Complexity Reduction for Near Real-Time High Dimensional Filtering and Estimation Applied to Biological Signals

Complexity Reduction for Near Real-Time High Dimensional Filtering and Estimation Applied to Biological Signals

A dissertation presented

by

Manish Gupta

to

The School of Engineering and Applied Sciences

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

in the subject of

Applied Mathematics

Harvard University

Cambridge, Massachusetts

May 2016

Dissertation advisors
Roger W. Brockett, Elizabeth B. Klerman

Author
Manish Gupta

Complexity Reduction for Near Real-Time High Dimensional Filtering and Estimation Applied to Biological Signals

Abstract

Real-time processing of physiological signals collected from wearable sensors that can be done with low computational power is a requirement for continuous health monitoring. Such processing involves identifying underlying physiological state x from a measured biomedical signal y, that are related stochastically: $y = f(x, \varepsilon)$ (here ε is a random variable). Often the state space of x is large, and the dimensionality of y is low: if $y \in R^N$ and $x \in S$ then $|S| >> N$, since the purpose is to infer a complex physiological state from minimal measurements. This makes real-time inference a challenging task. We present algorithms that address this problem by using lower dimensional approximations of the state. Our algorithms are based on two techniques often used for state dimensionality reduction: (a) *decomposition* of the form $x = x_1 \oplus x_2$ (variables can be grouped into smaller sets), and (b) *factorization* of the form $x = x_1 \otimes x_2$ (variables can be factored into smaller sets). The algorithms are computationally inexpensive, and permit online application. We demonstrate their use in dimensionality reduction by successfully solving two real complex problems in medicine and public safety.

Motivated originally by the problem of predicting cognitive fatigue state from EEG (**Chapter 1**), we developed the *Correlated Sparse Signal Recovery* (CSSR) algorithm and successfully applied it to the problem of elimination of blink artifacts in EEG from awake subjects (**Chapter 2**). Finding the decomposition $x = x_A \oplus x_{NA}$ into a low dimensional representation of the artifact signal x_A is a non-trivial problem and currently there are no online real-time methods accurately solve the problem for small N (dimensionality of y). By using a skew-Gaussian dictionary and a novel method to represent group statistical structure, CSSR is able to identify and remove blink artifacts even from few (e.g. 4-6) channels of EEG recordings in near real-time. The method uses a Bayesian framework. It results in more effective decomposition, as measured by spectral and entropy properties of the decomposed signals, compared to some state-of-the-art artifact subtraction and structured sparse recovery methods. CSSR is novel in *structured sparsity:* unlike existing group sparse methods (such as block sparse recovery) it does not rely on the assumption of a common sparsity profile. It is also a novel *EEG denoising* method: unlike state-of-the art artifact removal technique such as independent components analysis, it does not require manual intervention, long recordings

or high density (e.g. 32 or more channels) recordings. Potentially this method of denoising is of tremendous utility to the medical community since EEG artifact removal is usually done manually, which is a lengthy tedious process requiring trained technicians and often making entire epochs of data unuseable. Identification of the artifact in itself can be used to determine some physiological state relevant from the artifact properties (for example, blink duration and frequency can be used as a marker of fatigue). A potential application of CSSR is to determine if structurally decomposed cortical EEG (i.e. *non-spectral*) representation can instead be used for fatigue prediction.

A new E-M based active learning algorithm for ensemble classification is presented in **Chapter 3** and applied to the problem of detection of artifactual epochs based upon several criteria including the sparse features obtained from CSSR. The algorithm offers higher accuracy than existing ensemble methods for unsupervised learning such as similarity- and graph-based ensemble clustering, as well as higher accuracy and lower computational complexity than several active learning methods such as Query-by-Committee and Importance-Weighted Active Learning when tested on data comprising of noisy Gaussian mixtures. In one case we were to successfully identify artifacts with approximately 98% accuracy based upon 31-dimensional data from 700,000 epochs in a matter of seconds on a personal laptop using less than 10% active labels. This is to be compared to a maximum of 94% from other methods. As far as we know, the area of active learning for ensemble-based classification has not been previously applied to biomedical signal classification including artifact detection; it can also be applied to other medical areas, including classification of polysomnographic signals into sleep stages.

Algorithms based upon state-space factorization in the case where there is unidirectional dependence amongst the dynamics groups of variables (the "Cascade Markov Model") are presented in **Chapters 4**. An algorithm for estimation of factored state of the form $x = x_A \otimes x_B$ where dynamics follow a Markov model, from observations $y = Hx + \varepsilon$, is developed using E-M (i.e. a version of Baum-Welch algorithm on factored state spaces) and applied to *real-time* human gait and fall detection. The application of factored HMMs to gait and fall detection is novel; falls in the elderly are a major safety issue. Results from the algorithm show higher fall detection accuracy (95%) than that achieved with PCA based estimation (70%). In this chapter, a new algorithm for optimal control on factored Markov decision processes is derived. The algorithm, in the form of decoupled matrix differential equations, both is (i) computationally efficient requiring solution of a one-point instead of

two-point boundary value problem and (ii) obviates the "curse of dimensionality" inherent in HJB equations thereby facilitating real-time solution. The algorithm may have application to medicine, such as finding optimal schedules of light exposure for correction of circadian misalignment and optimal schedules for drug intervention in patients.

The thesis demonstrates development of new methods for complexity reduction in high dimensional systems and that their application solves some problems in medicine and public safety more efficiently than state-of-the-art methods.

Contents

Previously Published Work and Patents

Large portions of Chapters 2 and 3 have appeared in the following publication:

"On-line EEG Denoising Using Correlated Sparse Recovery", Gupta, M., Beckett S.A., Klerman, E.B, Proceedings of the IEEE 2016 10th International Symposium on Medical Information and Communication Technology (ISMICT)

Some of the work outlined in Chapters 5 and 6 resulted in the following patents:

"METHOD AND APPARATUS FOR DETECTING MODE OF MOTION WITH PRINCIPAL COMPONENT ANALYSIS AND HIDDEN MARKOV MODEL", Saeed S. Ghassamzadeh, Lusheng Ji, Robert Raymond Miller, II, Manish Gupta, Vahid Tarokh, U.S. Patent Number 20150161516, Issued 06/2015.

"METHOD AND APPARATUS FOR DETECTING DISEASE REGRESSION THROUGH NETWORK-BASED GAIT ANALYSIS", Saeed S. Ghassamzadeh, Lusheng Ji, Robert Raymond Miller, II, Manish Gupta, Vahid Tarokh, U.S. Patent Number 20150157274, Issued 06/2015.

"METHOD AND APPARATUS FOR USING GAIT ANALYSIS TO DETERMINE A HEALTH QUALITY MEASURE", Saeed S. Ghassamzadeh, Lusheng Ji, Robert Raymond Miller, II, Manish Gupta, Vahid Tarokh, U.S. Patent Number 20150161511, Issued 06/2015.

"METHOD, COMPUTER-READABLE STORAGE DEVICE AND APPARATUS FOR PROVIDING AMBIENT AUGMENTED REMOTE MONITORING", Saeed S. Ghassamzadeh, Lusheng Ji, Robert Raymond Miller, II, Manish Gupta, Vahid Tarokh, U.S. Patent Number 20150157279, Issued 06/2015.

Acknowledgments

I have been very privileged to have Prof. Roger W. Brockett and Prof. Elizabeth B. Klerman as my co-advisors, whose intellect and day-to-day guidance has been extremely valuable all along. I am also very grateful to Prof. Vahid Tarokh whose ideas have been the motivation for several algorithms presented in this dissertation. I am indebted to the Division of Sleep and Circadian Disorders at the Brigham and Women's Hosptial (BWH) and the Division of Sleep Medicine at Harvard Medical School for providing me with resources, data and support that led to the work in the Chapters 1-4 of this dissertation, and to the AT&T Research team (Dr. Saeed S. Ghassamzadeh, and Dr. Lusheng Ji), collaboration with whom resulted in the work of Chapters 5-6. The expertise provided by the registered polysomnographic technicians at BWH, in particular Scott A. Beckett, Brandon J. Lockyer and Daniel R. Mobley, has been extremely valuable in not only understanding of the EEG collection process but also validation of some of the results of Chapter 3 and 4. I am also grateful for the numerous technical discussions with several faculty members and post-docs at BWH, Harvard and M.I.T., in particular Dr. Andrew J. Phillips, Dr. Melissa St. Hillaire, Dr. Ali Belabbas, Dr. James P. Butler, Dr. Samuel Patz, Dr. S.C. Samuel Kuo, Dr. Demba Ba and Dr. Emery Brown. Finally, none of this work would have been possible without the care and support of staff members Kathleen A. Masse, John Girash and Julie Holbrook at Harvard and Jennifer L. Opp at BWH.

Chapter 1

A Motivating Problem - Can EEG Predict PVT ?

Cognitive fatigue, defined as a "state of reduced mental alertness that impairs performance" [80], is a major cause of road accidents [72, 137]. The National Highway Traffic Safety Administration (NHTSA) reports that each year driver fatigue results in about 1550 deaths, 71,000 injuries, and $12.5 billion in monetary losses [149, 186] and that 30 million drivers nod off or fall asleep while driving every year resulting in an average of about one crashes every two minutes nationwide [190]. According to the U.S. National Sleep Foundation, 54% of adult drivers report having driven while drowsy and about 28% of them have actually fallen asleep at the wheel [50]. In 2010, the American Automobile Association estimated that 1 out of 6 traffic accidents resulting in death and 1 out of 8 resulting in serious injury were due to drowsy driving [211]. European studies indicate similar statistics: 25-30% of driving accidents in the UK are drowsiness related [87], about 35% drivers in the Netherlands and 70% drivers in Spain have reported falling asleep while driving [165]. Similar catastrophic occurrences have been reported in other areas of transportation. For example, pilot operation under fatigue has been determined causal in several major aircraft crashes (e.g., Colgan Air Flight 3407, 2010) [206]. The National Sleep Foundation estimates that one out of five air pilots, one out of six train operators and one out of seven truck drivers report "near misses" due to sleepiness [89]. The problem of fatigue is equally disastrous in other professions: 1 in 5 physicians report making fatigue related mistakes leading to serious patient injury and 1 in 20 report a patient death [19]. These alarming facts have been the motivation behind several automated fatigue monitoring systems over the past 20 years including some commercial sys-

tems, several of which are reviewed in Chapter 2. However, according to the authors of an assessment of driver fatigue management technology [16], an "ideal" system should be able to, on an *individual* basis not only *monitor* alertness in *real-time* but also *predict* fatigue based upon factors causing it, and few systems satisfy both these requirements.

Fatigue is often the result of acute sleep deprivation (i.e., single extended wake episodes), chronic sleep restriction (i.e., multiple days with insufficient sleep), and/or adverse circadian phase (i.e., being awake during biological night time, which is times when the endogenous circadian system is promoting sleep) [55, 59, 205, 93, 232, 106]. The nonlinear interactions of acute sleep deprivation, chronic sleep restriction and circadian phase causes reduced subjective and objective alertness and vigilance, suboptimal neurobehavioral performance[37, 7], memory decrements[99], reduced situation awareness [164], episodes of automatic behavior[56], and reduction in accuracy and correctness in decision-making [79] in humans. In fact, even 24 hours of sleep deprivation results in vigilant attention performance comparable to that of operating under influence of 0.1% blood alcohol [62], a blood alcohol level that is illegal for operation and driving in the US. Peak drowsy driving accidents are observed during specific circadian phases (e.g., 2:00-6:00am (biological night) and 2:00pm-4:00pm (usual nap time)) [192]. 24-30 hours of continuous wake is associated with a drop in clinical performance of physicians from the 50th to the 7th percentile [59, 178] and an increase in medical errors by 36% [54, 58, 154, 153]. The National Transportation Safety Board (NTSB) concluded that the critical factors in predicting crashes due to fatigue are the duration of the most recent sleep period and the amount of sleep in the previous 24h [160, 50]. However, often the extent of impairment is underestimated by the sleep- deprived or sleep-restricted individual [219],[220, 221]. This necessitates objective assessment and prediction of neurobehavioral performance for individuals with work schedules that include extended wake duration and working during the biological nighttime, such as truck drivers, plant operators, air traffic controllers, medical residents, security officers and airline personnel [119]. While several mathematical models of neurobehavioral performance under sleep deprivation and sleep restriction have been developed [114, 88, 244, 130] and used [159, 64, 185] (reviewed in [102],[127] and [75]) by institutions such as NASA, the focus of many models is to predict group mean performance, rather than an individual's performance even though the need for capturing interindividual differences in models of fatigue assessment and performance prediction is well established [75, 73, 225].

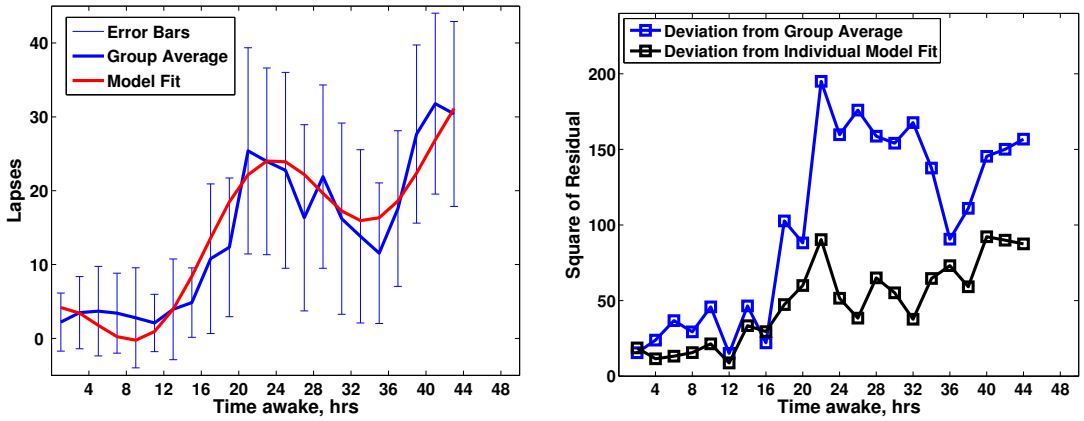
Differences amongst individuals are not only substantial but also consistent [141]

which is indicative that individual factors other than situational are at play in determining fatigue response. Using an intra-class correlation coefficient (ICC) metric, it was demonstrated that under sleep deprivation conditions, interindividual differences accounted for over 50% of the total variance in a PVT (Psychomotor Vigilance Task) measure [223]. Such differences may arise from genetic polymorphisms [99]. Of note, these trait-like differences are dependent on the measure considered [222, 141]. Our own analysis on several data sets from multiple sleep deprivation studies shows (Figs 1.1 (a)) substantial and consistent inter-individual differences on both PVT lapse rate, the rate at which a subject fails to respond within 500ms of presentation of stimulus, as well as median PVT reaction time (Fig 1.2) , and that these differences are not captured by mathematical models which work well at a group level (Fig 1.3).

Thus, in order to perform individual prediction of fatigue using a physiology-inspired model, fitting parameters to that model on a group or even individual basis is inadequate. In addition to the impact of acute sleep deprivation, chronic sleep restriciton and adverse circadian phase as modeled by known models, there are additional causal factors that contribute to departure from this model on an individual basis. Considering the value of being able to accurately predict individual fatigue, we started out by asking this question: *Can we develop models that incorporate individual physiological data such as EEG that will improve upon individualized predictions of fatigue (in particular, PVT lapses) than what can already be predicted using existing mathematical models ?* The inspiration for this hypothesis was from the fact that EEG has been shown to be a definitive marker of individual sleepiness and alertness, so perhaps there is enough information in the signals that can model additional causal factors of fatigue per individual. Tools and methods to be able to answer this question was the motivation behind a significant portion of the work outlined in this thesis. First we formalize the problem mathematically.

1.1 Problem Statement

Let us denote by $y(t)$ a outcome of the psychomotor vigilance test (PVT) at time t, where t is measured from some reference base time in the past. For the purposes of our study, we will assume $y(t)$ is a measure of a the response times during a PVT test. Some such measures include median reaction time (RT), number & percentage of lapses (where RT $>$ 500ms), 5 and 10% slowest RT, 5 and 10% fastest RT, mean RT, standard deviation of RT, maximum RT, and number of anticipations (where RT $<$ 0).-In this report we will be



(a) (b)

Figure 1.1: **Group-averaged performance metrics have high per-participant variability**. (A) Average and variability of number of lapses, averaged across 33 participants, over 44 hours of wake duration (red line), error bars (blue) and least-squares fit using the a mathematical (the *two-process*) model (black line). (B) Mean of residuals, computed per participant, from the group average of number of lapses compared with residuals from the model fit per participant, as a function of time awake. The difference suggests that the population model is a poor for for individuals. The data of these results were collected in a constant-routine protocol.

focussing on number of lapses, so a precise definition of $y(t)$ is : $y(t) \in \mathbf{N}$ denotes the number of lapses during a standard PVT conducted at time t (in hours), where a standard PVT is one that lasts for a specified number minutes (usually two) with a fixed number of visual stimuli.

We will also denote by $u(t) \in \{0, 1\}$ the state of an individual - sleep or awake -.at time t, which is a given input. Here 0 represents a "sleep state" and 1 represents a "wake state". This is a gross simplification since it is often not easy to make a clear distinction between these states (for example, a person may be awake but may experience a "microsleep", or a person may be awake momentarily at several times while sleeping, in which case a precise knowledge of $u(t)$ may not be available). For for purposes of this study we will assume that such an input is available to us.

We expect that there is a relationship between y and u and that this relationship is individual specific.This can be modeled as an input-output dynamical system $\mathbf{M}$, which will be individual specific with some state variable x - representing the physiological and

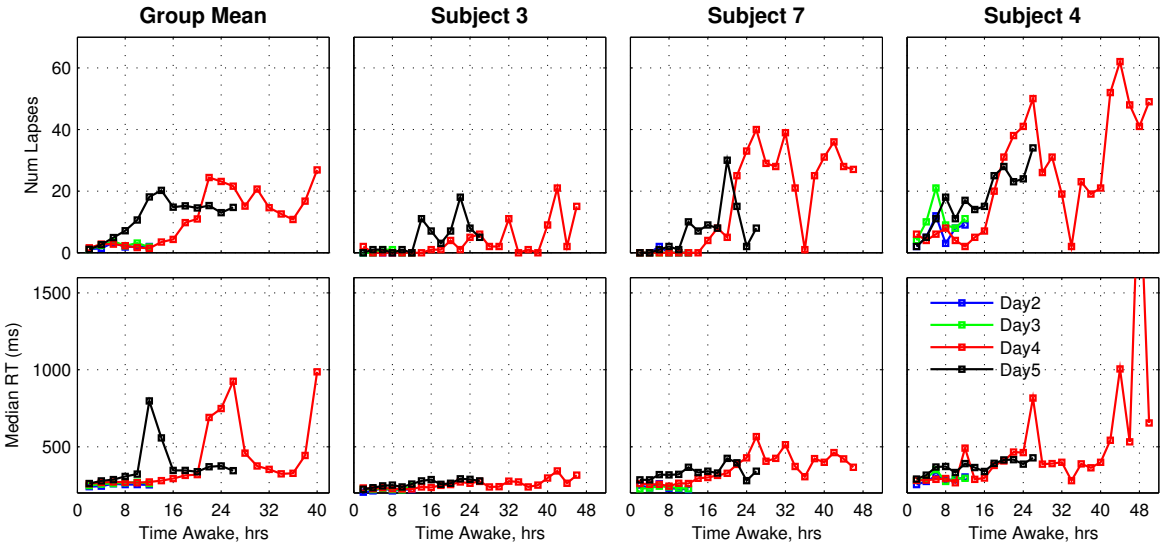


Figure 1.2: **Inter-individual differences are consistent.** Lapses (top panels) and median reaction time (RT) (bottom panels) for three participants on four different days (Days 2-5) of a constant-routine protocol [25, 99]. From left to right panels: Group Mean in the left panels, and data from three participants. The variability amongst individuals is consistent across days and performance metrics.

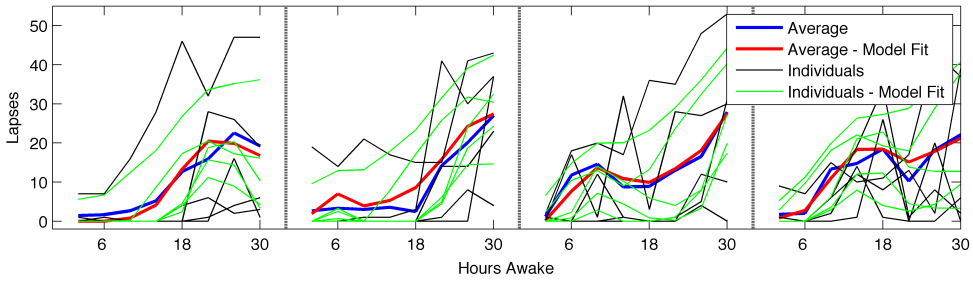


Figure 1.3: **Mathematical models fit average data but not individual data**. Number of PVT lapses for four wake periods in participants over 32 hours of wake duration per wake period. Individual data (black) are a poor fit (green) to the mathematical model. But when data are averaged over all nine participants (blue) the best fit using the mathematical model (red) shows a good approximation. The mathematical model used here is the *two-process* model (1.1) and data are fitted using least-squares. The data of these results were collected in a forced desynchrony protocol [25, 99].

environmental state. For our purposes we restrict our interest to the class of input functions $u(t)$ of the form $H(t - T_0)$ where T_0 is a fixed given value, and H represents the Heaviside function, and we are only interested in the performance output $y(t)$ for $t > T_0$ in response to this class of functions. (This means we are restricting our problem to the case of a continuous

wake episode from a fixed waking time, and that there is no chronic sleep debt or circadian shift). We can assume without loss of generality that $T_0 = 0$ (note, we are not claiming that the system is time-invariant, we are simply choosing our reference base time in such a way that $T_0 = 0$). Let the response $y(t)$ to input $H(t)$ given an initial state $x(t) = x_0$for an individual a be denoted by $p(t, x_0, a)$. *We would like to know $p(t, x_0, a)$ for every t given an initial state x_0 of an individual a. Since that is our only goal, we can completely ignore the dynamics and instead focus only on the function $p(t, x_0, a)$.*

In reality the state x_0 includes many factors which can not be measured and instead what is intended is to represent the unmeasurables probabilistically. Denoting by s_0 the initial values of a set of measurable state variables, our goal is to find a function $p(t, s_0, a, \mu)$ where μ is a random variable. Of course there are many such functions. We want the one that has the most predictive power in the sense that it minimizes the expected value of some function of the difference between the observed performance and that predicted by the model.

The determination of p is to be based on data collected from subjects. We can use a parametric form of p that is widely used in sleep and circadian research (extensions of the two-process model described below) and use the data to fix the parameters in some least squares way. Calling together all parameters (such as those specific to the individual, those determined by the initial state s_0 and those corresponding to the probability model used for μ) as θ, we write this parametrized performance function as the stochastic process $p_\theta(t)$, with pdf $\phi_\theta(t, \cdot)$.

We wish to address the following problems:

A) Given T and performance measurements $\mathbf{y}_T = \{y(t) : t = 1, 2, 3...T\}$, find an estimate $\widehat{\theta}$ that minimizes a certain loss function $l(\mathbf{y}_T, p_\theta)$. Then we call $\widehat{y}(T+k) = \mathbf{E}_{\phi_{\widehat{\theta}}(t)}(p_{\widehat{\theta}}(T+k))$,where $\mathbf{E}_\phi(.)$ is the expectation operator with respect to a distribution ϕ, as the $k-$step prediction of $y(T+k)$. We call the prediction error $E(T, k) = \widehat{y}(T+k) - y(T+k)$. We expect that $E(T, k)$ decreases with T. We want to characterize functional behavior of $E(T, k)$ with T based upon experimental data.

B) Suppose we are given additional EEG measurements $\mathbf{z}_T = \{z(t) : t = 1, 2, 3...T\}$. Then we seek to find another parameterized performance function $g_\upsilon(t)$ as above, and we ask if in doing so can we find an estimate $\widehat{\upsilon}$ of υ such that it gives us better prediction of $y(t)$ given $\mathbf{y}_T, \mathbf{z}_T$. Specifically, if EEG is indeed helpful, we would expect prediction error to decrease more rapidly with T than in (A). *In our analysis we restricted z to be the power spectral density of EEG.*

As shown in the next few sections, our conclusion to (B) was in the negative. That is, our parameterized performance function $g_v(t)$ did not improve prediction beyond $f_\theta(t)$. This lead us to look into perhaps if experimental data quality led to these results, which motivated the work in Chapter 2 and 3 of this thesis. Toward the end of this discussion, we will address the question (B) posed above from a causality point of view.

1.2 Problem (A): Baseline PVT Prediction

According to the *Two-Process model* [28, 204] two separate processes, process "S" or sleep homeostasis, and process "C", the circadian system that regulate multiple physiological functions including the timing of sleep in mammals. If k is the time in hours since the last wake, then according ot this model the performance metric is given by[1]

$$f(k,\theta) = \alpha(1 - \kappa e^{-\rho T_s(k-1)}) + \beta \sum_{i=1}^{5} a_i \sin(i\frac{2\pi T_s}{\tau_c}(k-1) + \phi) \tag{1.1}$$

where constants $T_s = 24, T_c = 2, a_1 = 0.9700,\ a_2 = 0.2200,\ a_3 = 0.0700, a_4 = 0.0300, a_5 = 0.0010$, and $\boldsymbol{\theta} = \{\alpha, \kappa, \rho, \beta, \phi\}$ are positive and real valued parameters[2].

Considering how well the physiologically motivated simple *two-process* model (1.1) fits group level data, we chose as to model our performance process $p_{\boldsymbol{\theta}}(t)$ as

$$p_{\boldsymbol{\theta}}(k) = f(k, \boldsymbol{\theta}) + \varepsilon_k \tag{1.2}$$

where ε_k is a standard Gaussian noise, and use a batch non-linear least squares method to estimate θ (described below). Performance prediction using a few baseline measurements is a promising prospective, the feasibility of which has been demonstrated in the context of diffusion modeling [177] and neuroimaging studies [47]. Individual prediction based upon prior performance measurements using the Two-Process model (1.1) has been attempted in [224] and using AR modeling in [183] and [184]. The fully Bayesian approach used in [224] is not computationally conducive to a real-time adaptive approach, and that of [183] and [184] requires too many baseline data points (e.g., at least 24h of wake data) before

[1] Process "C" is periodic but not exactly a sinusoid, and modeled using first five Fourier components of the periodic signal. Process "S" or homeostatic pressure is postulated to reflect brain adenosine concentration levels which typically follows multiplicative dynamics and hence the exponential form.

[2] α and β govern the basal level of performance and relative contribution of each process to performance, ρ is a rate constant of the sleep homeostat during wake, κ is the initial homeostat, and ϕ is the circadian phase angle (timing of the circadian process relative to the clock wake time). Parameters κ, ρ,β are individual specific (i.e., fixed for an individual) and α, ϕ depend on the initial state of the individual.

any reasonable inference can be made. Thus we employed a simpler non-linear least squares method for performance time series prediction. We show that a simplified approach is at least as accurate as those presented in [224], [183] and [184] and requires fewer measurements than what is required in [183] and [184]. Using our model, the behavior of the mean square error of performance predictions with the number of past measurements and prediction horizon on 44h of continuous wake data collected from 33 participants was characterized.

1.2.1 Data Collection

33 healthy young adults, free from medical, psychiatric and sleep disorders as determined by history, screening and physical examination participated in a Constant Routine (CR) protocol that lasted up to 44 hours [78]. The CR includes constant wake and posture in dim light and lasted 44h. Starting 2 hours after awake, the participants were given a ten minute Psychomotor Vigilance Task (PVT) every two hours, and their response times were recorded. Within each 10-minute PVT trial, approximately 100 stimuli are presented with a random latency between 2-11 seconds; the speed of reaction time (RT) to push a button is recorded and the number of lapses (defined as when RT $>$ 500ms) were determined.

1.2.2 Prediction Algorithms

Let us denote by y_t^s the number of lapses in a performance test (PVT) at time t, where t is measured from some reference base time in the past, for a participant labelled s. We denote by $\mathbf{Y}_T^s = \{y_1^s, y_2^s, ..., y_T^s\}$ (we will drop the superscript s for participant for clarity of notation if there is no ambiguity). Given a T and a $k > 0$, we write an estimate of y_{N+k} given $\mathbf{Y}_N$ is written as $\widehat{y}_{N+k}|\mathbf{Y}_N$ or, more succinctly, as $\widehat{y}_{N+k|N}$. We will call this the the $k-$*step prediction after* N *observations.* As stated in the problem statement, based upon experimental data, we want to characterize the prediction error $E(T,k) = \widehat{y}(T+k) - y(T+k)$, in particular its behavior with T and k.

(i) Batch Nonlinear Regression (BNLR) Algorithm Here we assume, in (1.2) $\varepsilon_k \sim N(0, \sigma^2)$ and a prior model $\boldsymbol{\theta} \sim N(\boldsymbol{\theta}_0, \mathbf{R})$ where $\boldsymbol{\theta}_0$ is the mean parameter vector, and $\mathbf{R}$ is the covariance matrix. Then the estimate $\widehat{\theta}$ that maximizes the posterior density $p(\theta|\mathbf{Y}_T)$ is the least squares estimate

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \sum_{k=1}^{T} \|y_k - f(k, \boldsymbol{\theta})\|_Q^2 + \|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_{\mathbf{R}}^2 \tag{1.3}$$

where notation $\|X\|_M^2 = X^T M^{-1} X$ for a positive definite symmetric matrix M,and $Q = \sigma^2 I$, as a form that generalizes to vector valued observations y_k. Then the $k-$step ahead prediction is $E(y_{T+k}|\widehat{\theta}) = f(T+k, \widehat{\theta})$.

(ii) Modified Batch Nonlinear Regression (MBNLR) Algorithm. By expanding an iteration step in the Gauss-Newton minimization procedure of (1.3) in the BNLR algorithm, it can be shown that

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_0 + \mathbf{R} f'(\boldsymbol{\theta}_i)^T (f'(\boldsymbol{\theta}_i)\mathbf{R} f'(\boldsymbol{\theta}_i) + \mathbf{Q})^{-1} \left(\mathbf{y} - f(\boldsymbol{\theta}_i) - f'(\boldsymbol{\theta}_i)(\boldsymbol{\theta}_0 - \boldsymbol{\theta}_i)\right) \tag{1.4}$$

This looks somewhat analogous to the Kalman filter update equation, of the form

$$\begin{aligned} \boldsymbol{\theta}_{i+1} &= \boldsymbol{\theta}_0 + \mathbf{K}_i(\mathbf{y} - f(\boldsymbol{\theta}_\mathbf{i}) - f'(\boldsymbol{\theta}_i)(\boldsymbol{\theta}_0 - \boldsymbol{\theta}_i)) \\ \mathbf{K}_i &= \mathbf{R} f'(\boldsymbol{\theta}_i)^T (f'(\boldsymbol{\theta}_i)\mathbf{R} f'(\boldsymbol{\theta}_i) \end{aligned}$$

If we run this iteration to convergence, we get an estimate $\widehat{\boldsymbol{\theta}}_N$ after a sequence of N measurements. If, instead, after $N+1$ measurements, we use this estimate $\widehat{\boldsymbol{\theta}}_N$ instead of $\boldsymbol{\theta}_0$ in the equation above, we obtain exactly the Iterated Extended Kalman Filter (IEKF). This motivates a modified form of the BNLR algorithm, where the update equation (1.4) after convergence of the current iteration process uses $\widehat{\boldsymbol{\theta}}_N$ instead of θ_0 for the next estimator (after $N+1$ measurements). We call this the Modified Batch Nonlinear Regression (MBNLR) algorithm, which is equivalent to batch non-linear least squares solution to the state-space model:

$$\begin{aligned} y_k &= f(k, \boldsymbol{\theta}_k) + \varepsilon_k \\ \boldsymbol{\theta}_{k+1} &= \boldsymbol{\theta}_k + \boldsymbol{\eta}_k \end{aligned} \tag{1.5}$$

where ε_k are i.i.d. and distributed as $N(0, \sigma^2)$ and $\boldsymbol{\eta}_k \sim N(0, \mathbf{R})$ are also iid, and ε_k and $\boldsymbol{\eta}_k$ are uncorrelated. That is, in our above observation model (1.2 we allow a random drift in the parameter values after each subsequent measurement. Note that the predictor above is the the solution to a non-linear state estimation problem which could be solved using approximate solutions such as Extended Kalman Filter (EKF) or Particle Filtering (PF). EKF is a single step of the IEKF; both EKF and PF are approximate and hence suboptimal to our batch algorithm that uses $\widehat{\boldsymbol{\theta}}_N$ in the update equation in order to solve the estimation problem for the state model (1.5).The steps of the MBNLR algorithm are summarized below:

1. Set $\widehat{\boldsymbol{\theta}}_0 = \boldsymbol{\theta}_0$ (which is given).

2. For $N \geq 1$, given measurements $y_1, y_2 ... y_N$ compute the estimate $\widehat{\boldsymbol{\theta}}_N$ recursively as

$$\widehat{\boldsymbol{\theta}}_N = \arg\min_{\boldsymbol{\theta}} \sum_{k=1}^{N} (y_k - f(k, \boldsymbol{\theta}))^2 + \sigma^2 \left\| \boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}_{N-1} \right\|_{\mathbf{R}}^2$$

3. Form the $k-$step ahead prediction

$$\widehat{y}_{N+k|N} = f(N + k, \widehat{\boldsymbol{\theta}}_N)$$

The minimization procedures above were implemented using quasi-Newton algorithm and the hyperparameters $\sigma^2, \mathbf{R}$, $\boldsymbol{\theta}_0$ above were estimated using a stochastic E-M algorithm (an alternative is to use particle filtering).

