangleq |U^T \beta| \tag{5.20}$$

Since U is orthonormal, $||\mathbf{Z_{N_2}^{indep}} - \alpha|| = ||U^T(\mathbf{Z_{N_2}} - \beta)|| = ||\mathbf{Z_{N_2}} - \beta||$ and so

$$P({d_2}^2(\mathbf{X}) \leq \Delta^2) = P(||\mathbf{Z_{N_2}^{indep}} - \alpha||^2 < \Delta^2) \tag{5.21}$$

Now, it is easy to see that

$$\{||\mathbf{Z_{N_2}^{indep}} - \alpha||^2 < \Delta^2\} \subseteq \cap_j B_j, \ B_j = \{(\mathbf{Z_{N_2,j}^{indep}} - \alpha_j)^2 < \Delta^2\}$$

The events $\{B_j\}$ are independent since elements of the vector $\mathbf{Z_{N_2}^{indep}}$ are independent. Using (5.12), $P(B_j) = P(\alpha_j - \Delta < \mathbf{Z_{N_2,j}^{indep}} < \alpha_j + \Delta) = [\Phi(\frac{\alpha_j + \Delta}{\sigma_j}) - \Phi(\frac{\alpha_j - \Delta}{\sigma_j})]$ where $\sigma_j^2 = S_{j,j}$.

$$\text{Thus, } \ P({d_2}^2(\mathbf{X}) \leq \Delta^2) \leq P(\cap_j B_j) = \prod_{j=1}^{M_2} [\Phi(\frac{\alpha_j + \Delta}{\sigma_j}) - \Phi(\frac{\alpha_j - \Delta}{\sigma_j})] \tag{5.22}$$

Finally, combining (5.14), (5.17) and (5.22), we get

$$P(E_1) \leq \prod_{j=1}^{M_2} [\Phi(\frac{\alpha_j + \Delta}{\sigma_j}) - \Phi(\frac{\alpha_j - \Delta}{\sigma_j})] + g_{M_1}(k) \tag{5.23}$$

5.3.3 Extension to Non-Gaussian Distributions

Now the analysis for one-dimensional ANS can be extended to the case of non-Gaussian distributions which are symmetric and unimodal[4]. Assume that the distribution of $\mathbf{X}$ is symmetric about its mean, μ_1, and has covariance matrix Σ_1. Let $F_1(.)$ be the cumulative distribution function (cdf) and $f_1(.)$ the probability distribution function (pdf) of $\frac{N_1^T(X-\mu_1)}{\sqrt{N_1^T\Sigma_1 N_1}}$ i.e. it is the cdf of $N_1^T\mathbf{X}$ after location normalization to zero mean and scale normalization to unit variance. Similarly, let $F_2(.)$ and $f_2(.)$ be the cdf and pdf of $\frac{N_2^T(X-\mu_1)}{\sqrt{N_2^T\Sigma_1 N_2}}$. Then it is easy to see that $P({d_1}^2(\mathbf{X}) > \Delta^2) = 2(1-F_1(k))$. Also $\mathbf{Z}$ defined in (5.10), has cdf F_2. Hence

$$\begin{aligned} P(E_1) &\leq F_2(\frac{\alpha+\Delta}{\sigma}) - F_2(\frac{\alpha-\Delta}{\sigma}) + 2(1-F_1(k)) \\ &= \int_{\frac{\alpha}{\sigma}(1-\frac{\Delta}{\alpha})}^{\frac{\alpha}{\sigma}(1+\frac{\Delta}{\alpha})} f_2(z)dz + 2\int_k^\infty f_1(z)dz \end{aligned} \tag{5.24}$$

Now the distribution of X is unimodal implies that f_1 and f_2 are unimodal. Thus assuming f_1 is not heavy tailed, the second term is small (for k large enough). Also if Δ/σ is small or if α/σ is large, and f_2 is not heavy tailed, the first term is small as well. It is easy to see that if the pdf of X is lighter tailed than Gaussian (sub-Gaussian), then the upper bound on $P(E_1)$ will be smaller than that for Gaussian distributed classes. Hence *for all sub-Gaussian, unimodal, symmetric distributions, the PCNSA performance will be better than for Gaussian distributed classes.*

[4] The M_i-dimensional ANS analysis is more difficult to extend because it hinges on the assumption that dependent Gaussian variables can be made independent by a linear transformation.

5.4 Comparison with Subspace Linear Discriminant Analysis (SLDA)

5.4.1 Subspace Linear Discriminant Analysis (SLDA)

As discussed in Section 5.1, SLDA [56] is a linear classification algorithm. It first computes a PCA space for the training data of all classes taken together as one sample. In PCA space, it performs linear discriminant analysis, i.e. it computes the most discriminant directions, W^{LDA}, as

$$W^{LDA} = \arg \max_{W:W^TW=1} \frac{(W^T\Sigma_b W)}{(W^T\Sigma_w W)}, \tag{5.25}$$

where $\Sigma_b = (\sum_{i=1}^{K}(\mu_i - \bar{\mu}))/K$ and $(\Sigma_w = \sum_{i=1}^{K} \Sigma_i)/K$. The classification metric is

$$d_i(\mathbf{X}) = ||W^{LDA^T}(\mathbf{X} - \mu_i)||. \tag{5.26}$$

The error event for a two class problem (one dimensional W^{LDA}) is $E_1 \triangleq \{{d_2}^2(\mathbf{X}) < {d_1}^2(\mathbf{X})|\mathbf{X} \in C_1\}$. The error probability for LDA follows directly using Gaussian hypothesis testing [65] and has also been discussed in [73]:

$$\begin{aligned} P(E_1) = 1 - \Phi(\frac{\hat{\alpha}}{\hat{\sigma}}) \quad &= \quad \int_{\frac{\hat{\alpha}}{\hat{\sigma}}}^{\infty} \mathcal{N}(z;0,1)dz \quad \text{where} \\ \hat{\alpha} \triangleq \frac{|W^{LDA^T}(\mu_2 - \mu_1)|}{2}, \; \hat{\sigma} \quad &\triangleq \quad \sqrt{W^{LDA^T}\Sigma_1 W^{LDA}}. \end{aligned} \tag{5.27}$$

5.4.2 Classification Performance Comparison

Looking at expressions (5.12) and (5.27), it is clear that the PCNSA error probability can be made small if either the class means' distance along ANS is large compared to standard deviation of other classes along ANS ($\alpha/\sigma \to \infty$) or if it is large compared to ANS standard deviation of the class itself ($\Delta/\alpha \to 0$). While LDA necessarily requires class means' distance along the classification directions to be large compared to the standard deviation of all classes ($\frac{\hat{\alpha}}{\hat{\sigma}} \to \infty$). We now compare the error probability expressions when using PCNSA and LDA for a best and a worst case situation for LDA. We make some simplifying assumptions to reduce the number of variables.

Simplifying Assumptions

We assume a two dimensional PCA space and each class having a one dimensional ANS and one direction of maximum variance. Also, we assume that the eigenvalues of covariance matrices of both classes are equal, i.e. $\lambda_{max,1} = \lambda_{max,2} = \lambda_{max}$ and $\lambda_{ANS,1} = \lambda_{ANS,2} = \lambda_{min}$ [5]. Now by the linear separability assumption, $||\mu_1 - \mu_2||$ should be of the order of $\sqrt{\lambda_{max}}$, we take $||\mu_1 - \mu_2|| = \sqrt{\lambda_{max}}$. With these assumptions, the error probability expressions can be reduced to a function of three variables: the condition number, $R = \lambda_{max}/\lambda_{min}$, the angle between N_1 and N_2, denoted by ψ and the angle made by the the vector $(\mu_1 - \mu_2)$ (line joining the means) with N_2, denoted by θ. In two dimensions these two angles automatically fix the angle between the direction of $(\mu_1 - \mu_2)$ and N_1. We study the variation of error probability as a function of R and θ for two extreme values of ψ, $\psi = 0^o$ (case 1) and $\psi = 90^o$ (case 2) which correspond to best case and worst case scenarios for LDA. We show that PCNSA works well in both these extreme cases as long as the assumptions of Section

[5] PCNSA actually requires $\lambda_{max,2}/\lambda_{ANS,1}$ and $\lambda_{max,1}/\lambda_{ANS,2}$ to be large. Our assumption combines both these into a single variable $R = \lambda_{max}/\lambda_{ANS}$.

5.2.1 are satisfied and fails completely when they are not.

Qualitative Comparison

We first provide a qualitative comparison of the two cases ($\psi = 0, 90^o$) using figure 5.1(a) and (b). In both figures, the condition number R is set to a large value (assumption 1 of Section 5.2.1). We have $\theta \approx 0$ in figure 5.1(a) and $\theta \approx 45^o$ in 5.1(b), both being far from 90^o (assumption 2 of Section 5.2.1). Case 1 shown in Figure 5.1(a) is a best case scenario for both PCNSA and LDA since the Y axis is the ANS direction for both classes and the common LDA direction (W^{LDA}) is close to the Y axis (ANS direction for either class). As the variance of both classes along W^{LDA} is small, the LDA works very well. Also the variance of class 1 along ANS of class 2 (and vice versa) is small and $\theta = 0^o$ is far from 90^0. Hence the performance of PCNSA will also be very good in this case. The class boundaries defined by PCNSA and SLDA in the figure are almost coincident.

But in case 2 shown in Figure 5.1(b), the maximum variance direction of one class coincides with the ANS of the other. This is the worst case for LDA but PCNSA works very well in this case. In fact, this case demonstrates the need for the PCNSA algorithm. Here, the Y axis is ANS direction for class 1 but a maximum variance direction for class 2 and vice versa for X axis. Thus W^{LDA} is along the direction ($\mu_1 - \mu_2$) (direction AB in the figure). Along W^{LDA} both classes have a large enough variance. So LDA has a high error probability in this case. The region for the LDA error event E_1^{LDA} is the region of ellipse 1 to the right of line PR and for E_2^{LDA} it is the region of ellipse 2 below line PR. But PCNSA still works well because the integration region for E_1^{NSA} is only those parts of ellipse 1 that are closer to μ_2 (point B) along N_2 (X axis) than to μ_1 (point A) along N_1 (Y axis) and similarly for E_2^{NSA}. Thus the error region is the small overlap region of the two ellipses (region PQRS) for both E_1^{NSA} and E_2^{NSA}.

Quantitative Comparison: Error Probabilities as a function of R and θ

Now in case 1 ($\psi = 0^o$), $N_1 = N_2 = [0\ 1]^T$. Using the simplifying assumptions and definitions (5.11), $\Sigma_1 = \Sigma_2 = diag\{\lambda_{max}, \lambda_{min}\}$, $\alpha = \sqrt{\lambda_{max}} \cos\theta$ and $\sigma = \sqrt{\lambda_{min}}$. $R = \lambda_{max}/\lambda_{min}$ is the condition number of either class's covariance matrix. Substituting in (5.12), we get

$$P(E_1^{NSA}) \leq \int_{\sqrt{R}\cos\theta - k}^{\sqrt{R}\cos\theta + k} \mathcal{N}(z; 0, 1) dz \overset{\triangle}{=} P(E^{NSA\ bound}) \tag{5.28}$$

and the same expression for $P(E_2^{NSA})$ so that $P(E_{avg}^{NSA}) = P(E_1^{NSA})$. We also evaluate $P(E^{LDA})$ using (5.27). MATLAB is used to evaluate W^{LDA} for different values of R and θ. Both $P(E^{NSA,bound})$ and $P(E^{LDA})$ are plotted in figure 5.2(a), for $\theta \in [0, 90^o]$, and $R = 10^3, 10^4, 10^5$. This is a best case scenario for both SLDA and PCNSA as long as θ is bounded away from 90^o (assumption 2 of section 5.2.1 satisfied). We have for both NSA and LDA

$$\lim_{R\to\infty} P(E^{NSA/LDA}, R, \theta) = 0, \quad \forall \quad |\theta| < \theta_0 < 90^o$$

$$\begin{aligned} \text{But,} \ \lim_{\theta\to 90^o} \lim_{R\to\infty} P(E^{NSA\ bound}, R, \theta) &= 1 \\ \text{while,} \ \lim_{\theta\to 90^o} \lim_{R\to\infty} P(E^{LDA}, R, \theta) &\approx 0.31 \end{aligned} \tag{5.29}$$

i.e. when θ tends to 90^o, PCNSA fails completely while the performance of LDA degrades gracefully [6].