1.2.3 Prediction Results

We discuss only a few relevant results, more details on the results are on the Online Supplemental Material Given a step length k and observation window T, for the estimate $\widehat{y}_{T+k|T}$ of y_{T+k} we define the *prediction error* $E(k,T) = y_{T+k} - \widehat{y}_{T+k|T}$ and define *Prediction Root Mean Square Error* (PRMSE) as

$$PRMSE(T) = \left(\frac{1}{N-T} \sum_{k=1}^{N-T} (E(k,T))^2 \right)^{\frac{1}{2}} \tag{1.6}$$

where N is the total number of observations available. Let $f_{T+k} = f(T+k, \widehat{\theta}_N)$, that is, the value predicted from the Two-Process model based upon the best estimate (given the entire set of observations) Then we define *prediction-fitness error* $E^{pf}(k,T) = f_{T+k} - \widehat{y}_{T+k|T}$ and *fitness error* $E^{err}(k,T) = y_{T+k} - f_{T+k}$. Note that $E(k,T) = E^{pf}(k,T) + E^{err}(k,T)$ and while component E^{pf} is a measure of the algorithm performance given a particular model (in our case, the Two-Process model), E^{err} is a measure of how well the chosen model fits the given data. Measures analogous to PRMSE, namely $PRMSE^{pf}(T)$ and $PRMSE^{err}(T)$ are defined using (1.6) replacing E with E^{pf} and E^{err} respectively.

For each individual, we tested both algorithms BNLR and MBNLR to make k step ahead predictions given T observations, for $k = 1, 2, 3...32$ hours and $T = 1, 2, 3...40$ hours. . We set the number $N_1 = 6$h to be the shortest time before we can make any prediction of any kind. The total number of observations available for validation corresponds to $N = 44$h. Results were validated by comparing the prediction estimates with the experimental data. A continuous real-time prediction task for a fixed step size k was simulated by using data

sequentially up to time $T = 1, 2, 3...40$ hours to estimate model parameters that are then used to derive the estimate at time $T + k$. Fig. 1.4 shows the prediction results for three participants for 2h,6h and 10h steps ahead. A visual inspection shows vast variation in the difference between predicted results and experimental data across participants.

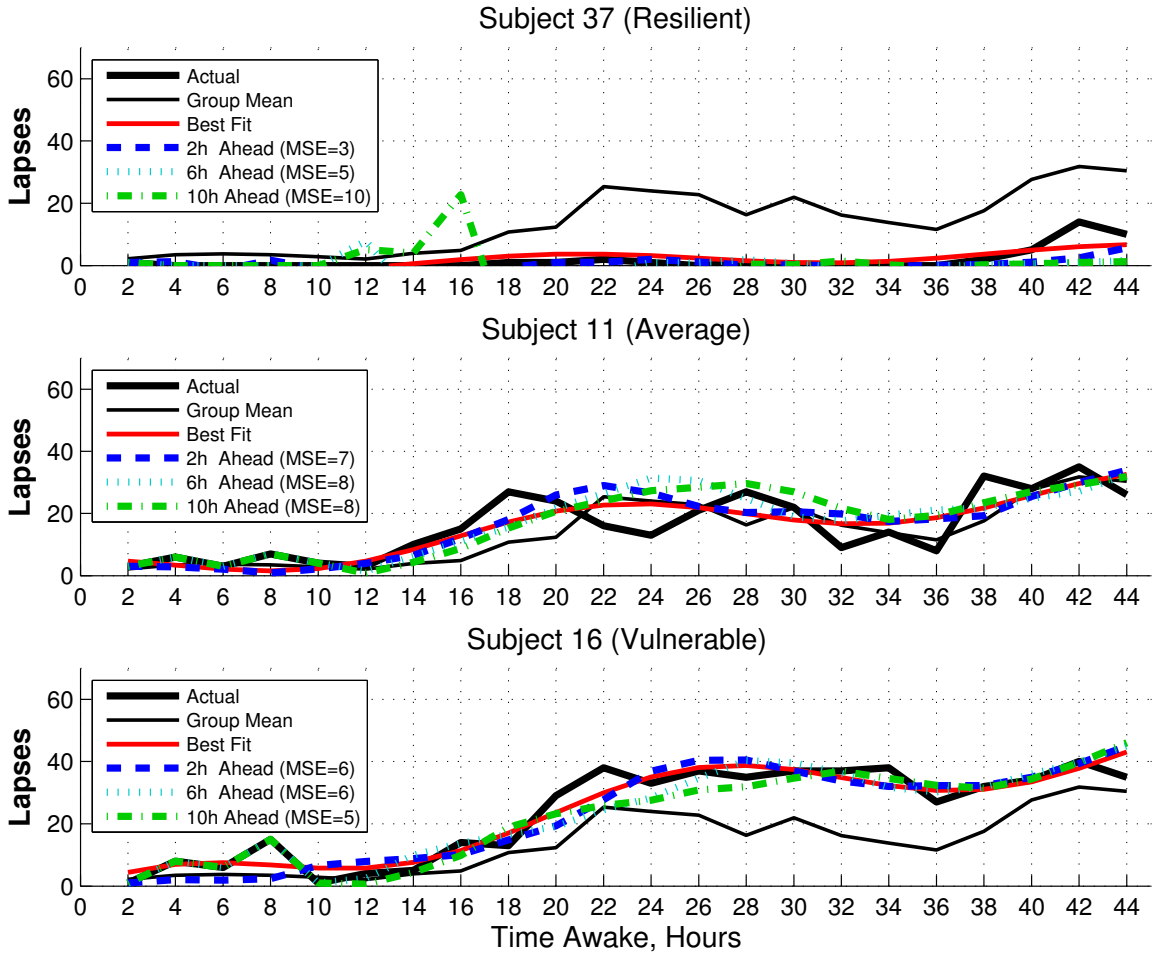


Figure 1.4: k-step ahead prediction based upon modified Batch NLS algorithm for some of the research participants. The individual and group mean values, and the best fit using the Two-Process model are also shown in each panel. The prediction results are for step sizes k = 2,6, and 10h.

The behavior of $PRMSE^{pf}$(1.6) with observation length T gives us an indication of how well our algorithm's estimates converge to the true model parameters. Only in 21 out of the 33 participants we see overall decrease in $PRMSE$ and $PRMSE^{pf}$ with observation length (Fig 1.6) and these participants also showed convergence after 34 hours of observa-

tions (Fig 1.5). This suggests that in the model comprising of the Two-Process model and assumption of Gaussian measurement noise is a poor fit for 12 out of 33 participants' data. When averaged over *all* participants, however, the error shows a decrease with observation length (Fig 1.7).

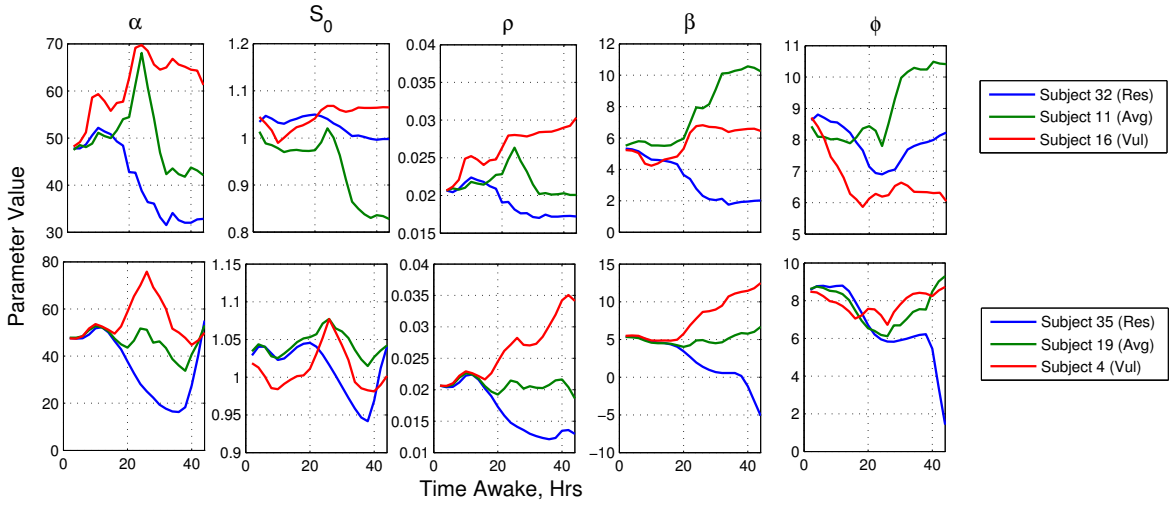


Figure 1.5: Convergence of parameters with number of observations in the MBNLR Algorithm in six participants.

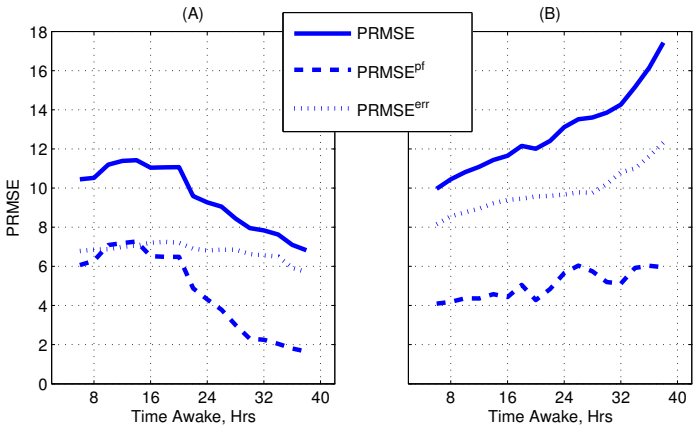


Figure 1.6: **The $PRMSE$, $PRMSE^{mf}$ and $PRMSE^{err}$, averaged across participants per group and step lengths, as a function of observation window.** (A) $PRMSE$ averaged over participants for which the algorithm converges (21 out of 33) (B) $PRMSE$ for participants (12 out of 33) for which the algorithm does not converge.

The BNLR algorithm was applied to data digitized from published data in two

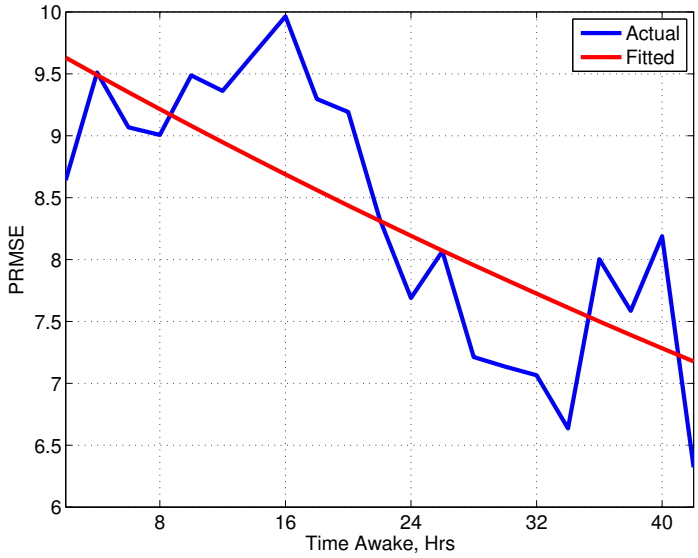


Figure 1.7: The median PRMSE of lapses across subjects and step lengths and the best exponential fit using the fit $M_b e^{-T/T_b}$ as a function of observation window T.

studies of PVT performance prediction. In [224], a Bayesian forecasting procedure is used to make 24h performance predictions for types of participants: Resilient, Average, and Vulnerable (Fig. 1.8) after observing 12h, 20h, 28h, 36h and 44h of wake performance data. Our algorithm performs worse for the Vulnerable participant when more observations are available, but better when fewer are available, which is preferable. Our algorithms also tend to do better with prediction of lapses in the Resilient individual. When averaged over all participants and observation periods, our methods have an overall improvement of 18% in RMSE lapses.

In [184] a linear method is used, and the digitized results for 10h-step predictions were compared with those from the BNLR algorithm for three participants, one in each category (Fig 1.9). Averaged over the three participants we see an overall 15% improvement in prediction performance.

1.2.4 Discussion

While possible implementations of the algorithm include particle filtering and non-linear Kalman filters (e.g., scented or extended Kalman filter), our simple non-linear least squares algorithms do better in prediction of PVT lapses from baseline predictions than existing ones ([183] , [184],[224]). The behavior of MSE with observation length (Fig 1.6A) is

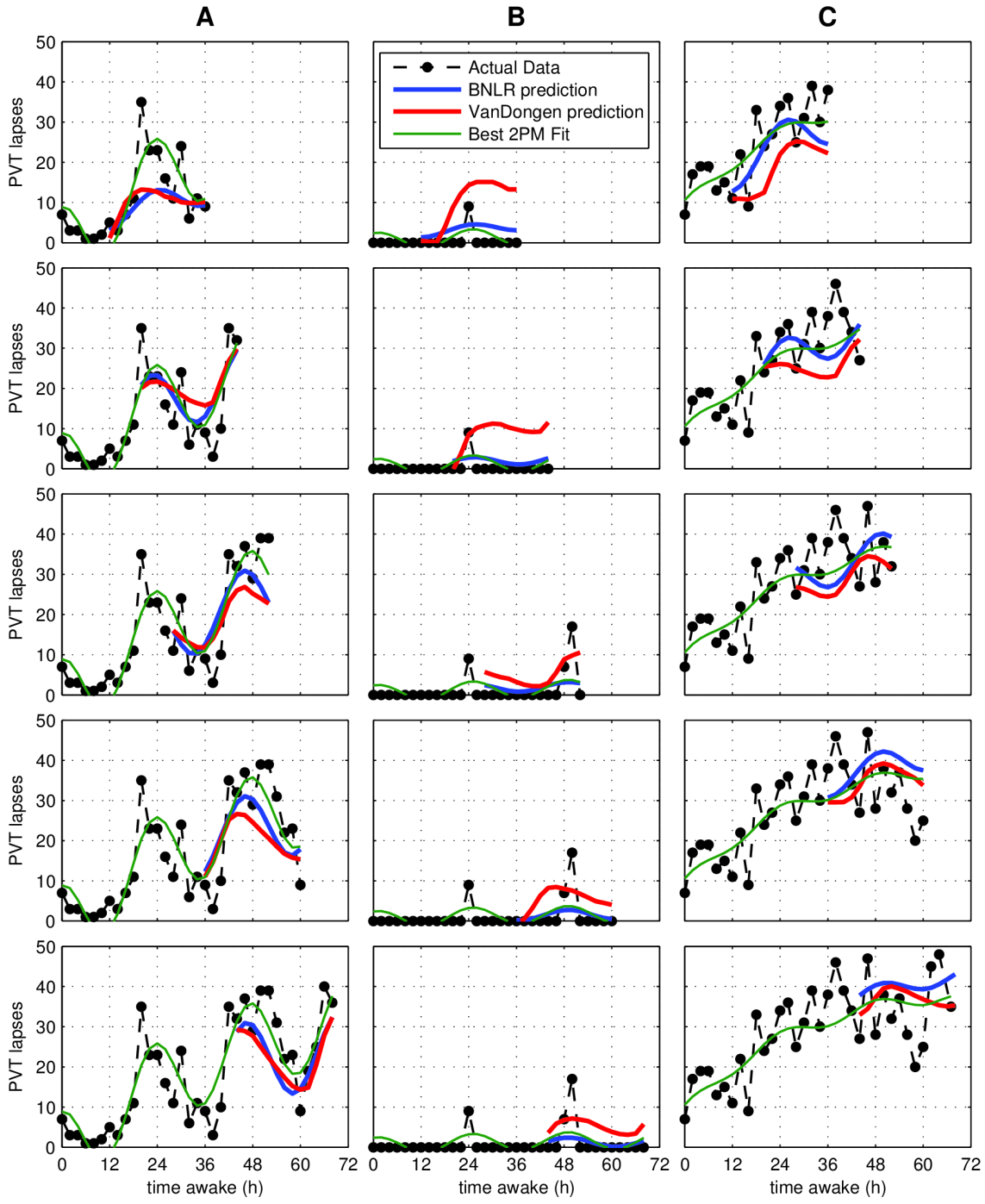


Figure 1.8: **Prediction using BNLR and method used in [224].** Results are adapted from Figure 2 of [224] are overlaid on top of results from BNLR algorithm. Each column of panels represents the (A) Average, (B) Resilient, and (C) Vulnerable participant. The black circles denote the experimental number of lapses, and the blue and green solid lines show 24-hour predictions from BNLR and the algorithm used in [224], respectively. The top panels show prediction results after observation of 12h of awake data, and each subsequent row shows results from observing and additional 8h of data. The best Two-Process model fit is shown in solid green.

indicative of at least two sources of variability: (1) trait variability that is reflected in the per individual parameters which can not be inferred from a small number of baseline observations, and (2) inter-trial variability that is not captured by the noise observation model. The fact that a stochastic model (such as the drift model implemented using the MBNLR algorithm) does not account for any additional MSE is indicative of a fundamental limitations in the Two-Process model to account for this variability. If the individual parameter differences were the only source of MSE in prediction, then one would have expected that MSE would initially decrease with observation length but at some point reach a constant level. However, that does not seem to be the case from the continuing downward trend as shown in Fig 1.6A.Our next goal in the future is to incorporate EEG measurements to test whether this additional information causes the estimation error to decrease faster than what we report above, thereby indicative of better individualized predictions.

To make our conclusion concrete regarding our original goal of how well can PVT measurements alone be predicted (without use of an additional measurement such as EEG). To quantify our hypothesis, we have fitted the following characteristic curve: median MSE of lapses vs. observation length:

$$PRMSE(n) = M_b \exp(-\frac{n}{T_b})$$

where M_b is the maximum (baseline) PRMSE with no observations (measured in lapses), and n is the observation window in hours, and T_b is a characteristic constant measured in hours. Here the MSE is averaged taken over all step (horizons), and the median is taken over all subjects. For the CR study above, our batch algorithm gives (Fig 1.7) us $T_b = 34h$, and $M_b = 9.7$ lapses.

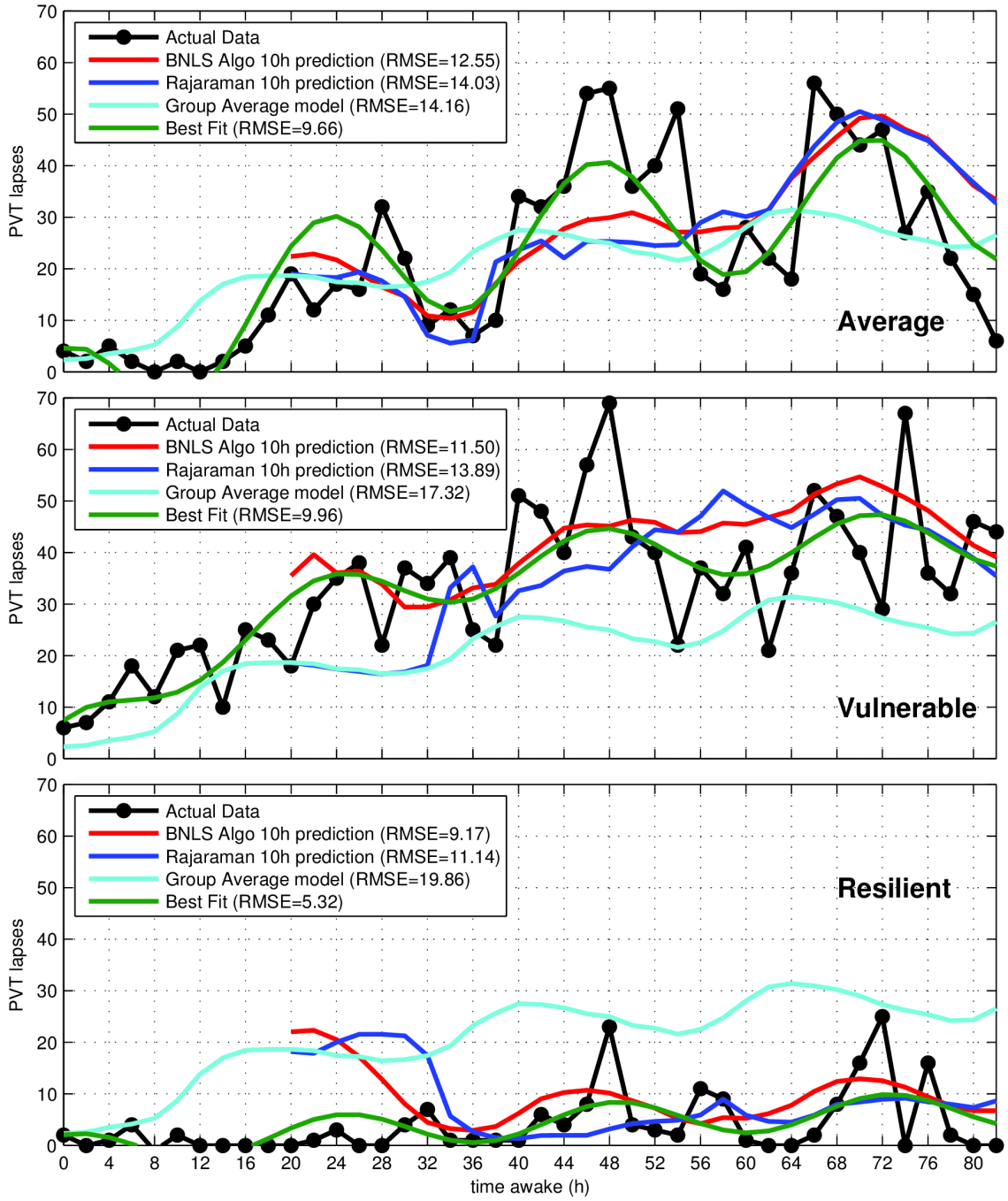


Figure 1.9: **10h-ahead performance prediction using BNLR and method used in [184]**Results adapted from Figure 7 of [184] are overlaid on top of results from our BNLR algorithm for the same data. Individualized predictions are for three participants: Average (top), Vulnerable (middle) and Resilient (bottom). The black circles denote the experimental number of lapses, and the red and blue solid lines show 24-hour predictions from BNLR and algorithm used in [184]. The best Two-Process model fit is shown in solid green, and the group average is shown in cyan.

1.3 Problem (B): EEG-Based PVT Prediction

Thus, performance measurements alone are insufficient to predict future measurements on an individual basis using models that predict well at a group level. We also saw that inter-individual differences are consistent and hence based in physiology rather than an artifact of varying testing environments. It has been shown that EEG has been directly shown to be correlated to performance decrement with sleep deprivation and shows circadian modulation [35] and that they differ substantially between individuals. It is thus plausible that EEG contains the information needed to infer these inter-individual differences [32]. We consider some linear models using EEG power spectrum to predict performance by using data collected under controlled conditions of sleep deprivation, sleep restriction and adverse circadian phase and will address problem (B) posed in Section 1.1. The main conclusion of our analysis was that while EEG spectrum on a group level exhibits specific wake-time behavior, unless some additional information - such as individual phase - is provided, spectral based measurements do not improve estimates of performance lapses on a per individual basis. A comprehensive survey of other studies on EEG-based fatigue detection is provided at the end of this chapter. Details on the analyses and some other problems such as (i) impact of chronic sleep debt on EEG spectrum, (ii) use of EEG for classification instead of prediction tasks are provided on the *Online Supplemental Material*.

1.3.1 Previous Studies

It is well known that the homeostatic process of sleep/wake regulation is reflected by changes in electroencephalographic slow wave activity (SWA, 0.75-4.5Hz) in non-rapid-eye-movement (NREM) sleep ([3]). SWA has been known to increase in the beginning of sleep and decrease with duration of sleep. It has also been known that increased theta activity is associated with increased drowsiness, and that absence of SWA and presence of alpha (8-12Hz) and beta (>20Hz) is required for memory and attention ([35]). In fact, this has been shown to be the most predictive to date of all physiological indicators of fatigue [82, 228, 11].

Changes in EEG spectral power with varying degrees of alertness have been well documented, though the exact nature of these changes is not agreed upon universally. In general, high *Delta* band activity (0.5-4 Hz) is observed during sleep and transition to drowsiness. *Theta* band activity (4-7 Hz) is generally associated with decreased information processing and is shown to replace *Alpha* band activity (7-14 Hz) at the onset of sleep and thus associated with early stages of drowsiness.*Alpha* band activity increases with increased "relaxed wake-

fulness" (a non-receptive state) and decreases in the occipital derivations as one gets tired. Usually higher frequency activity such as *Beta* band (15-30 Hz) indicates increased wakefulness and alertness. While several automated fatigue detection systems are based upon simple monitoring of relative activity in these bands (such as ratio of *Alpha* + *Theta* to *Beta* activity), the nature of these changes in band power activity with vigilance states is not uniform across studies. In [6, 217] *Alpha* bursts and appearance of *Theta* waves were shown indicative of lapses in alertness. In [22], an increase in *Delta* and *Theta* activities with decreased *Beta* activity was observed during fatigue whereas in [138] a slight increase in *Alpha* and *Beta* and a significant increase in *Delta* and *Theta* was reported. In [5], the proportion of band powers was calculated during the eyes open (EO) and eyes closed (EC) segments of the Karolinska Drowsiness Test (KDT): during active/awake condition, the EO segment is dominated by *Beta*, and the EC segment by *Alpha*; but with increased levels of sleepiness, the EO segment has increased *Alpha* and *Theta*, and the EC has increased *Theta* and reduced *Alpha*. Based upon this, in [208] a sleepiness test measuring the ratio of *Alpha* during EC/EO segments (*Alpha* Attenuation Test or AAT) was developed. In [122] a linear relationship between KSS (Karolinska Sleepiness Scale, a subjective scale of sleepiness) and EEG power in *Theta* and *Alpha* power during EO and a linear decrease in *Alpha* (but not *Theta*) during EC was demonstrated. EEG site specific changes are also found in various band powers, with increased low frequency components being most dominant in the occipital areas [150]. In [113, 83] the ratios $(Alpha+Theta)/Beta$, $Alpha/Beta$, $(Theta+Alpha)/(Alpha+Beta)$, $Theta/Beta$ were all found to increase in all brain areas except the temporal lobe during a monotonous driving task, with the occipital lobe $(Theta+Alpha)/Beta$ being the most significant. Particular characteristics of *Alpha* activity have been reported in [137, 193] with onset of drowsiness: *Alpha* attenuates for a few seconds and then reappears again, and this alteration repeats and finally disappears at sleep; frontal *Alpha* activity increases, lasting for 1-10s, while occipital *Alpha* simultaneously decreases. In addition, some characteristics of sleep state appear even with eyes open when one is fatigued, such as sleep spindles [203] and k-complexes [170]. In [176] EEG was shown to have an acute increase of the *Alpha* waves, decrease in gamma waves and significant increase in Kullback-Leilbler (KL) entropy from the first to last five minutes of a long monotonous fatigue-inducing driving task. Specific patterns of changes in EEG with varying degrees of sleep deprivation and circadian phase were studied in [35, 36]; they concluded that with wakefulness, *Delta* and *Theta* power increases without circadian modulation, *Alpha* activity is unchanged but is circadian modulated, *Sigma* and *Beta* changes

with circadian modulation, and that all these changes exhibit site (e.g., frontal vs. occipital) specific differences. However these results were qualitative and no specific quantitative conclusions were made.

Due to the non-stationarity of the EEG signal, measures of the EEG spectrum other than FFT have been studied in relation to alertness levels. Changes in dominant frequency and frequency variability in each band [237, 201], inter and intra-hemispheric cross spectral density [229], sample entropy, approximate entropy and Kolmogorov complexity [46, 123, 83, 42, 33] were shown to convey more information in their ability to distinguish alertness states than band power alone. In particular, it has been shown that the signal entropy increases with fatigue and that just before an alertness lapse the cross-entropy of the EEG signal in $Alpha/Gamma$ band power decreases while *Alpha* phase synchronization increases (i.e. *Alpha* bursts) [175]. Approximate entropy of EEG was statistically significantly is lower during Stage IV (i.e. deep) sleep and higher during wake and REM sleep [33]. Other non-linear measures such as fractal dimensions, bispectrum [250], Gabor transform [181] and Hilbert-Huang transform [43] have also been explored in the context of fatigue level. Most of the above studies, however, are not conducive to implementations in real-time [16] and have been *correlational* rather than *causal*.

Systems vary in degree of technologies used regarding feature extraction, feature reduction and classification. FFT-based power spectral density measures (most commonly of which are *Theta*, *Alpha* and *Beta* band powers [119, 145, 149, 38], dominant frequency, average power of peak, center of gravity of frequency and frequency variability [237, 201]) alone usually do not result in high degree of accuracy due to the inherent assumption of stationarity. Band energy powers derived using discrete wavelet transform (DWT) have shown to be more reliable that FFT based automated EEG classification, and as such wavelet methods (including wavelet packet transform or WPT) have become the de-facto in EEG-based fatigue analysis [151, 247, 243, 125, 8, 134, 209, 42, 27, 97, 104, 140, 242]. The observation of *Alpha* bursts just before driving lapses led many authors to use non-linear features such as entropy measures approximate entropy (ApEn) ([175, 42, 151]), wavelet entropy [242], Kolmogorov complexity [151], cross-spectral density [229] and fractal dimension [157] which have reported varying degrees of accuracy for classification. Most technologies augment the above methods (FFT, WPT or non-linear) with PCA [119, 145, 147], ICA [149, 150, 148], Kernel-PCA (KPCA) [151, 247, 243, 143], mapping constructive agglomerative (MCA) [27], and fuzzy mutual-information (MI) based feature extraction [140] for feature reduction. Classification

and prediction is usually done by unsupervised machine learning methods such as support vector machines (SVM), or supervised methods such as linear regression, linear discriminant analysis (LDA), artificial neural networks (ANN) and hidden Markov models (HMM).

However, there are several issues with the above systems. Most systems do not use standard objective fatigue measures but instead use subjective, manual or custom ones derived from the experimental protocol for testing. For example, [237, 8, 134, 209, 229, 42, 140] drowsiness state is determined manually by an expert EEG neurologist, in [175] by an observer of subject behavior, and in [242, 143] by an observer of a video stream of the user. Such methods are not extendable to operational conditions. Some recent studies use polysomnographic databases of EEG recordings that are pre-classified into "sleepy" and "awake" states [27, 104, 97], but these databases contain sleep study data that are not necessarily indicative of driver fatigue states. Many studies assume monotony of a task will automatically induce fatigue and thus use time-on task as a measure of fatigue [113, 174, 247, 236], or assume that switching to a specific type of task will induce fatigue [151]. Systems developed using protocols where it is monotony rather than underlying causal factors (e.g., sleep loss and adverse circadian phase) that induce fatigue would be expected to perform poorly in real life situation with a sleep-deprived driver [201]. In some cases the experimental protocol, such as a simulated virtual-reality driving task, dictates the metric to be used such as lane deviation [145, 149, 147, 150, 148] or reaction time to crash/avoidance [140, 157]. Special tasks with customized metrics are used in [38, 119, 201] and subjective measures such as KSS and SSS are used in [243]. The manual, non-standard or subjective nature of these metrics makes it harder to evaluate these systems in real life conditions. Furthermore, *virtual* task metrics are often not indicative of real life fatigue [192].

Most studies study the association between EEG and alertness at only unknown circadian phase and wake duration; we do not know (i) how long the participants had been in the time zone of study or at what time they were studied relative to their habitual wake time (both are needed to determine relative circadian timing) (ii) how long they had been awake when they were studied (possible sleep deprivation), and/or (iii) how much sleep they had received in the past week (possible sleep restriction). Very few systems (such as [236, 108, 48]) take into account these fatigue-inducing factors. Causal models such as we propose should make predictions *in addition* to monitoring of fatigue.

Many systems are based upon a large number (e.g., 20-64) of EEG channel inputs, which is impractical in a real-time continuous monitoring situation. Our experimental pro-

tocols used to develop our models will rely upon only 4-6 channels of EEG. Another issue is that group models derived from unsupervised methods perform poorly for individuals. While there exist some supervised models [119, 145, 149, 147, 150, 148, 174, 146] that address the issue of individual specificity, they require training and testing on the same individual; the long time of training that is necessary makes this approach impractical. In [174, 146] a novel unsupervised method that computes Mahalanobis distance from a probability distribution fit based upon baseline data has been used to obviate need for training sets to build subject-specific models, though some tuning parameters for such models must still be preselected per individual. To our knowledge there has been no previous report of prediction of performance measures such as lapses or median reaction time using EEG measurements on a per individual basis.

1.3.2 Data Collection & Processing

EEG data were collected from inpatient studies of nine healthy young individuals [49] during a forced desynchrony (FD) [57, 41] protocol, consisting of 12 cycles of 42.85-hour days with 3:1 wake:sleep ratio. Starting 2 hours after awake, the subjects are given the PVT every four hours, their response times are recorded, and number lapses determined. The Karolinska drowsiness test (KDT) was performed at 2-h intervals starting 4h after scheduled wake time. During this test subjects are instructed to relax and fixate on a 5-cm black dot 1m away attached to a computer screen for 4min, followed by 1 min with eye closure. EEG measurements using a z-line (Fz,Cz,Pz,Oz) were made roughly starting at 10h after awake, along with electrooculogram (EOG) measurements. All signals were digitized using a 12-bit AD converter, stored at a sampling rate of 256Hz and digitally filtered at 35Hz and 0.5Hz. The EEG signals during the 4-min eyes open segment of the KDT were visually inspected for eye blinks, slow eye movements and small body movements. Two second epochs containing muscle activity, eye blinks, slow eye movements and microsleeps were marked as artifiacts and stored on a separate artifact channel. Differential signals (Fz-Cz and Pz-Oz) were then formed and subject to the Welch's method of power spectral density estimation [215, 12] using non-overlapping window segments of 1s. Based upon spectral power, some epochs were marked as outliers and further rejected as artifactual epochs. In this manner for each KDT episode, we were able to obtain from 10-80 artifact free epochs. The estimated power spectrum is averaged across all epochs within a given KDT, the assumption being that the EEG signal is stationary during a KDT episode of 3min. The computed power spectrum is

z-scored across the entire protocol duration for each individual separately to account for scalp differences. This is then binned into five frequency bands: delta (0.5-4Hz), theta (4.5-8Hz), alpha (8.5-12Hz), beta-1 (12.5-15Hz), beta-2 (15.5-20Hz) ([35]).

Based upon the circadian phase at awakening and the endogenous period for each participant the time at each administration of PVT and KDT is assigned a 60^0 phase bin and a 4-hr time awake bin, so that all data could be analyzed identically across subjects for effects of phase and homeostat separately. Binning was done per week (a total of three weeks), so that this way the EEG power density in each frequency band was allocated across 6 x 6 x 3 bins. When analyzing data, it is first averaged in each bin within each subject in order to give equal weight to each bin.

1.3.3 Prediction Algorithms

As before, we let $\mathbf{Y}_T^s = \{y_1^s, ...y_T^s\}$ be the set of PVT lapse measurements up to and including at T for subject s. Let the corresponding EEG spectrum measurements be $\mathbf{X}_T^s = \{x_1^s, ...x_T^s\}$ where x_i^s is a 10 dimensional vector, comprising of the z-score of the power spectrum value in the five bands in the two derivations (Fz-Cz, Pz-Oz). We are interested in the $\widehat{y}_{T+k} = E(y_{T+k}|\mathbf{Y}_T, \mathbf{X}_T)$, i.e. *$k-$step prediction after N observations.* As in Problem (A) if assume a static regression model of the form, where $\theta \in R^p$ is a vector of (possibly unknown) parameters,

$$\begin{aligned} y_k &= f(k,\theta) + \varepsilon_k \\ x_k &= g(k,\theta) + \nu_k \end{aligned} \tag{1.7}$$

where $\varepsilon \sim^{iid} N(0,\sigma^2)$ and $\nu_k \sim^{iid} N(0,\Sigma)$ and a prior model $\theta \sim N(\theta_0, R)$ then the estimate $\widehat{\theta}$ that maximizes the posterior density $p(\theta|\mathbf{Y}_T, \mathbf{X}_T)$ is the solution to the least squares

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \sum_{k=1}^{T} \|y_k - f(k,\boldsymbol{\theta})\|^2_{\boldsymbol{\sigma}^2 I} + \sum_{k=1}^{T} \|x_k - g(k,\boldsymbol{\theta})\|^2_{\boldsymbol{\Sigma}} + \|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|^2_{\mathbf{R}} \tag{1.8}$$

from which the $k-$step ahead prediction $E(y_{T+k}|\widehat{\theta}) = f(T+k, \widehat{\theta})$ can be computed. Hyper-parameters $(\sigma^2, \Sigma, R, \theta_0)$ above can be estimated using EM as before.

Linear Prediction (LPM-1) Algorithm: No known physiologically motivated parametric forms for f, g are available. Due to the limited number of data points available, we have not used AR type models either. Instead, we use a linear model for f, g as follows. Instead we do

PCA over both y, x from 8 (out of 9) subjects and all wake periods to compute eigenfunctions $E_d = \begin{bmatrix} F_d & G_d \end{bmatrix}$ (where F_d, G_d are the components corresponding to y and x respectively), and expand f, g using the first D dominant principal components $\theta = (\alpha_1, \alpha_2, ...\alpha_D)$

$$\begin{aligned} f(k,\theta) &= \sum_{d=1}^{D} \alpha_d F_d(k) \\ g(k,\theta) &= \sum_{d=1}^{D} \alpha_d G_d(k) \end{aligned} \tag{1.9}$$

Then (1.8) becomes a linear least squares problem which can be solved in closed form using pseudo-inverse.

Linear Prediction (LPM-2) Algorithm This is similar to the above, but instead PCA is performed independently on y, x to give eigen functions F_d, G_d and the D_1, D_2 dominant components respectively are used to expand f, g:

$$\begin{aligned} f(k,\boldsymbol{\alpha}) &= \sum_{d=1}^{D_1} \alpha_d F_d(k) \\ g(k,\boldsymbol{\beta}) &= \sum_{d=1}^{D_2} \beta_d G_d(k) \end{aligned}$$

where $\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 & ... & \alpha_{D_1} \end{bmatrix}, \boldsymbol{\beta} = \begin{bmatrix} \beta_1 & ... & \beta_{D_2} \end{bmatrix}$. We also assume the linear regression model

$$\boldsymbol{\alpha} = \boldsymbol{\Gamma}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{1.10}$$

where $\boldsymbol{\epsilon} \sim N(0, S)$.The resulting least squares problem, using notation $\mathbf{F} = \begin{bmatrix} F_1 & ... & F_{D_1} \end{bmatrix}$, $\mathbf{G} = \begin{bmatrix} G_1 & ... & G_{D_2} \end{bmatrix}$,

$$\widehat{\boldsymbol{\theta}} = (\widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\Gamma}}) = \arg \min_{(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\Gamma})} \sum_{k=1}^{T} \left(\left\| y_k - \mathbf{F}^T(k)\boldsymbol{\alpha} \right\|^2_{\boldsymbol{\sigma}^2 I} + \left\| x_k - \mathbf{G}^T(k)\boldsymbol{\beta} \right\|^2_{\boldsymbol{\Sigma}} \right) + \left\| \boldsymbol{\alpha} - \boldsymbol{\Gamma}\boldsymbol{\beta} \right\|^2_S \tag{1.11}$$

is solved in two steps: (i) Compute the projections $\widetilde{\boldsymbol{\alpha}}^s, \widetilde{\boldsymbol{\beta}^s}$ of *all* observations $\mathbf{X}_N, \mathbf{Y}_N$ for subjects $s = 1..8$ onto the basis functions $\mathbf{F}, \mathbf{G}$ and use it to find the least squares estimate $\widehat{\boldsymbol{\Gamma}}$ of $\boldsymbol{\Gamma}$ i.e.

$$\widehat{\boldsymbol{\Gamma}} = \arg\min_{\boldsymbol{\Gamma}} \sum_{s=1}^{S} \left\| \widetilde{\boldsymbol{\alpha}}^s - \boldsymbol{\Gamma}\widetilde{\boldsymbol{\beta}}^s \right\|^2$$

and then (ii) solve the linear least squares problem

$$\widehat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \sum_{k=1}^{T} \left(\left\| y_k - \mathbf{F}^T(k)\widehat{\boldsymbol{\Gamma}}\boldsymbol{\beta} \right\|^2_{\boldsymbol{\sigma}^2 I} + \left\| x_k - \mathbf{G}^T(k)\boldsymbol{\beta} \right\|^2_{\boldsymbol{\Sigma}} \right)$$

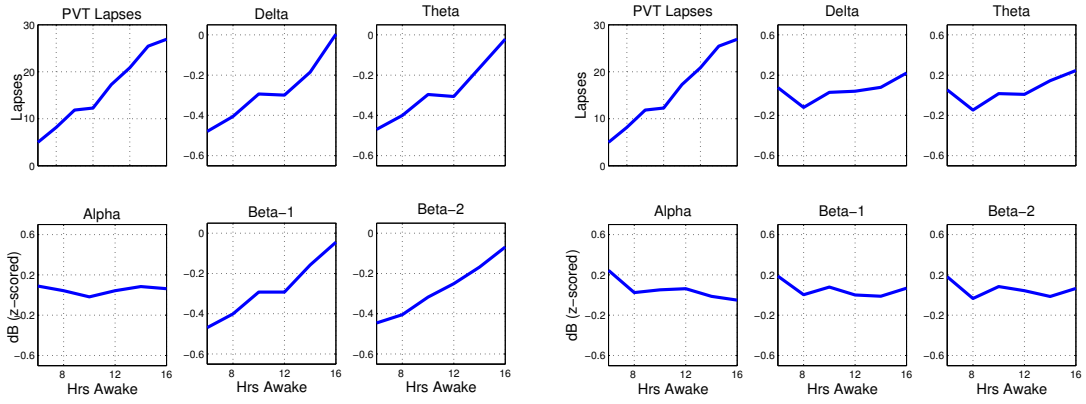
The prediction estimate is then given by $\widehat{y}_{T+k} = \mathbf{F}^T(T+k)\widehat{\mathbf{\Gamma}}\widehat{\boldsymbol{\beta}}$.

Per Phase Linear Prediction (LPM2-Phase) Algorithm We noted that the above algorithm performs well only we restrict our analysis to any *single wake period* (*or any single phase*), then the linear regression model is a better fit. Thus, the above algorithm is implemented on a *per phase* basis. We call this implementation **LPM2-Phase.**

Note that using the above PCA representation, the best possible estimate with any algorithm is $\widehat{y}_{T+k} = \mathbf{F}^T(T+k)\widetilde{\boldsymbol{\alpha}}^s$ which can be used to evaluate the performance of the prediction algorithms above.

1.3.4 Experimental Results

When averaged over all subjects, the time course spectrum over the duration of the protocol (12 periods of 32.5hr days) shows that EEG power exhibits homeostatic, circadian and chronic debt dependence in a manner similar to that of PVT lapses (Fig 1.11). Homeostatic dependence is strikingly similar, especially in the theta, delta and beta bands of the frontal deviation (Fig 1.10), as is the circadian variation when separated (Fig 1.12).



(a) Fz-Cz (b) Pz-Oz

Figure 1.10: Time course of PVT lapses and spectrum in the five bands (delta, theta, alpha, beta-1, beta-2) for (a) Fz-Cz derivation, (b) Pz-Oz derivation, averaged across all subjects, weeks and wake periods to eliminate chronic and circadian effect.

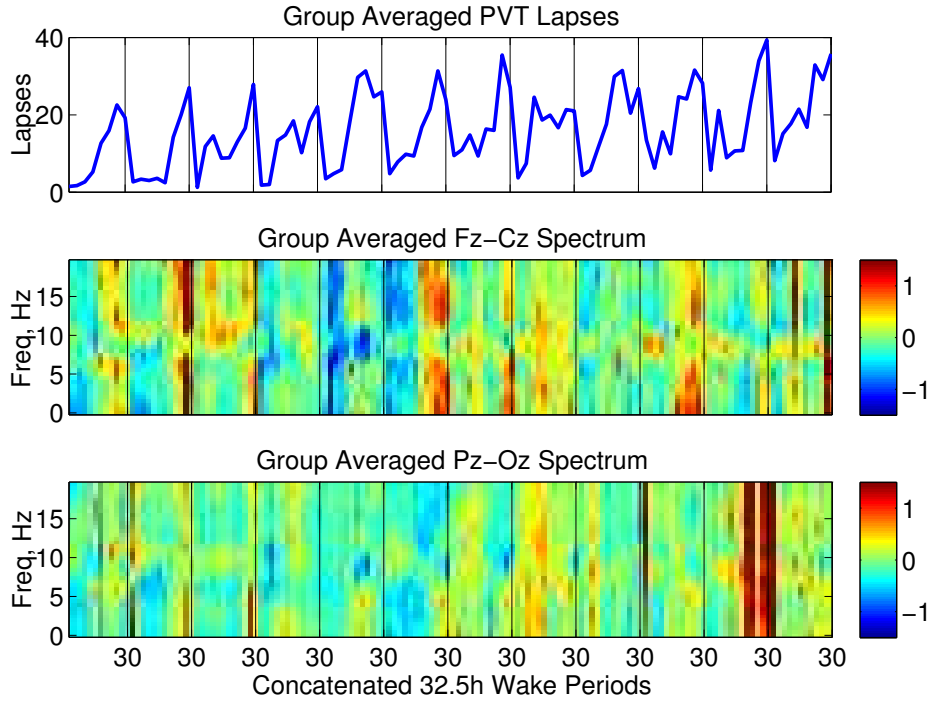


Figure 1.11: (Top) Time course of PVT lapses, and (Middle/Bottom) time frequency spectrogram of EEG collected during KDT, concatenated across all wake periods of the protocol (12 periods of 32.5hr days each), averaged over subjects.

However, this correspondence between PVT and EEG spectrum in all bands and derivations breaks down at an individual level (Fig 1.13). The correlation between PVT lapses and the power spectrum at all frequencies *per subject* is fairly poor, while that of *group* is quite high.

1.3.5 Prediction Results

Since computation of the PCs in our algorithm requires us to use a subset of subjects as training subjects (we used 8 out of 9), individual prediction results from using the LPM2-Phase algorithm are shown for one (testing) subject in Fig 1.15. We determined that using $D_1 = 1, D_2 = 2$ gaves us best results, and that LPM-1 and LPM-2 gave poor prediction results (and not shown). More details of results is in the *Online Supplemental Material*. The performance metric $PRMSE(T)$ for an observation window T used here is the same as in

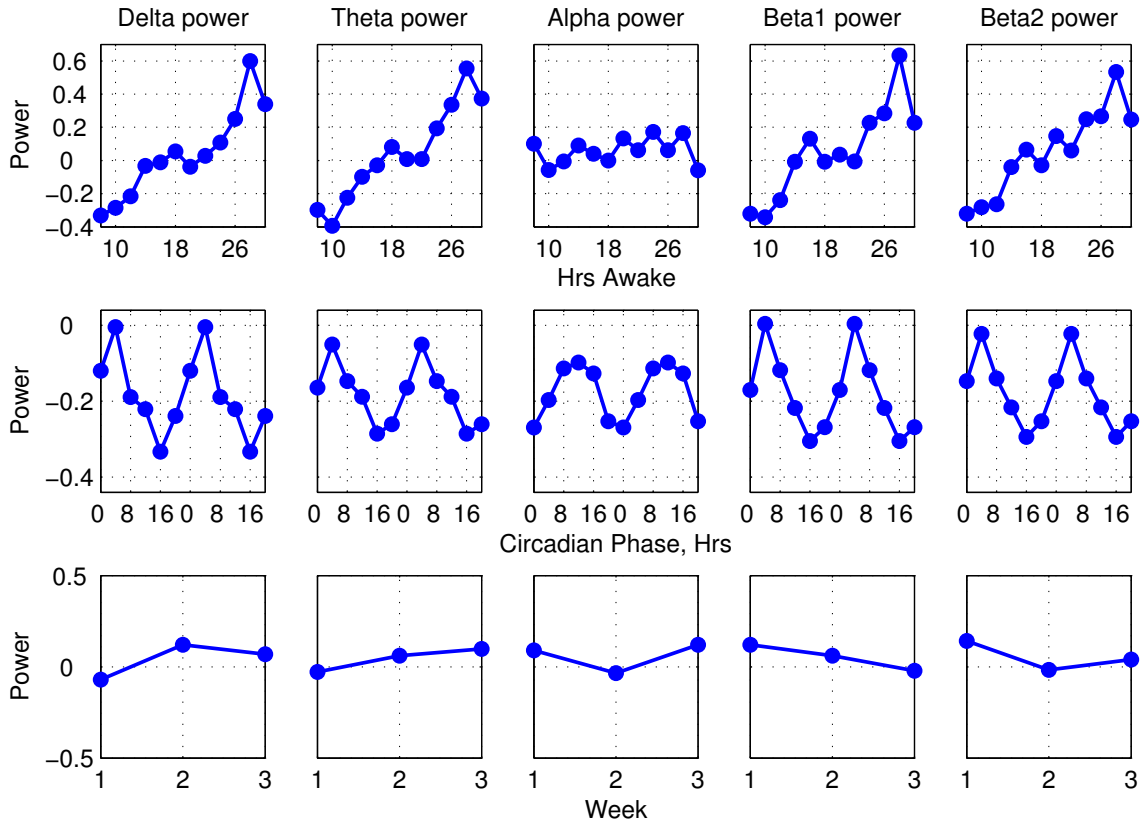
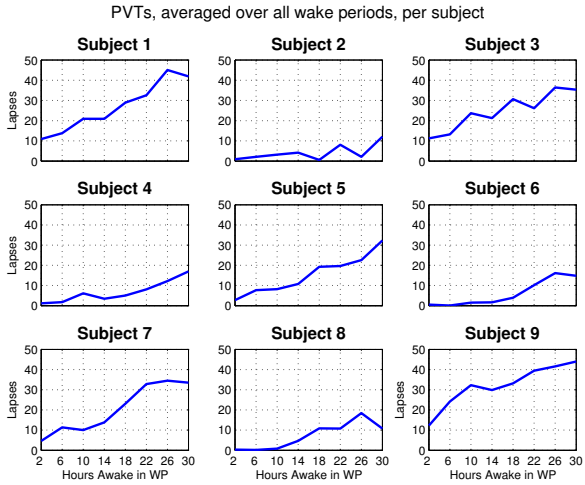


Figure 1.12: Separation of circadian, wake-dependent and chronic effects on EEG power spectrum in the five bands for the Fz-Cz derivation, when averaged over all subjects.

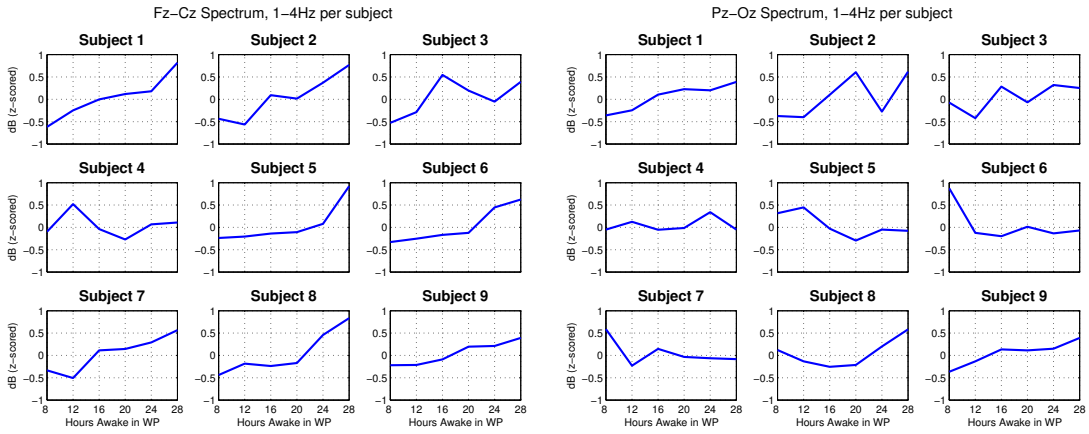
Section 1.2.3 A. We characterized the behavior of $PRMSE$ with T for the test subject. This was done several times by randomly selecting one subject as the test subject. The result of two such tests shows (Fig 1.16) that the prediction error does not decline faster when using EEG vs. when using baseline PVT measurements alone. Since the algorithm LPM2-Phase requires a knowledge of phase for computation of parameters, the prediction can be done only during any single given wake period. That is, the total horizon length is always less than 32.5 h for our prediction task.

1.3.6 Discussion

Our first two models LPM1 and LPM2 fail to predict inter individual differences in PVT based upon EEG spectrum. A scatter plot showing the relationship between the



(a) PVT



(b) Fz-Cz, Delta

(c) Pz-Oz, Delta

Figure 1.13: Time course of PVT lapses (top) and delta spectrum in the Fz-Cz and Pz-Oz derivations (bottom) for the nine individuals in the study. Data are averaged over weeks and wake periods to eliminate chronic and circadian effect.

dominant (first principal) coefficients of PVT and spectrum is shown in Fig 1.17. If a linear model that estimates subject specific parameters (θ in the model (1.7)) using both PVT (y_k) and spectrum (x_k) is to do better than just when measuring y_k alone, we would expect some degree of correlation between these first principal coefficients. However, the fact that there is no degree of correlation is indicative that no measurement of x_k provides additional information that will improve our estimate of θ insofar as its impact on prediction of y_{k+1}. However, we observe that when such an analysis is done on a *per wake period* (or per phase), we do

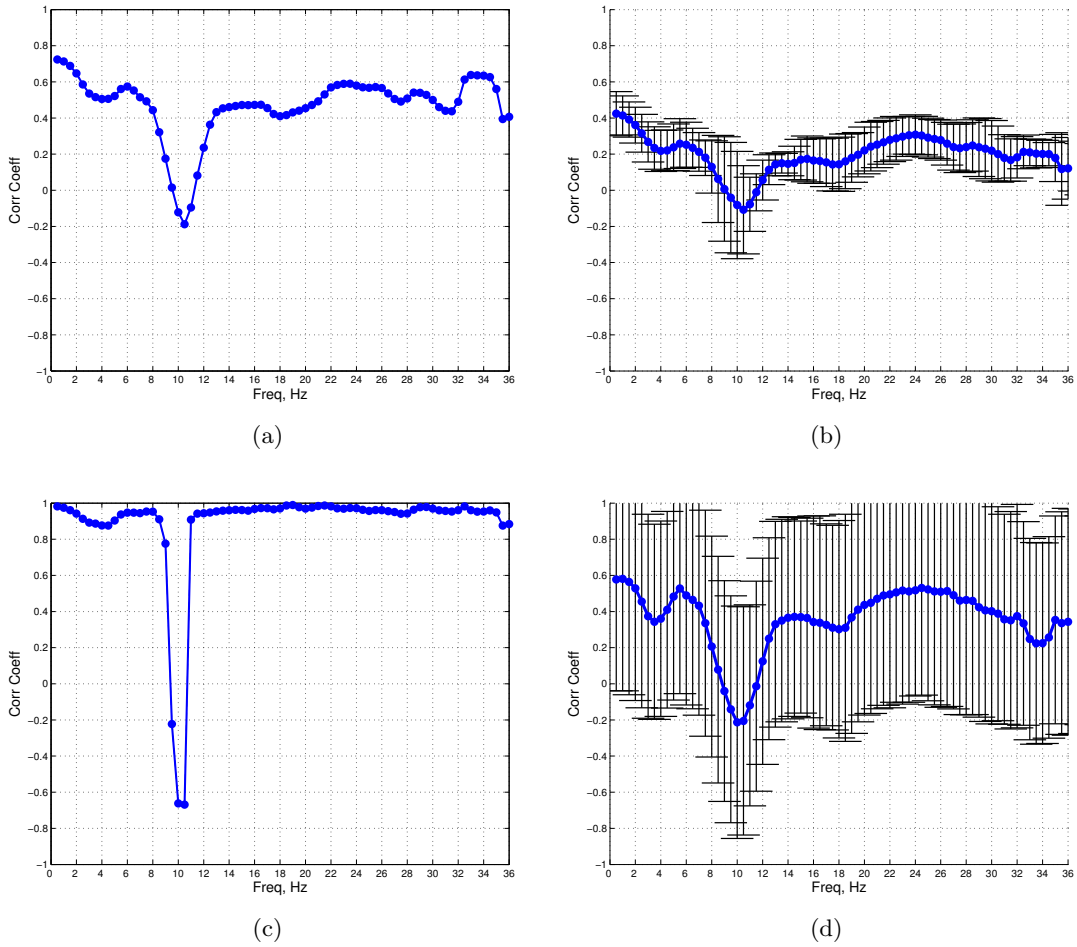


Figure 1.14: Cross-Correlation at zero log of PVT lapses with Fz-Cz spectrum at various frequencies (a) between *group mean* values (b) between *individual* values, then averaged over subjects. (c),(d) are same as (a),(b) except that values are first averaged over all phases and weeks to eliminate circadian and chronic effect. Error bars indicate variation in per-subject correlations.

observe some degree of correlation (FIG CorrWP11), Thus, within a wake period, we can make better inferences about inter-subject variations from spectral analysis than by measuring PVT alone, and hence in some cases LPM2-Phase is able to predict PVT better when making use of EEG. However, this model requires that specific circadian phase information be known, and thus is applicable only a per wake period basis, an assumption that is hard to satisfy in real world applications since it is not always possible to know an individual's circadian phase. Furthermore, the MSE is of the order of 20 lapses, which is considerably

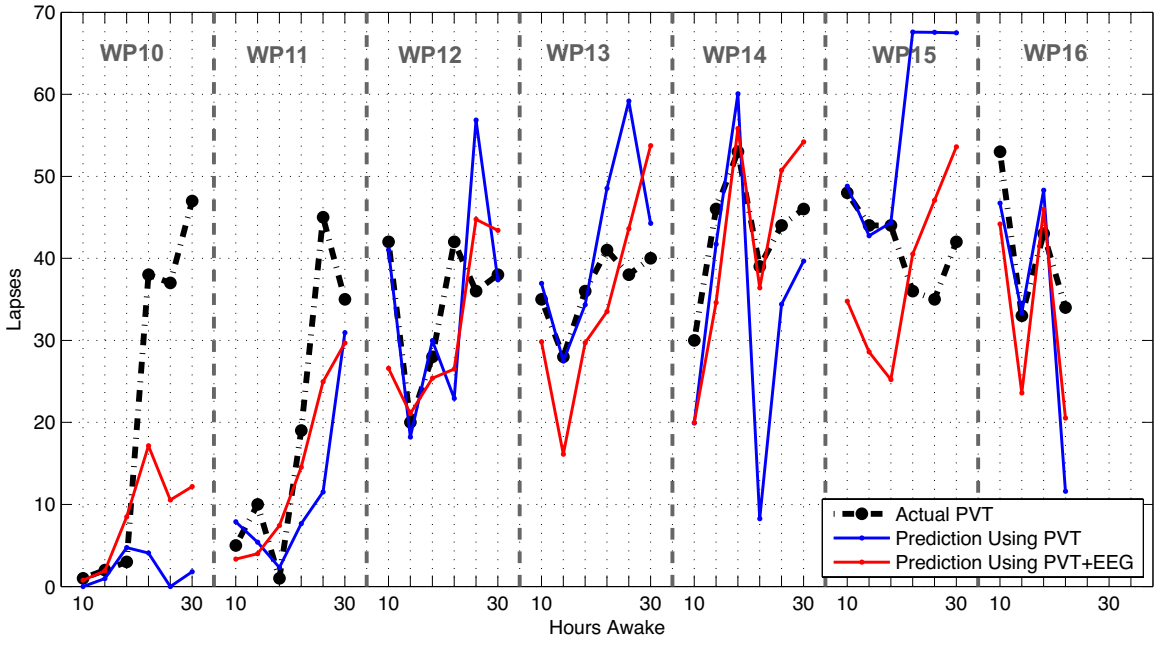


Figure 1.15: Individual PVT prediction when using an observation window of $T = 12h$. The model was estimated using data from eight subjects that was then applied to the prediction task for the ninth subject. Model is estimated *per* wake period, and the results for six wake periods (concatenated) are shown above. The PVT only estimation corresponds to using only the first term in the minimization (1.11) and the PVT+EEG based estimation corresponds to using all three terms in (1.11).

high considering the normal range of lapses (0-60).

1.4 Discussion of Results

We saw that EEG power spectrum in various bands, on a group level, exhibits specific pattern with wake time depending on chronic sleep debt and circadian phase.These patterns correlate strongly with the corresponding variation in performance measures, and thus prediction of performance based upon EEG seemed plausible. However, this correlation breaks down on a per subject level. While models of performance measures based upon inter subject patterns seem applicable to intra-subject modeling, the same does not seem true for EEG power spectrum.

There are several possible explanations as to why we didn't get the positive result we were hoping for (problem (B) vs (A) as discussed in Section 1.1):

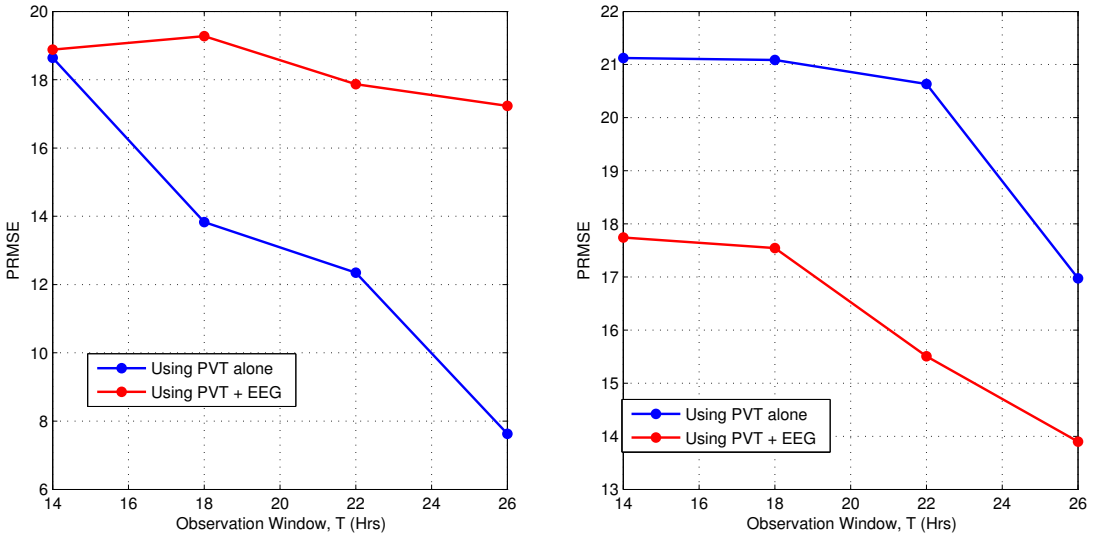


Figure 1.16: Characterization of $PRMSE$ with observation window T when using LPM2-Phase algorithm for PVT lapse prediction with and without EEG measurements. In each test, eight out of nine subjects randomly selected to used to estimate the model parameters, and prediction is tested on the remaining subject. The two cases (a), (b) above correspond to two such randomized tests.

1. **Data Quality:** The EEG data from KDT episodes was assumed to be free of artifacts. However, a re-examination of some of the data revealed us that not all artifacts were actually removed. The presence even a small number of artifacts can result in changes in spectrum that are comparable to what would be expected due to fatigue or sleep deprivation.

2. **Variable Number of Epochs:** The current manual process of artifact removal results in discarding an epoch that contains any artifact. Not only does this result in loss of possibly useful EEG data, but also the effect of variable number of epochs. We observed that anywhere from 5% to 90% of epochs were retained from trial to trial. The mean number of epochs per KDT episode was 50 with a standard deviation of 22. This makes deriving of any conclusion that compares trials based upon their spectra problematic. In fact, the variability in estimated spectrum resulting from use of variable number of epochs is of the same order as the change in spectrum that would be expected due to sleep deprivation (Fig 1.18).