Now in case 2 ($\psi = 90^o$), $N_1 \perp N_2$ i.e. $N_1 = [0\ 1]^T$ and $N_2 = [1\ 0]^T$. So $\Sigma_1 = diag\{\lambda_{max}, \lambda_{min}\}$ while $\Sigma_2 = diag\{\lambda_{min}, \lambda_{max}\}$. Again using the simplifying assumptions and (5.11), $\sigma = \sqrt{N_2^T \Sigma_1 N_2} = \sqrt{\lambda_{max}}$ and $\alpha = \sqrt{\lambda_{max}} \cos\theta$. This gives

$$P(E_1^{NSA}) \leq \int_{\cos\theta - \frac{k}{\sqrt{R}}}^{\cos\theta + \frac{k}{\sqrt{R}}} \mathcal{N}(z; 0, 1) dz. \tag{5.30}$$

For LDA, $\Sigma_w = \frac{\Sigma_1 + \Sigma_2}{2} = diag\{\frac{\lambda_{max} + \lambda_{min}}{2}, \frac{\lambda_{max} + \lambda_{min}}{2}\}$ so that W^{LDA} is along $(\mu_1 - \mu_2)$ i.e. $W^{LDA} = [\cos\theta\ \ \sin\theta]^T$. Thus we have

$$P(E_1^{LDA}) = \int_{\frac{\sqrt{R}}{2(\sqrt{R\cos^2\theta + \sin^2\theta})}}^{\infty} \mathcal{N}(z; 0, 1) dz. \tag{5.31}$$

The expressions for $P(E_2)$ for both PCNSA and LDA have the "cos" replaced by "sin". Case 2, as also discussed earlier, is the worst case for LDA. The average error probabilities are plotted in figure 5.2(b). The LDA error probability in this case converges to a non-zero value which depends on θ, i.e. we get (using (5.31)),

$$\lim_{R \to \infty} P(E^{LDA}, R, \theta) = \frac{\int_{\frac{\sec\theta}{2}}^{\infty} \mathcal{N}(z; 0, 1) dz + \int_{\frac{cosec\theta}{2}}^{\infty} \mathcal{N}(z; 0, 1) dz}{2} \tag{5.32}$$

The above limit is approximately the LDA curve (dotted line) shown in figure 5.2(b). PCNSA still works very well in this case, i.e. we have (using (5.30))

$$\lim_{R \to \infty} P(E^{NSA}, R, \theta) = 0 \quad \forall\ \ \theta \tag{5.33}$$

although the rate of convergence is much slower than in case 1.

[6] The LDA limit is an approximate numerically evaluated value

Discussion

From the above analysis, we conclude that PCNSA fails for small values of R (no null space) or when the distance between class means projected along ANS becomes small ($\theta \rightarrow 90^o$). We have included checks in steps 3 and 4 of our algorithm (section 5.2.2) to avoid these two situations. For all other cases, its performance is superior or as good as SLDA as long as the query data follows the training data distribution. *By evaluating the error probability expressions, one can choose between LDA and PCNSA for a given application or even use different algorithms for distinguishing different class pairs in a multi-class classification problem.*

5.4.3 Comparing Size of Training Data

In real applications, the model is never exact and so the ANS calculation is never exact. Finding the approximate null space directions requires a large amount of training data to correctly find directions along which there is almost no variation. The size of the training data set per class should be at least two to three times the dimension of the PCA space to correctly estimate the lowest eigenvalues (and corresponding eigenvectors) of the class covariance matrix. SLDA can do with lesser training data and PCA requires the least. This fact has been observed experimentally and is plotted in figure 5.3. Performance of PCNSA when compared with LDA in real applications is not as good as that predicted by the analytical expressions. As part of future work, we hope to do a perturbation analysis similar to that done in [73] to compare robustness to model error of both algorithms. Another relevant work is [74] which compares PCA and SLDA training data size by evaluating them on many face databases.

5.4.4 Comparing 'New' (Untrained) Class Detection Ability

Since PCNSA defines a class specific metric, 'new' (untrained) classes can be detected most easily using PCNSA. When a query belongs to a trained class its distance from the class mean along that class's approximate null space is a very sharp minimum while a query belonging to a new class will have no such sharp minimum. Detecting new classes is more difficult with LDA because trained classes will also not have very sharp minimum distances from their own class means along the LDA directions. The new class detection strategy used by us is discussed in Section 5.6.2. Also, the PCNSA class-specific metric does not require any knowledge of the second class and so can be used for binary hypothesis testing problems where the statistics of the not null hypothesis (H_1) are not known. We have discussed its application to abnormal activity detection (where "abnormality" is not characterized) in Section 5.6.4.

5.5 Relation to Multispace KL

Multispace KL (MKL) [58] when used for classification, separates all classes into subsets of similar classes and for each subset derives a principal component subspace representation. For classification of a query, it first finds the subspace (subset) from which the distance of the query is a minimum and in that subspace finds the class mean that is closest to the query in Euclidean norm. The distance from space defined in [58] is equivalent to the distance in

ANS space defined by us. PCNSA if put in this framework, defines one subspace for each class and classifies queries by finding the subspace (and hence the class) to which the query is closest.

In fact MKL is exactly equivalent to performing NSA to choose the nearest subspace (subset) and then using PCA to choose the nearest class within the subset. We can extend the error probability analysis of Section 5.3 to analyzing the classification error probability of MKL. The error in choosing the correct subspace is $P(E^{NSA})$ with ANS dimension $M_i = n-k$ (Using notation from [58] where k is the subspace dimension and n is the original data dimension). The bound for this error, $P(E^{NSA\ bound})$, for a two class problem is given by (5.23). The error in classification within the subspace is the error in classification using Euclidean distance in PCA space. The classification error (given class i) using MKL would be

$$P(E_i^{MKL}) = P(E_i^{NSA}) + (1 - P(E_i^{NSA}))P(E_i^{PCA}) \leq P(E_i^{NSA\ bound}) + P(E_i^{PCA}). \quad (5.34)$$

5.6 Experiments and Results

We first present a Monte Carlo verification of correctness of the PCNSA upper bounds derived in Section 5.3. We then discuss the new (untrained) class detection problem and a heuristic solution to it using PCNSA. We propose a modification of PCNSA which we call progressive-PCNSA that varies the dimension of ANS on the fly and also detects new classes. We then compare performance of PCNSA, progressive-PCNSA, SLDA and PCA for three image classification applications - *object recognition*, *feature matching for image registration* and *face recognition under large pose/expression variation.* Finally we show applications of PCNSA to two video classification problems - *abnormal activity detection* and *action retrieval*, using a shape dynamical framework proposed by us in another work [11].

5.6.1 Synthetic Data Verification

We simulated $L = 3$ and $L = 20$ dimensional PCA space data at random and evaluated the PCNSA error probabilities to verify the correctness of the error probability bounds both using one dimensional ANS and multidimensional ANS. We also generated the mean and covariance matrices at random. Each element of the mean was i.i.d. normally distributed with mean zero and variance one. The covariance matrix was generated as CC^T where the matrix C had i.i.d. standard normal entries. The tightness of the PCNSA error bound depended on the smallness of the ANS eigenvalues. Also, one could obtain the tightest possible bound by varying the value of k until the smallest value of error probability was obtained. Another observation is that as the PCA space dimension, L, is increased the performance of PCNSA improves. This is because as the number of eigenvalues is increased, there is greater chance that the ratio of lowest to highest eigenvalue is small (or equivalently R is large). We found the for $L = 3$, SLDA performed better than PCNSA more times, but for $L = 20$ the opposite is true. For $L = 20$, PCNSA outperformed SLDA (when using analytically calculated PCNSA upper bound for PCNSA) 86% of times and 99.9% of the times when using Monte-Carlo simulation data.

5.6.2 New Class Detection and Progressive-PCNSA

New Class Detection

A common problem in most classification applications is to detect when a query does not belong to any of the classes for which the classifier has been trained. We call such a query as belonging to a 'new' class. Since PCNSA uses a class-specific metric, its ability to detect 'new' classes is better. We use the following heuristic idea to test for a 'new' class: If distances from two or more classes are roughly equal, we conclude that the query belongs to a 'new' class. This is because a query will have a very sharp minimum in its own class's ANS and if there is no such sharp minimum, then one can say that it does not belong to any of the trained classes. We classify a query $\mathbf{X}$ as belonging to a 'new' class if the minimum distance ($d_c(\mathbf{X})$) is greater than a threshold t times the distance from any other class ($d_i(\mathbf{X})$,$i \neq c$) with $t < 1$ i.e.

$$d_c(\mathbf{X}) > t d_i(\mathbf{X}) \ \ \forall i \neq c, \quad t < 1 \tag{5.35}$$

$$\text{or equivalently} \ \ \frac{d_c(\mathbf{X})}{\min_{i \neq c} d_i(\mathbf{X})} > t \tag{5.36}$$

The value of t governs the false alarm probability. If we define H_0 *as the hypothesis that the query belongs to one of the K trained classes* and H_1 *as the hypothesis that it belongs to an untrained ('new') class*, then false alarm is the event that the algorithm decides in favor of H_1 ('new' class) when actually H_0 is true (query comes from a trained class) [75]. One could use Neyman Pearson's lemma [75] to choose t to minimize the miss probability (probability of wrongly classifying a query from a 'new' class as belonging to one of the K trained classes) given a maximum value of false alarm probability. But since H_0 is a complex hypothesis, this is analytically intractable and hence we choose the value of t experimentally.

Progressive-PCNSA

Progressive-PCNSA is a modification of the PCNSA algorithm to choose the number of ANS directions on the fly and to also use the above new class detection strategy. In practice, when the number of classes is large, quite often there is no one single direction of the ANS of class i which satisfies (5.3) for all $j \neq i$. As a practical solution to this problem, we vary the dimension of ANS of all classes from M_l to M_h and evaluate the ratio given in the left hand side of (5.36) until it is less than t. The class c for which the distance defined in (5.4) is minimized and the ratio is less than t is chosen as the most likely class. If this does not happen for any class, for any value of $M \in [M_l, M_h]$, we declare the query as belonging to a 'new' class. The stepwise classification procedure is as follows:

1. Vary ANS dimension from $M = M_l, M_l + 1, ...M_h$. For each value of M,

   - Evaluate $d_i(\mathbf{X})$ for all classes using (5.4) and with W_i^{NSA} the M trailing eigenvectors of Σ_i. Find the minimum distance $d_c(\mathbf{X})$ and the corresponding class c.

   - Evaluate (5.36). If it is true, then try the next higher dimension of ANS for all classes. If it is false, then *declare the current minimum distance class 'c' as the most likely query class* and stop.

2. If (5.36) is true for all values of $M \in [M_l, M_h]$, *declare the query as belonging to a 'new' class.*

5.6.3 Image Classification Experiments

We compare the performance of PCNSA, SLDA and PCA for object recognition, facial feature matching and face recognition. We show the superior performance of PCNSA for new class detection by leaving a few classes untrained and testing for data from those classes. There are three kinds of classification errors

-**Misclassification error given H_0 :** A query from trained class i gets wrongly classified as trained class j, $i \neq j$.

-**False Alarm (Type I error) given H_0 :** A query from any 'trained' class gets wrongly classified as 'new'.

-**Miss (Type II error) given H_1 :** A query from a 'new' class gets wrongly classified as some 'trained' class. Correct new class detection probability is $1 - P(\text{Miss})$.

The total error probability is $P(H_0) * (P(\text{Misclassification}|H_0) + P(\text{False Alarm}|H_0)) + P(H_1) * P(\text{Miss})$. The new class detection threshold, t, can be adjusted based on the requirements of the problem, if the application can tolerate false alarms but is sensitive to misses and misclassification, t can be reduced. For PCNSA, if a really low value is used for t, the misclassification probability can be reduced to zero. Thus test data classified as 'not new' can be used as labeled data for training thus increasing the amount of training data and consequently improving the performance of PCNSA which requires large amounts of training data. This idea is motivated by the discriminant-EM idea described in [76] for LDA. In the experiments described below, two kinds of tests were performed: - **Test (a) :** First assuming no possibility of a 'new' class. In this case the only type of error is misclassification error.

and **Test (b) :** Allowing possibility of 'new' class (hypothesis H_1) and setting $P(H_1)$ to a non-zero value so that all three types of errors can occur.

Object Recognition

The algorithm was tested on the Columbia Object Image Library (COIL-20) which contains 20 different objects and 72 views of each object taken at 5 degree apart orientations. Due to the entirely different covariance matrix structures of different objects, PCA and SLDA do not work as well as PCNSA. 'Leave 10 out' testing was done by choosing 10 frames per class at a time for testing and the rest 62 for training and in this way a total of 1400 tests were carried out by choosing different test and training samples every time. Sample images from the 20 classes are shown in figure 5.4(a). Table 5.1 (a) and (b) tabulate the results for the two types of tests described above. A total of 1400 queries were tested each time. In (a), $P(H_1) = 0$ while in (b), only three classes were trained and 17 were left 'untrained' so that $P(H_0) = 0.15, P(H_1) = 0.85$. As can be seen from the table, progressive-PCNSA performs best in terms of all three errors followed by SLDA and then PCA. The 'miss' probability using prog-PCNSA/ PCNSA is almost half that of SLDA or PCA thus indicating the superior new class detection ability of PCNSA.