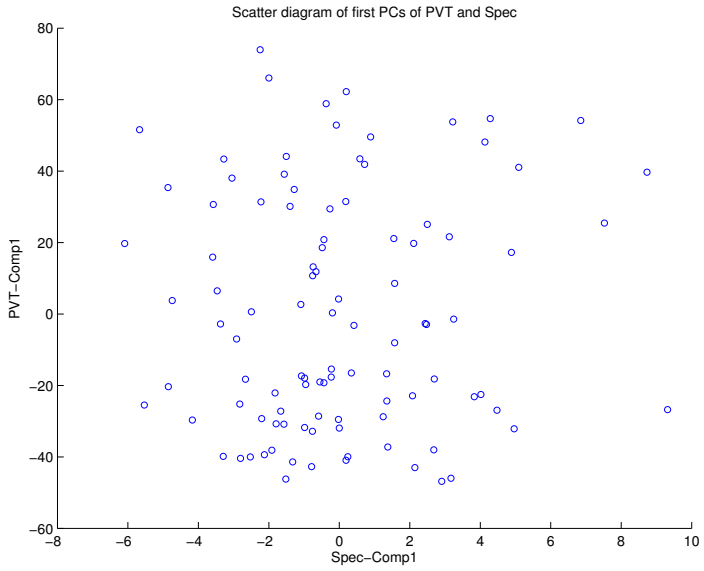
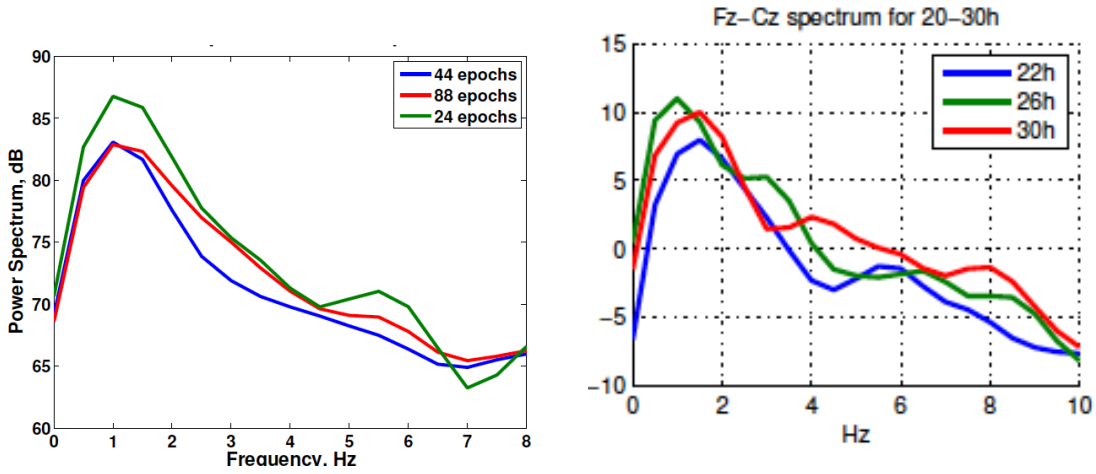


Figure 1.17: Scatter plot of the coefficient of the dominant principal component of PVT vs the coefficient of the dominant principal component of the spectrum, when the PCA is done using data across all subjects and wake periods, showing that the two are uncorrelated.

3. **Stationarity Assumption:** The inherent assumption of quasi-stationary or Gaussian nature of EEG signals in use of power spectral density as a measure of information present in EEG may not be a valid assumption.

To address the above issues, we develop methods that (i) achieve artifact removal without any data loss, resulting in the same number of epochs per trial, and (ii) extract *structural* features from an EEG signal that does not depend on the quasi-stationary assumption inherent in spectral methods. These methods are the subject of the next two chapters (Chapter 3 and Chapter 4).



(a) (b)

Figure 1.18: (a) Fz-Cz Spectral power density estimation for a trial when using different number of (randomly selected) epochs. (b) Fz-Cz Spectral power density for a subject when awake for 22,26 and 30h. In this case, each trial had the same number of epochs.

Chapter 2

Correlated Sparse Signal Recovery: Algorithm and Examples

Abstract

We address the problem of structured sparse signal recovery when only certain statistical properties describing the structure of the signal, rather than exact signal structure, are available. Such problems arise, for example, in recovery of eye movement artifacts in waking EEG recordings, where the artifactual signals exhibit a morphology (structure) that can be described by a temporal correlation amongst components in a predefined dictionary. Such signals can not be efficiently represented using standard structured models that assume a common sparsity profile of fixed groups of components. Using a Gaussian prior model that is typical of sparse Bayesian learning (SBL), we demonstrate that this type of statistical structure can be modeled using a correlation matrix in the prior joint probability distribution of coefficients, without having to assume a fixed group structure. We derive an E-M based algorithm for learning sparse coefficients in this Bayesian paradigm, and illustrate, using simple toy examples motivated by the artifact problem, how a priori statistical signal structure can be efficiently incorporated this way.

2.1 Introduction

Sparse recovery when the sparse coefficients exhibit a certain interdependency as in the example of *structured sparsity* which is useful in distinguishing two signals both of

which have sparse components but differ in component structure. While several algorithms for structured sparsity exist [18, 231, 241, 109, 77]), most are based upon *group sparsity* - that is, non-zero (sparse) coefficients are assumed to coexist in groups. Thus, a common sparsity profile across the group is assumed, and this group structure is often required to be known a priori. Modifications that allow for overlapping groups and hence a more flexible structure also exist but their implementation is often inefficient as they require remapping of the overlapping groups to a larger set of non-overlapping groups [249]. Moreover, such a priori grouping is not always possible. For example, if one is interested in making use of the temporal and spatial structure in the shape of an eye blink artifact that distinguishes it from ambient EEG, the large variety in the shapes of blinks makes such a priori grouping implausible. Use of fixed structural relationships can often lead to *overfitting* (Fig 2.1). Instead of a *fixed* structure, we would need to model this using a *statistical* structure amongst the coefficients, such as a priori correlation, which is readily incorporated as an a priori correlation amongst coefficients when using a Bayesian approach to sparse representation. While a model using correlations has previously been proposed in [202], their estimation requires a mean field approximation of the inter coefficient relationships which is not applicable to our example of blink artifacts due to the asymmetry of these relationships.

In our approach, we extend the standard Sparse Bayesian Learning model (SBL) of [234] to include correlations in a prior Gaussian distribution for the coefficients. In doing so, we represent the covariance matrix using a *separation strategy* where the correlation matrix is assumed and the variances are estimated using an E-M approach. The idea of estimating variances to determine sparsity is similar to in [234] and [245]. Our model is more general, however, and includes the cases in [234] and [245] as special cases.

Our algorithm, that we call Correlated Sparse Signal Recovery (CSSR), is developed in this part of the Chapter and illustrated using two examples motivated by EEG blink artifacts. Full description of construction and implementation of the dictionary and application of CSSR to eye blink artifacts is in Part II of the chapter.

2.2 Background

2.2.1 Sparse Bayesian Recovery

The basic model of sparse signal recovery, or compressed sensing is to recover the source vector $x \in R^M$, given the measurement vector $y \in R^N$ and a known dictionary matrix

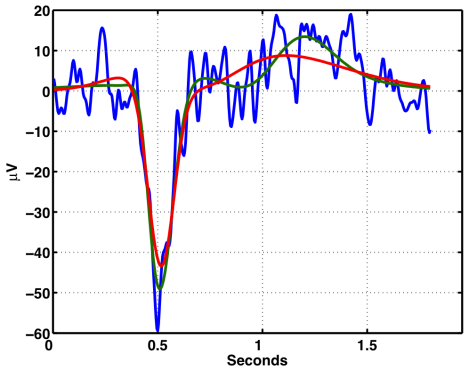


Figure 2.1: Use of standard methods of group sparsity often results in "overfitting". We are interested in isolating the artifact (red) from the raw signal (blue) but the assumption of fixed sparsity can match portions of the ambient signal (green).

$H \in R^{M \times N}$ such that x is sparse in H, which are related as

$$y = Hx + \varepsilon$$

where $\varepsilon \in R^N$ is an unknown noise vector. Typically H is overcomplete, in that $M >> N$ and that the sparsity $K = \|x\|_0 << M$. The sparse recovery problem is the inversion problem:

$$\widehat{x} = \arg\min_x \left\{ \|y - Hx\|_2^2 + \lambda \|x\|_0 \right\}$$

where λ is a scalar that controls the relative importance applied to the Euclidean error and sparseness terms. An l_1 regularized approximation is often used with $\|x\|_0$ replacing by $\|x\|_1$ for the NP-hard problem above:

$$\widehat{x} = \arg\min_x \left\{ \|y - Hx\|_2^2 + \lambda \|x\|_1 \right\} \tag{2.1}$$

The above problem can be interpreted from a Bayesian perspective: we have a prior belief that x is sparse in the basis H and the objective is to provide a posterior belief for the coefficients x given the measured data y. A widely used "sparsity promoting" prior on x is the Laplacian prior,

$$x \sim \exp(-\|x\|_1)$$

Then, if we assume $\varepsilon \sim N(0, \sigma^2 I)$ then the posterior density $x|y$ is

$$p(x|y) \sim \exp\left\{ -\frac{1}{2\sigma^2} \|y - Hx\|_2^2 \right\} \exp\{-\|x\|_1\}$$

so that the L_1 problem (2.1) is equivalent to the MAP problem

$$\widehat{x} = \arg\max_x p(x|y)$$

with $\lambda = 2\sigma^2$.

However, since the Laplacian prior is not conjugate to the Gaussian likelihood, Bayesian inference in its form above can not be carried further in closed form [115]. This problem has been addressed previously in the SBL paradigm [234] where a parameterized form of the prior is proposed:

$$p(x; \lambda_1, ...\lambda_M) = \prod_{i=1}^{M} N(x_i; 0, \lambda_i)$$

where the notation $N(\xi; \mu, \sigma^2)$ refers to a univariate Normal density in variable ξ with mean μ and variance σ^2. In [234] it was shown that when the parameters $\lambda_1, ...\lambda_M$ are estimated using E-M from the data y, several of the parameters $\lambda_k \longrightarrow 0$ and the remaining non-zero λ_k correspond to the sparse coefficients and that it provided recovery equivalent to that between L_0 and L_1. We use the above approach since it is easy to adapt to our correlation model without compromising its analytic capability. Using non-conjugate priors or hierarchical Bayesian models are analytically intractable, require MCMC type approaches [51, 74] and computationally infeasible in real time. The Bayesian approach also facilitates computation of confidence intervals for estimated coefficients.

2.2.2 Structured Sparsity

As mentioned in the Introduction, structured sparsity refers to recovery of sparse signals when the non-zero coefficients have a particular structure. The most common approach is where the signals share a common coefficient support set - an approach also known as *block sparsity* or *group sparsity*. This approach was originally developed for sensor networks and MIMO networks [18] where frequency-sparse acoustic signals recorded by an array of microphones all contain the same Fourier frequencies but vary in amplitudes and delays. In the model of [245, 246], block sparsity is addressed from a Bayesian perspective by extending the SBL model where the block structure is incorporated in the prior model for x.In this case, the space of coefficients is partitioned into L blocks, with each x_k within block l having a common variance λ_l which is then estimated using E-M. This was also extended to incorporate a correlation structure amongst a block itself, however, the assumption of common sparsity across a block was still assumed. In our algorithm described below, we relax

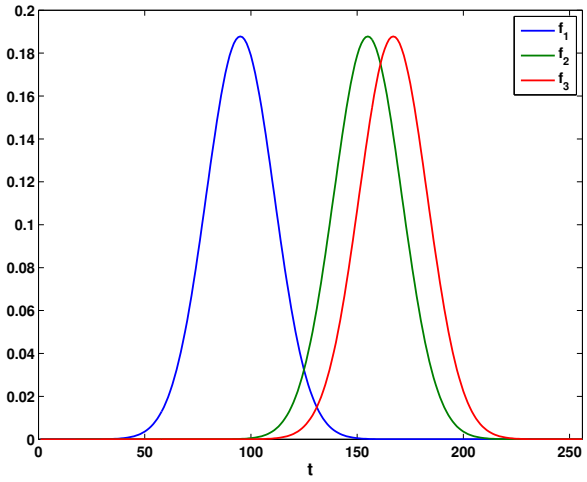


Figure 2.2: Three atoms of the dictionary used in Toy Example 1.

the assumption of common sparsity and show that this model is a special case of our more general approach.

2.3 Examples

The following two toy examples will be used to demonstrate our approach.

Example 1. A simple dictionary of three atoms is used to create a signal. We wish to recover this signal when there are two sources of noise: (1) a signal constructed from the dictionary itself and (2) Gaussian noise. The length of the signals is 256. The three atoms f_1, f_2 and f_3 of the dictionary are shifted Gaussian functions of variance of 16 with means 95, 155 and 167 (Fig 2.2). The signal to be recovered is $y_0(t) = 2f_2 - 4f_3$. We assume the noise is $y_n(t) = 2f_1 + 0.1\varepsilon$ where ε is Gaussian noise with mean zero and variance 1. The measured signal $y(t) = y_0(t) + y_n(t)$. The original signal $y_0(t)$ and measured signal $y(t)$ are shown in Fig 2.3. While simple, this toy example still illustrates a basic features of an EEG signal with a blink artifact. $y_0(t)$ can be considered to be the artifact signal that needs to be recovered, and $y_n(t)$ is the background EEG. Since $y_n(t)$ was constructed using an from the dictionary, SBL does not to recover $y_0(t)$ accurately (Fig 2.3,red). We will see that our algorithm is able to recover the original signal with sufficient accuracy.

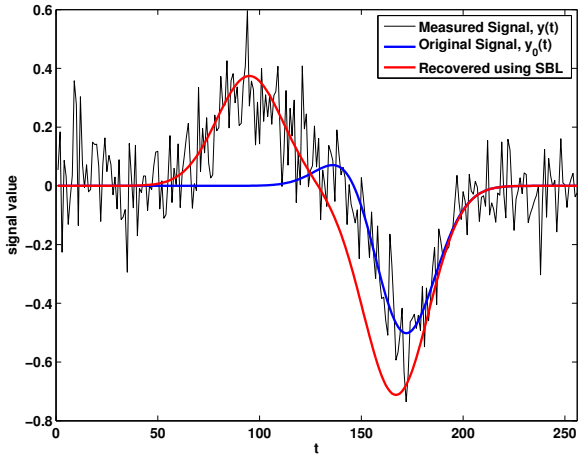


Figure 2.3: Actual and measured signals for Toy Example 1. The recovered signal using SBL is also shown to illustrate the "overfitting" issue.

Example 2. A blink artifact signal $y_0(t)$ is simulated by fitting a real 2s EEG epoch containing artifact using OMP and three components from the overcomplete blink dictionary discussed in Part II. The epoch chosen was one for which spectral distortion after $y_0(t)$ is subtracted is minimal so that $y_0(t)$ mimics a real blink as closely as possible. A 2 second epoch of artifact-free EEG signal $y_n(t)$ is then added to create the measured signal $y(t) = y_0(t) + \lambda y_n(t)$, where the value λ is adjusted for desired SNR (Fig 2.4). By synthesizing an EEG signal this way, we are able to quantify the performance of our algorithm. A subdictionary H (of the full dictionary) consisting of 29 elements, with Gaussian elements at fixed scale with translations at intervals of 8 units, or 8/256 s (sampling rate is 256Hz) is used to test various sparse recovery algorithms. SBL is not able to recovery $y_0(t)$ accurately (Figure 2.4) since in addition to $y_0(t)$, the signal $y_n(t)$ also contains significant sparse elements.. Note that to make the example more realistic we constructed y_0 with a much larger dictionary than H - that is, not all elements used to construct y_0 are contained in H.

2.4 Algorithm Development

2.4.1 Model

We wish to recover source vector $x \in R^M$, given the measurement vector $y \in R^N$ and a known dictionary matrix $H \in R^{M \times N}$ such that x is sparse in H, which are related as

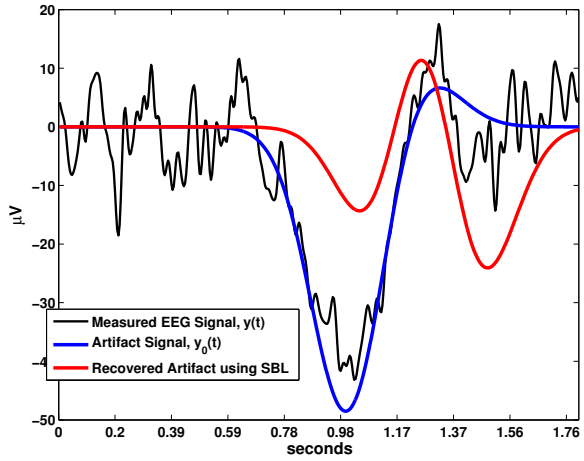


Figure 2.4: Actual and measured signals for Toy Example 2. The recovered signal using SBL is also shown to illustrate the "overfitting" issue.

in:

$$y = Hx + \varepsilon$$

where $\varepsilon \in R^N$ is Gaussian and iid, i.e. $\varepsilon \sim N(0, \sigma^2 I_N)$ with σ^2 being the noise variance and $I_N \in R^{N \times N}$ is the identity matrix. The prior density on coefficients x is assumed zero-mean multivariate normal i.e.$x \sim N(0, C)$ where the covariance matrix $C \in R^{M \times M}$ is symmetric positive semi-definite. σ^2 and C are considered hyperparameters that are estimated from the data. Estimating C and σ^2 typically will result in overfitting due to the large number of parameters to be learned. Hence some structure for C needs to be assumed. Assuming C to be a diagonal matrix $\Gamma = diag\{\gamma_1, ...\gamma_M\}$ results in the SBL model [234]. If C is assumed to have a block diagonal form

$$C = \begin{bmatrix} \gamma_1 B_1 & & \\ & ... & \\ & & \gamma_l B_L \end{bmatrix}$$

where $B_1, ...B_L$ are symmetric positive semidefinite matrices then we get the Block Sparse Bayesian Learning (BSBL) model [245], in which block l represented by $\gamma_l B_l$ is assumed to have a common sparsity level. Usually this model also results in overfitting so typically one assumes each block has fixed size $n = M/L$ and $C = \Gamma \otimes B$ where $B \in R^{n \times n}$ and $\Gamma = diag(\gamma_1, ...\gamma_L)$. However, as mentioned previously, these models are inadequate for our purpose because of the common sparsity assumption.

We assume, instead, that the coherence structure can be captured by a *correlation*

matrix $R \in R^{M\times M}$. Writing $\Lambda = diag(\lambda_1, ...\lambda_M)$ where λ_k is the standard deviation of component k, we use the following decomposition of the covariance matrix:

$$C = \Lambda R \Lambda$$

The above decomposition, sometimes also called *separation strategy* [20], allows us to express our prior beliefs about the temporal dependence between the sparse components of x via the correlation matrix R and we let Λ be estimated from the data. As in other sparse Bayesian models, if x is sparse then most λ_i are close to zero. In the algorithm below we assume that R is known a priori. As will be discussed in the examples below, a complete knowledge of R is often difficult and some parameterization will be necessary.

We are interested in finding an estimate $\widehat{x}$ of the sparse signal x. The MAP estimate is (where we use notation $p(\xi;\mu)$ to represent the probability distribution of variable ξ given parameters μ):

$$\widehat{x} = \max_x p(x|y;\sigma^2,\Gamma)$$

Since $\varepsilon \sim N(0,\sigma^2 I)$ the conditional density of $y|x$ is $p(y|x) = N(Hx,\sigma^2 I)$. From the conjugate prior density $x \sim N(0,\Lambda R\Lambda)$ the posterior density of x can be shown to be $p(x|y) = N(\mu_{x|y},\Sigma_{x|y})$ where

$$\begin{aligned} \mu_{x|y} &= \frac{1}{\sigma^2}\Sigma_{x|y}H^T y \\ \Sigma_{x|y} &= (\Lambda^{-1}R^{-1}\Lambda^{-1} + \frac{1}{\sigma^2}H^T H)^{-1} \end{aligned} \tag{2.2}$$

so that the MAP estimate $\widehat{x}$ is given by

$$\widehat{x} = \mu_{x|y} = \frac{1}{\sigma^2}\Sigma_{x|y}H^T y$$

The problems is now to estimate the parameters σ^2 and Λ.

2.4.2 E-M Based Estimation of σ^2 and Λ

From the conjugacy property above the marginal density of y can be derived to be $p(y;\sigma^2,\Lambda) = N(0,\Sigma_y)$ where

$$\Sigma_y = H\Lambda R\Lambda^T H^T + \sigma^2 I$$

and the likelihood of y is

$$L(y;\sigma^2,\Lambda) = \log p(y) = -\frac{1}{2}(\log|\Sigma_y| + y^T\Sigma_y^{-1}y) \tag{2.3}$$

from which the the ML estimates Λ^*, σ^* of Λ, σ can be computed:

$$(\Lambda^*, \sigma^*) = \arg\max_{\Lambda,\sigma} L(y; \sigma^2, \Lambda) \tag{2.4}$$

Instead of maximizing the likelihood function directly, we use the E-M algorithm to compute the ML estimates by treating x as a hidden variable.

E-M Steps for estimating Λ

E-Step. We form the Q function, at the t^{th} iteration

$$\begin{aligned} Q(\Lambda, \Lambda^{(t)}) &= E_{x|y,\Lambda^{(t)}} \log p(y, x; \Lambda) \\ &= E_{x|y,\Lambda^{(t)}} \log p(y|x; \Lambda) p(x; \Lambda) \\ &= E_{x|y,\Lambda^{(t)}} \log N(y; Hx, \sigma^2 I) + E_{x|y,\Lambda^{(t)}} \log N(x; 0, \Lambda R \Lambda) \end{aligned}$$

M-step. If $\Lambda^{(t)}$ refers to the value of Λ at the t^{th} iteration then

$$\begin{aligned} \Lambda^{(t+1)} &= \arg\max_{\Lambda} Q(\Lambda, \Lambda^{(t)}) \\ &= \arg\max_{\Lambda} E_{x|y,\Lambda^{(t)}} \log N(y; Hx, \sigma^2 I) + E_{x|y,\Lambda^{(t)}} \log N(x; 0, \Lambda R \Lambda) \\ &= \arg\max_{\Lambda} E_{x|y,\Lambda^{(t)}} \log N(x; 0, \Lambda R \Lambda) \\ &= \arg\max_{\Lambda} E_{x|y,\Lambda^{(t)}} \log \left((2\pi)^{-M/2} |\Lambda R \Lambda|^{-1} \exp\left\{ -\frac{1}{2} x^T (\Lambda R \Lambda)^{-1} x \right\} \right) \\ &= \arg\min_{\Lambda} \log |\Lambda R \Lambda| + E_{x|y,\Lambda^{(t)}} (x^T \Lambda^{-1} R^{-1} \Lambda^{-1} x) \\ &= \arg\min_{\Lambda} 2 \log |\Lambda| + tr(\Lambda^{-1} R^{-1} \Lambda^{-1} S^{(t)}) \end{aligned}$$

where

$$S^{(t)} = \Sigma_{x|y}^{(t)} + \mu_{x|y}^{(t)} \mu_{x|y}^{(t)T}$$

and values $\Sigma_{x|y}^{(t)}$ and $\mu_{x|y}^{(t)}$ are obtained by using the value $\Lambda^{(t)}$ in (2.2). In derivation of the last step above, we have used the fact that If $A = A^T$ and ξ is a random variable with $E(x) = \mu, cov(X) = \Sigma$ then $E(x^T A x) = tr(A(\Sigma_x + \mu\mu^T))$. Writing $\Lambda^{-1} = L$ we have the M-step as

$$L^{(t)} = \arg\min_{L} \left(-2 \log |L| + tr(L R^{-1} L S_t) \right)$$

Writing the diagonal matrix L in terms of the vector $l = diag(L) = [l_1 \; l_2 \; ... \; l_M]^T$, using the identity for a diagonal matrix L and arbitrary matrices A, B

$$tr(LALB) = l(A \circ B^T) l^T$$

(where (where $\circ$ denotes Schur product), since $S^{(t)}$ is symmetric, we get the M-step

$$\begin{aligned} l^{(t+1)} &= \arg\min_l \left(\left(-2\sum_{i=1}^{M} \log l_i \right) + l(R^{-1} \circ S^{(t)}) l^T \right) \\ &= \arg\min_l \left(2e^T(\log l) + l(R^{-1} \circ S^{(t)}) l^T \right) \end{aligned} \tag{2.5}$$

where $e = [1\ 1\ ...\ 1]^T \in R^M$ and $\log(l)$ is the elementwise logarithm of the vector l.

The minimization step above can be done using gradient descent, since the gradient of the function f above to be minimized can be computed as ($1./l$ denotes elementwise quotient of l):

$$\begin{aligned} \nabla f &= 2(1./l) - 2(R^{-1} \circ S^{(t)}) l \\ &= diag(-2L^{-1} + 2S^{(t)} L R^{-1}) \end{aligned} \tag{2.6}$$

It should be noted that setting $\nabla f = 0$ for the special cases of $R = I_M$ and $R = \Gamma \otimes B$ results in the iteration formulae in SBL and BSBL [245] respectively.

E-M Steps for estimating σ^2

To simplify notation, we write $\lambda = \sigma^2$.

E-Step. We form the Q function, at the t^{th} iteration

$$\begin{aligned} Q(\lambda, \lambda^{(t)}) &= E_{x|y,\lambda^{(t)}} \log p(y, x; \lambda) \\ &= E_{x|y,\lambda^{(t)}} \log p(y|x; \lambda) p(x; \lambda) \\ & \quad E_{x|y,\lambda^{(t)}} \log N(y; Hx, \lambda I) + E_{x|y,\lambda^{(t)}} \log N(x; 0, \Lambda R \Lambda) \end{aligned}$$

M-Step.

$$\begin{aligned} \lambda^{(t+1)} &= \arg\max_\lambda Q(\lambda, \lambda^{(t)}) \\ &= \arg\max_\lambda E_{x|y,\lambda^{(t)}} \log N(y; Hx, \lambda I) + E_{x|y,\lambda^{(t)}} \log N(x; 0, \Lambda R \Lambda) \\ &= \arg\max_\lambda E_{x|y,\lambda^{(t)}} \log N(y; Hx, \lambda I) \\ &= \arg\max_\lambda E_{x|y,\lambda^{(t)}} \log(2\pi\lambda)^{-N/2} \exp\{-\frac{1}{2\lambda} \|y - Hx\|^2\} \\ &= \arg\min_\lambda \left(N \log \lambda + \frac{1}{\lambda} E_{x|y,\lambda^{(t)}} \|y - Hx\|^2 \right) \end{aligned}$$

To compute the second term we write, where $\mu_{x|y}$ is as in (2.2),

$$\|y - Hx\|^2 = \left\| y - H\mu_{x|y} + H(x - \mu_{x|y}) \right\|$$

Since x, y are independent under the distribution $x|y$, we have $E_{x|y,\lambda^{(t)}}\left[(y-H\mu)H(x-\mu)\right] = 0$ so that

$$\begin{aligned} E_{x|y,\lambda^{(t)}}\left\|y-Hx\right\|^2 &= E_{x|y,\lambda^{(t)}}\left\|y-H\mu_{x|y}\right\|^2 + E_{x|y,\lambda^{(t)}}H(x-\mu_{x|y}) \\ &= \left\|y-H\mu_{x|y}\right\|^2 + E_{x|y,\lambda^{(t)}}(x-\mu_{x|y})^T H^T H(x-\mu_{x|y}^T) \\ &= \left\|y-H\mu_{x|y}\right\|^2 + tr(H^T H\Sigma_{x|y}) \end{aligned}$$

where for the latter result we have used the fact that for a random variable ξ with zero mean and variance Σ if $A = A^T$ then $E(xAx^T) = tr(A\Sigma)$. Thus the M-step becomes

$$\lambda^{(t+1)} = \arg\min_{\lambda} M\log\lambda + \frac{1}{\lambda}\left(\left\|y-H\mu_{x|y}^{(t)}\right\|^2 + tr(H^T H\Sigma_{x|y}^{(t)})\right)$$

where we have emphasized the $\lambda^{(t)}$ dependence of $\mu_{x|y}$ and $\Sigma_{x|y}$. Since the term in the parentheses does not depend on λ we can explicitly compute this minimum to be

$$\lambda^{(t+1)} = \frac{1}{N}\left(\left\|y-H\mu_{x|y}^{(t)}\right\|^2 + tr(H^T H\Sigma_{x|y}^{(t)})\right)$$

The term in the parenthesis can be further simplified. Setting $C = \Lambda R\Lambda$, we have from (2.2)

$$H^T H = \lambda\left(\Sigma_{x|y}^{-1} - C^{-1}\right)$$

so that we have the iteration formula

$$\begin{aligned} \lambda^{(t+1)} &= \frac{1}{N}\left(\left\|y-H\mu_{x|y}^{(t)}\right\|^2 + \lambda^{(t)} tr(I-\Sigma_{x|y}C^{-1})\right) \\ &= \frac{1}{N}\left(\left\|y-H\mu_{x|y}^{(t)}\right\|^2 + \lambda^{(t)}\left(M - tr(\Sigma_{x|y}^{(t)}C^{-1})\right)\right) \end{aligned} \tag{2.7}$$

2.4.3 Algorithm Steps

The above analysis motivates an iterative algorithm using the steps (2.2), (2.5) and (2.7) till convergence is reached.The minimization in (2.5) is attained using gradient descent, where the gradient is given by (2.6). The largest K coefficients are retained. Most of the entries in the estimated Λ approach zero, and a thresholding technique can be used instead of assuming a fixed sparsity K. Once the hyperparameters $\widehat{\lambda}, \widehat{\lambda}_1, \widehat{\lambda}_2, ..., \widehat{\lambda}_M$ are estimated, the MAP estimate $\widehat{x}$ is given by (2.2), that is:

$$\widehat{x} = \frac{1}{\widehat{\lambda}}\widehat{\Sigma}_{x|y}H^T y \tag{2.8}$$

where

$$\widehat{\Sigma}_{x|y} = (\widehat{\Lambda}^{-1}R^{-1}\widehat{\Lambda}^{-1} + \frac{1}{\lambda}H^TH)^{-1}$$

where $\widehat{\Lambda} = diag(\widehat{\lambda}_1, \widehat{\lambda}_2, ..., \widehat{\lambda}_M)$. The corresponding estimate $\widehat{y}$ of the recovered signal is then given by

$$\widehat{y} = H\widehat{x}$$

The computation of $\Sigma_{x|y}^{-1}$ in (2.2) above can be made more efficient by using the the matrix inversion lemma (here $C = \Lambda R \Lambda$) thereby reducing the inversion of an $M \times M$ matrix to the inversion of an $N \times N$ matrix (note $M << N$). We call our algorithm **Correlated Signal Sparse Recovery** (CSSR).

2.5 Results

2.5.1 CSSR on Example 1

For the example described earlier, we have $N = 256$ and $M = 3$, and set $K = 2$ and $y = y_0 + y_n$ where $y_0 = Hx_0, y_n = Hx_n + 0.1\varepsilon$ with $H = \begin{bmatrix} f_1 & f_2 & f_3 \end{bmatrix}, x_0 = \begin{bmatrix} 0 & 2 & -4 \end{bmatrix}^T$ and $x_n = \begin{bmatrix} 2 & 0 & 0 \end{bmatrix}^T$.We let R be the parameterized vector

$$R = \begin{bmatrix} 1 & \rho & \mu \\ \rho & 1 & \gamma \\ \mu & \gamma & 1 \end{bmatrix}$$

where ρ, μ, γ are such that R is positive definite. The CSSR algorithm was used to estimate the recovered signal $\widehat{y}$ for various values of the parameters ρ, μ, γ and the mean square error $MSE = \|y_0 - \widehat{y}\|_2$, is used to compare the algorithm recovery performance for various parameter combinations. Note that the parameter values $\rho = \mu = \gamma = 0$ corresponds to SBL. We call the space $P = \{(\rho, \mu, \gamma) : -1 \leq \rho \leq 1, -1 \leq \mu \leq 1, -1 \leq \gamma \leq 1, R > 0\}$.

MSE as a function on P is depicted in Figures 2.5 and 2.6. For $\rho = \mu = \gamma = 0$ the MSE is 2.67 (corresponding to SBL). However, we notice on the entire $P-$space only two distinct values, MSE=2.67 and MSE=0.17 are observed. The variation of MSE on P is also shown using a different view in Figures 2.7 and 2.8. We call these two spaces P_A and P_B respectively. That is, MSE for $(\rho, \mu, \gamma) \in P_A$ is 2.67 and MSE for $(\rho, \mu, \gamma) \in P_B$ is 0.17. Note that the parameters for SBL, corresponding to $\rho = \mu = \gamma = 0$ are in P_A.

The results of the recovered signal are shown for the two cases where $R \in P_A$ and $R \in P_B$ are shown in Figure 2.9. The recovery for the latter case is almost exact, compared

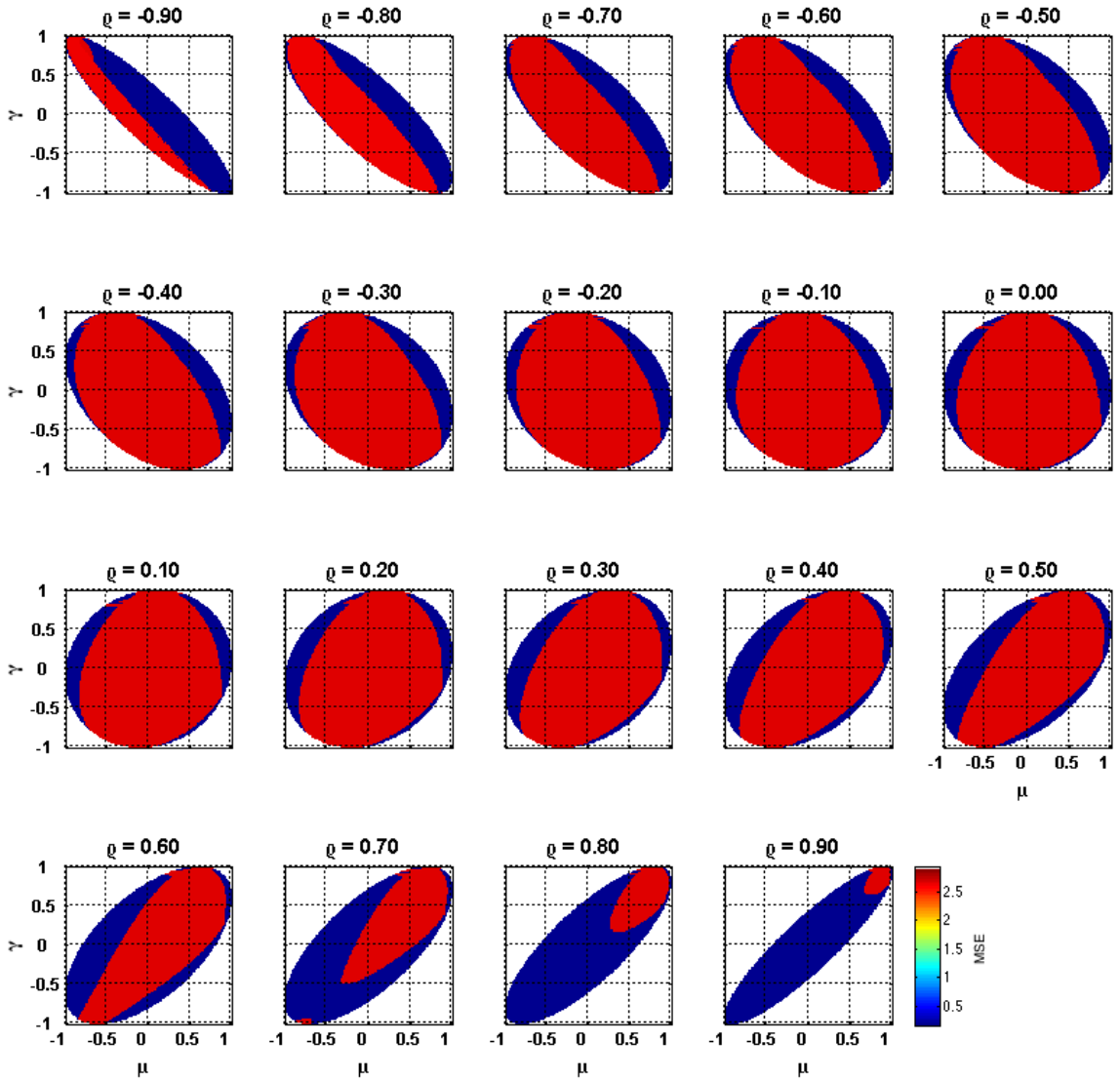


Figure 2.5: Variation of MSE with μ and γ for various values of ρ in Example 1. Note that only two distinct values of MSE are observed on the entire (ρ, μ, γ) space, MSE=2.67 (red), MSE=0.17 (blue).

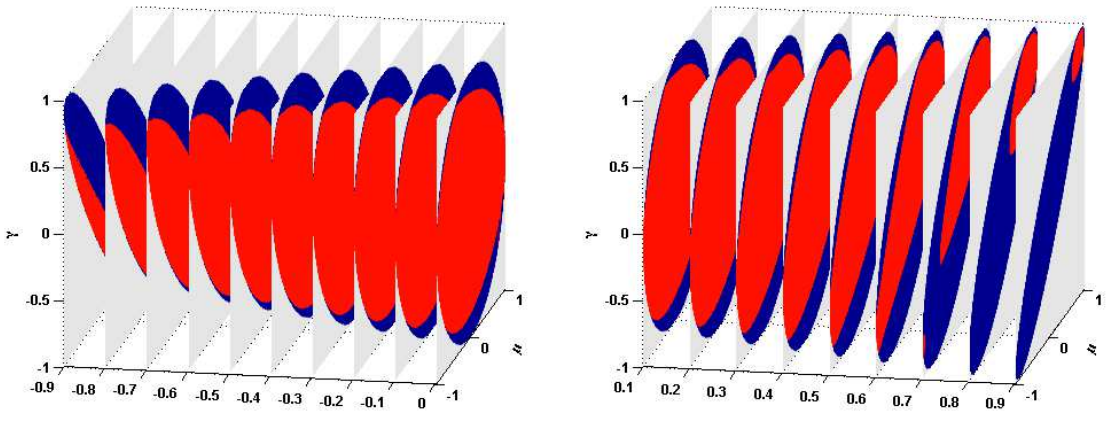


Figure 2.6: Same as 2.5 except shown as a slice plot. Value of ρ is on the x-axis.

to the former case which also includes SBL as a special case. When $R \in P_A$, the two dominant recovered components are f_1 and f_3 whereas when $R \in P_B$ the components f_2, f_3 are recovered. In the absence of a priori correlation information, or with improper a priori correlation as in P_A, the likelihood (2.3) is maximized when the variance $\lambda_1^2 > \lambda_2^2$ (as before, λ_k is the standard deviation of component x_k), which results in the selection of component f_1 selected over f_2, whereas with proper a priori correlation as in P_B the likelihood is maximized with $\lambda_2^2 > \lambda_1^2$ leading to f_2 being selected over f_1. The noise in this example (the left most hump) is such that, in the absence of a priori correlation, the measured signal is more likely if $|x_2|$ is significant. However, the presence of a priori correlation (in P_B) negates this likelihood.

This example, though somewhat simple, illustrates our basic hypothesis that the use of a priori correlations can improve structured signal recovery.

2.5.2 CSSR on Example 2

For this example, we use a signal of length $N = 461$, a dictionary H with $M = 29$ elements and sparsity $K = 3$ since y_0 is constructed was constructed using three sparse coefficients. Note, however, to make the example more realistic, we constructed y_0 with a much denser dictionary (i.e., not all elements used to construct y_0 are contained in our dictionary H). The results of recovering $\widehat{y_0}$ based on SBL and CSBL using $R = R_1$ (specified below) for two levels of $\lambda = 0.75$ and $\lambda = 0.375$ are shown in Fig 2.10. The values of PMSE,

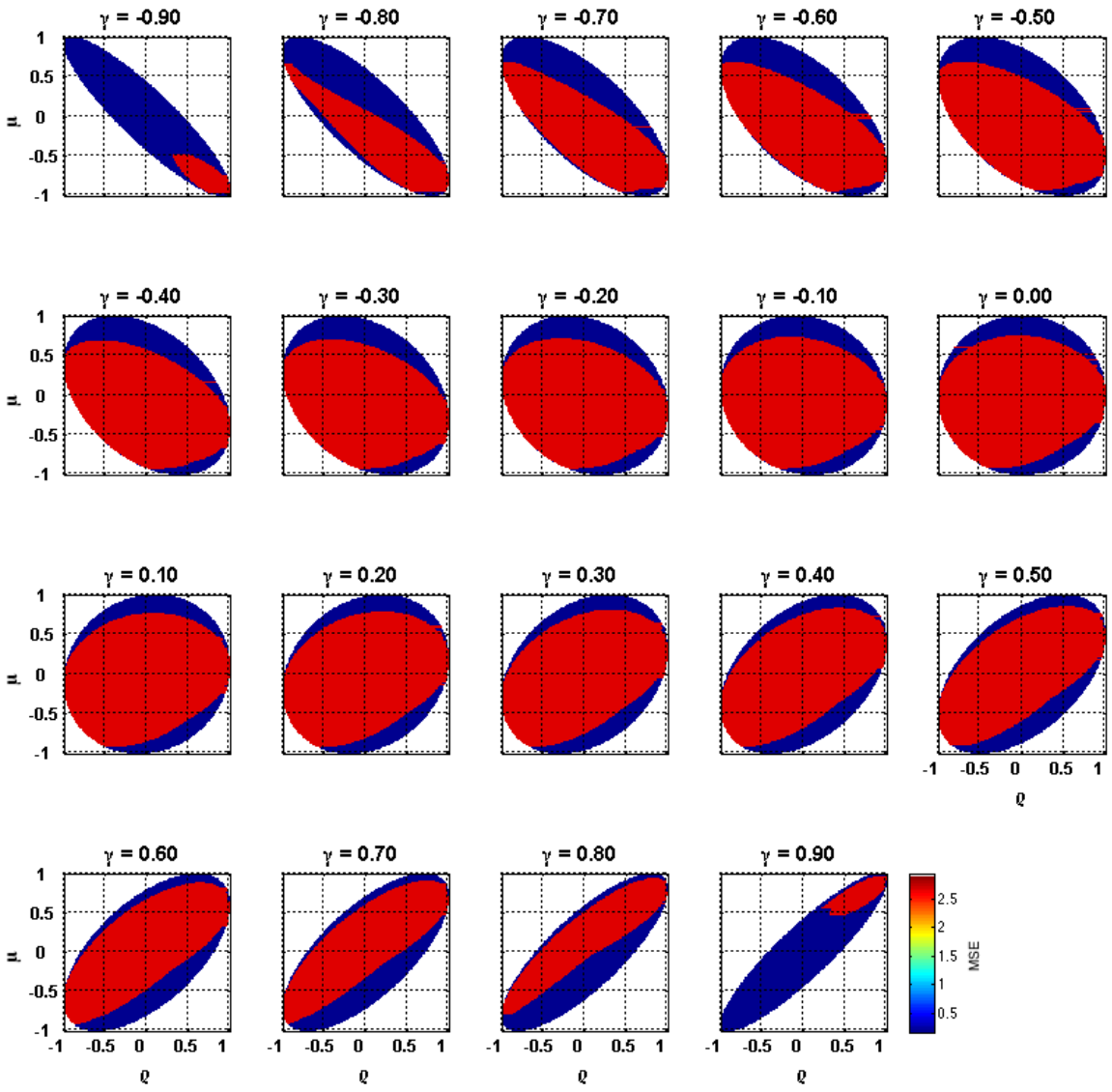


Figure 2.7: Variation of MSE with ρ and μ for various values of γ in Example 1. Note that only two distinct values of MSE are observed on the entire (ρ, μ, γ) space, MSE=2.67 (red), MSE=0.17 (blue).

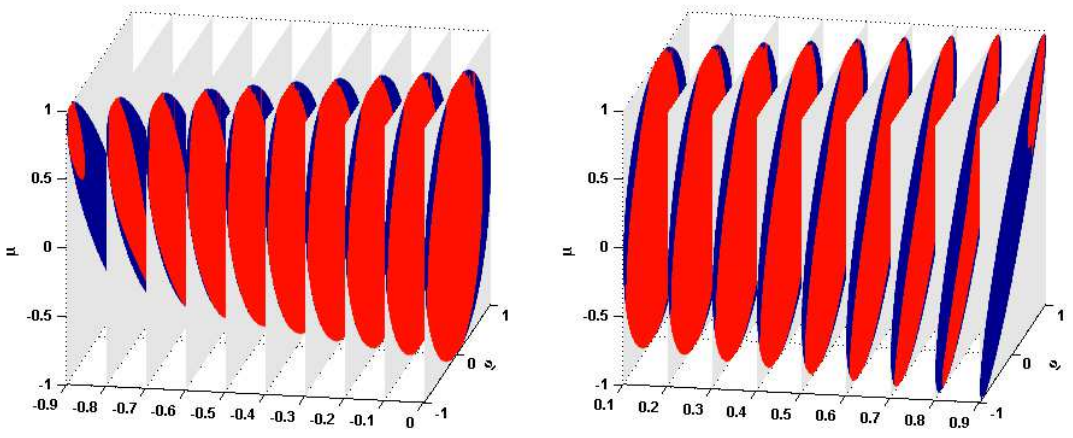


Figure 2.8: Same as 2.7 except shown as a slice plot. Value of γ is on the x-axis.

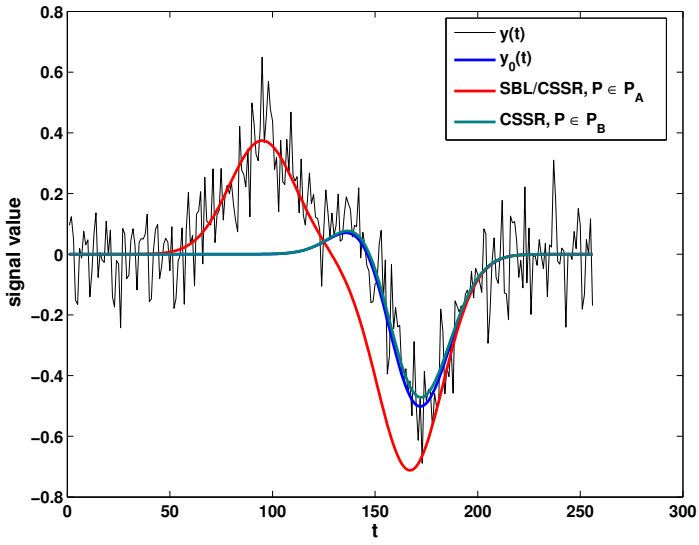


Figure 2.9: Comparison of signal recovery for Example 1 using CSSR with $R \in P_A$ (includes SBL) and $R \in P_B$. A properly chosen correlation matrix can result in almost perfect signal recovery in the presence of Gaussian noise as well as structured noise.

Table 2.1: PMSE values for Example 2

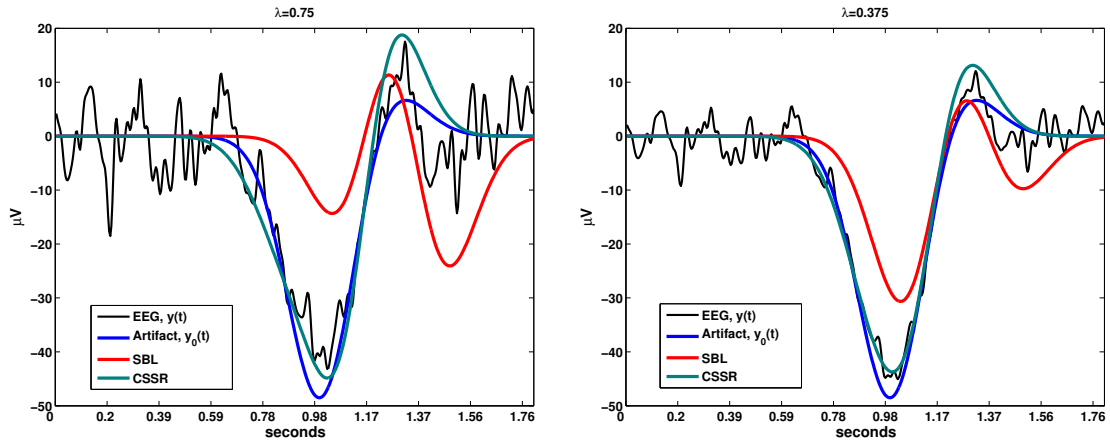
	$\lambda = 0.75$	$\lambda = 0.375$
SBL	91.60	49.23
CSSR	21.60	14.65

where PMSE (percentage MSE) is defined below, are shown in Table 2.1.

$$PMSE = \frac{\|y_0 - \widehat{y}\|_2}{\|y_0\|_2} \times 100$$

Motivated by the results from Example 1 we chose R_1 in Example 2 based on the following logic. Without any correlations, SBL matches signal components which are sparse in H, including the ones to the right of but not part of the actual blink. However, if we specify a positive correlation amongst dictionary elements corresponding to translation levels past 1.25 seconds, those components become insignificant when maximizing the likelihood (2.3). Accordingly we set R_1 with a positive correlation of 0.9 amongst elements of H corresponding to translation levels past 1.25 seconds (this is translation level of 1.25 x 256 = 320 units). While this choice was made heuristically, it illustrates our hypothesis that specifying a priori correlations can favor selection of actual signal components over noisy components that are also sparse in the chosen dictionary.

I



(a) (b)

Figure 2.10: Comparison of signal recovery for Example 2 using SBL and CSSR with $R = R_1$ for (a) $\lambda = 0.75$ and (b) $\lambda = 0.375$

2.6 Discussion and Conclusions

We have shown that the proper use of a correlation in the prior joint probability density of coefficients can improve Bayesian sparse recovery of a noisy signal when both the signal and some noise components are sparse in a chosen dictionary, but exhibit different temporal relationships amongst each other. The differing temporal relationships is captured in the a priori correlation structure. Using two simple examples, both motivated by the application of sparse representation of blink artifacts in EEG signals, we have successfully demonstrated that proper choice of correlation matrix in a Gaussian model can lead to discrimination between noisy and signal components. Our approach is an alternative to standard group sparsity models of structured compressed sensing, where a common sparsity profile is assumed - an assumption that is limited where the structure can be only specified statistically rather than exactly, such as EEG artifacts.

As in other standard sparse Bayesian recovery approaches, the prior variances of the coefficients are estimated using E-M. We derived the E-M update steps (2.2,2.5,2.7) when a prior correlation R is assumed amongst the sparse coefficients. Convergence of the algorithm is guaranteed from E-M convergence properties. However, several issues still need to be addressed: E-M convergence can be slow, and a gradient based approach to achieve the ML optimization (2.4) can yield faster convergence. An approach similar to that of [213] can be used. In our implementation, need to provide the correlation matrix R a priori. The choice of this can be difficult and, as shown in Example 1, counterintuitive. A better way to estimate R needs to be developed, preferably from the data itself. Since this algorithm is applied on streaming data (e.g. EEG), the structure of R can be learnt over multiple samples of data. An alternative would be to use a Markov model to model the relationahip amongst coefficients instead. A rigorous analysis of the algorithm is also currently lacking, including precise conditions when our algorithm will perform better recovery than standard SBL.

Our approach can also be easily extended to incorporate correlation amongst different channels, thereby providing an alternate model for the multiple measurement vector problem. In this case the spatial correlation would be modeled in addition to temporal correlation in the a priori specification of R. In Part II of this chapter, the application of the above approach to real-time EEG signals and artifact elimination will be presented.

Chapter 3

Correlated Sparse Signal Recovery: Application to EEG Denoising

3.1 Introduction and Motivation

Waking EEG is a physiological marker of vigilance [82, 228, 11] that can be used for automatic detection of times of decreased performance in safety sensitive environments such as driving [149] and medical practice [154]. Several methods of EEG spectral [138],[208],[176] [229] and complexity [46, 123, 83, 42, 33] analysis for fatigue detection have limited applicability in real-time environments as they are based upon a large number (e.g. 32-64) of EEG channels. Methods using few EEG channels [125, 8, 134, 209] and some even permitting wireless streaming to smartphones [147, 146, 165] promise practical non-invasive solutions but they either suffer from low accuracy or require additional signal inputs such as EOG, ECG, EMG or PPG to maintain a high accuracy rate. Furthermore, most of these analyses were done assuming that the EEG had been cleaned of artifacts. The dominating presence of artifacts in real data make predictions from these algorithms virtually unreliable [39]. The deleterious impact of artifacts that is exacerbated in ambulatory environments [81] is a serious obstacle in the field of mobile EEG monitoring.

State-of-the art techniques for removal of artifacts have several pitfalls [124]. Semi-manual methods require trained staff and are expensive. Simplistic filtering or thresholding techniques are unreliable. Both these methods can result in up to 90% data loss [120] since most methods require discarding of the entire epoch of data even if a small section contains the artifact. Artifact subtraction methods using regression [197], ICA [67] or PCA [195] require

lengthy and high density (32+ channel) recordings or require some manual intervention, thus limiting their applicability for real-time and non-invasive monitoring. Furthermore, selective subtraction of components can result in corruption of the desired EEG signal. We have developed a novel technique using sparse coding to identify and subtract certain artifact types such as eye movements from EEG. This was motivated by the observation that recorded cortical EEG signals can be considered to be the projection of the sum total of the electrical field from entire cortical neural activity whereas signals such as those from eye movements and ECG, being the result of transient electric fields from localized dipole sources exhibit temporal and spatial structure. Thus it is plausible that artifact signals are sparse in an appropriate dictionary whereas cortical EEG signals are not. This hypothesis was tested by manually extracting several blink artifact signals from real EEG recordings, and were shown to be indeed sparse in a dictionary comprising of skew Gaussian shaped elements. However, use of standard sparse recovery methods such as orthogonal matching pursuit (OMP) or sparse Bayesian learning (SBL), and their extension to multiple channels - joint matching pursuit (JMP) and block sparse Bayesian learning (BSBL) - when applied to selectively fit eye blink artifacts resulted in significant spectral distortion of the recovered EEG signal. The reason for this is that some components of non-artifactual EEG can also be sparse in an artifact dictionary. In the first part of this chapter, we developed a *structured sparsity* method called CSSR which makes use of temporal and spatial correlation to define artifact structure in addition to shape morphology. In this part of the chapter, we propose CSSR as a tool for near real-time automated artifact elimination in low density EEG with minimal data loss. We demonstrate that CSSR can solve the aforementioned problem of corruption resulting from overfitting when using standard sparse recovery methods.

3.2 Methods

3.2.1 Data (Experiments)

Six-channel (Fz, C3, Cz, C4, Pz, Oz) EEG and two-channel (VEOG, HEOG) EOG recordings of healthy awake individuals during the Karolinska Drowsiness Test (KDT) and

Psychomotor Vigilance Test (PVT) performance testing of a chronic sleep restriction inpatient protocol (T20CSR) were used as data for testing and parameter validation of our algorithms. Two-second (2s) epochs containing artifacts during the KDT portions of these recordings were identified and marked manually by a Registered Polysomnographic Technician (RPSGT). Two such recordings, each having KDT portions approximately 14 minutes long, were used for blink shape extraction for dictionary validation. Four recordings, with a total of 1.45 hours of KDT portions and a total of 0.83 hours of PVT portions were used to test the artifact extraction algorithms.

3.2.2 Blink Extraction and Modeling

Blink Shape Extraction. For each RPSGT-identified artifact, both the (i) 2s epoch containing the artifact and (ii) beginning/end of the artifact are marked by the RPSGT. However, at times the 2s may contain data more than just the artifact, and at times the segment between the beginning and end is too short to reflect the entire temporal characteristics of an artifact. Typically, a blink will result in a slight upward deflection, followed by a strong downward deflection and a subsequent short upward deflection in the frontal electrodes, but the precise artifact markers delineated by the RPSGT contain the strong deflection (the *main lobe)* but do not always contain the short leading and trailing deflections (the *side lobes*) . We thus use the following procedure to be able to fully extract the electrode deflections resulting from a blink:

1. Using heuristic criteria such as peak amplitude, 2s epochs (as identified by RPSGT) containing potential blinks were identified, and those confirmed to be blinks by visual inspection retained. Epochs were extended beyond the 2s boundary if the signal did not reach baseline at the boundaries.

2. Using a peak detection procedure, all episodes containing multiple blinks are eliminated, all peaks are aligned, and data are normalized (Fig. 3.1)

3. A semi-automated procedure is used to determine the precise start and end of each blink (Fig 3.2). Candidates for boundaries are determined based upon (i) smoothing the signal and determining where it reaches baseline, and (ii) using matching pursuit to fit the signal using three atoms of a non-translationally invariant dictionary comprising of Gaussian shaped elements. The alignment of peaks of step 2 allows this matching

process to fit the actual blink rather than spuriously fitting ambient EEG signal deflections. Once candidate boundaries are determined, manual inspection is used to decide the precise boundary.

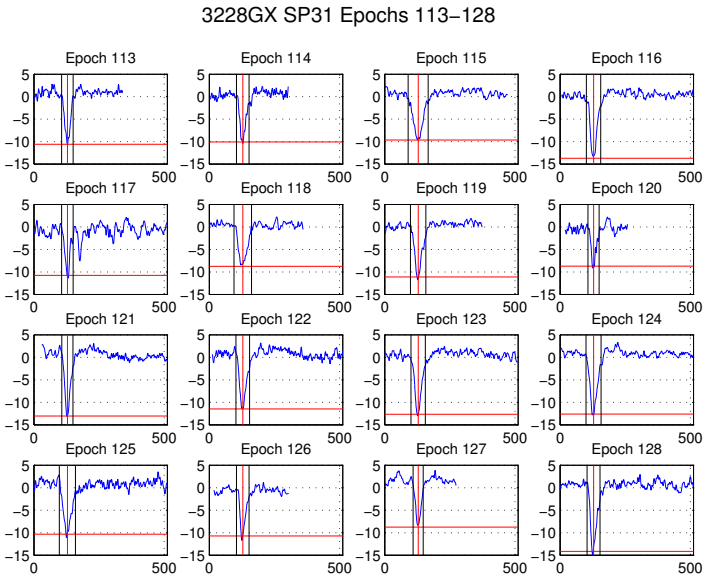


Figure 3.1: Aligning of the extended blink episodes so that the peak occurs at a specific offset within an epoch (in this case offset is chosen to be 166 samples). Example shows a few blink artifact epochs from Fz channel. The black vertical lines show the portion marked by the RPSGT, the vertical red line shows the identified peak. Before aligning, the 2s epoch is extended if necessary to include portions of the blink not included in the epoch. The overall signal length is variable depending on the location of the peak within the original 2s epoch. The x-axis is the sample number, and the signal amplitude in microV is shown on the y axis.

Shape Modeling. The signals corresponding to the precise blink portion of the artifact for all channels in the chosen recordings above were extracted using the procedure described above. The extracted blink signals show markely different shapes across epochs for the same channel (Fig 3.3), and for the same epoch across different channels (Fig 3.2.2). This leaves us with the non-trivial problem of how to model a blink shape appropriately. Taking the mean blink signals across all epochs even after careful alignment of the peak of the main lobe does not account for the main lobe shape variation (variable rate of fall and rise of the signal) and also almost completely flattens out the side lobes (Fig 3.2.2), and thus using this as a template for matching a blink will result in distortion of the ambient EEG signal. We explored another option - dynamic modeling of the blink signal. Linear dynamic models did

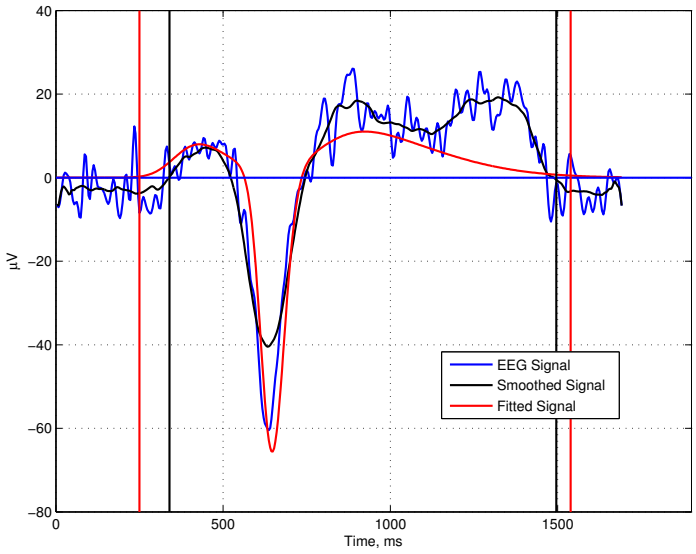


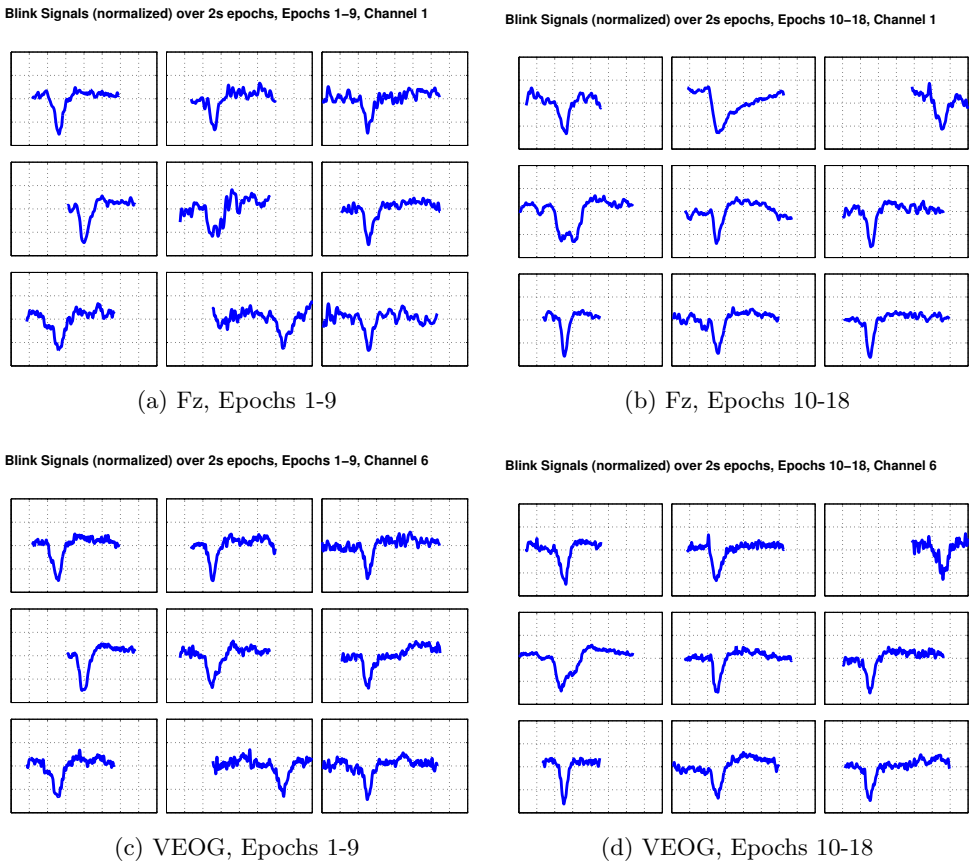
Figure 3.2: Determining candidate boundaries using smoothed (black) and fitted (red) versions of the signal. The final determination of the boundary is made visually.

not give us a good fit but due to the striking similarity to an (inverted) action potential signal, we explored a non-linear dynamic model. Using the FitzHugh-Nagumo model gave a fit (Fig 3.5) that was limited by being unable to capture the left side lobe and the shape of the falling phase of the deflection correctly (inverted in the figure). Thus fitting such a dynamic model will also result in distortion of the recovered EEG.

3.2.3 Skew Gaussian (SG) Dictionary Construction & Validation

Due to the difficulty in shape modeling and variability of shapes as explained above, we analytically constructed a dictionary whose atoms can represent the gamut of blinks. We determined that using standard wavelets such as Daubechies basis, or a Gabor basis [163, 135, 43, 21], or dictionary comprising of only Gaussian elements and their derivatives was not sufficient to correctly model the blink shapes (data not shown), so we constructed a new dictionary comprising of *skew Gaussian* elements (defined below) and their first and second derivatives. We call our dictionary *Skew Gaussian (SG)* dictionary. This describes details on construction of this dictionary.

Definitions. Below $\phi(x)$ denotes a standard normal, and $\Phi(x)$ the CDF of a standard normal i.e. $\Phi(x) = \int_{-\infty}^{x} \phi(\xi) d\xi$.