Feature Matching for Image Registration

Image registration is an important problem in 3D model alignment and baseline stereo. The first step in image registration is detecting features and obtaining feature correspondences between two or more frames. [14] uses a cornerfinder algorithm followed by k-means clustering for facial feature detection. Feature matching is posed as a posterior hypothesis testing problem, i.e. detected features are matched to seven pre-trained facial feature classes (shown in figure 5.4(b)) and the probability of correct match to a class is taken to be proportional to correlation with mean of that class. We propose to replace this correlation match by distance in PCA, SLDA or PCNSA space which would use both class mean and covariance information. We show that using PCNSA space for obtaining distance measures gives lowest error rates. Also new feature detection is very important here, since as the face moves, new (previously occluded) features can appear. Tests were done using 110 training images per class to train the PCA, LDA or PCNSA spaces and 10 images to test and twelve such iterations were run with different training and test data (total 840 tests). Results for tests (a) and (b) described above are shown in table 5.2. In (b), only three classes are trained and four are left untrained so that $P(H_0) = 0.43, P(H_1) = 0.57$.

Also, variation with reduced training data sizes is shown in figure 5.3. As discussed in Section 5.4.2, PCA works well even for really small training data sizes per class followed by LDA while PCNSA requires much more training data to obtain correct directions of minimum intra-class variation.

Face Recognition

Face recognition has been discussed very briefly only as an example of an 'apples from apples' type application where LDA and PCNSA perform equally well. The algorithms were tested on two standard face databases: The UMIST face database[77] which consists of 23 images of each person taken in different poses is shown in figure 5.5(a). The Yale database which has 11 images per subject: one per facial expression or configuration - center-light, w/glasses, happy, left-light, w/no glasses, normal, right-light, sad, sleepy, surprised, and wink is shown in figure 5.5(b) [7]. The face images can be downloaded from http://images.ee.umist.ac.uk/danny/database.html and

http://cvc.yale.edu/projects/yalefaces/yalefaces.html respectively. The 'one leave out' strategy was adopted for testing, i.e. all except one image of each class were used in the training and the one left out image was used as the query. When there are no new classes (tables 5.3(a) and 5.4(a)), PCNSA and LDA perform equally well, PCNSA is superior for UMIST while LDA is superior for the Yale database. But in test (b) where we train three classes and test on data from all fifteen, PCNSA has a significantly better new class detection ability and hence outperforms the other two.

5.6.4 Video Classification/Retrieval Experiments

Abnormal Activity Detection

Abnormal activity detection in the fully observed case (discussed in section 4.1.1) can be treated as a two class classification problem with only statistics of the normal class known. We have proposed in chapters 3 and 4, a shape based dynamical model for modeling activity

[7] The reason only these two databases were used is that they had enough training data per class to obtain reliable ANS representations for each class. When training data is small, performance of PCNSA deteriorates very fast.

performed by a group of moving landmarks (here people). We repeat here some of the details of the problem formulation. A normal activity frame and an abnormal activity frame are shown in figure 4.1. We represent a stationary shape activity by a mean shape plus a linear dynamical model in the tangent space [9] at the mean shape. The dynamics in tangent space is modeled by a linear Gauss-Markov (G-M) model, $v_t = Av_{t-1} + n_t$. We define an ANS for Σ_n which is the covariance matrix of the system noise, n_t. The training algorithm using a "normal activity sequence" is:

$$\{Y_t\}_{t=1}^T \longrightarrow \{w_t\}_{t=1}^T \longrightarrow S_\mu, \{z_t\}_{t=1}^T \longrightarrow \{v_t = [I - S_\mu S_\mu^*] z_t\}_{t=1}^T \longrightarrow A, \Sigma_v, \Sigma_n \text{ where}$$

Y_t is the configuration vector at time t, w_t is the preshape obtained after translation and scale normalization of Y_t, S_μ is the Procrustes mean shape obtained after Generalized Procrustes Analysis [9] on the preshapes, z_t is the shape obtained after aligning the preshapes, w_t, to S_μ [9]. v_t are the tangent coordinates in the tangent space at S_μ, A is the autoregression matrix, Σ_v, Σ_n are the the covariance matrices of v_t and n_t [11].

We used sum square distance from mean (zero) in ANS of Σ_n over a subsequence of past frames as the activity metric to detect an abnormal activity at a given time. For L=20 past frames the activity metric at time t is

$$d_{20}(t)^2 = \sum_{\tau=t-20}^{t} ||W^{NSA^T}(v_\tau - Av_{\tau-1})||^2. \tag{5.37}$$

We observed in [11] that this detected abnormality faster than both full Euclidean distance and full Mahalonobis distance (log likelihood under the Gauss Markov model)[8] [11]. Plots of

[8] In this application, the data dimension was originally quite small (8 dimensional) and hence dimension-

the activity metric as a function of time for normal activity and two kinds of abnormalities are shown in figure 5.6.

Action Retrieval

We show here an application of PCNSA to action retrieval in the landmark shape dynamical framework of chapter 3. We use motion capture data (which provides locations of 53 human joints in a set of frames) to learn Procrustes mean shapes and PCNSA spaces of tangent space at the Procrustes means, for three different actions - "walking", "brooming", and "sitting". In this case the landmarks were the different joints. Also, for 53 joint locations the dimension of tangent to shape space was quite large , $2 * 53 - 4 = 102$ and hence dimensionality reduction is required in this case. For each class, we define the PCA space by projecting all classes into its tangent space and then define a Gauss-Markov model in this reduced dimension space. The algorithm is as follows:

1. For each class i, learn the Procrustes mean shape and tangent space.

2. For each class i,

   - Project data from all classes into its tangent space and learn a $L = 20$-dim PCA space, W_i^{PCA} for the class.

   - Project training data of class i into this PCA space, to learn the Gauss-Markov

ality reduction using PCA was not required. Also, the "abnormal" class was not characterized, so we could not apply PCA to increase between class variance. For the same reason, LDA could not be used for this application.

model, $A_i, \Sigma_{n,i}, \Sigma_{v,i}$ in PCA space. Project the autoregression matrix A_i back into full tangent space to get $A_{full,i} = W_i^{PCA} A_i W_i^{PCA^T}$.

- Learn W_i^{NSA}, the ANS projection matrix of $\Sigma_{n,i}$. Combine both PCA and NSA projection matrices to obtain $W_i^{project} = W_i^{PCA} W_i^{NSA}$.

3. Given a test sequence,

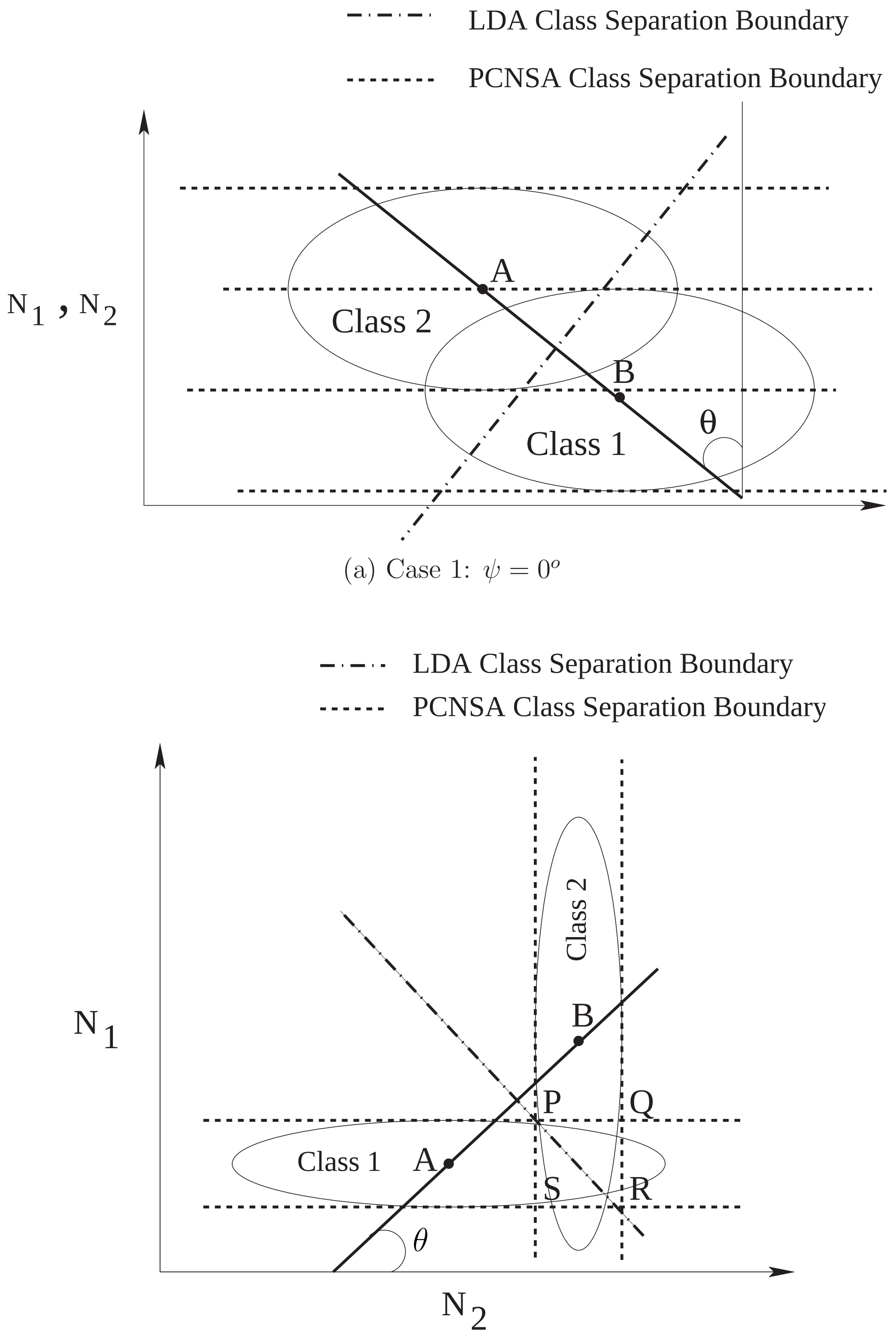
- For each class i, project the sequence into its tangent space to obtain $\{v_{t,i}\}$.

- Choose the most likely class c as $c = \arg\min_i d_i$ where

$$d_i = \sum_{\tau=t-20}^{t} ||W_i^{project^T}(v_{\tau,i} - A_{full,i} v_{\tau-1,i})||^2 \tag{5.38}$$

We used one sequence of each of the actions to learn the mean shape, PCA space, G-M model parameters, and ANS for each class. We then used different instances of walking, brooming and sitting actions as queries and attempted to retrieve the closest action to the given action. We show the distances in table 5.5. The query actions were prowl-walk, 2 brooming sequences, crawl, jog, 2 sitting sequences, 3 walking sequences and a sad-walk sequence (shown in first row). We have underlined the distance of a query from its closest action. As can be seen, for all the 5 walk sequences, the "walk" sequence is correctly retrieved. Also for the two broom sequences and two sit sequences, the correct action is retrieved. For crawl, which is a new class, the minimum distance (dmin) and second largest distance (dmin2) are quite close, so using the new class detection method given in (5.35) with $t = 0.5$, it gets classified as a new class.