(a) Fz, Epochs 1-9

(b) Fz, Epochs 10-18

(c) VEOG, Epochs 1-9

(d) VEOG, Epochs 10-18

Figure 3.3: Sample blink shapes showing variation across epochs. Amplitudes are normalized to 1 and each epoch is 2s in duration.

A real-valued *skew Gaussian* function $f_\theta : R \to R$ parameterized by $\theta = (\tau, \alpha, \kappa)$ is defined as

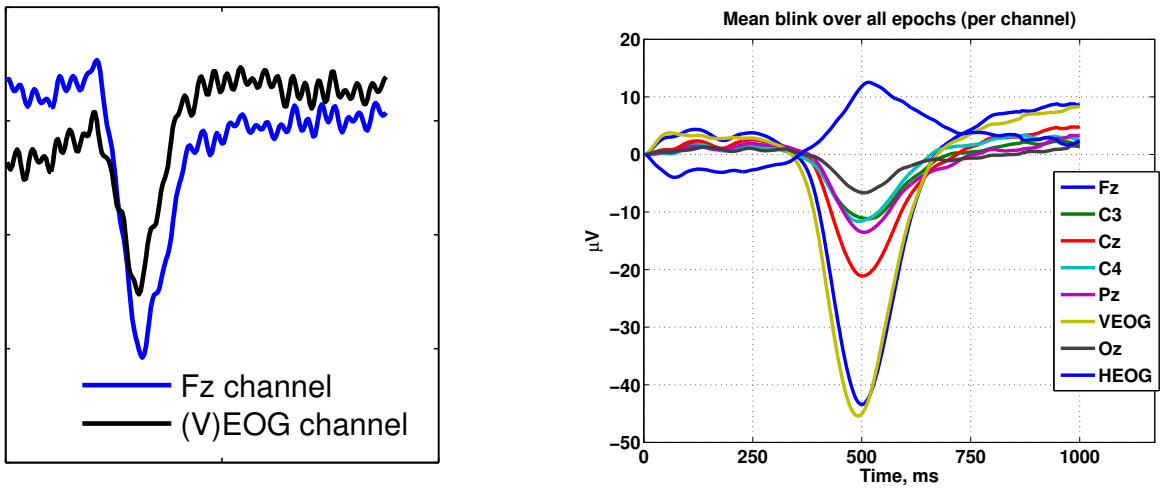
$$f_\theta(x) = C(\theta) g(\frac{x-\tau}{\alpha}, \kappa)$$

where the skew Gaussian kernel $g(x, \kappa)$ is defined as

$$g(x, \kappa) = 2\phi(x)\Phi(\kappa x)$$

and $C(\theta)$ is a normalizing constant such that $\|f_\theta\| = 1$. The parameters τ, α, κ are the *translation, scale* and *skew* parameters, respectively.

The SG dictionary $D_\infty = D_\infty^0 \cup D_\infty^1 \cup D_\infty^2$ where the subdictionaries $D_\infty^0, D_\infty^1, D_\infty^2$ are defined as follows. $D_\infty^0 = \{f_\theta, \theta \in R^3\}$, $D_\infty^1 = \{f'_\theta, \theta \in R^3\}$ and $D_\infty^2 = \{f''_\theta, \theta \in R^3\}$ where f'_θ and f''_θ denote the first and second derivatives, respectively. The subscript on D_∞ is used in the notation to emphasize that the dictionary is infinite. Explicit expressions for f'_θ and



(a) Blinks in Single Epoch

(b) Averaged Blink (over epochs)

Figure 3.4: (a) Sample blink shapes showing variability in different EEG channels for the same epoch. Fz, C3, Cz, C4, Pz and Oz refer to different placements of electrodes on the scalp; the Fz signal is closest to the eye. VEOG and HEOG are the EOG electrodes placed near the eye. (b)Averaging of the extracted blink signals after aligning of the peak in the main lobe. Using the mean signal does not account for variation in the rate of fall and rise of the main deflection or the side lobes.

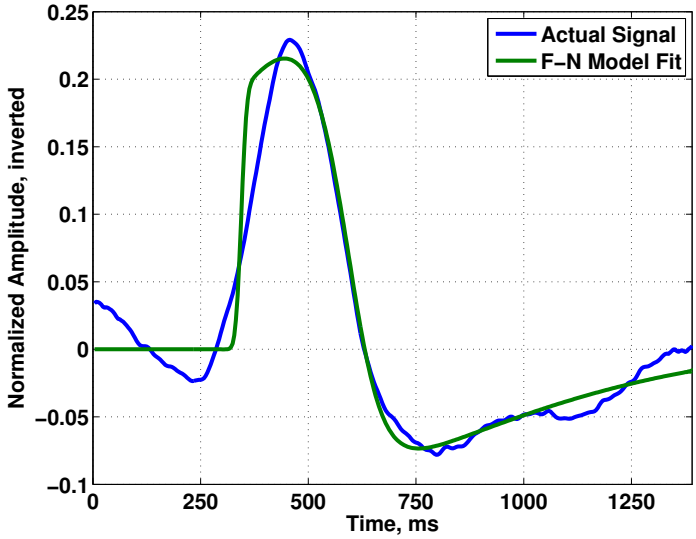


Figure 3.5: Dynamic modeling of the blink shape using the FitzHugh Nagumo (F-N) model of action potential (AP) does not capture a blink in its entirety. The blink is inverted in this figure to show its apparent similarity to AP.

f''_θ can be computed using the facts

$$\begin{aligned} \phi'(x) &= -x\phi(x) \\ \phi''(x) &= (x^2-1)\phi(x) \end{aligned}$$

so that

$$\begin{aligned} g'(x,\kappa) &= -2x\phi(x)\Phi(\kappa x) + 2\kappa\phi(x)\phi(\kappa x) \\ g''(x,\kappa) &= 2(x^2-1)\phi(x)\Phi(\kappa x) - (4\kappa x + 2\kappa^3 x)\phi(x)\phi(\kappa x) \end{aligned}$$

We chose the additional dictionaries D^1_∞, D^2_∞ due to the tri-lobed nature of the blink shape.

Implementation of Finite Dictionary. Only a finite subset of D_∞ is required to form a complete basis. In fact, we use a subset that forms an over-complete basis to minimize the dimensionality of the subspace spanned by blink artifacts in this overcomplete basis. Naturally, translation τ (time center) and scale α (time spread) are restricted by the maximum length of the epoch. In our case if x denotes time in seconds, then for a 2s epoch, we restrict $|\tau| \leq 2$ and $|\alpha| \leq 2$. We can further restrict skew κ to $|\kappa| \leq \kappa_{\max}$ as there will be no advantage to highly skewed Gaussian atoms (i.e., beyond a certain skew value) in modeling blinks. We must further subsample the dictionary atoms to be able to get a finite $D \subset D_\infty$; the open question remains as to how to obtain the optimal sampling ? The larger the dictionary, the lesser the number of base vectors needed to represent a signal. However, the performance of most sparse recovery algorithms is bound by dictionary size. If we restrict to uniform sampling, then the question of choosing an optimal dictionary becomes that of finding the optimal values of $\{\kappa_{\max}, \Delta\tau, \Delta\kappa, \Delta\alpha\}$ such that restricting D_∞ to the set of atoms $f_\theta, f'_\theta, f''_\theta$ where $\theta \in \{\ |\tau| \leq 2.\ |\alpha| \leq 2,\ \ |\kappa| \leq \kappa_{\max}, \tau = n_\tau\Delta\tau,\ \alpha = n_\alpha\Delta\alpha,\ \kappa = n_\kappa\Delta\kappa$ for $n_\tau, n_\alpha, n_\kappa \in Z\}$ results in minimal number of atoms to match a blink in our test data set while maximizing algorithm performance.

Finding these optimal values is a highly non-trivial problem, and obviously dependent on the algorithm of choice, and the input signals. We use a heuristic method to determine the values $\kappa_{\max}, \Delta\tau, \Delta\kappa, \Delta\alpha$. We further restricted the set of atoms to those that have compact support on the domain corresponding to a single epoch. We use a soft criteria of "compactness" i.e. we consider f_θ (defined on the domain $x \in [0,T]$ with $T = 2$) as compact if

$$\left| \overline{f_\theta}|_{[0,\varepsilon]} - \overline{f_\theta}|_{[T-\varepsilon,T]} \right| \leq RMS(f_\theta)/3 \tag{3.1}$$

where $RMS(f)$ is the root mean square value of the function f on the specified domain $[0, T]$, and $f|_A$ denotes the restriction of f to the domain A and $\overline{f}$ is the mean value of f.

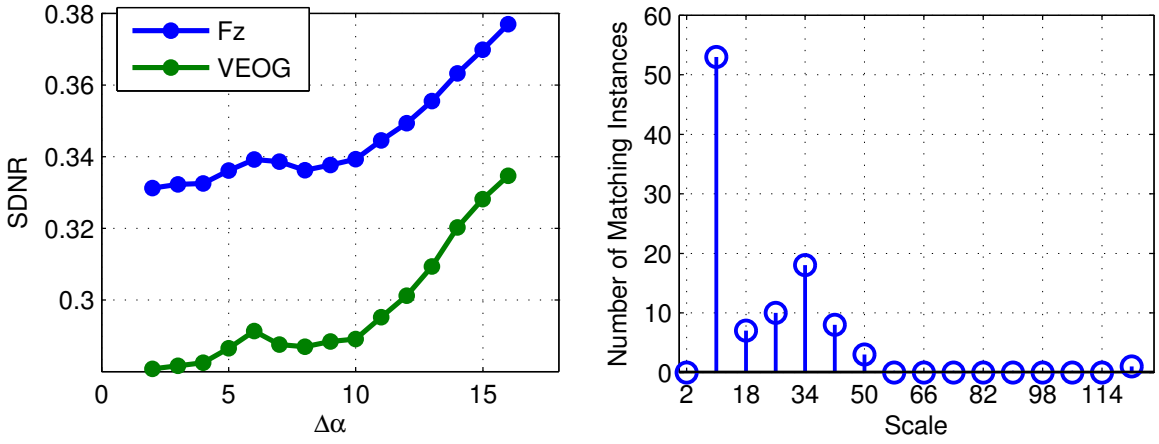
Heuristic Method to Determine Dictionary Sampling. About 100 epochs containing a single blink episode from a single recording are used. SG dictionaries are constructed with various samplings and the variation of mean SDNR over epochs from a single iteration of the Matching Pursuit algorithm with parameters $\{\kappa_{\max}, \Delta\tau, \Delta\kappa, \Delta\alpha\}$ is observed. We use the maximum $\Delta\tau, \Delta\kappa, \Delta\alpha$ and minimum $\kappa_{\max}$ value beyond which change in SDNR is not significant. Instead of exploring the entire parameter space, we vary one parameter at a time, keeping the remaining constant. Since $\Delta\tau$ determines levels of translation of the elements, we treat this differently since we can expect SDNR to be pretty much linear with $\Delta\tau$ (which is indeed the case as shown below). We thus first fix the translation level at $\tau = \tau_0 = 0.6289$, and re-align the epochs so that the peaks of each blinks are centered at τ_0. This way we can test the impact of all other parameters $\Delta\kappa, \Delta\alpha, \kappa_{\max}$.

Variation of mean SDNR (over all epochs) with $\Delta\alpha$ for two channels with fixed $\tau = \tau_0, \kappa = 0$ (Fig 3.2.3) shows no significant improvement in SDNR for $\Delta\alpha \leq 8/256$. Hence we fix $\Delta\alpha = 8/256$. Next, fixing $\alpha = 0.391$, mode of the distribution of α with $\Delta\alpha = 8/256$ over matching blinks (Fig 3.2.3), we used the variation of SDNR with $\Delta\kappa$ (Fig 3.2.3, 3.2.3) to set a value of $\Delta\kappa = 1$. Using $\Delta\kappa = 1, \Delta\alpha = 8/256, \tau = \tau_0$ in a similar fashion we determined $\kappa_{\max} = 7$ (Fig 3.2.3, 3.2.3)).

Next, fixing $\kappa = [0\ 1], \alpha = [10/256\ \ 34/256]$, based upon the modes of the distributions (Fig 3.2.3, 3.2.3), variation of mean SDNR with $\Delta\tau$ was determined on the unaligned blink epochs, showing linear behavior as expected (Fig 3.2.3). We chose $\Delta\tau = 2/256$.

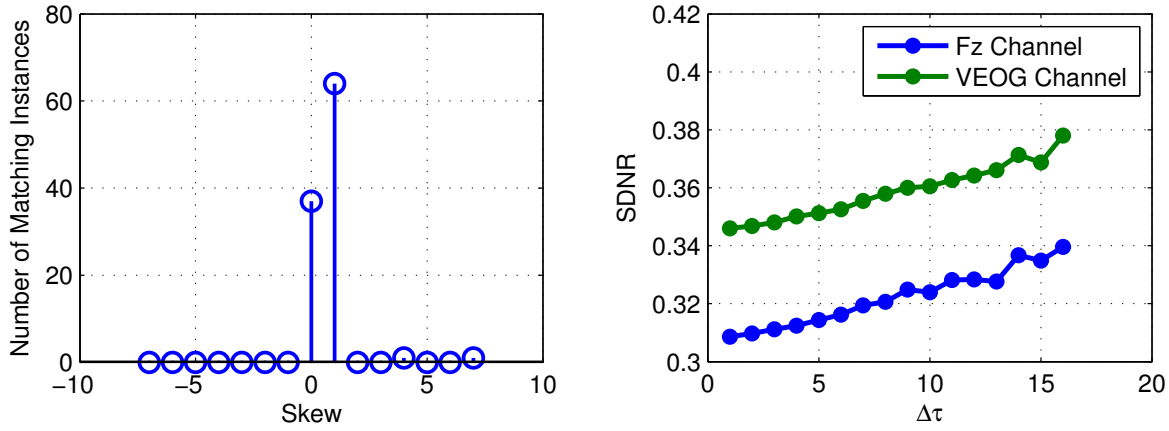
4. Constructed Dictionary. Using the above values for $\{\kappa_{\max}, \Delta\tau, \Delta\kappa, \Delta\alpha\}$ and the criterion of compact elements (3.1), a dictionary comprising of a total of 212649 elements was constructed. A few sample atoms in the dictionary are shown in Fig. 3.10. An animation showing all dictionary atoms is available at Online Supplemental Material.

5. Dictionary Validation. The procedure above for determining the optimal dictionary includes validation of our chosen dictionary against the extracted blink shapes. A visual depiction showing some sample fits is in Fig 3.11.



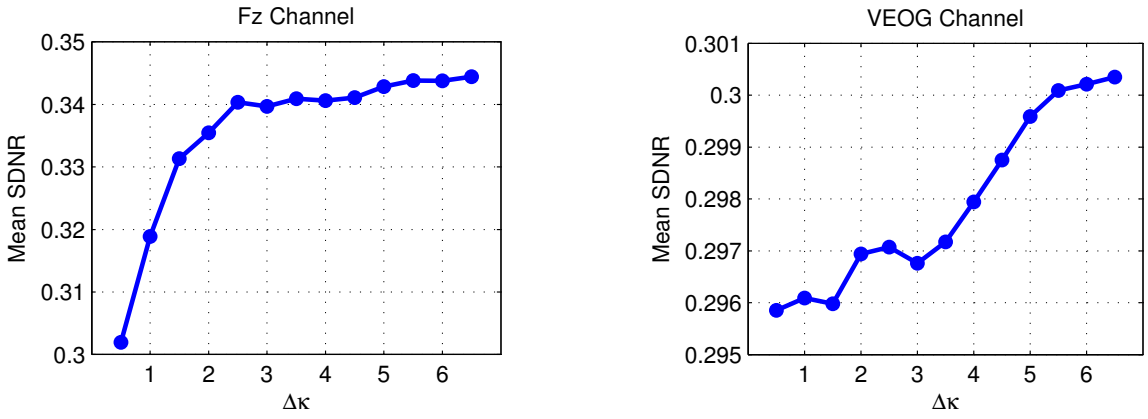
(a) (b)

Figure 3.6: (a) Variation of mean SDNR over epochs of aligned blinks with $\Delta\alpha$ with $\tau = 0.6289$ and $\kappa = 0$. Values of $\Delta\alpha$ are shown in samples, and actual values are those scaled by the sampling rate (256). (b)Distribution of α for matching blinks in (aligned) test blink set. Values of α are shown in samples, actual values are those scaled by the sampling rate (256). This shows that the mode of the distribution is $\alpha = 10/256 = 0.391$.



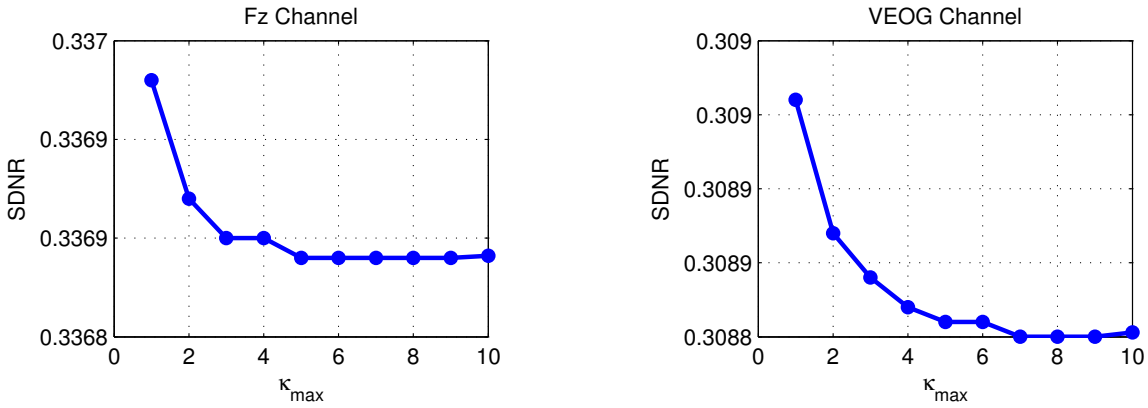
(a) (b)

Figure 3.7: (a) Distribution of κ for matching blinks in (aligned) test blink set. This shows that the mode of the distribution is $\kappa = 1.$. (b)Variation of mean SDNR over epochs of aligned blinks with $\Delta\tau$ with α set to two values [10/256 34/256] and κ set to two values [0 1]. Values of $\Delta\tau$ are shown in samples, and actual values are those scaled by the sampling rate (256). The linear behavior shows that SDNR can be arbitrarily reduced by choosing a smaller $\Delta\tau$. In order to limit the dictionary to a reasonable size we used $\tau = 2$. (Note, we must have $\tau \geq 1/256$, the sample length).



(a) Fz Channel (b) VEOG Channel

Figure 3.8: (Variation of mean SDNR over epochs of aligned blinks with $\Delta\kappa$ with $\tau = 0.6289$ and $\alpha = 0.391$ for (a) Fz channel (b) VEOG channel.



(a) Fz Channel (b) VEOG Channel

Figure 3.9: (Variation of mean SDNR over epochs of aligned blinks with $\kappa_{\max}$ with $\tau = 0.6289$, $\Delta\alpha = 8/256, \Delta\kappa = 1$ for (a) Fz channel (b) VEOG channel.

3.2.4 Algorithms

Using the above dictionary, greedy (CSSR, OMP, JMP) as well as Bayesian versions (SBL, BSBL) of sparse recovery algorithms are applied to the KDT (1.45 hours total) and PVT (0.83 hours total) sections of four recordings. The algorithms are applied to each epoch independently. Once sparse coefficients (sparsity level is chosen per algorithm, described below) are obtained, a k-means clustering algorithm is applied on a feature set comprising of (i) the dominant sparse coefficient amplitude for an epoch and (ii) the maximum signal ampli-

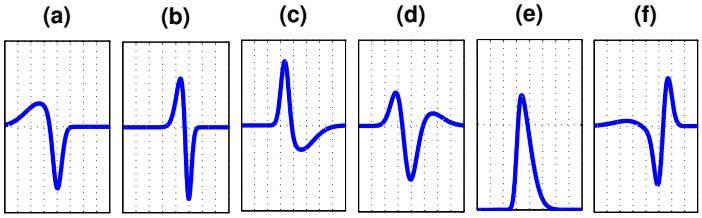


Figure 3.10: Sample atoms (normalized) in constructed dictionary. Each atom represents a signal epoch 2s long. Sub-dictionary and parameters for the atoms shown above are (a) D^1, $\alpha = 0.3438, \tau = 1, \kappa = -4$ (b) D^1,$\alpha = 0.125, \tau = 1.234, \kappa = -2$ (c) D^1, $\alpha = 0.3125, \tau = 0.8359, \kappa = 4$ (d) D^2, $\alpha = 0.3125, \tau = 0.8394, \kappa = 2$ (e) $D^0, \alpha = 0.2188, \tau = 0.7578, \kappa = 4$ (f) D^2, $\alpha = 0.4062, \tau = 1.3281, \kappa = -4$. .

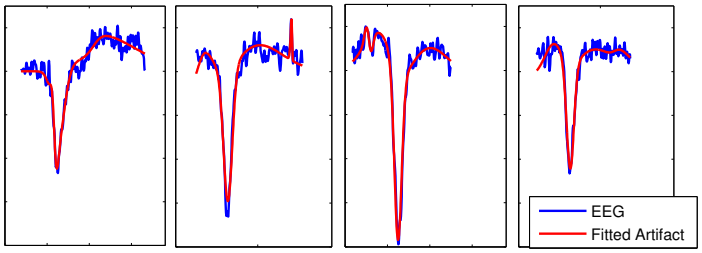
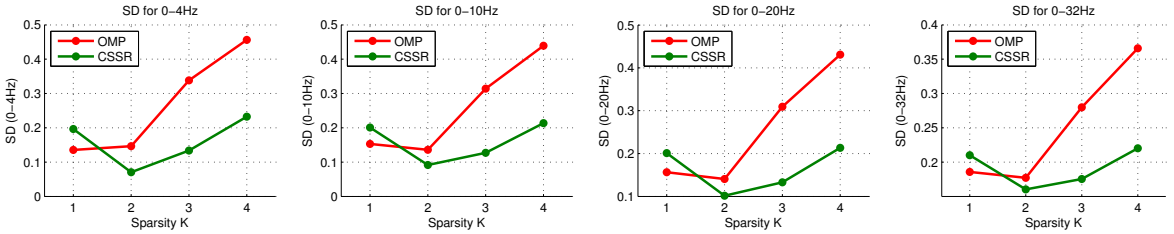


Figure 3.11: Validation of dictionary atoms by fitting atom combinations to real blink EEG signals. Each epoch is 2s in duration .

tude over an epoch to cluster epochs into artifactual vs non-artifactual categories. Recovered blinks are then subtracted from the original signals to obtain artifact free EEG in the epochs classified as artifactual.

In addition, a naive application of ICA was done on these recordings using the logistic infomax ICA algorithm of Bell & Sejnowski ([66, 215, 65, 187]) using the EEGLAB toolbox [67, 136]. Application of ICA this way on short epochs (upto 2s each) for low density EEG recordings (8 electrodes or less) has previously been reported in [254]. In our implementation, determination of which components correspond to artifact components was made based upon the correlation of each component with the EOG channels (a procedure also used in [118]). The purpose was to compare our algorithm to a naive application of ICA as a denoising rather than artifact detection tool, and so we ran ICA only on epochs containing blink artifacts.



(a) SD_{0-4Hz} (b) SD_{0-10Hz} (c) SD_{0-20Hz} (d) SD_{0-32Hz}

Figure 3.12: Spectral Distortion (SD) vs Sparsity Level (K) for OMP and CSSR. (a) SD over 0-4Hz (b) SD over 0-10Hz (c) SD over 0-20Hz (d) SD over 0-32Hz. All graphs indicate that $K = 2$ is a reasonable choice.

Choice of Sparsity Level K. For algorithms CSSR, OMP/SBL and JMP/BSBL the parameter K, number of sparse components was fixed heuristically. We ran the algorithms on approximately 7mins of KDT data from a single recording that contained blink artifacts using various values of K, and compared the spectral distortion (SD, see definition below) for various frequency ranges. From this we made the determination to fix $K = 2$. (Figs 3.2.4,3.2.4, 3.2.4, 3.2.4).

3.2.5 Performance Metrics

When used as a denoising tool, three performance metrics, *signal denoising ratio* (SDNR), *spectral distortion* (SD) and *multi-scale entropy preservation* (MSEP), defined as follows, were used to compare the performance of CSSR with denoising techniques (JMP, OMP, SBL, BSBL, ICA). To evaluate our technique for *artifact detection* we compared artifactual epochs identified by our method with those manually marked by an RPSGT to determine the accuracy, sensitivity and specificity of our classification.

1. Signal Denoising Ratio (SDNR) If y is original data for an epoch, and $\widehat{y}$ is the fitted artifact signal (that is, $y - \widehat{y}$ is the denoised signal), then the SDNR of that epoch is defined as

$$SDNR = \frac{\|y - \widehat{y}\|_2}{\|y\|_2}$$

SDNR is indicative of the extent to which is denoised. A smaller SDNR indicates a better fit. However, SDNR is not an accurate measure of how well a denoising algorithm performs for our purposes since it is highly sensitive to overfitting. Thus, an algorithm that overfits can often result in lower values of SDNR.

2. Spectral Distortion (SD) For a frequency range Ω, the spectral distortion over a recording is defined as

$$SD(\Omega) = \left(\frac{\int_{\omega \in \Omega} \left[P_{ra}(\omega) - P_{na}(\omega) \right]^2 d\omega}{\int_{\omega \in \Omega} \left[P_a(\omega) \right]^2 d\omega} \right)^{\frac{1}{2}}$$

where $P_{ra}(\omega)$ is the power spectral density (PSD) of the denoised signal over artifactual epochs, $P_{na}(\omega)$ is the PSD of the original signal over non-artifactual epochs and $P_a(\omega)$ is the PSD of the original signal over artifactual epochs. We use the multi-taper method to compute the PSD per epoch and then take the average value over all the relevant epochs. The integration above is approximated by summation over the frequencies over which PSD is computed. For our purposes, since blink artifacts mostly distort the delta power band (1-4Hz) of the EEG signal, we will be interested in $SD_{1-4\ Hz}$. Since we are not interested in frequencies above 32 Hz for most EEG analysis, we will also be interested in the total $SD_{1-32\ Hz}$. *Smaller* values of SD are better. SD is a measure of how much spectral power in the EEG signal is lost as a result of the denoising process. Thus SD can be useful in determining the quality of denoised signal in the frequency domain. Previous uses of spectral distortion to measure quality of the denoised signal has been reported in [40, 238, 81, 92]. Note that since averages are taken over epochs, overfitting and underfitting errors can cancel each other, and thus one can get a small SD value even if an algorithm performs poorly. As will be seen in section 3.3.1 ,algorithms such as OMP that consistently tend to overfit or JMP that consistently tend to underfit can be compared using this metric. However, for algorithms such as ICA that sometime overfits and sometime underfits, use of this metric can be misleading.

3. Multi-scale Entropy Preservation (MSEP) For a scale range S, the multi-scale entropy preservation (MSEP) over a recording is defined as

$$MSEP(S) = \left(\frac{\int_{s \in S} \left[M_{ra}(s) - M_{na}(s) \right]^2 ds}{\int_{s \in S} \left[M_a(s) \right]^2 ds} \right)^{\frac{1}{2}}$$

where $M_{ra}(s)$ is the entropy of the denoised signal over artifactual epochs at scale s, $M_{na}(s)$ is the entropy of the original signal over non-artifactual epochs, $M_a(s)$ is the entropy of the original signal over artifactual epochs. Here entropy at a particular scale is computed as in the multiscale entropy (MSE) method described in [52]. Comparison of MSE over denoised and noiseless portions is an indicator of how much the original complexity of the

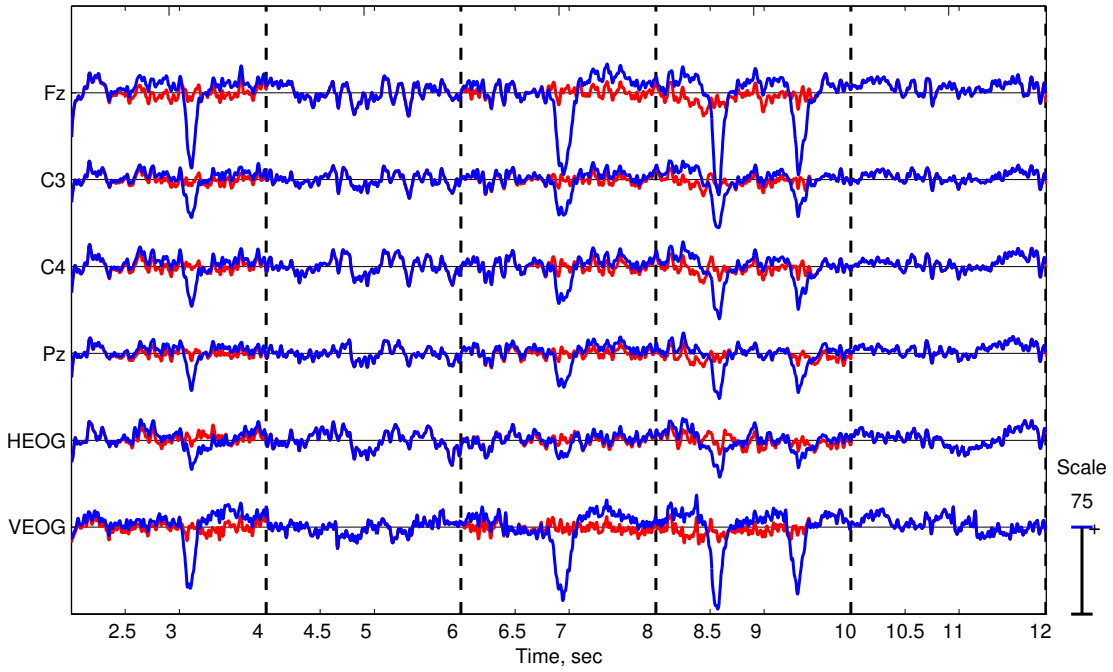


Figure 3.13: Snapshot of multi-channel EEG recording with original (blue) and reconstructed (red) after blink artifact elimination using CSSR.

signal is maintained by the denoiser. A *smaller* MSEP value is indicative of higher level of complexity preservation and is better. MSE has been shown to be a good indicator of measure of complexity in biological signals [52]. MSEP can thus be useful in determining the quality of the denoised signal as far as complexity is concerned.

Note that the two metrics, SD and MSEP as defined above, that we use to evaluate reconstructed signal quality capture two of the most commonly used measures of EEG, namely spectral power density and signal complexity [215].

3.3 Results

3.3.1 EEG Denoising In Real Recordings

Reconstructed EEG signals after blink artifacts were removed using CSSR were inspected visually by an RPGST and confirmed to be artifact free. A visual snapshot is shown in Fig 3.13.

Comparison of Algorithms. We compared the results of denoising using CSSR with (i) other sparse recovery techniques (OMP/SBL, JMP/BSBL) on the same dictionary and (ii) ICA (applied as described in Section 3.2.4), a common EEG denoising technique. CSSR does not suffer from the problems of over-fitting as in OMP/SBL (Fig 3.14(a)), that of under-fitting as in JMP/BSBL (Fig 3.14(a)), and does not remove useful EEG components in addition to noise as with ICA (Fig 3.14(b)). The problem of over and underfitting is also apparent in one example of a multi-channel view of an epoch (Fig 3.15 and Fig 3.16).

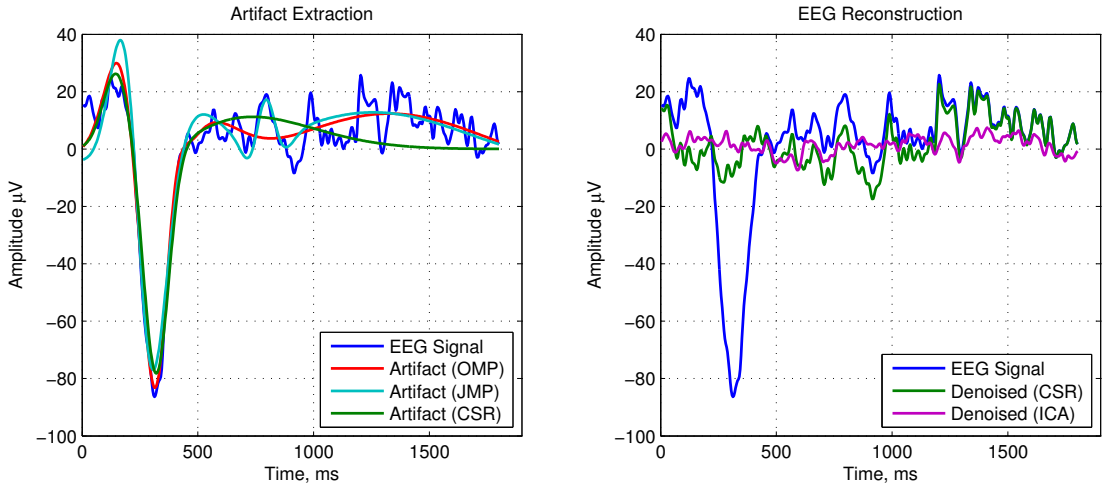


Figure 3.14: Comparision of CSSR, OMP, JMP and ICA algorithms for artifact elimination. Single epoch showing (a) matched artifact (b) reconstructed EEG. OMP, JMP and ICA often result in over or underfitting of the actual artifact resulting in corruption of the ambient EEG signal. (Not all algorithms are shown on both plots for visual clarity).

Performance Metrics. Comparison of power spectrum of the denoised signals with that of artifact-free epochs shows that CSSR algorithm exhibits the least spectral distortion both overall (Fig 3.3.1) and in the lower frequency (1-4Hz, Delta) range, which is the frequency range most impacted by eye blinks (Fig 3.3.1). The spectral distortion is shown by frequency band in Fig 3.3.1. The spectral distortion caused by CSSR in the delta range is the least. Comparison of entropy of denoised signals at various scales for the algorithm again shows the CSSR leads to maximal entropy preservation (Fig 3.3.1). Performance metrics - SD and MSEP for all epochs in a single recording - for CSSR, OMP, JMP and ICA are shown in Table 3.1. The values of SDNR over all artifactual epochs are also shown, though as mentioned previously these values are not truly indicative of denoising algorithm performance

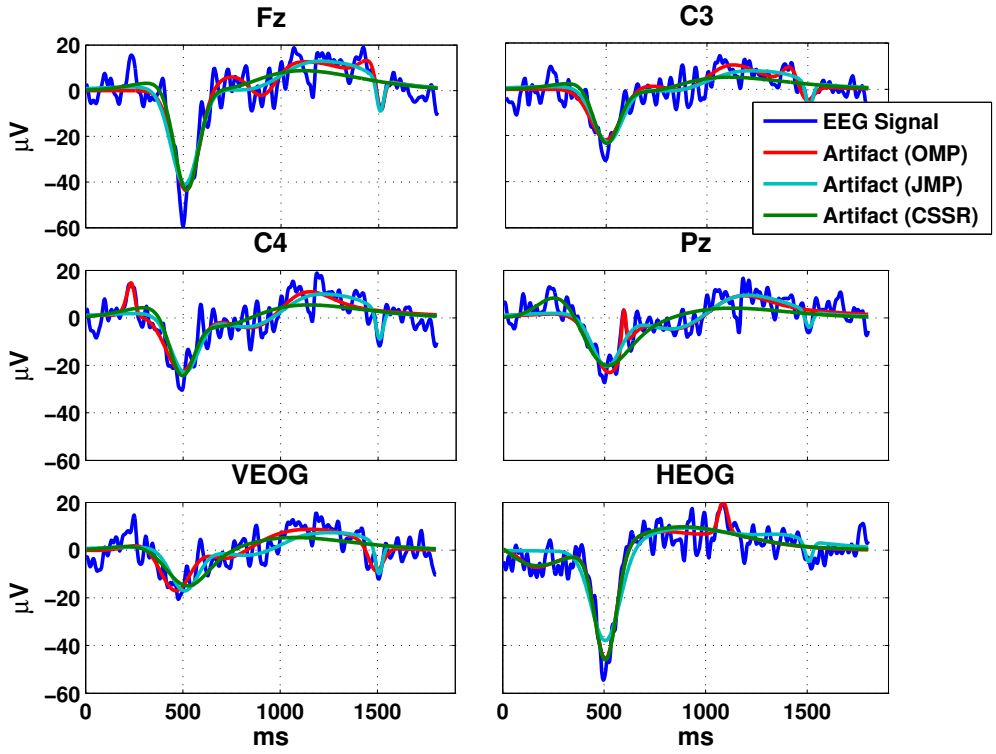
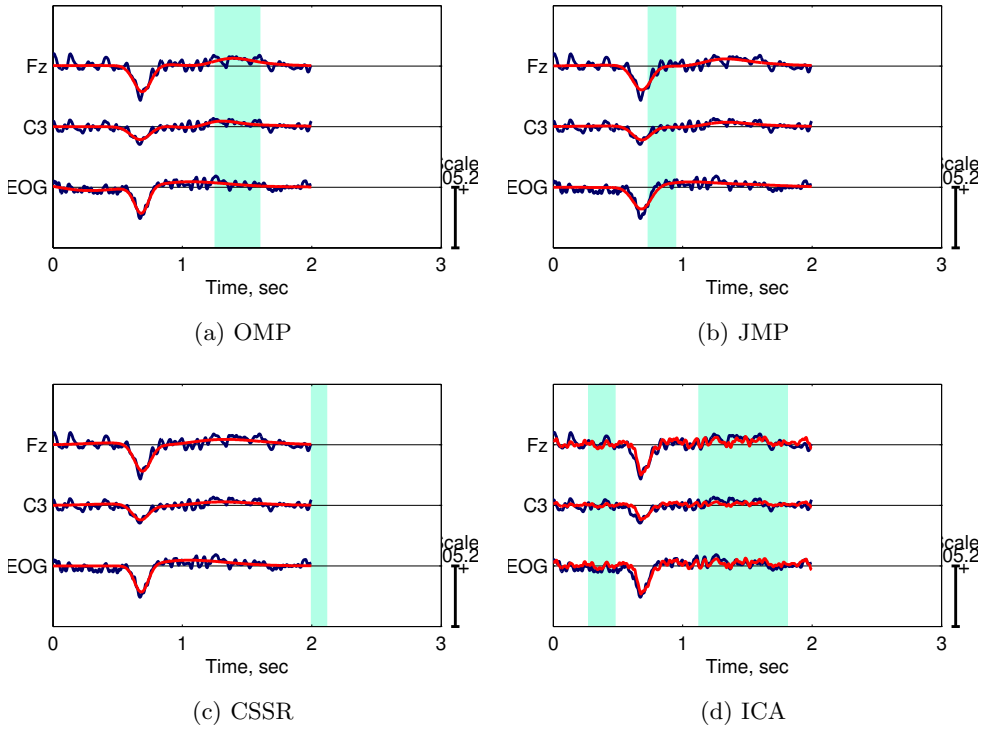


Figure 3.15: Comparsion of CSSR, OMP and JMP for matching a blink artifact in a single epoch. By using correlations across channels as well as by learning temporal relationships between sparse coefficients, CSSR is able to obviate over-fitting (OMP) and under-fitting (JMP).

as overfitting results in lower SDNR so that lower SDNR is not always indicative of better denoising performance.

Table 3.1: Denoising performance metrics for algorithms on the Fz channel in a single recording.

	$SD_{\text{1-32 Hz}}$	$SD_{\text{1-4 Hz}}$	$MSEP$	$SDNR$
OMP/SBL	0.1901	0.2412	5.0406	0.3883
CSSR	0.1058	0.0226	0.9991	0.4810
JMP/BSBL	0.1471	0.1586	3.9164	0.4280
ICA	0.2727	0.0438	9.2524	0.4714



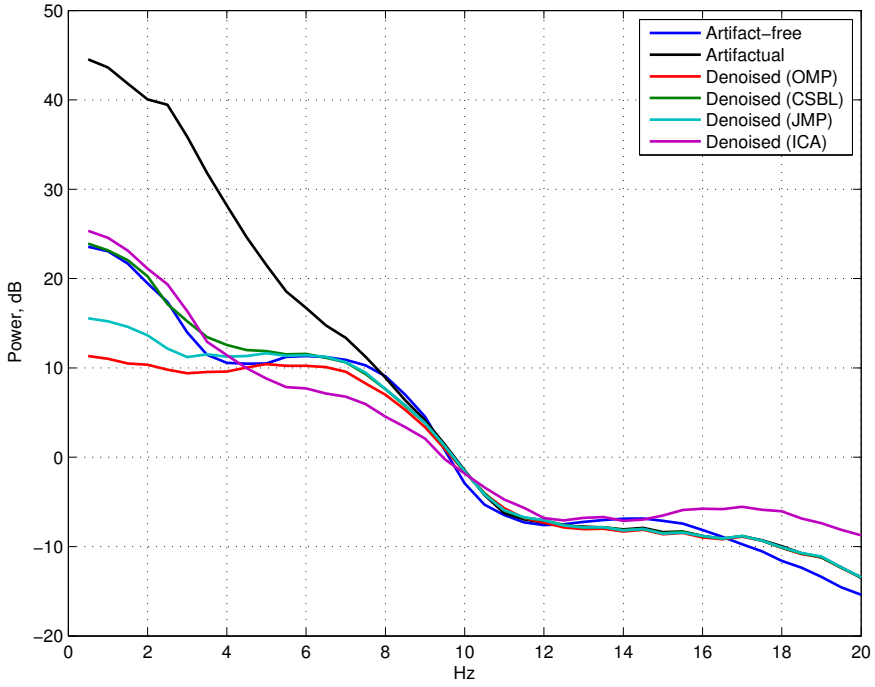
(a) OMP (b) JMP

(c) CSSR (d) ICA

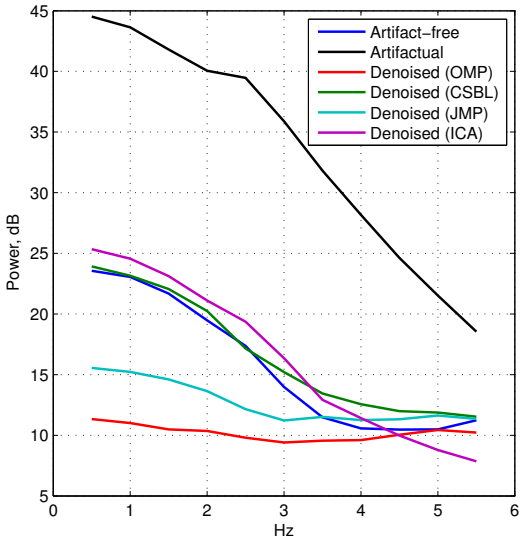
Figure 3.16: Detailed view of fitted artifacts in multi-channel EEG when using (a) OMP (b) JMP (c) CSSR (d) ICA (as described in Section 3.2.4, showing over and underfitting. Some portions where which over or underfitting is apparent are highlighted in cyan.

3.3.2 Detection Using k-means Clustering

Using k-means using the largest sparse coefficient (from CSSR, OMP or JMP) and maximum signal amplitude shows a clear separation of epochs with and without blinks (Fig 3.20). Confusion matrices of classified epochs for a single recording, when compared with manually (RPSGT) identified artifacts, are shown for CSSR, OMP and JMP in Fig 3.19. Our method showed a specificity of 96% and sensitivity of 97%.

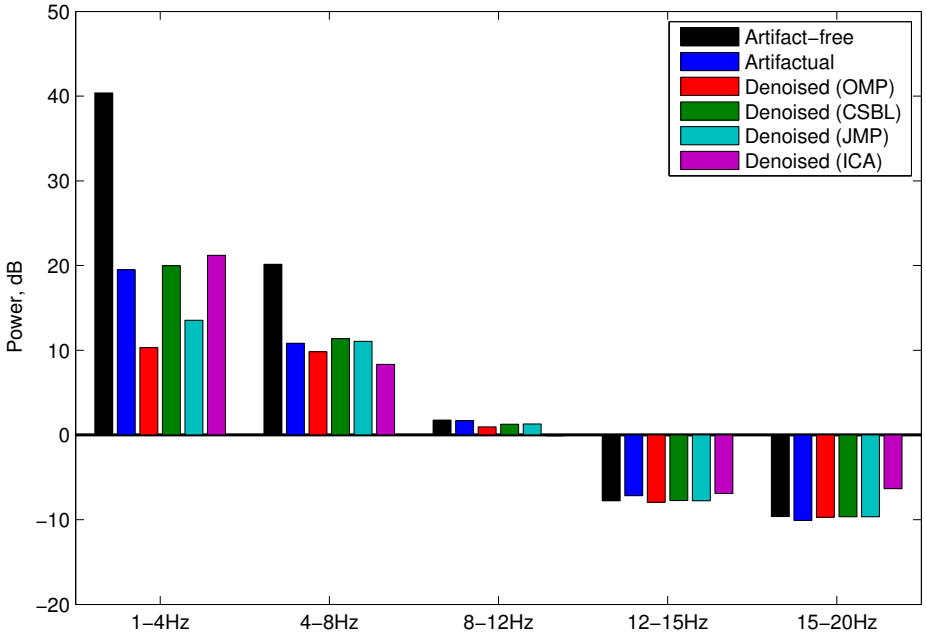


(a)

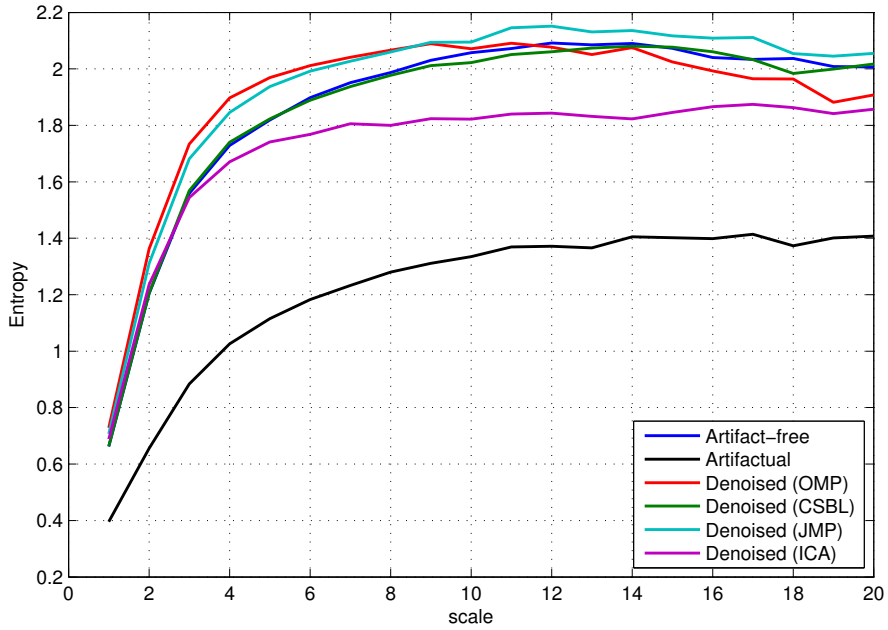


(b)

Figure 3.17: (a) Comparison of power spectra of reconstructed EEG epochs for various algorithms with that of artifact-free EEG epochs, over 0-20Hz. CSSR shows spectrum that is closest to that of artifact-free EEG when compared to all other algorithms. As noted in the section on performance metrics, due to the averaging process when computing spectrum over all epochs, over and under fitting errors (as in ICA) may cancel each other and the actual distortion may not be apparent from the averaged spectral plot. (b) Spectral power zoomed over 0-6Hz to show distortion caused by various algorithms in the low frequency range.

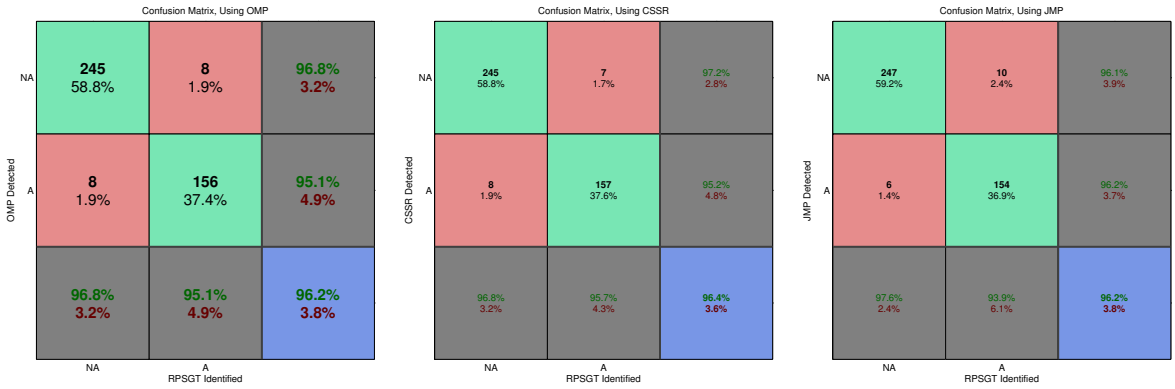


(a)



(b)

Figure 3.18: (a) Aggregated spectrum over the five bands delta (1-4Hz), theta (4-8Hz), alpha (8-12Hz), low beta (12-15Hz), high beta (15-20Hz) of artifactual, non-artifactual and reconstructed epochs showing relative spectral distortion of algorithms. (b) Multi-scale entropy after elimination of artifacts and reconstruction of EEG across all epochs in a single recording, showing entropy preservation properties of algorithms.



(a) OMP (b) CSSR (c) JMP

Figure 3.19: Confusion matrices when using (a) OMP (b) CSSR (c) JMP as an artifact detection tool. "A" denotes artifactual epoch, "NA" denotes non-artifactual. The four upper left boxes show the number of epochs in each category, and the lower and right boxes (in gray) indicate the relative percentages. The blue box indicates the overall accuracy.

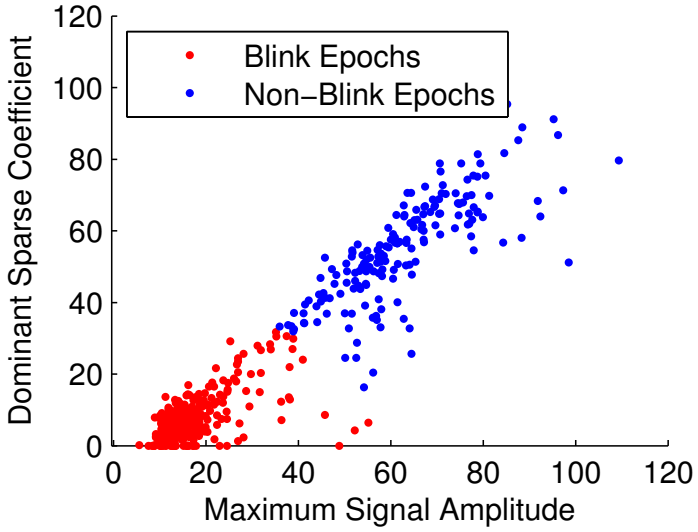


Figure 3.20: Clustering of epochs into ones with and without blinks estimated sparse coefficient.

3.4 Conclusions and Discussion

Our methodology making use of temporal and spatial correlations eliminates structured artifacts and reconstructs cortical EEG with high fidelity in both linear (spectral) and non-linear (entropy) measures. Compared to existing methods of artifact detection our method (i) allows us to preserve EEG instead of discarding data in epochs containing artifacts; (ii) permits fully on-line implementation since it is epoch-based, unlike ICA which requires long recordings; (iii) requires only few (4-6) channels of EEG facilitating practicality and non-invasiveness; (iv) is fully automated requiring no manual intervention. We have shown that an application of ICA as described in Section 3.2.4 is outperformed by our epoch-based algorithm in its ability to faithfully reconstruct blink artifact free EEG. Our near real-time artifact elimination technique may enable using EEG in ambulatory medical applications for monitoring cognitive states during wake and even detecting sporadic events such as seizure onset.

Our alternative to standard group sparsity models of structured compressed sensing has applicability beyond EEG artifacts. We have shown that the proper use of a correlation in the prior joint probability density of coefficients can improve Bayesian sparse recovery of a noisy signal when both the signal and noise have sparse components in a dictionary. Extensions to our current work include application to other artifact types, run-time optimization of the algorithm to be operable in mobile environments, strategies on learning the correlation matrix R from data and applications to learning other (non-artifactual) structural information in EEG such as measures of sleepiness.

Chapter 4

Active Learning for Ensemble Clustering: Application to EEG Epoch Classification

4.1 Abstract

Active learning is a machine learning paradigm where the learner can ask for specific examples to be labeled rather than being handled pre-labeled data; this has been shown to dramatically reduce the number of labeled examples required which especially useful when labeling is expensive. However, it is known that often active learning fails when the input data are noisy or highly non-separable. We have developed a computationally and conceptually simple method, Output-based Active Selection (OAS), for selecting samples for active learning that makes use of predicted output from base learners as augmented features. Using synthetic noisy Gaussian mixtures we show our approach outperforms several state-of-the-art algorithms on both supervised and semi-supervised E-M based base learners. Besides computational simplicity, our algorithm is amenable to use for ensemble classification. We have applied our technique to the task of EEG epoch classification from an ensemble of automated artifact detectors, and show that our method boosts the overall accuracy of the ensemble from 91% (the maximum accuracy of any individual member) to 97.5% with only 10% active labels, which is better than what is achievable using standard ensemble clustering methods.

4.2 Motivation

As discussed in Chapter 1, the presence of a small number of artifacts in EEG can change the power spectral density and confound analysis of wake EEG for use as a potential marker of sleepiness. This problem was partially solved in Chapter 3 where we developed a method to identify and remove eye movement artifacts using clustering methods on sparse features. It was only natural to apply this method on the dataset discussed in Chapter 1 to remove eye movement artifacts from EEG, and subsequently clean remaining types of artifacts using standard methods such as thresholding and filtering. Application of these methods, however, presented difficulty. This dataset had already been partially cleaned of some artifactual epochs manually by an RPSGT which resulted only a small fraction of the epochs from the original recordings available for further analysis. Methods in Chapter 3 assumed that the EEG had been already cleaned of artifacts other than eye movements and that all remaining data set containing eye movement artifacts was available. This permitted use of clustering methods such as k-means over all epochs. When these standard clustering methods were applied followed by thresholding/filtering, there were far too few epochs remaining to be able to do any meaningful statistical analysis such as spectral estimation.

The above problem is an example of a *multiple classifier* or *ensemble* system, where the classification task is based upon the output of multiple classifiers. In the above example, the methods of clustering on sparse features, thresholding and filtering form an ensemble of three classifiers. The method described above is *one* way of combining these outputs to produce a final outcome: an epoch is classified as positive for artifacts if any of the three individual methods yield a positive outcome. This methodology of combining ensemble outputs exaggerates the false positive rates of individual classifiers: if each classifier has a false positive rate of ε then the overall false positive rate (FPR) with an ensemble of size N is $1-(1-\varepsilon)^N$ which approaches 100% if $\varepsilon > 0$. On the other extreme, one can comprehend a combination method where an artifact is detected if *all* of the methods detect a positive, but this exaggerates the false negative rate (FNR): if each classifier has a false negative rate of δ then the overall false negative rate is $1-(1-\delta)^N$.Application of the first method above to the example above resulted in an FPR of 15% and the second method gave FNR of 22% which are unacceptable for our requirements: we need small FPR in order to retain as many epochs as possible, an even more stringent necessity considering the sparsity of available epochs to begin with; and low FNR to minimize the confounding impact of artifacts on power spectral

density. It is thus clear that, especially when none of the individual classifiers have high accuracy in by themselves, we need a better way of combining the output of classifiers in an ensemble.

Ensemble classification is an active area of research and several advanced techniques [71, 70, 69, 144, 180, 189]) exist to improve the classification accuracy of an ensemble beyond what is achievable using any single classifier. However, most of these techniques work by re-training the individual classifiers - e.g. by manipulating the training set - so as to produce different output given the same input, an assumption inapplicable to our example above. The thresholding and filtering based classifiers are *not retrainable*. Several methods in medical decision making also fall in this category. Consider, for example, a large set of EEG recordings that need to be scored for sleep (i.e. classified into sleep stages) at a laboratory for a sleep research analysis with scores from the following three sources: (i) a built-in automated sleep scorer that is the output of the EEG recording system itself (ii) annotations for a subset of the recordings from a technician who is no longer available, (iii) an open source third party tool for automated sleep scoring. We have three methods for sleep scoring: the recording system, the technician and the open source tool. For none of these do we have any prior knowledge of their reliability other than that they do better than random classification. This an example of an ensemble classification problem: combining the output of multiple classifiers to produce an accurate classification where some of the members (the recording system and the technician) are *not re-trainable.*

Two types of existing methods that address the problem of ensemble clustering with non-retrainable members were tested on an ensemble of six different artifact detection methods each with classification accuracy $< 91\%$. (i) *Combination* methods ([251]) that combine the outputs of individual classifiers in some optimal way yielded an accuracy of 91%, and (ii) *passive transformation* methods, and in particular ones that assume there is an uber expert available to provide adjudication on at least a small fraction of the samples which can be used to train a machine learning algorithm yielded an accuracy of 93% when 10% of randomly chosen training samples were classified by an expert and 94% with 70% training samples. Considering the significant cost of labeling by an expert this is not very promising. However, when the examples to be labeled by the expert were chosen carefully, even 17% of labeled samples gave us accuracy 95.4%. This motivated us to consider the area of *active learning* (AL) in which examples to be labeled by the expert are chosen carefully. By adapting several existing AL algorithms to an ensemble setting, the resulting *active transformation*

method yieled an accuracy of 94.5% with 10% labeled samples. We went on to consider the following question: is it possible develop an active learning algorithm that can further improve the classification accuracy of an ensemble ? This is precisely the subject of this chapter.

4.3 Introduction

Active learning (AL) is a paradigm of machine learning based on the idea that learning can be greatly improved if a learner is allowed to choose the data *from* which it learns, as opposed to the traditional *passive* learning paradigm where examples are presented *to* the learner (4.1). Intuitively, the advantage comes from the assumption that labeling of data is often the most expensive operation in the learning process and thus overall cost can be reduced if the learner is able to request only those labels that improve its learning ability. AL is closely related to semi-supervised learning (SSL), where learner is able to learn from a vast pool of unlabeled data in addition to a small number of labeled examples. AL goes one step further in that it is able to determine which labels need to be acquired. While both SSL and AL address the issue of expensive labeling, SSL makes use of information it can derive from unlabeled instances by selecting instances with least ambiguity or most confidence, whereas AL explores the unknown aspects of the unlabaled data and selects instances with most ambiguity or least confidence [198, 90]. In other words, while SSL attempts to select instances that are most *representative*, AL selects instances that are most *informative*[218].

AL has been an active area of research since the early 1990s though most AL algorithms work well when data are separable (non-noisy). However, for non-separable (noisy) data, some of these algorithms based upon generative base learners can do well if the noise distribution is known but break down when the noise distribution is unknown, i.e. when there is *agnostic* noise. In some such cases AL performs worse than passive learning. AL that uses uncertainty sampling often fails on noisy data ([198, 4]) because the chosen active samples are not always representative of the unlabeled dataset and the algorithm can become overconfident about unexplored but representative regions. Several algorithms have been developed that balance the trade-off between "informativeness" and "representativeness" of the sample chosen to be queried to obviate the problem of sampling bias inherent in uncertainty based sampling. One class of problems uses SSL for the purpose of being able to exploit representativeness of unlabeled data using a base SSL classifier. However, these algorithms are applicable only in specific application contexts ([107, 169, 214, 218, 240, 34]), work with only

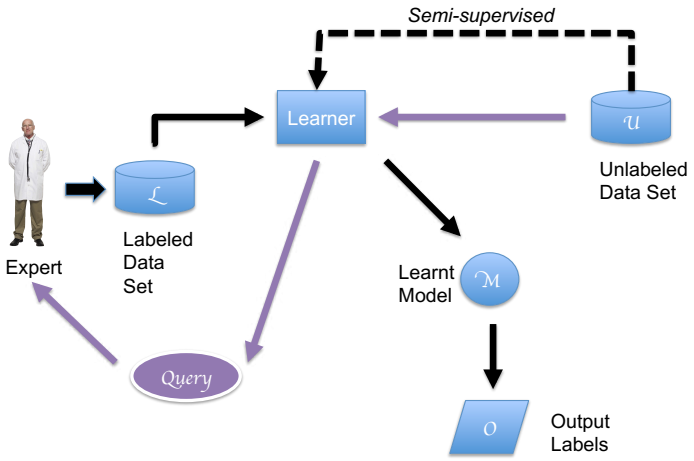


Figure 4.1: The paradigms of *(passive) supervised, (passive) semi-supervised*, and *active* learning. In *(passive) supervised learning* (solid black arrows), an examples are pre-selected at random to be labeled by an expert that are then used as examples to train a learner. In *(passive) semi-supervised learning* (black dashed arrow) the unlabeled samples are also used to train a learner. In *active learning* (purple arrows) the learner determines a subset of unlabeled examples to be be queried the expert to be used as its training examples.

specific base learners ([200, 196, 128, 76]), are computationally expensive ([200, 253, 103]) or do not guarantee performance in noisy cases ([60, 61]). In fact, several of these approaches showed no improvement in accuracy improvement over standard uncertainty based AL when applied to our ensemble problem.

Current AL methods base the active selection current unclassified and labeled *inputs*. We have developed a low complexity algorithmic framework termed *Output based Active Selection (OAS)* (Fig 4.6) that combines these inputs with predicted *output* to give an informative active measure and improves accuracy in noisy cases, in particular ensemble classification. Our idea starts with the observation that uncertain samples can often be noisy, and so acquiring labels for them and retraining the classifier can actually make its prediction worse. However, acquiring labels for such noisy samples may be obviated if the classifier's own output were also used in determining active samples. To make this intuition clear, consider the scenario shown in a simple toy example in Fig 4.2. The best hypothesis (Fig 4.2, right panel) shows one noisy data point (the red circle above the blue line). Starting with the shown initial training samples, standard US-based AL queries a noisy data point based on uncertainty which results in the classifier learning an inaccurate hypothesis in the subsequent iteration. As shown in this example, after two iterations the learnt model will have misclassi-

fied 6 out of 14 samples. However, in the case of OAS, the two most uncertain samples would have the same predicted output, so instead the algorithm chooses to query two uncertain samples with different predicted outputs resulting in no misclassification. While this example is a bit contrived, it conveys the idea that by combining model prediction uncertainty with actual predictions, the effect of querying of pathological samples can be mitigated.

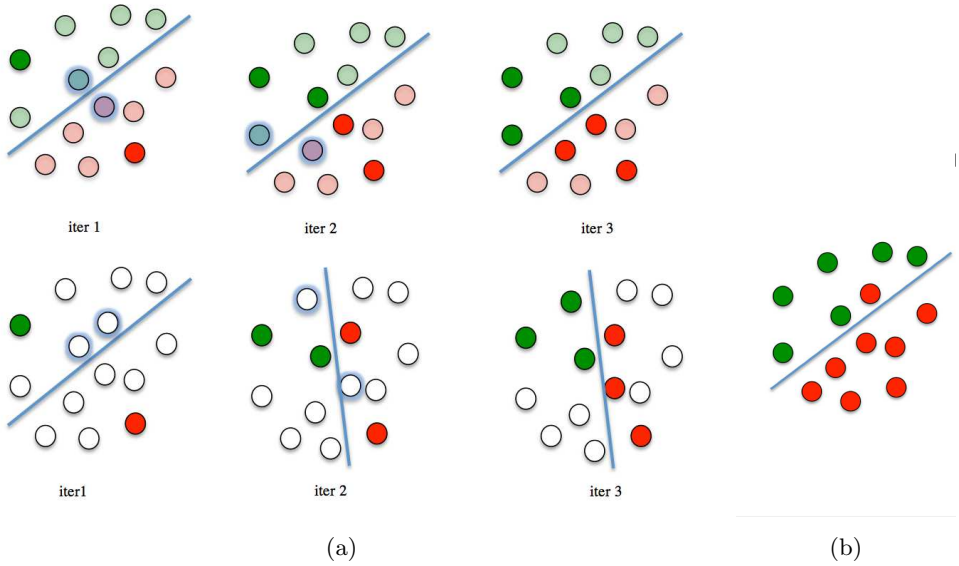


Figure 4.2: A toy example demonstrating OAS. (a) *Bottom* Three iterations of traditional uncertainty based sampling (US) that uses only unlabeled features to determine active candidates. Solid green and red filled circles indicate already labeled samples, white circles are unlabeled samples, blue solid line is the current learnt hypothesis, and shadowed white circles are active candidates for the current iteration. *Top* Three iterations of output-based active selection (OAS). In addition to the information used by US above, the current predicted output (light green and red), is used to determine the active candidates. (B) The actual labels (red and green) and the optimal hypothesis amongst the class of linear separators (blue line).