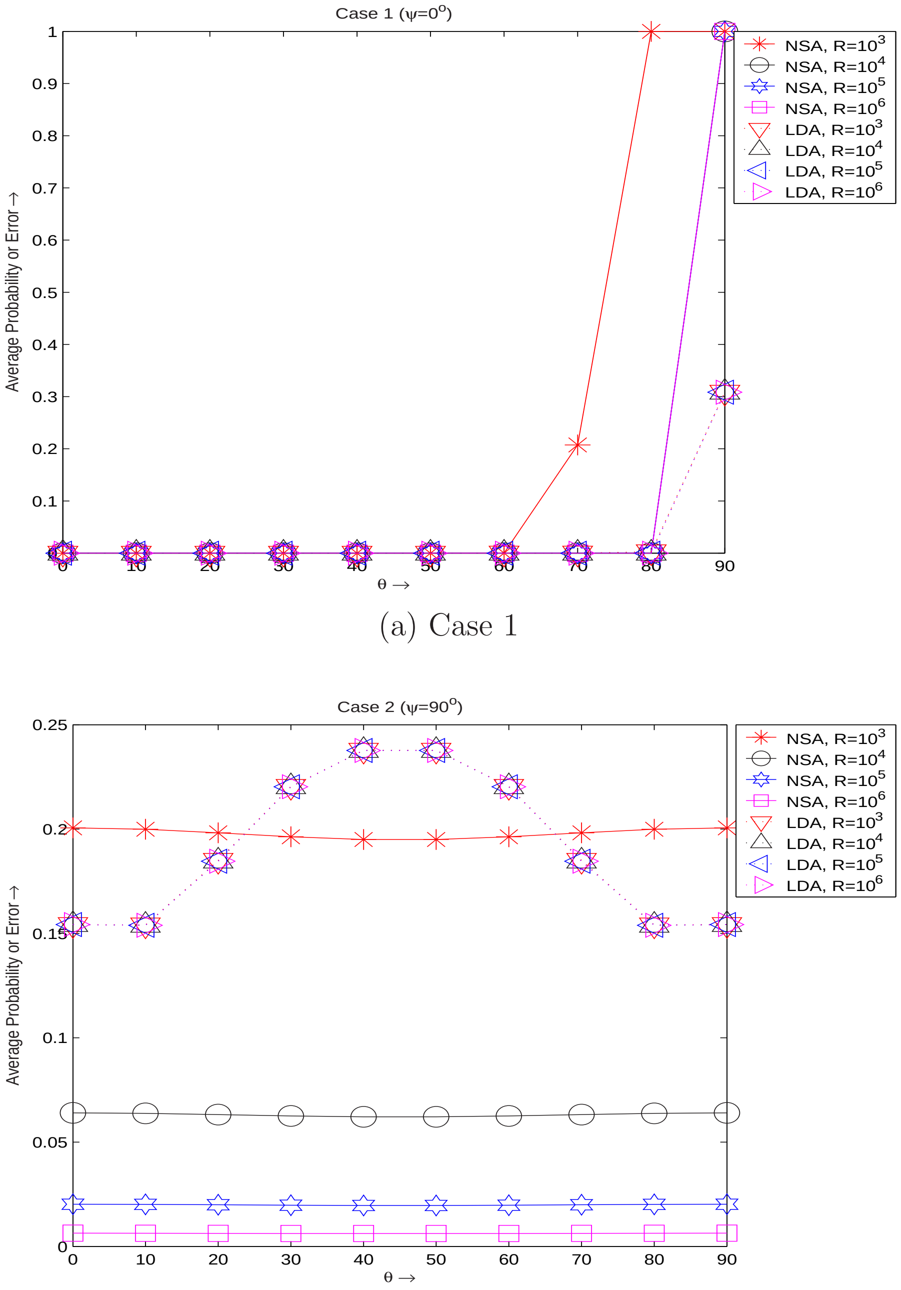
Now for a large database retrieval application or in fact for any *classification problem involving large number of classes*, we can *select subsets of classes with similar within-class* *covariance matrices and obtain ANS for each subset (as in [58]). PCNSA can then be used to choose the subset to which the query is closest and LDA to classify within the subset.*


LDA Class Separation Boundary
PCNSA Class Separation Boundary
N1 , N2
A
Class 2
B
θ
Class 1
LDA Class Separation Boundary
PCNSA Class Separation Boundary
Class 2
N1
B
P
Q
Class 1
A
S
R
θ
N2

(a) Case 1: $\psi = 0^o$

(b) Case 2: $\psi = 90^o$

Figure 5.1: θ is the angle between the line AB and the Y-axis in (a) and between AB and the X-axis in (b). **Case 1** with ANS directions (Y-axis) of both classes coinciding is shown in (a). **Case 2** is shown in (b). Y axis is ANS for class 1 & maximum variance direction for class 2, vice versa for X axis.



(a) Case 1

(b) Case 2

Figure 5.2: Average probability of error as a function of θ for different values of condition number R for (a) Case 1 (b) Case 2. As can be seen the LDA error probability does not vary much with R in either case (curves for all R values are coincident) and also does not degrade much as $\theta \rightarrow 90^o$.

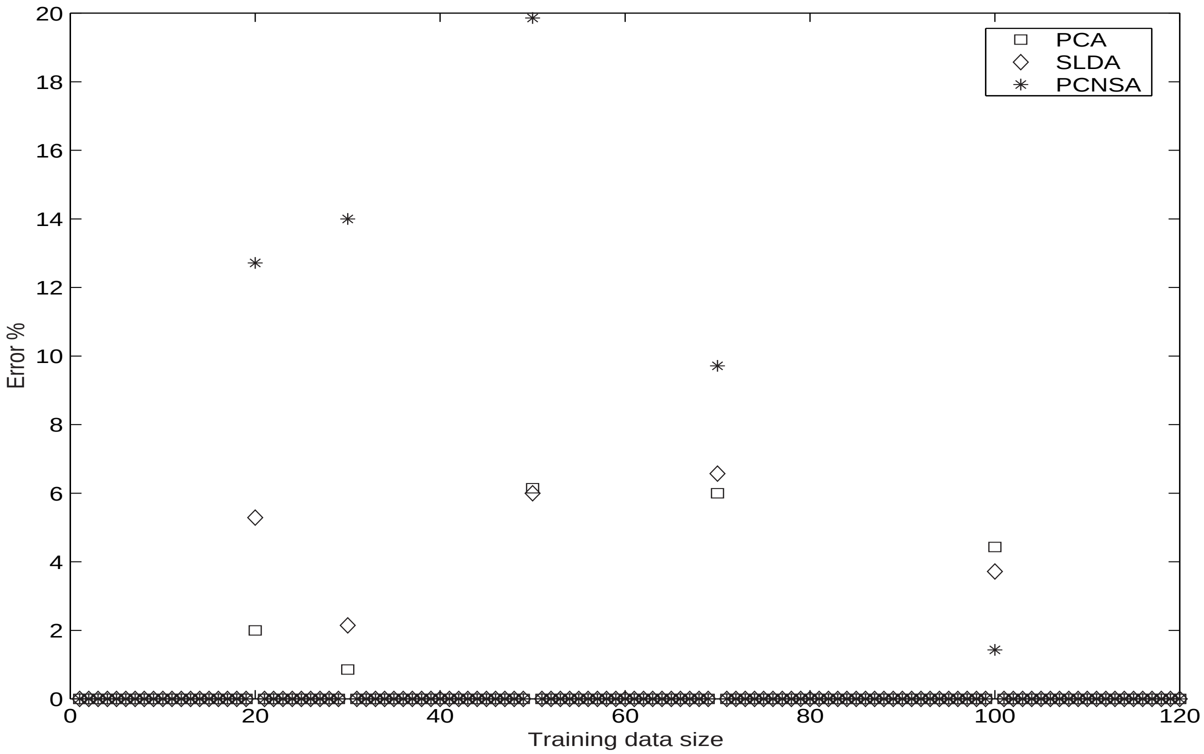
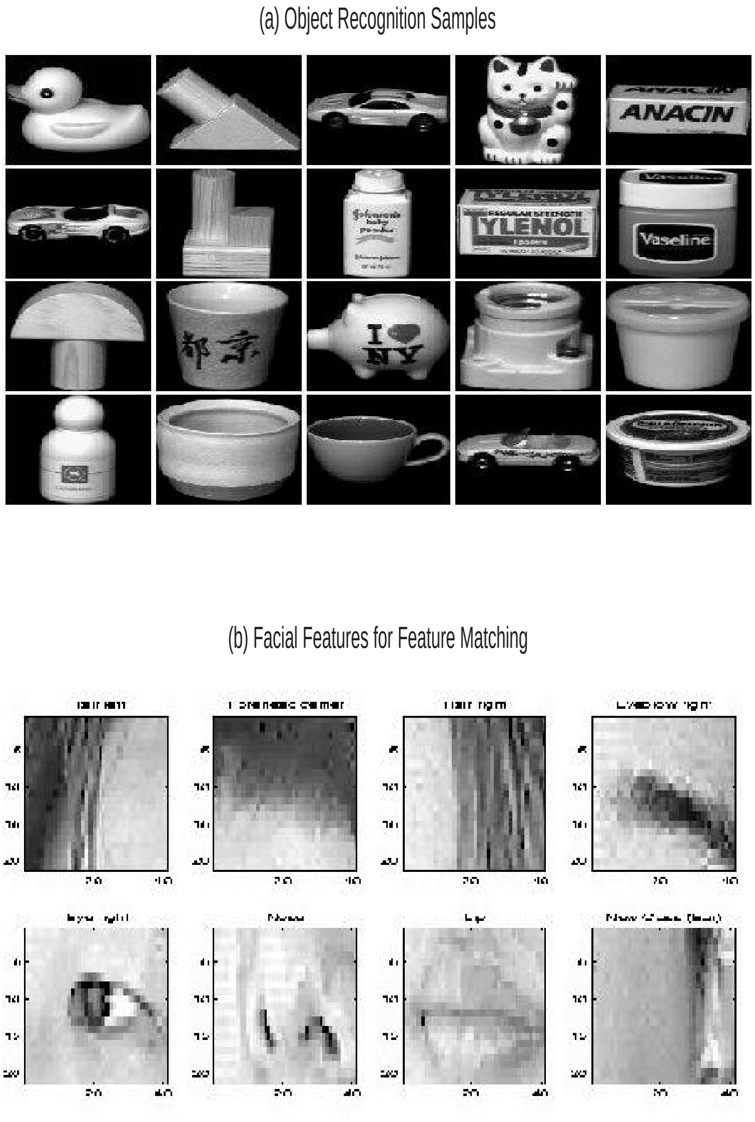


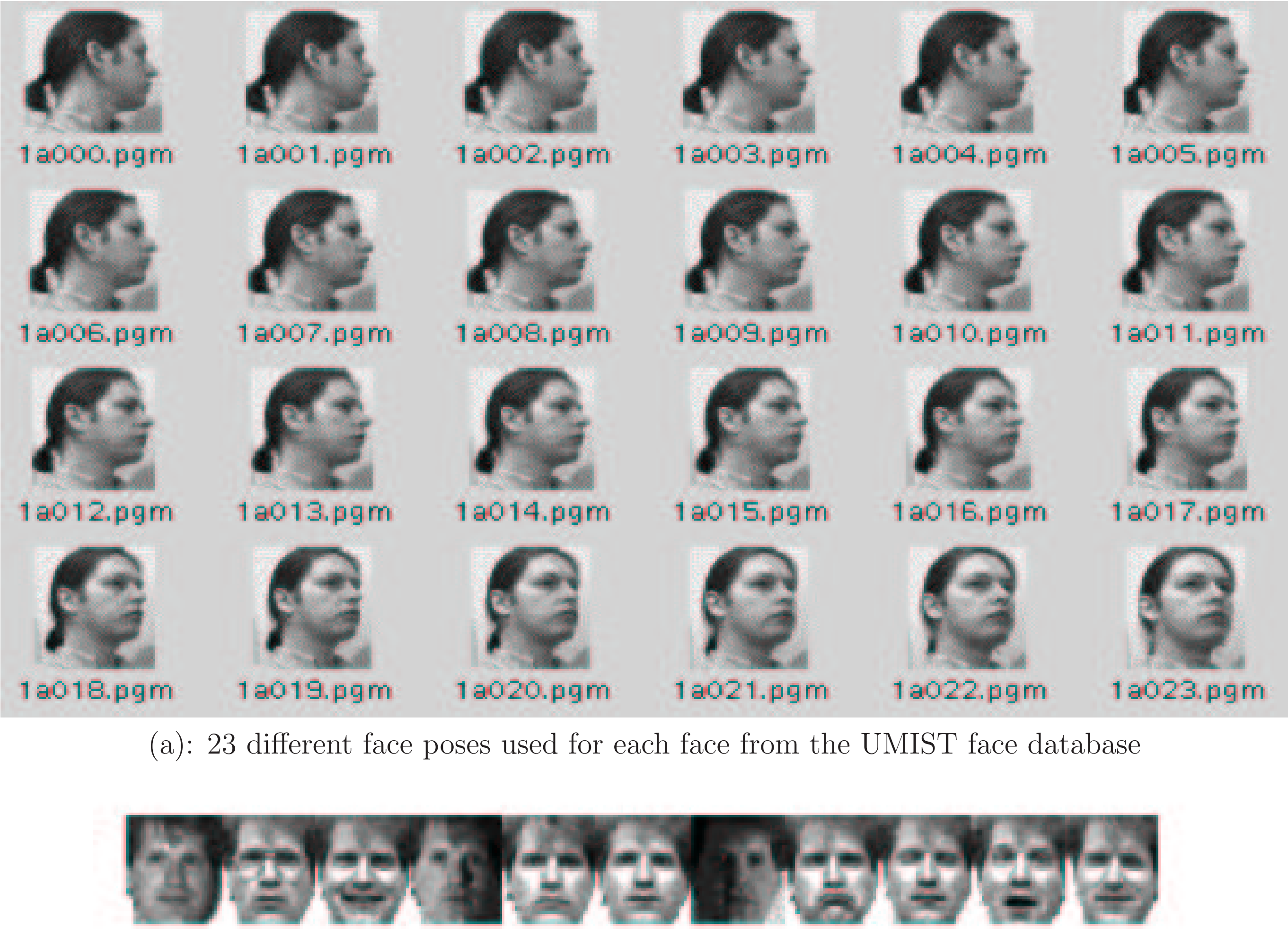
Figure 5.3: Error probability variation with reduced training data sizes per class



(a) Object Recognition Samples

(b) Facial Features for Feature Matching

Figure 5.4: (a) Object recognition classes (b) Facial Feature Matching classes



(a): 23 different face poses used for each face from the UMIST face database

(b): 11 facial expressions used for each face in the Yale face database

Figure 5.5: Face recognition databases

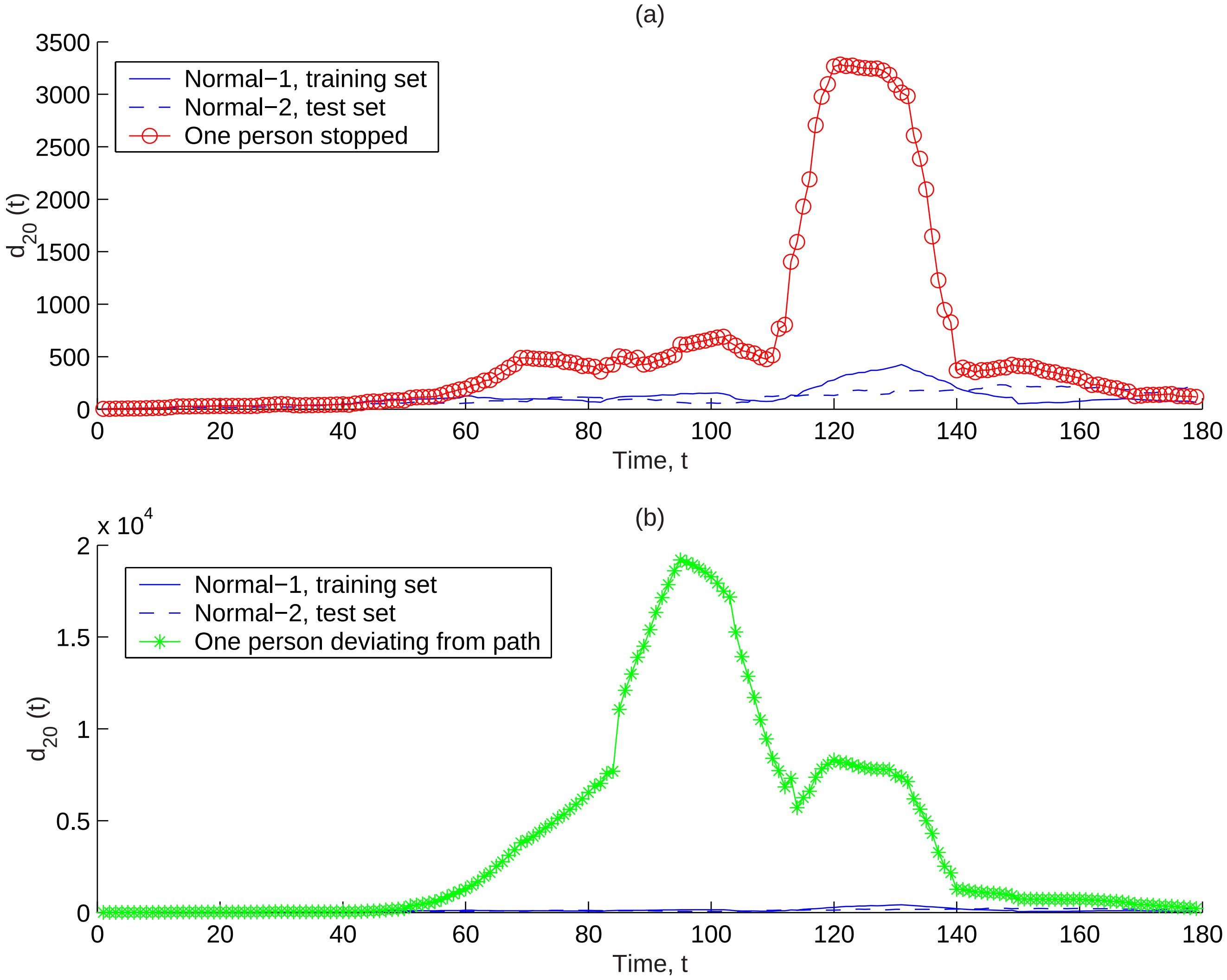


Figure 5.6: Detecting Abnormal Activities: The blue solid and dotted plots in both figures are the activity metric for a normal activity as a function of time. The red -o plot in (a) is for a temporal abnormality introduced at $t = 5$ and the green -* plot in (b) is for a spatial abnormality introduced at $t = 5$.

(a) No New, New Class Detection Disabled

Error Probabilities	PCA	SLDA	progressive-PCNSA	PCNSA
Misclassification = Total Error	0.125	0.053	0.020	0.046

(b) 17 New, $P(H_0) = 0.15, P(H_1) = 0.85$, New Class Detection Enabled

Error Probabilities	PCA	SLDA	progressive-PCNSA	PCNSA
Misclassification $\|H_0$ (R1)	0.0476	0.0048	0.0000	0.0429
False Alarm $\|H_0$ (R2)	0.0333	0.0143	0.0381	0.0286
Total Error $\|H_0$ $R3 = (R1 + R2)$	0.0810	0.0190	0.0381	0.0286
Miss $\|H_1$ (R4)	0.5496	0.6924	0.3731	0.3739
Total Error $R3 * P(H_0) + R4 * P(H_1)$	0.4793	0.5886	0.3229	0.3286

Table 5.1: Object Recognition results: (a) shows results for training and testing on data from 20 trained classes (no new). (b) shows results for training 3 object classes and using data from all 20 for testing. Correct new class detection probability is $1 - P(\text{Miss})$. For (a), a 50 dimensional PCA space was used while for (b) a 20 dimensional PCA space was used. In (a), 15 LDA directions were used and ANS dimension was varied between 4 and 9 for progressive-PCNSA while in (b) 2 LDA dimensions were used and ANS dimension varied between 3 and 6.

(a) No New, New Class Detection Disabled

Error Probabilities	PCA	SLDA	progressive-PCNSA
Misclassification = Total Error	0.0548	0.0369	0.0226

(b) 4 New, $P(H_0) = 0.43, P(H_1) = 0.57$, New Class Detection Enabled

Error Probabilities	PCA	SLDA	progressive-PCNSA
Misclassification $\|H_0$ (R1)	0.0028	0.0028	0.0000
False Alarm $\|H_0$ (R2)	0.1333	0.0528	0.0500
Total Error $\|H_0$ $R3 = (R1 + R2)$	0.1361	0.0556	0.0500
Miss $\|H_1$ (R4)	0.0500	0.6250	0.4083
Total Error $R3 * P(H_0) + R4 * P(H_1)$	0.0857	0.3810	0.2548

Table 5.2: Facial Feature Matching results: (a) shows results for training and testing on data from the 7 trained classes (no new). For (a) a 50 dimensional PCA space was used while for (b) a 20 dimensional PCA space was used. In (a), 6 LDA directions were used and ANS dimension was varied between 3 and 9 for progressive-PCNSA while in (b) 2 LDA dimensions were used and ANS dimension varied between 3 and 6.

(a) No New, New Class Detection Disabled

Error Probabilities	PCA	SLDA	progressive-PCNSA
Misclassification = Total Error	0.1182	0.0061	0.0030

(b) 12 New, $P(H_0) = 0.20, P(H_1) = 0.80$, New Class Detection Enabled

Error Probabilities	PCA	SLDA	progressive-PCNSA
Misclassification $\|H_0$ (R1)	0.0000	0.0000	0.0000
False Alarm $\|H_0$ (R2)	0.8030	0.0000	0.0303
Total Error $\|H_0$ $R3 = (R1 + R2)$	0.8030	0.0000	0.0303
Miss $\|H_1$ (R4)	0.1136	0.8636	0.1023
Total Error $R3 * P(H_0) + R4 * P(H_1)$	0.2515	0.6909	0.0879

Table 5.3: Face Recognition Results (UMIST database): (a) shows results for training and testing on data from 15 trained classes (no new). (b) shows results for training 3 classes and using data from all 15 for testing. The false alarm and miss probabilities of LDA in (b) are skewed because new class detection thresholds were kept constant at the same value for all the three applications (not optimized for each application separately). PCNSA new class detection is not sensitive to the type of application (numerical values of class variances) and hence performs equally good new class detection for all applications.

(a) No New, New Class Detection Disabled

Error Probabilities	PCA	SLDA	progressive-PCNSA
Misclassification = Total Error	0.0100	0.0000	0.0000

(b) 12 New, $P(H_0) = 0.20, P(H_1) = 0.80$, New Class Detection Enabled

Error Probabilities	PCA	SLDA	progressive-PCNSA
Misclassification $\|H_0$ (R1)	0.0000	0.0000	0.0000
False Alarm $\|H_0$ (R2)	0.0833	0.0000	0.0667
Total Error $\|H_0$ $R3 = (R1 + R2)$	0.0833	0.0000	0.0667
Miss $\|H_1$ (R4)	0.4083	0.5625	0.1708
Total Error $R3 * P(H_0) + R4 * P(H_1)$	0.3433	0.4500	0.1500

Table 5.4: Face Recognition Results (Yale database): (a) shows results for training and testing on data from 15 trained classes (no new). (b) shows results for training 3 classes and using data from all 15 for testing.

	bprowl-walk	broom1	broom3	crawl	jog1	sit1	sit2	walk1	walk2	walk3	walk-sad1
walk	1.43e-4	5.09e-4	4.58e-4	4.11e-4	2.36e-4	1.01e-3	2.08e-4	0.03e-6	1.60e-4	4.20e-5	2.20e-5
broom	5.65e-4	2.00e-6	2.90e-5	7.52e-4	4.36e-4	4.16e-3	1.95e-3	1.36e-4	2.05e-4	2.44e-4	3.31e-4
sit	9.66e-4	5.18e-4	3.53e-4	1.17e-3	6.33e-4	3.00e-6	4.70e-5	2.23e-4	3.69e-4	5.40e-4	3.67e-4

Table 5.5: Retrieving actions using PCNSA in tangent to shape space: The distances of the query sequences (top row) in ANS space of tangent to mean shape (equation (5.38)) of each of the 3 database sequences (in leftmost column) are shown. The underlined distance in each column corresponds to the closest match to query.

Chapter 6

Summary and Future Directions

6.1 Summary

In chapter 2, we have proposed statistics for slow and drastic change detection in stochastic state space models when change parameters are unknown. We use a PF to estimate the posterior probability distribution of the state at time t (X_t) given observations up to t ($Y_{1:t}$), $Pr(X_t \in dx|Y_{1:t}) \overset{\triangle}{=} \pi_t(dx)$. We propose here a statistic called ELL which is able to detect slow changes. ELL is the conditional Expectation of the negative Log-Likelihood of the state at time t ($[-\log p_t(X_t)]$), given past observations, $Y_{1:t}$. It is evaluated as the expectation under π_t of $[-\log p_t(X_t)]$.

Now, the PF is optimal for the unchanged system and hence when estimating π_t for the changed system, modeling error is present. Also the particle filtering error (error due to finite number of Monte Carlo samples or particles) is much larger. But using stability results from [5], we have shown that the approximation errors are stable (monotonically decreasing) with time and number of particles (in section 2.4). We have also shown in section 2.5, that the bound on the error is proportional to the rate of change. Thus for slow changes, the estimation error in π_t is small i.e. ELL is approximated correctly for such changes. Hence the approximate value of ELL detects the slow change as soon as it becomes "detectable" (defined in Definition 5 of section 2.3.2). ELL fails to detect drastic changes because of large estimation error in evaluating π_t. But large estimation error in evaluating π_t also corresponds to a large value of OL (or tracking error) which can be used for detecting such changes. We discuss this in Section 2.6. The application of ELL and OL (or tracking error) to abnormal activity detection and activity segmentation is discussed in chapter 4.

In chapter 3, we proposed stochastic state space models for the changing configuration of moving landmarks. We split the deformation/motion of the configuration into scaled Euclidean motion plus nonrigid shape deformations. In applications where the shape is stationary, the deformation model is simply a zero mean AR model in tangent plane to shape space at the mean shape. This idea is an extension of landmark shape analysis described in [9] for static hypothesis testing applications to modeling dynamics of a sequence of shapes. But for a non-stationary shape activity, the shape moves on the shape manifold. Dynamics is defined by a linear Gauss-Markov model on the shape "velocity" (time derivative of shape), which can be a random walk or an AR model depending on the problem. The "velocity" at a point on a manifold is defined in the tangent space to the manifold at that point. When the shape is not stationary but is only slowly varying, one can model the mean shape as being piecewise constant (instead of changing it at each time instant).

In this work, we treat an object as a point object or landmark and we model "activities" performed by groups of moving objects as moving and deforming landmark shapes. An HMM as described above is learnt for an activity using training data. In chapter 4, we have shown an application of the stationary shape activity model to tracking normal activities for which the mean shape is constant over time and to detect abnormal activity. Abnormal activity is defined as a slow or drastic change from normal shape dynamics with change parameters unknown. We have used a combination of the statistics defined in chapter 2 (ELL and tracking error) for abnormality detection. The nonstationary shape activity model is useful for detecting and also tracking abnormal behavior (it is a more flexible model which can track unmodeled shape changes as well). When using NSSA, more abnormalities get detecting using ELL i.e. get detected before loss of track. A long activity sequence can be segmented into stationary pieces using a piecewise stationary shape activity model and ELL to detect the segmentation boundaries.

The last part of the thesis is a linear subspace algorithm for pattern classification, which we call Principal Components' Null Space Analysis. PCNSA was motivated by PCA and it approximates the optimal Bayes classifier for Gaussian distributions with unequal covariance matrices. We have derived classification error probability expressions for PCNSA and compared its performance with that of subspace LDA both analytically and experimentally. Results have been shown for abnormal activity detection, human action retrieval and object and face recognition.

6.2 Future Directions

As part of future work, we intend to find practical examples of non-linear systems which satisfy the assumptions of theorems 1 and 2 in chapter 2. Also, we would like to study in more detail the implications of the Alpha function bound on ELL error. We are working on applying the CUSUM algorithm [15] to the case of unknown change parameters (as discussed in section 2.3.4) and evaluating its performance. We are also working on using ELL for neural signal processing where the goal is to detect how quickly an animal's brain responds to changes in stimuli provided to it. Also, we would like explore application of ELL to network or traffic congestion detection which also starts as a slow change.

When defining landmark configuration dynamics, we have currently dealt with the problem of varying number of landmarks in an adhoc fashion. As part of future work, we would like to study this problem in greater detail since it is a very important practical issue. To do this we would like to be able to deal with a time varying dimension of the shape manifold (treating the dimension as a Poisson process). Approaches in literature which embed a lower dimensional manifold as a boundary of the higher dimensional manifold will be explored.

Currently we have dealt with 2D shapes, the same approaches can be extended to 3D shapes. Also, the shape activity framework can be used in conjunction with a measurement model as a tracker to obtain new observations (discussed in section 4.2). Activity segmentation using PSSA and ELL to detect the segmentation boundaries has application in segmenting long video sequences containing multiple moving objects (for e.g. a traffic video) into stationary pieces. Our approach is sensor independent and observations can instead be obtained from acoustic, radar, infra-red or any other sensors and we would like to explore these applications in the future.

We would also like to improve performance of PCNSA for classification and make it more robust. Some ideas to do this are: (a) By evaluating the error probability expressions, one can choose between PCNSA and SLDA for a given application or even use different algorithms for different class pairs in a multi-class problem. (b) Perform a class-specific LDA i.e. find for each class, directions which not only minimize its variance but also maximize its distance from means of all other classes. (c) For classifying between a large number of classes, one can use ideas similar to [58] as discussed above in Section 5.6.4. (d) Kernel methods [61, 62] can be used to transform non-linearly separable data into higher dimensions where it becomes linearly separable and then PCNSA can be performed in kernel space. (e) A more systematic method than progressive-PCNSA (discussed in Section 5.6.2) can be developed for choosing valid ANS directions and new class detection thresholds. (f) Discriminant EM [76] can be used to increase the size of the training data set. We also hope to do a perturbation analysis for PCNSA similar to that done in [73] for LDA.

6.3 Contributions

Now in this last section, we briefly summarize the contributions of this thesis:

1. We have proposed the ELL statistic for slow change detection in general HMMs. We have extended PF stability theorems to prove stability (asymptotic stability under stronger assumptions) of errors in ELL approximation.

2. We have compared performance of ELL with OL for slow and drastic changes. Finally, we have also shown that the bound on ELL approximation error is an increasing function of "rate of change" with all increasing derivatives and discussed its implications.

3. Stochastic dynamical models for landmark shapes (random walk of shape velocity on the shape manifold), the stationary, nonstationary and piecewise stationary cases, are an extension of landmark shape analysis [9] to modeling dynamics.

4. The ideas of modeling a changing configuration of a group of moving point objects as a deforming shape is new. We have applied the landmark shape dynamical models to represent group activity and tracked it using a particle filter, defined abnormal activity as a change from normal shape dynamics and used ELL and tracking error for abnormality detection.

5. Also, the idea of using ELL along with PSSA model for activity segmentation and activity sequence identification and the idea of using the shape activity models for tracking to obtain observations is new.

6. The PCNSA classification algorithm, its classification probability analysis and its application to abnormal activity detection, action retrieval and face and object recognition.

Chapter 7

Appendix: Proofs of Chapter 2

Proof of Theorem 1.1:

- E_{x,Y_t} being a compact and proper subset of E_t (assumption (iv)) implies that there exists $M_t < \infty$, s.t. $[-\log p_t(x)] \leq M_t$ for all $x \in E_{x,Y_t}$. Because of the uniform compactness $M^* = \sup_t M_t < \infty$. Or in other words, $[-\log p_t(x)]$ is uniformly bounded by M^* for all t.

- First consider normal observations. Since assumptions (ii) and (iii) hold and since $[-\log p_t(x)] \leq M^*$ (bounded), we can apply the lemma 2.2 (for uniformly mixing kernels). Taking $\phi(x) = \frac{[-\log p_t(x)]}{M^*}$ [1], $\mu_t = \pi_t^0, \mu_t^N = \pi_t^{0,N}$, $\epsilon_k = \epsilon^0$, $\forall k < t_c$, $\epsilon_k =$

[1] Note that $\phi(x) \leq 1 \; \forall x \in E_{x,Y_t}$ and both posterior distributions μ_t, μ_t' are zero outside E_{x,Y_t}. Hence the inner product over E is equal to the inner product taken over the set E_{x,Y_t}.

$\epsilon^{c,0}$, $\forall k \geq t_c$, we get:

$$E_{Y_{1:t}}[\Xi_{pf}[|K(\pi_t^{0,N}:p_t) - K(\pi_t^0:p_t)|]] = M^* E_{Y_{1:t}}[\Xi_{pf}[|(\pi_t^{0,N} - \pi_t^0, \frac{[-\log p_t(x)]}{M^*})|]] \leq \frac{M^*\beta^*}{\sqrt{N}} \tag{7.1}$$

Taking $N \to \infty$, first equation of (2.16) follows.

- For changed observations [2],

$$|K_t^c - K_t^{c,0,N}| \;\; \leq \;\; |K_t^c - K_t^{c,0}| + |K_t^{c,0} - K_t^{c,0,N}| \tag{7.2}$$

  – Since (iii) holds, we can apply lemma 1 with $\epsilon = \min\{\epsilon^c, \epsilon^{c,0}\}$ and $\tau = \max\{\tau^c, \tau^{c,0}\}$. We take $\phi(x) = \frac{[-\log p_t(x)]}{M^*}$, $\mu_t = \pi_t^c, \mu_t' = \pi_t^{c,0}$, $R_k = R_k^c, \forall t_c \leq k \leq t_f$, $R_k = R_k^{c,0}, \forall k > t_f$, and consider $t \geq t_f + 3$. Then we get

$$|K_t^c - K_t^{c,0}| \leq M^*(\tau)^{(t-t_f-3)} \sum_{k=t_c}^{t_f} (\tau)^{(t_f-k)} \delta_k \leq 2M^*(t_f - t_c + 1)(\tau)^{(-t_f-3)}\tau^t \triangleq LM^*\tau^t \tag{7.3}$$

  The second inequality follows from inequality (2.12) and the fact that $\tau < 1$. For uniformly mixing kernels, ϵ and hence also τ are nonrandom (independent of $Y_{1:t}$) and so we can take $E_{Y_{1:t}}[.]$ in (7.3) and the RHS remains unchanged. Now taking $t \to \infty$, we get

$$\lim_{t\to\infty} E_{Y_{1:t}}[|K_t^c - K_t^{c,0}|] = 0 \tag{7.4}$$

  which means that given any error $\Delta > 0$, we can choose a t_Δ s.t. $\forall t \geq t_\Delta$, $E_{Y_{1:t}}[|K_t^c - K_t^{c,0}|] \leq \Delta/2$.

[2] For ease of notation, we denote $K(\pi_t^c : p_t)$ by K_t^c, $K(\pi_t^{c,0,N} : p_t)$ by $K_t^{c,0,N}$ and so on

- Now fix $t = t_\Delta$, and apply lemma 2.2 (for uniformly mixing kernels) to $|K_t^{c,0} - K_t^{c,0,N}|$ with $\mu_t = \pi_t^{c,0}, \mu_t^N = \pi_t^{c,0,N}$, $R_k = R_k^0, \forall k < t_c$, $R_k = R_k^{c,0}, \forall k \geq t_c$ and $\epsilon_k = \min\{\epsilon^0, \epsilon^{c,0}\}$. Then we get:

$$E_{Y_{1:t}}[\Xi_{pf}[|K_{t_\Delta}^{c,0} - K_{t_\Delta}^{c,0,N}|]] \leq \frac{M^* \beta^*}{\sqrt{N}} \tag{7.5}$$

  Taking $N \to \infty$, we get $\lim_{N\to\infty} E_{Y_{1:t}}[\Xi_{pf}[|K_{t_\Delta}^{c,0} - K_{t_\Delta}^{c,0,N}|]] = 0$.

Now since β^* is constant with time, the above convergence is uniform in t and so we can take $\lim_{t,N\to\infty}$ simultaneously. Thus taking $\lim_{t,N\to\infty} E_{Y_{1:t}}[\Xi_{pf}[.]]$ in (7.2), we get the result.

Proof of Theorem 1.2:

Since assumption (iv) (of Theorem 1.1) does not hold, $[-\log p_t(x)]$ is not bounded in this case. But we can approximate it by the increasing sequence of bounded functions $[-\log p_t^M(x)] = \min\{[-\log p_t(x)], M\}$. So we have $\lim_{M\to\infty}[-\log p_t^M(x)] = [-\log p_t(x)]$ pointwise in x.

- First consider normal observations.

$$|K_t^0 - K_t^{0,M,N}| \leq |K_t^0 - K_t^{0,M}| + |K_t^{0,M} - K_t^{0,M,N}| \tag{7.6}$$

  - Applying Monotone Convergence Theorem (MCT) [64](page 87), with $\mu = \pi_t^0$, $f_M(x) = [-\log p_t^M(x)]$ [3], we get

$$\lim_{M\to\infty} |K_t^0 - K_t^{0,M}| \ = \ \lim_{M\to\infty} |(\pi_t^0, [-\log p_t^M(x)]) - (\pi_t^0, [-\log p_t(x)])| = 0, \ a.s. \tag{7.7}$$

[3] Since p_t is a pdf, $\sup_x p_t(x) < \infty$. So it is easy to see that $C_t = \inf_x[-\log p_t^M(x)] > -\infty \ \forall M$, and hence we can apply MCT [64] in this case

Since the above result holds $a.s.$ over observation sequences, it also holds in mean [68], i.e.

$$\lim_{M\to\infty} E_{Y_{1:t}}[|K_t^0 - K_t^{0,M}|] \;=\; 0 \tag{7.8}$$

Now by assumption (iv)$'$, the above convergence is uniform in t. Thus given an error Δ, one can choose an M_Δ (independent of t) large enough s.t. $\forall M \geq M_\Delta$, $|K_t^{0,M} - K_t^0| < \Delta/2$.

- Now fixing $M = M_\Delta$, one can apply Theorem 1.1 (all assumptions required for it hold) with $M^* = M_\Delta$ and $p_t = p_t^{M^*}$ to get that $\lim_{N\to\infty} E_{Y_{1:t}}[\Xi_{pf}[|K_t^{0,M_\Delta} - K_t^{0,M_\Delta,N}|]] = 0$, uniformly in t.

Thus taking $\lim_{M\to\infty}(\lim_{N\to\infty} E_{Y_{1:t}}[\Xi_{pf}[.]])$ in (7.6), we get the result.

- For changed observations,

$$|K_t^c - K_t^{c,0,M,N}| \leq |K_t^c - K_t^{c,M}| + |K_t^{c,M} - K_t^{c,0,M,N}| \tag{7.9}$$

  - We can again apply MCT [64] to get $\lim_{M\to\infty} E_{Y_{1:t}}[|K_t^c - K_t^{c,M}|] = 0$ uniformly in t (by assumption (iv)$'$). Thus given an error Δ, one can choose an M_Δ (M_Δ is independent of t because of uniform convergence with t), s.t. $\forall M \geq M_\Delta$, $|K_t^{c,M} - K_t^c| < \Delta/3$.
  - Applying Theorem 1.1, with $M^* = M_\Delta$, and $p_t = p_t^{M_\Delta}$, we can show that $\lim_{t,N\to\infty} E_{Y_{1:t}}[\Xi_{pf}[|K_t^{c,M_\Delta} - K_t^{c,0,M_\Delta,N}|]] = 0$ [4].

[4] We can apply Theorem 1.1 here because M_Δ is independent of time

Thus taking $\lim_{M\to\infty}(\lim_{t,N\to\infty} E_{Y_{1:t}}[\Xi_{pf}[.]])$ in (7.9), we get the result.

Proof of Theorem 1.3:

By assumption (iv)$''$, we have a compact posterior state space, E_{x,Y_t}, which is a proper subset of E_t, and this implies that $[-\log p_t(x)] < M_t,\ \forall x \in E_{x,Y_t}$. Thus the total error can be split as (similar to proof of Theorem 1.1)

$$|K_t^c - K_t^{c,0,N}| = |K_t^c - K_t^{c,0}| + |K_t^{c,0} - K_t^{c,0,N}| \tag{7.10}$$

Now using (7.3) with $M^* = M_t$, we get

$$|K_t^c - K_t^{c,0}| \leq L M_t \tau^t \tag{7.11}$$

But by assumption (iv)$''$, the increase of M_t is atmost polynomial i.e. $M_t = bt^p$ for some finite p and b. It is simple to show that $M_t\tau^t$ goes to zero as t goes to infinity (apply L'Hospital's rule p times). This implies that $\lim_{t\to\infty}|K_t^c - K_t^{c,0}| = 0$.

By lemma 2.1,

$$\Xi_{pf}[|K_t^{c,0} - K_t^{c,0,N}|] \leq \frac{M_t\beta_t^{c,0}}{\sqrt{N}}. \tag{7.12}$$

Thus taking $\lim_{t\to\infty}(\lim_{N\to\infty}\Xi_{pf}[.])$ in 7.10, we get the result[5].

Proof of Theorem 2.1:

The proof is similar to that of theorem 1.2 but there are two differences. First, now the

[5] Note that because of M_t in RHS of (7.12), the convergence with N is not uniform in t. So we apply lemma 2.1 to get a.s. convergence (but it is not uniform with t

kernels $R_k^{c,0}$ are not uniformly mixing but only mixing. In this case we have for $t > t_f + 3$, $\theta_t^{c,0} = \tau_t^{c,0}\theta_{t-1}^{c,0}$. Thus $\theta_t^{c,0}$ is eventually strictly monotonically decreasing since $\tau_t^{c,0} < 1$ always. But the decrease is not exponential since $\tau_t^{c,0}$ is time varying and hence we cannot show convergence to zero of $\theta_t^{c,0}$. Also, now $\theta_t^{c,0}$ is a function of $Y_{1:t}$. Hence we need to take $E_{Y_{1:t}}[\theta_t^{c,0}]$. But since $\theta_t^{c,0}(Y_{1:t})$ is everywhere positive, it is trivial to show that $E_{Y_{1:t}}[\theta_t^{c,0}]$ is also eventually monotonically decreasing.

The second difference here is that since $R_k^{c,0}$ is not uniformly mixing, the convergence with N is not uniform in t.

Proof of Theorem 2.2:

Now we have a bounded posterior state space at each t, i.e. $[-\log p_t(x)] < M_t, \ \forall x \in E_{x,Y_t}$. Thus the total error can be split as

$$|K_t^c - K_t^{c,0,N}| = |K_t^c - K_t^{c,0}| + |K_t^{c,0} - K_t^{c,0,N}| \tag{7.13}$$

Applying lemma 1,

$$|K_t^c - K_t^{c,0}| \leq M_t \theta_t^{c,0} \tag{7.14}$$

Applying lemma 2.2 gives

$$\Xi_{pf}[|K_t^{c,0} - K_t^{c,0,N}|] \leq \frac{M_t \beta_t^{c,0}}{\sqrt{N}} \tag{7.15}$$

Taking $\lim_{N\to\infty} \Xi_{pf}[.]$ in (7.13), we get the result.

Proof of Theorem 2.3:

$$|K_t^c - K_t^{c,0,M,N}| \leq |K_t^c - K_t^{c,M}| + |K_t^{c,M} - K_t^{c,0,M}| + |K_t^{c,0,M} - K_t^{c,0,M,N}| \qquad (7.16)$$

- Applying MCT [64] once again, we can say that given an error $\Delta > 0$, there exists an $M_{t,\Delta}$ (since (iv)$'$ does not hold, it depends on t), s.t. $|K_t^c - K_t^{c,M_{t,\Delta}}| < \Delta$.
- Now since $M_{t,\Delta}$ is a function of t, we cannot apply Theorem 2.1. But for this value of M, applying lemma 1, we get $|K_t^{c,M_{t,\Delta}} - K_t^{c,0,M_{t,\Delta}}| < M_{t,\Delta}\theta_t^{c,0}$.
- Given Δ and $M_{t,\Delta}$, and applying lemma 2 we get, $\Xi_{pf}[|K_t^{c,0,M_{t,\Delta}} - K_t^{c,0,M_{t,\Delta},N}|] < \frac{M_{t,\Delta}\beta_t^{c,0}}{\sqrt{N}}$.

Combining the above three statements and taking $\lim_{N\to\infty} \Xi_{pf}[.]$ in (7.16), we get equation (2.21).

Proof of Lemma 3:

We need to show that $f(x,y) = \alpha_1(x, \alpha_2(y))$ is an Alpha function, given that $\alpha_1(x,z), \alpha_2(y)$ are Alpha functions of $[x,z]$ and y respectively. Consider the more general case, let

$$f([x,y]) = \sum_{j=1}^{m} \alpha_1^j(x, \alpha_2^j(y)) \qquad (7.17)$$

and show that f is an Alpha function, given that $\alpha_1^j, \alpha_2^j, j = 1, 2, ..m$ are Alpha functions of their arguments. We prove this as follows: We show the following two facts

1. $\nabla_{x,y} f(x,y)$ (gradient of f) is an increasing function and
2. $\nabla_{x,y} f(x,y)$ can also be written as a sum of compositions of Alpha functions i.e it has the same form as $f(x)$ defined in (7.17).

Because of statement 2, the statements 1 and 2 can now be applied on ∇f to show that ∇f is an increasing function and that $\nabla \nabla f$ can also be expressed as (7.17). This recursive process can be continued forever to show that all derivatives of f are increasing (or that f is an Alpha function).

Proof of statement 1: Now

$$\nabla_{x,y} f(x,y) = \sum_{j=1}^{m} [\alpha_{1x}^j(x, \alpha_2^j(y)) + \alpha_{1z}^j(x, \alpha_2^j(y))\alpha_{2y}^j(y)] \tag{7.18}$$

where α_{1x}^j is partial w.r.t x and so on. Now it is easy to see that both the terms above are increasing functions.

Proof of statement 2: From (7.18), it is easy to write ∇f as a sum of compositions of Alpha functions. Setting $\tilde{m} = 2m$ and $\tilde{\alpha}_1^{2j} = \alpha_{1x}^j(x,z)$, $\tilde{\alpha}_2^{2j} = \alpha_2^j(y)$, $\tilde{\alpha}_1^{2j+1} = \alpha_{1z}^j(x,z)\alpha_{2y}^j(y)$, $\tilde{\alpha}_2^{2j+1} = \alpha_2^j(y)$, we have expressed ∇f in exactly the same form as (7.17). We have used here the facts that derivative of an Alpha function is also an Alpha function (follows from the definition) and that the product of two Alpha functions is also an Alpha function (simple to prove using an argument exactly like the one used here).

Proof of Lemma 4:

For ease of notation, denote $\sup_x \psi_{k,Y_k}(x) \overset{\triangle}{=} S$. We first prove the following three inequalities below and then apply them to bound δ_k, ρ_k. Note that $R_{k,Y_k} = R_{k,Y_k^c}^c$ when applying lemma 1 (model error bound) but $R_{k,Y_k} = R_{k,Y_k^c}^0$ when using lemma 2 (PF error bound for incorrect model).

$$||R_{Y_k^c}^0(\pi_{k-1}^{c,0}) - R_{Y_k^c}^c(\pi_{k-1}^{c,0})||$$

$$\begin{aligned}
&\leq \int_x \int_{x'} |R^0_{Y^c_k}(x,x') - R^c_{Y^c_k}(x,x')|\pi^{c,0}_{k-1}(x)dx'dx \\
&\leq \sup_x \int_{x'} |R^0_{Y^c_k}(x,x') - R^c_{Y^c_k}(x,x')|dx' \\
&\triangleq D_R(R^0_{Y^c_k}, R^c_{Y^c_k}) = D_{Q,k} \qquad (7.19)
\end{aligned}$$

Also,

$$|A_k - R^0_{k,Y^c_k}(\pi^{c,0}_{k-1})(E)| = |R^c_{k,Y^c_k}(\pi^{c,0}_{k-1})(E) - R^0_{k,Y^c_k}(\pi^{c,0}_{k-1})(E)|$$

$$\begin{aligned}
&\leq \textstyle\int_{x'} | \int_x (R^0_{Y^c_k}(x,x') - R^c_{Y^c_k}(x,x'))\pi^{c,0}_{k-1}(x)dx|dx' \\
&= ||R^0_{Y^c_k}(\pi^{c,0}_{k-1}) - R^c_{Y^c_k}(\pi^{c,0}_{k-1})|| \overset{(a)}{\leq} D_{Q,k} \qquad (7.20)
\end{aligned}$$

Inequality (a) follows from of (7.19).

Next, we lower bound $A_k = C - (C - A_k)$:

$$\begin{aligned}
C - A_k = |C - A_k| &\leq ||R^c_{k,Y^c_k}(\pi^c_{k-1} - \pi^{c,0}_{k-1})|| \\
&\overset{(b)}{\leq} \frac{\lambda^c_{k,Y^c_k}(E)||\pi^c_{k-1} - \pi^{c,0}_{k-1}||}{\epsilon^c_k} \triangleq \frac{\tilde{D}_{k-1}}{\epsilon^c_k} \\
\text{Thus,}\ A_k &\geq C - \frac{\tilde{D}_{k-1}}{\epsilon^c_k} \qquad (7.21)
\end{aligned}$$

(b) follows from Lemma 3.5 of [5] and mixing property of R_k.

Now we use the above inequalities to bound δ_k:

$$\begin{aligned}
\delta_k &= \sup_{\phi:||\phi||_\infty \leq 1} |(\pi^{c,0}_k - \bar{R}^c_{Y^c_k}(\pi^{c,0}_{k-1}), \phi)| \\
&\leq ||\pi^{c,0}_k - \bar{R}^c_k \pi^{c,0}_{k-1}|| = ||\bar{R}^0_{Y^c_k}(\pi^{c,0}_{k-1}) - \bar{R}^c_{Y^c_k}(\pi^{c,0}_{k-1})|| \\
&\overset{(c)}{\leq} \frac{||R^0_{Y^c_k}(\pi^{c,0}_{k-1}) - R^c_{Y^c_k}(\pi^{c,0}_{k-1})|| + |A_k - R^0_{k,Y^c_k}(\pi^{c,0}_{k-1})(E)|}{A_k} \\
&\overset{(d)}{\leq} \frac{2D_{Q,k}}{A_k} \overset{(e)}{\leq} \frac{2D_{Q,k}}{C - \frac{\tilde{D}_{k-1}}{\epsilon^c_k}} \qquad (7.22)
\end{aligned}$$

Inequality (c) is an application of inequality (6) of [5] (given in (2.40)), (d) follows by combining (7.19) and (7.20) and (e) follows from (7.21).

Now consider ρ_k:

$$\begin{aligned}\rho_k &\overset{(f)}{\leq} \frac{S}{{\epsilon_k^{c,0}}^2 R^0_{k,Y_k^c}(\pi^{c,0}_{k-1})(E)}\\ &\overset{(g)}{\leq} \frac{S}{{\epsilon_k^{c,0}}^2 (A_k - D_{Q,k})} \overset{(h)}{\leq} \frac{S}{{\epsilon_k^{c,0}}^2 (C - \frac{\tilde{D}_{k-1}}{\epsilon_k^c} - D_{Q,k})}\end{aligned}$$

Inequality (f) follows from Remark 5.10 of [5] (given in (2.39)), (g) follows from (7.20) and assumption (2.26); (h) follows from (7.21) and assumption (2.26).

Also note that it is easy to see that $f(z) = a/(b - cz)$ and also $f(z) = az$ is an Alpha function. Thus the bound on δ_k is an Alpha function of $\tilde{D}_{k-1}$ and $D_{Q,k}$. The bound on ρ_k is an Alpha function of $\frac{1}{\epsilon}$ since $f(z) = z^2$ is an Alpha function; it is an Alpha function of $D_{Q,k}$ and $\tilde{D}_{k-1}$ since $f(z) = a/(b - cz)$ is an Alpha function.

BIBLIOGRAPHY

[1] W.E.L. Grimson, L. Lee, R. Romano, and C. Stauffer, "Using adaptive tracking to classify and monitor activities in a site," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Santa Barabara, CA, 1998, pp. 22–31.

[2] T. Huang, D. Koller, J. Malik, G. Ogasawara, B. Rao, S. Russell, and J. Weber, "Automatic symbolic traffic scene analysis using belief networks," in *American Association for Artificial Intelligence Conference*, 1994, pp. 966–972.

[3] J. Spletzer, A. Das, R. Fierro, C. Taylor, V. Humar, and J. Ostrowski, "Cooperative localization and control for multi-robot manipulation," in *Proceedings of the Conference on Intelligent Robots and Systems (IROS 2001)*, 2001.

[4] N.J. Gordon, D.J. Salmond, and A.F.M. Smith, "Novel approach to nonlinear/nongaussian bayesian state estimation," *IEE Proceedings-F (Radar and Signal Processing)*, pp. 140(2):107–113, 1993.

[5] LeGland F. and Oudjane N., "Stability and Uniform Approximation of Nonlinear Filters using the Hilbert Metric, and Application to Particle Filters," *Technical report, RR-*

4215, INRIA, 2002.

[6] D.F. Kerridge, "Inaccuracy and inference," *J. Royal Statist. Society, Ser. B*, vol. 23 1961.

[7] Rudolf Kulhavy, "A geometric approach to statistical estimation," in *IEEE Conference on Decision and Control (CDC)*, Dec. 1995.

[8] D.G. Kendall, D. Barden, T.K. Carne, and H. Le, *Shape and Shape Theory*, John Wiley and Sons, 1999.

[9] I.L. Dryden and K.V. Mardia, *Statistical Shape Analysis*, John Wiley and Sons, 1998.

[10] N. Vaswani, A. RoyChowdhury, and R. Chellappa, "Activity recognition using the dynamics of the configuration of interacting objects," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Madison, WI, June 2003.

[11] N. Vaswani, A. RoyChowdhury, and R. Chellappa, "Statistical shape theory for activity modeling," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2003.

[12] R. Chellappa, C. Wilson, and S. Sirohey, "Human and machine recognition of faces: A survey," in *Proc IEEE*, 1995, pp. 705–740.

[13] A. Samal and P. Iyengar, "Automatic recognition and analysis of human faces and facial expressions: A survey," in *Pattern Recognition*, 1992, pp. 65–77.

[14] A. Roy Chowdhury, R. Chellappa, and Trish Keaton, "A probabilistic correspondence algorithm using shape cues and prior information," .

[15] M. Basseville and I Nikiforov, *Detection of Abrupt Changes: Theory and Application*, Prentice Hall, 1993.

[16] C. Andrieu, A. Doucet, S.S. Singh, and V.B. Tadic, "Particle methods for change detection, system identification, and control," *Proceedings of the IEEE*, vol. 93, pp. 423– 438, March 2004.

[17] B. Azimi-Sadjadi and P.S. Krishnaprasad, "Change detection for nonlinear systems: A particle filtering approach," in *American Control Conference*, 2002.

[18] Shaohua Zhou and Rama Chellappa, "Probabilistic human recognition from video," in *European Conference on Computer Vision*, Copenhagen, Denmark, May 2002, pp. 681–697.

[19] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*, Academic Press, 1988.

[20] D. Ocone and E. Pardoux, "Asymptotic stability of the optimal filter with respect to its initial condition," *SIAM Journal of Control and Optimization*, pp. 226–243, 1996.

[21] Rami Atar and Ofer Zeitouni, "Lyapunov Exponents for Finite State Nonlinear Filtering," *SIAM Journal on Control and Optimization*, vol. 35, no. 1, pp. 36–55, 1997.

[22] F. LeGland and L. Mevel, “Exponential forgetting and geometric ergodicity in hidden markov models,” *Mathematics of Control, Signals and Systems*, pp. 63–93, 2000.

[23] R. Atar, “Exponential stability for nonlinear filtering of diffusion processes in a non-compact domain,” *Ann. Probab.*, pp. 1552–1574, 1998.

[24] A. Budhiraja and D. Ocone, “Exponential stability of discrete time filters for bounded observation noise,” *System and Control Letters*, pp. 185–193, 1997.

[25] P. DelMoral, “Non-linear filtering: Interacting particle solution,” *Markov Processes and Related Fields*, pp. 555–580, 1996.

[26] C.T. Zahn and R.Z. Roskies, “Fourier descriptors for plane closed curves,” *IEEE Transactions on Computers*, vol. C-21, pp. 269–281, 1972.

[27] David F. Rogers and J. Alan Adams, *Mathematical Elements for Computer Graphics*, WCB/McGraw-Hill, 1990.

[28] Fumin Zhang, Michael Goldgeier, and P. S. Krishnaprasad, “Control of small formations using shape coordinates,” in *International Conference on Robotics and Automation (ICRA)*, 2003.

[29] J.T. Kent, “The complex bingham distribution and shape analysis,” in *Journal of the Royal Statistical Society, Series B*, 1994, pp. 56:285–299.

[30] Y. Zhou, L. Gu, and H. Zhang, "Bayesian tangent space model: Estimating shape and pose parameters via bayesian inference," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Madison, WI, June 2003.

[31] C.G. Small, *The Statistical Theory of Shape*, Springer, New York, 1996.

[32] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham, "Training Models of Shape from Sets of Examples," in *British Machine Vision Conference*, 1992, pp. 9–18.

[33] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham, "Active shape models: Their training and application," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, January 1995.

[34] S. Soatto and A.J. Yezzi, "Deformotion: Deforming motion, shape average and the joint registration and segmentation of images," in *European Conference on Computer Vision*, Copenhagen, Denmark, May 2002, p. III: 32 ff.

[35] A. Srivastava, "Prior models for bayesian filtering in subspace tracking," in *First IEEE Sensor Array and Multichannel Signal Processing Workshop*, 2000.

[36] A. Chiuso and S. Soatto, "Monte-carlo filtering on lie groups," in *IEEE Conference on Decision and Control*, 2000.

[37] A. Srivastava, W. Mio, E. Klassen, and S. Joshi, "Geometric analysis of continuous planar shapes," .

[38] P. Thomas Fletcher, Conglin Lu, and Sarang Joshi, "Statistics of shape via principal geodesic analysis on lie groups," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003.

[39] S. Kurakake and R. Nevatia, "Description and tracking of moving articulated objects," in *International Conference on Pattern Recognition*, 1992, pp. I:491–495.

[40] T. Starner and A. Pentland, "Visual recognition of american sign language using hidden markov models," in *Proc. Intl. Workshop on Face and Gesture Recognition*, 1995.

[41] C. Bregler, "Learning and recognizing human dynamics in video sequences," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1997, pp. 568–574.

[42] A. Bobick and Y. Ivanov, "Action recognition using probabilistic parsing," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1998.

[43] A. Roy Chowdhury and R. Chellappa, "A factorization approach for event recognition," in *CVPR Event Mining Workshop*, Madison, WI, June 2003.

[44] L. Torresani and C. Bregler, "Space-time tracking," in *European Conference on Computer Vision*, Copenhagen, Denmark, May 2002.

[45] L. Zelnik-Manor and M. Irani, "Event based analysis of video," in *IEEE International Conference on Computer Vision*, 2001.

[46] V. Parmeswaran and R. Chellappa, "Action recognition based on view invariant spatio-temporal analysis," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Madison, WI, June 2003.

[47] D. Ponceleon T. Syeda-Mahmood, "Recognizing action events from multiple viewpoints," in *IEEE Workshop on Detection and Recognition of Events in Video*, 2001.

[48] A. Bissacco, A. Chiuso, Y. Ma, and S. Soatto, "Recognition of human gaits," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.

[49] A. Doucet, N. deFreitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*, Springer, 2001.

[50] J.P. MacCormick and A. Blake, "A probabilistic contour discriminant for object localisation," *IEEE International Conference on Computer Vision*, Mumbai, India, June 1998.

[51] H. Moon, R. Chellappa, and A. Rosenfeld, "3d object tracking using shape-encoded particle propagation," *IEEE International Conference on Computer Vision*, 2001.

[52] G. Qian and R. Chellappa, "Structure from motion using sequential monte carlo methods," *International Journal of Computer Vision*, pp. 5–31, August 2004.

[53] D. Schulz, W. Burgard, D. Fox, and A. Cremers, "Tracking multiple moving targets with a mobile robot using particle filters and statistical data association," in *Proc. of the IEEE International Conference on Robotics and Automation*, 2001.

[54] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol 3, no. 1 1991.

[55] P. Belhumeur, J. Hespanha, and D. Kreigman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, July 1997.

[56] W. Zhao, R. Chellappa, and P.J. Phillips, "Subspace linear discriminant analysis for face recognition," *IEEE Trans. on Image Processing*, 1999.

[57] D.L. Swets and J.J. Wengs, "Using discriminant eigenfeatures for image retrieval," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pp. 831–836, August 1996.

[58] R. Cappelli, D. Maio, and D. Maltoni, "Multispace KL for Pattern Representation and Classification," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pp. 977–996, September 2001.

[59] N. Vaswani, "A linear classifier for gaussian class conditional distributions with unequal covariance matrices," in *International Conference on Pattern Recognition*, 2002.

[60] H. Murase and S.K. Nayar, "Visual learning and recognition of 3-d objects from appearance," *International Journal of Computer Vision*, pp. 5–24, 1995.

[61] S. Li and Q. Fu et al, "Kernel machine based learning for multiview face detection and pose estimation," in *International Journal of Computer Vision*, July 2001, pp. 674–679.

[62] V. Roth and V. Stainhage, “Nonlinear discriminant analysis using kernel functions,” in *Neural Information Processing Systems 12, MIT Press*, 2000, pp. 568–574.

[63] J. Mao and A. Jain, “Artificial neural networks for feature extraction and mutlivariate data projection,” *IEEE Trans. on Neural Networks*, pp. 296–316, March 1995.

[64] H.L. Royden, *Real Analysis*, Prentice Hall, 1995.

[65] G. Casella and R. Berger, *Statistical Inference*, Duxbury Thomson Learning, second edition, 2002.

[66] T. Cover and J. Thomas, *Elements of Information Theory*, Wiley Series, 1991.

[67] P. Fearnhead, “Sequential monte carlo methods in filter theory,” in *PhD Thesis, Merton College, Universoty of Oxford*, 1998.

[68] A. Papoulis, *Probability, Random Variables and Stochastic Processes*, McGraw-Hill, Inc., 1991.

[69] Q Zheng and S. Der, “Moving target indication in lras3 sequences,” in *5th Annual Fedlab Symposium College Park MD*, 2001.

[70] T. Kailath, A.H. Sayed, and B. Hassibi, *Linear Estimation*, Prentice Hall, 2000.

[71] D. Crisan and A. Doucet, “A survey of convergence results on particle filtering for practitioners,” *IEEE Trans. Signal Processing*, vol. 50, no. 3, pp. 736–746, 2002.

[72] N. Vaswani, “Change detection in partially observed nonlinear dynamic systems with unknown change parameters,” in *American Control Conference (ACC)*, 2004.

[73] W. Zhao, R. Chellappa, and N. Nandhakumar, “Empirical performance analysis of linear discriminant classifiers,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1998.

[74] Alex M. Martinez and Avinash C. Kak, “Pca versus lda,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, February 2001.

[75] H. Vincent Poor, *An Introduction to Signal Detection and Estimation*, Springer, second edition.

[76] Y. Wu, Q. Tian, and T. Huang, “Discriminant-em algorithm with application to image retrieval,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2000.

[77] D. Graham and N. Allinson, “Characterizing virtual eigensignatures for general purpose face recognition,” in *Face Recognition: From Theory to Applications*. 1998, pp. 446–456, Springer.

[78] N. Vaswani, “Bound on errors in particle filtering with incorrect model assumptions and its implication for change detection,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2004.

[79] N. Vaswani and R. Chellappa, “Classification probability analysis of principal component null space analysis,” in *International Conference on Pattern Recognition*, 2004.

[80] N. Vaswani, A. RoyChowdhury, and R. Chellappa, ““Shape Activity”: A Continuous State HMM for Moving/Deforming Shapes with Application to Abnormal Activity Detection,” *Accepted to IEEE Trans. on Image Processing*, 2004.

[81] N. Vaswani and R. Chellappa, “Principal component null space analysis for image/video classification,” *Submitted to IEEE Trans. on Image Processing*, 2004.

[82] N. Vaswani, “Slow and drastic change detection in general hmms using particle filters with unknown change parameters,” *In preparation for IEEE Trans. on Signal Processing*, 2004.