We developed algorithms OASL and OASSL on the aforementioned idea for both SL and SSL base learners and outperformed state-of-the-art AL methods on simulated Gaussian mixtures with and without agnostic nosie. We also developed adaptive versions (OASL-A, OASSL-A) that switch to passive learning when it is detected that active querying is of no additional help. Our implementation uses an E-M based classifier internally; this classifier is intrinsically based on ensemble inputs. This makes our algorithms amenable to the problem of ensemble clustering discussed in section 4.2. When applied to the problem of ensemble classification for the EEG artifact detection, OASL-A demonstrated 97.5% accuracy which outperforms all other methods we tried for this problem.

While AL spans a diversity of application areas in natural language processing [171] bioinformatics [107, 85] and even pedestrian detection [2] , applications to the areas of biomedical signal processing are fairly limited. In [14, 188] AL was applied to seizure detection using intracranial EEG where a standard margin based AL algorithm was applied using an SVM base learner. To our knowledge the only application of AL for EEG artifact labeling is in [139, 235] where features extracted from EEG using an AR model are fed to a Query-based-committee (QBC) based active learning algorithm using support vector machine with radial basis function kernel (SVM-RBF) as the base learner. This problem, however, does not address our ensemble classification problem, and to our knowledge our approach is the only one to address active learning for ensemble clustering.

4.4 Formulation and Base Model

4.4.1 Definitions

We formalize the different types of problems we will be addressing. For our purposes we restrict our exposition to binary classification only. We are given a *feature space* $\mathbf{X} \subseteq \mathbb{R}^d$, a measurable set, and an *output space* $\mathbf{Y} = \{0, 1\}$. Features are then represented by the random variable $X \in \mathbf{X}$ and class (output) by $Y \in \mathbf{Y}$. Let $p_{XY}(x, y)$ be the (unknown) joint distribution of (X, Y) and $p_X(x)$ be the marginal density of X, and the set of *hypotheses* $H = \{h : \mathbf{X} \to \mathbf{Y}\}$. Technically we require h to be measurable.

Let $C : Y \times Y \to R$ be a given *loss function*. For example, the $0 - 1$ loss function is $C_0(y, y') = I_{y \neq y'}$ where I_A denotes the indicator function over the set A. Then the *Bayes' optimal classifier* is the hypothesis h^* defined as

$$h^* = \arg\min_{h \in H} \mathbb{E}_{X,Y} \left[C(Y, h(X)) \right]$$

where $\mathbb{E}_{X,Y}$ denotes the expectation operator with respect to the probability measure p_{XY}. We will drop the subscript X, Y when there is no confusion. For $C = C_0$ we get

$$\begin{aligned} h^* &= \arg\min_{h \in H} \Pr(h(X) \neq Y) \\ &= \arg\max_{y} p_{XY}(x, y) \end{aligned}$$

where $\Pr(A)$ above denotes the probability of set A with respect to the probability measure p_{XY}. For our binary classification task, one can injectively map any measurable h to a measurable set $G \subseteq \mathbf{X}$, $h(x) = I_G(x)$. Then the Bayes optimal classifier can be equivalently

formulated in terms of the measurable set G^*

$$G^* = \arg \min_{G \subseteq \mathbf{X}} \Pr(I_G(X) \neq Y)$$

Since p_{XY} is not known, one needs to form an estimate $\widehat{h}$ of h^* that minimizes the excess error in *some specified subclass of hypotheses*, where excess error is

$$\epsilon(\widehat{h}) = \mathbb{E}(C(Y, \widehat{h}(X)) - \mathbb{E}(C(Y, h^*(X))$$

Some definitions are as follows. In **Supervised Learning (SL),** the classifier $\widehat{h}$ is constructed given L labeled samples $\mathcal{L} = \{x_j, y_j\}_{j=1}^{L}$ where each $(x_j, y_j) \sim^{\text{iid}} p_{XY}$. This is sometimes also known as *passive learning.* In **Unsupervised Learning (UL),** the classifier $\widehat{h}$ is constructed given U unlabeled samples $\mathcal{U} = \{x_j\}_{j=1}^{U}$ where each $x_j \sim^{\text{iid}} p_X$. In the case of **Semi-Supervised Learning (SSL)**, the classifier $\widehat{h}$ is constructed given L labeled samples $\mathcal{L} = \{x_j, y_j\}_{j=1}^{L}$ and U unlabeled samples $\mathcal{U} = \{x_j\}_{j=L+1}^{U+L}$ where each $(x_j, y_j) \sim^{\text{iid}} p_{XY}$ and $x_j \sim^{\text{iid}} p_X$. In **Active Learning (AL)** we are given $\mathcal{U} = \{x_j\}_{j=1}^{U} \sim^{\text{iid}} p_X$ and we *select* $\mathcal{L} = \{x_j, y_j\}_{j=1}^{L}$ to design $\widehat{h}$ where each x_j in $\mathcal{L}$ is in $\mathcal{U}$. Active learning where $\widehat{h}$ is designed using *both* $\mathcal{U}$ and $\mathcal{L}$ is called **Active Semi-supervised Learning (ASSL).**

The problem of **Ensemble Clustering (EC)** is defined as follows. We are given an unlabeled set $\mathcal{U} = \{x_j\}_{j=1}^{N}$ and the output of C classifiers (sometimes called *weak classifiers*) for each sample j. The labels y_j are unknown. This output for sample j is assumed to be the 2-tuple (u_j, v_j) where $u_j \in \mathbf{Y}$ is the "crisp label" output and $0 \leq v_j \leq 1$ is the "soft" or "confidence" output (typically, classifier c' s estimate of $p(y_j = 1|x_j)$). Then we define our *ensemble* as the set $\mathcal{E} = \{h_1 ... h_C\}$ where each $h_c : \mathcal{U} \rightarrow \mathbf{Y}$ is defined by

$$h_c(x_j) = (u_j, v_j)$$

Note that ensemble members h_c are defined on the domain $\mathcal{U}$ and not the entire feature space $\mathbf{X}$. This means that the weak classifiers can not be re-trained; this distinction will be important for rest of our analysis. For instance, this means that several of standard ensemble methods which can re-generate ensemble members are not applicable to our problem at hand. The problem of *Ensemble Clustering (EC)* is to construct a hypothesis $\widehat{h}$ of h^* given $\mathcal{U}$ and $\mathcal{E}$. A particular type of EC is *Transformation-based* EC where the problem is transformed into an UL problem by defining $\widetilde{x}_j = (h_1(x_j), ..., h_C(x_j), x_j)$ and working on the transformed space $\widetilde{x}_j$.

We are now ready to define **Active Semi-supervised Learning for Ensemble Clustering (ASSL-EC)**. Given $\mathcal{U} = \{x_j\}_{j=1}^{N}$, ensemble $\mathcal{E} = \{h_1 ... h_C\}$, an oracle O :

$\mathbf{X} \to \mathbf{Y}$ that produces the true label y_j for any sample j, and the maximum number K of times the oracle can be queried for a label, we wish to (i) find a set of *active* samples $\mathcal{A} \subset \mathcal{U}$ with $|\mathcal{A}| \leq K$ and (ii) construct a semi-supervised classifier $\widehat{h}$ that uses $\mathcal{U} \backslash \mathcal{A}$ as its unlabeled set and $\mathcal{L} = \{(x_{i_k}, O(x_{i_k}\}_{k=1}^{K}$ as the labeled set to predict label $\widehat{y_j} = \widehat{h}(x_j)$ for $x_j \in \mathcal{U} \backslash \mathcal{A}$.

4.4.2 Bernoulli-Gaussian Mixture Model for Ensemble Learning

We describe the model that is used as the base learner in our algorithms. This model is an example of a *generative* classifier, i.e. where p_{XY} is estimated parametrically from input data, given labeled examples $\{(x_j, y_j)\}_{j=1}^{L}$, unlabeled features $\{x_j\}_{j=L+1}^{L+U}$, $\{u_j\}_{j=1}^{U}$ and $\{v_j\}_{j=1}^{U}$ where $x_j \in \mathbb{R}^D$, $y_j \in \{0, 1\}$, $u_j \in \{0, 1\}^K$ and $v_j \in \mathbb{R}^K$ with the assumptions:

1. (x_j, y_j, u_j, v_j) are jointly iid , that is $(x_i, y_i, u_i, v_i) \perp (x_j, y_j, u_j, v_j)$ for $i \neq j$
2. u_j, v_j are independent of x_j
3. u_j, v_j are only component-wise dependent: $u_{jk} \perp v_{il}$, $u_{jk} \perp u_{jl}, v_{jk} \perp v_{jl}$ for $k \neq l$.
4. x_j distribution is a 2 component Gaussian mixture: $(x_j | y_j = c) \sim N(x_j; \mu_c, \Sigma_c), c = 0, 1$
5. Distribution of (u_j, v_j) are determined as follows. For $c = 0, 1$ and $m = 0, 1$, (here $B(y; \pi)$ denotes a Bernoulli distribution with success probability π):

$$\begin{aligned} (u_{jk} | y_j = c) &\sim B(u_{jk}; \pi_{ck}) \\ (v_{jk} | y_j = c, u_{jk} = m) &\sim N(v_{jk}; \lambda_{cmk}, \sigma^2_{cmk}) \end{aligned}$$

The above assumptions fully specify the joint distribution $p(x, y, u, v)$ with parameters, for $c = 0, 1$ as α_c (mixture proportions), $\Sigma_c \in \mathbb{R}^{d \times d}, \mu_c \in \mathbb{R}^d, \pi_c \in [0\ 1], \lambda_c \in \mathbb{R}^{K \times 2}, \sigma_c \in R^{K \times 2}$ where we have written (with slight notation abuse) λ_c as the matrix of elements λ_{cmk} and σ_c as the matrix of elements σ_{cmk}. These parameters can be estimated by maximizing the likelihood of the data

$$L(x_1 ... x_{L+U}, u_1 .. u_{L+U}, v_1 .. v_{L+U}, y_1, .. y_L) = \log \prod_{j=1}^{L} p(x_j, u_j, v_j, y_j) \prod_{j=L+1}^{L+U} p(x_j, u_j, v_j)$$

using E-M with $y_j, j = L + 1 .. L + U$ as hidden data. We consider some special cases first.

1. No Ensemble Data, Unsupervised (GMM): This is the simplest case where we assume $L = 0$, $\{u_j\} = \phi$ and $\{v_j\} = \phi$. This is the case of the well-known Gaussian mixture model, and the E-M equations for this case is well known (see Online Supplemental Material). The E-step is

$$Q(\theta, \theta^t) = \sum_{j=1}^{N} \sum_{c=0}^{1} T_{j,c}^{(t)} \left[\log \alpha_c - \frac{1}{2} \log |\Sigma_c| - \frac{1}{2} (x_j - \mu_c)^T \Sigma_c^T (x_j - \mu_c) \right]$$

where

$$T_{j,c}^{(t)} = p(y_j = c | x_j, \theta^t) = \frac{N(x_j; \mu_c^{(t)}, \Sigma_c^{(t)}) \alpha_c^{(t)}}{\sum_{c=0}^{2} N(x_j; \mu_c^{(t)}, \Sigma_c^{(t)}) \alpha_c^{(t)}} \tag{4.1}$$

and the M-step is

$$\mu_c^{(t+1)} = \frac{\sum_{j=1}^{N} T_{j,c}^{(t)} x_j}{\sum_{j=1}^{N} T_{j,c}^{(t)}} \tag{4.2}$$

$$\Sigma_c^{(t+1)} = \frac{\sum_{j=1}^{N} T_{j,c}^{(t)} (x_j - \mu_c^{(t+1)})((x_j - \mu_c^{(t+1)})^T}{\sum_{j=1}^{N} T_{j,c}^{(t)}} \tag{4.3}$$

$$\alpha_c^{(t+1)} = \frac{\sum_{j=1}^{N} T_{j,c}^{(t)}}{\sum_{c=0}^{1} \sum_{j=1}^{N} T_{j,c}^{(t)}} \tag{4.4}$$

We will also be interested in the case where Σ_c is block diagonal, as in:

$$\Sigma_c = \begin{bmatrix} \Sigma_{1,c} & & \\ & ... & \\ & & \Sigma_{K,c} \end{bmatrix}$$

in which case the M-step for Σ_c is slightly modified to:

$$\Sigma_{k,c}^{(t+1)} = \frac{\sum_{j=1}^{N} T_{j,c}^{(t)} (x_{k,j} - \mu_{k,c}^{(t+1)})((x_{k,j} - \mu_{k,c}^{(t+1)})^T}{\sum_{j=1}^{N} T_{j,c}^{(t)}}$$

where $x_j = [x_{1,j} \;\; x_{2,j} \;\; ... \; x_{K,j}]$ and $\mu_j = [\mu_{1,j} \;\; \mu_{2,j} \;\; ... \; \mu_{K,j}]$.

2. No Ensemble Data, Semi-supervised (GMSS) In this case we have $\{u_j\} = \phi$ and $\{v_j\} = \phi$ but $L \neq 0$. Let $N = L + U$ Using variables $y = (y_{L+1} ... y_{L+U})$ as hidden variables, the E-M steps for estimation of parameters $\theta = (\alpha, \mu_0, \Sigma_0, \mu_1, \Sigma_1)$ are as follows (details on Online Supplemental Material). The E-step is

$$Q(\theta, \theta^t) = \sum_{j=1}^{N} \sum_{c=0}^{1} \gamma_{j,c}^{(t)} \left[\log \alpha_c - \frac{1}{2} \log |\Sigma_c| - \frac{1}{2} (x_j - \mu_c)^T \Sigma_c^T (x_j - \mu_c) \right]$$

where with $T_{j,c}^{(t)}$ as in (4.1) and

$$\gamma_{j,c}^{(t)} == \begin{cases} I(y_j = c) & \text{if} \quad 1 \leq j \leq L \\ T_{j,c}^{(t)} & \text{if} \quad L+1 \leq j \leq L+U \end{cases} \tag{4.5}$$

The M-step is

$$\begin{aligned} \mu_c^{(t+1)} &= \frac{\sum_{j=1}^N \gamma_{j,c}^{(t)} x_j}{\sum_{j=1}^N \gamma_{j,c}^{(t)}} \\ \Sigma_c^{(t+1)} &= \frac{\sum_{j=1}^N \gamma_{j,c}^{(t)} (x_j - \mu_c^{(t+1)})((x_j - \mu_c^{(t+1)})^T}{\sum_{j=1}^N \gamma_{j,c}^{(t)}} \\ \alpha_c^{(t+1)} &= \frac{\sum_{j=1}^N \gamma_{j,c}^{(t)}}{\sum_{c=0}^1 \sum_{j=1}^N \gamma_{j,c}^{(t)}} \end{aligned} \tag{4.6}$$

The case of block diagonal variances is handled analogous to the unsupervised case. For one of our AL implementations (importance-weighting) we will require the samples to have weights $w_1 ... w_N$, in which case we need to minimize the weighted likelihood

$$\log p(x, y|\theta) = \sum_{j=1}^N w_j \log p(x_j, y_j|\theta) \tag{4.7}$$

All the above analysis remains the same and we get the same formulae (4.6) if we use $w_j \gamma_{j,c}^{(t)}$ instead of $\gamma_{j,c}^{(t)}$. Note the weights need not be normalized because the formulae (4.6) are scale invariant.

3. No Ensemble Data, Supervised (GMS) In this case $\{u_j\} = \phi$ and $\{v_j\} = \phi$ and $L = 0$. In this case we can still use the learning rules (4.5) and 4.6) and since $\gamma_{j,c}^{(t)}$ does not depend on t we algorithm reduces to the standard maximum likelihood estimation using sample mean and sample variance for a multivariate Gaussian.

4. Ensemble Outputs Only - Unsupervised (BMM) In this case $L = 0$, $\{x_j\} = \phi$, $\{v_j\} = \phi$. That is, we only have the binary ensemble outputs $u_j \in [0,1]^K$. This case corresponds to the standard transformation based clustering ensemble algorithm [251, 96, 169]. As before, estimation is done using E-M with $\{y_j\}_{j=1}^U$ as hidden variables. The distribution is specified as $(u_j|y_j = c) \sim \prod_{k=1}^K B(u_{jk}; \mu_{ck})$ where $B(x;p)$ is the standard Bernoulli distribution i.e. $B(x;p) = p^x(1-p)^{1-x}$ and for short-hand we write $B(u_j; \mu_c) = \prod_{k=1}^K \mu_{ck}^{u_{jk}} (1-\mu_{ck})^{1-u_{jk}}$.

Then the parameters to be estimated are $\Theta = \{\alpha_c, \mu_c, c = 0, 1\}$ where $\mu_c = \{\mu_{c1}, ...\mu_{cK}\}$ and $\Sigma_c \alpha_c = 1$.The E-step in this case is

$$Q(\theta, \theta^t) = \sum_{j=1}^{N} \sum_{c=0}^{1} T_{j,c}^{(t)} \left[\log \alpha_c + \sum_{k=1}^{K} u_{jk} \log \mu_{ck} + (1 - u_{jk}) \log(1 - \mu_{ck}) \right] \tag{4.8}$$

where, as before,

$$T_{j,c}^{(t)} = p(y_j | u_j, \theta^t) = \frac{B(u_j; \mu_c^{(t)}) \alpha_c^{(t)}}{\sum_{c=0}^{1} B(u_j; \mu_c^{(t)}) \alpha_c^{(t)}}$$

and the M-step is

$$\begin{aligned} \mu_c^{(t+1)} &= \frac{\sum_{j=1}^{N} T_{j,c}^{(t)} u_j}{\sum_{j=1}^{N} T_{j,c}^{(t)}} \\ \alpha_c^{(t+1)} &= \frac{\sum_{j=1}^{N} T_{j,c}^{(t)}}{\sum_{c=0}^{1} \sum_{j=1}^{N} T_{j,c}^{(t)}} \end{aligned}$$

5. Ensemble Outputs Only - Semi-Supervised (BMSS). In this case $\{x_j\} = \phi$, $\{v_j\} = \phi$ but $L > 0$. By analogy to the GMM case, the M-step becomes (E-step same as 4.8)

$$\begin{aligned} \mu_c^{(t+1)} &= \frac{\sum_{j=1}^{N} \gamma_{j,c}^{(t)} u_j}{\sum_{j=1}^{N} \gamma_{j,c}^{(t)}} \\ \alpha_c^{(t+1)} &= \frac{\sum_{j=1}^{N} \gamma_{j,c}^{(t)}}{\sum_{c=0}^{1} \sum_{j=1}^{N} \gamma_{j,c}^{(t)}} \end{aligned} \tag{4.9}$$

where

$$\gamma_{j,c}^{(t)} == \begin{cases} I(y_j = c) & \text{if} \quad 1 \le j \le L \\ T_{j,c}^{(t)} & \text{if} \quad L + 1 \le j \le L + U \end{cases}$$

6. Ensemble Outputs Only - Supervised (BMS). As in the Gaussian case, the learning rules for BMSS (4.9) still apply, with $U = 0$ so that $\gamma_{jc}^{(t)}$ does not depend on t and it reduces to the standard MLE estimation of μ using sample mean and for that of α using class proportion.

7. Ensemble Outputs and Features, Unsupervised, Independent Case (BGMMi). Here we have $L = 0$ and we make the additional assumption that u_j are independent of v_j, x_j. Then, by writing $\widetilde{x}_j = (v_j \; x_j)$ we have u_j as a jointly Bernoulli mixture, and $(v_j \; x_j)$ as a jointly Gaussian mixture with block diagonal covariance. That is, $(u_j | y_j = c) \sim B(u_j; \pi_c)$

and $(\widetilde{x}_j|y_j = c) \sim N(x_j; \mu_c, \Sigma_c)$ with $\Sigma_c \in R^{(K+d)\times(K+d)}$as block diagonal, with blocks of size K and d. The parameters to be estimated are $\Theta = \{\alpha_c, \mu_c, \Sigma_c, \pi_c\}$ for $c = 0, 1$. Here $\mu_c \in R^{K+d}$ and $\pi_c \in \{0, 1\}^K$. Since all variables are independent, the solution is similar to the GM or Bernoulli cases with the E-step:

$$T_{j,c}^{(t)} = \gamma_{j,c}^{(t)} = \frac{N(\widetilde{x}_j; \mu_c, \Sigma_c)B(u_j; \mu_c^{(t)})\alpha_c^{(t)}}{\sum_{c=0}^{1} N(\widetilde{x}_j; \mu_c, \Sigma_c)B(u_j; \mu_c^{(t)})\alpha_c^{(t)}} \tag{4.10}$$

and the M-step:

$$\begin{aligned}
\pi_c^{(t+1)} &= \frac{\sum_{j=1}^{N} \gamma_{j,c}^{(t)} u_j}{\sum_{j=1}^{N} \gamma_{j,c}^{(t)}} \\
\mu_c^{(t+1)} &= \frac{\sum_{j=1}^{N} \gamma_{j,c}^{(t)} \widetilde{x}_j}{\sum_{j=1}^{N} \gamma_{j,c}^{(t)}} \\
\Sigma_c^{(t+1)} &= \frac{\sum_{j=1}^{N} T_{j,c}^{(t)} (\widetilde{x}_j - \mu_c^{(t+1)})((\widetilde{x}_j - \mu_c^{(t+1)})^T}{\sum_{j=1}^{N} T_{j,c}^{(t)}} \\
\alpha_c^{(t+1)} &= \frac{\sum_{j=1}^{N} \gamma_{j,c}^{(t)}}{\sum_{c=0}^{1} \sum_{j=1}^{N} \gamma_{j,c}^{(t)}}
\end{aligned} \tag{4.11}$$

8. Ensemble Outputs and Features, Semi-supervised, Independent Case (BGMSSi) As before, simply use the definition of $\gamma_{j,c}^{(t)}$ as in (4.5) with $T_{j,c}^{(t)}$ as defined in (4.10) and learning rule (4.11).

9. Ensemble Outputs and Features, Supervised, Independent Case (BGMSi) As before, simply use the definition of $\gamma_{j,c}^{(t)} = I(y_j = c)$ in the learning rule (4.11).

10. General Case (BGMM, BGMS and BGMSS) The definition of this case was at the beginning of this section. Recall, that the parameters of interest here are: α_c (mixture proportions), for $c = 0, 1$, α_c , $\Sigma_c \in \mathbb{R}^{d\times d}, \mu_c \in \mathbb{R}^d, \pi_c \in [0\ 1], \lambda_c \in \mathbb{R}^{K\times 2}, \sigma_c \in R^{K\times 2}$. We write

$$\begin{aligned}
p(u_j, v_j|y_j &= c) = \prod_{k=1}^{K} \left(N(v_{jk}; \lambda_{c1k}, \sigma_{c0k}^2)\pi_{ck}\right)^{u_{jk}} \left(N(v_{jk}; \lambda_{c1k}, \sigma_{c0k}^2)(1 - \pi_{ck})\right)^{1-u_{jk}} \\
&\equiv NB(u_j, v_j; \lambda_c, \sigma_c, \pi_c)
\end{aligned}$$

As before, writing

$$\begin{aligned} T_{j,c}^{(t)} &= \gamma_{j,c}^{(t)} = p(y_j|x_j,u_j,v_j,\theta^t) \\ &= \frac{NB(u_j,v_j;\lambda_c^{(t)},\sigma_c^{(t)},\pi_c^{(t)})N(x_j;\mu_c^{(t)},\Sigma_c^{(t)})\alpha_c^{(t)}}{\sum_{c=0}^{1} NB(u_j,v_j;\lambda_c^{(t)},\sigma_c^{(t)},\pi_c^{(t)})N(x_j;\mu_c^{(t)},\Sigma_c^{(t)})\alpha_c^{(t)}} \end{aligned}$$

the E step is

$$\begin{aligned} Q(\theta,\theta^t) &= E_{y|x,\theta^t}\log p(x,y|\theta) \\ &= \sum_{j=1}^{N}\sum_{c=0}^{1} T_{j,c}^{(t)}[\log\alpha_c - \frac{1}{2}\log|\Sigma_c| - \frac{1}{2}(x_j-\mu_c)^T\Sigma_1^T(x_j-\mu_c) + \\ &\quad \sum_{k=1}^{K} u_{jk}\left(\log\pi_{ck} - \frac{1}{2}\log\sigma_{c0k}^2 - \frac{1}{2\sigma_{c1k}^2}(v_{jk}-\lambda_{c0k})^2\right) + \\ &\quad \sum_{k=1}^{K}(1-u_{jk})\left(\log(1-\pi_{ck}) - \frac{1}{2}\log\sigma_{c1k}^2 - \frac{1}{2\sigma_{c1k}^2}(v_{jk}-\lambda_{c1k})^2\right)] \end{aligned}$$

For the M step we can optimize per parameter, giving us the following

$$\begin{aligned} \mu_c^{(t+1)} &= \frac{\sum_{j=1}^N \gamma_{j,c}^{(t)} x_j}{\sum_{j=1}^N \gamma_{j,c}^{(t)}} \\ \Sigma^{(t+1)} &= \frac{\sum_{j=1}^N \gamma_{j,c}^{(t)}(x_j-\mu_c^{(t+1)})((x_j-\mu_c^{(t+1)})^T}{\sum_{j=1}^N \gamma_{j,c}^{(t)}} \\ \alpha_c^{(t+1)} &= \frac{\sum_{j=1}^N \gamma_{j,c}^{(t)}}{\sum_{c=0}^1\sum_{j=1}^N \gamma_{j,c}^{(t)}} \\ \pi_c^{(t+1)} &= \frac{\sum_{j=1}^N \gamma_{j,c}^{(t)} u_j}{\sum_{j=1}^N \gamma_{j,c}^{(t)}} \\ \lambda_{c0k}^{(t+1)} &= \frac{\sum_{j=1}^N \gamma_{j,c}^{(t)} u_{jk} v_{jk}}{\sum_{j=1}^N \gamma_{j,c}^{(t)} u_{jk}};\ \lambda_{c1k}^{(t+1)} = \frac{\sum_{j=1}^N \gamma_{j,c}^{(t)}(1-u_{jk}) v_{jk}}{\sum_{j=1}^N \gamma_{j,c}^{(t)}(1-u_{jk})} \\ \sigma_{c0k}^{(t+1)} &= \frac{\sum_{j=1}^N \gamma_{j,c}^{(t)} u_{jk}(v_{jk}-\lambda_{c1k}^{(t+1)})^2}{\sum_{j=1}^N \gamma_{j,c}^{(t)} u_{jk}}; \sigma_{c1k}^{(t+1)} = \frac{\sum_{j=1}^N \gamma_{j,c}^{(t)}(1-u_{jk})(v_{jk}-\lambda_{c2k}^{(t+1)})^2}{\sum_{j=1}^N \gamma_{j,c}^{(t)}(1-u_{jk})} \end{aligned} \tag{4.12}$$

The semi-supervised case (BGMSS), in analogy with the previous cases, use the learning rule (4.12) replacing $\gamma_{j,c}^{(t)}$ as in (4.5) and in the weighted case using $w_j\gamma_{j,c}^{(t)}$ instead of $\gamma_{j,c}^{(t)}$. And in the supervised case (BGMS) use (4.12) replacing $\gamma_{j,c}^{(t)}$ with $I(y_j = c)$.

4.5 Existing Methods For Active Learning

This section reviews some existing AL algorithms that we evaluated. These algorithms are *batch pool-based* ([198]) that is, where are given the entire pool of unlabeled examples to choose from instead of a streaming sequence of examples and samples are chosen iteratively for active labeling in batches. Only algorithms that were readily adaptable to E-M base learners for purposes of Gaussian mixture classification were evaluated. Thie excludes several classes of algorithms such as expected model change, expected gradient length ([198]) and multi-view based ([252, 168, 230]). Hierarchical clustering ([60, 15]) is also not evaluated as it is known not to perform well in the presence of gnostic noise. For convenience, we categorize the evaluated algorithms into (i) selection based. (ii) committee based and (iii) importance weighted.

Selection Based AL Algorithms

A generic active learning algorithm is shown in 1, and a single iteration depicted in Fig 4.3. Here a base learner *Learn*, which can be supervised or semi-supervised, returns a trained model θ^* that is then used to determine what samples to actively query via a selection criterion *Select*. The labels returned by the *Oracle* for the selected samples are used as labeled examples to train the base learner in the next iteration. The different *Select* methods used are the following:

1. **Uncertainty:** For base learners (such as GMM, GMS) where the trained model θ has a classification rule based upon posterior probability $p_\theta(y|x)$ for a given sample x and the posterior probability is also an output of the model (in addition to the crisp label for the class), we can use a likelihood based uncertainty. That is, for a given sample x if the classification rule is

$$\widehat{y} = \arg \max_{y \in \{0,1\}} p_\theta(y|x) \tag{4.13}$$

then the samples $x_{i_1}...x_{i_B}$ for active labeling are given by (B is the batch size)

$$\begin{aligned} i_1 &= \arg \min_j p_\theta(\widehat{y}|x_j) \\ i_k &= \arg \min_{j \neq i_1 ... i_{k-1}} p_\theta(\widehat{y}|x_j) \text{ for } k = 2...B \end{aligned}$$

2. **Margin:** For base learners (such as SVM) where the model θ is computed in the form of a decision boundary in the feature space, the samples that are furthest away

from the optimal decision boundary are selected as active samples. More formally (see [110, 107, 129]), $\theta^*(x) = sgn(f^*(x))$ where f^*, the optimal decision function, is obtained using training samples $\{x_j, y_j\}_{j=1}^{L}$ as:

$$f^* = \arg\min_{f \in \mathcal{H}_K} \frac{\lambda}{2} \|f\|^2_{\mathcal{H}_K} + \sum_{j=1}^{L} l(y_j, f(x_j)) \tag{4.14}$$

where $\mathcal{H}_K$ is the Hilbert space reproduced by a kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, λ is a regularization parameter and $l(\cdot, \cdot)$ is a loss function. For example, if a kernel SVM is used, then f is the hyperplane function of the form $f(x) = \sum_{j=1}^{L} \alpha_i K(x, x_j)$, and l is the hinge loss $l(u, v) = \max(0, 1 - uv)$ function. In this case $|f^*(x_j)|$ is the distance of x_j from the optimal decision boundary. A simple margin selection rule is then

$$\begin{aligned} i_1 &= \arg\min_j |f^*(x_j)| \\ i_k &= \arg\min_{j \neq i_1 \ldots i_{k-1}} |f^*(x_j)| \text{ for } k = 2 \ldots B \end{aligned}$$

Other margin based selection rules such as min-max margin, ratio margin ([216]) are also possible but we have not implemented those.

3. **Minimum Expected Error:** Samples which minimize the expected prediction error of the trained model $\theta^*$are selected or active querying. We evaluated the Manifold Adaptive Experimental Design (MAED) algorithm (See [34] for details) that works for a manifold learning (ML) based learner..

4. **Information Density (Similarity):** *Information Density* of the unlabeled samples is defined as ([200])

$$\phi(x) = \frac{1}{U} \sum_{j=L+1}^{L+U} sim(x, x_j) \tag{4.15}$$

and a function (of the form ϕ^β) is multiplied with the uncertainty (4.13) or margin function (4.14). We have used a cosine based similarity ([200]):

$$sim(x, x_j) = \frac{\langle x, x_j \rangle}{\|x\| \, \|x_j\|} \tag{4.16}$$

Based upon the combinations of various *Learn* and compatible *Select* functions we have tested the algorithms listed in Table 4.5.

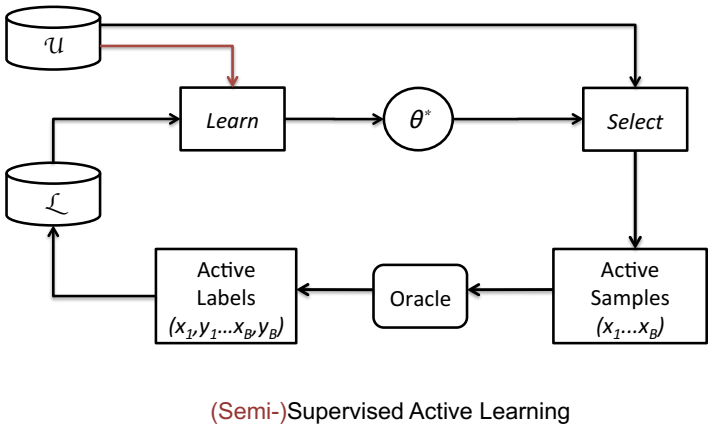


Figure 4.3: Schematic of a single iteration of a generic selection-based active learning algorithm. $\mathcal{U}$ is the unlabeled set, $\mathcal{L}$ is the labeled set, θ^* is the trained model. The red arrow from $\mathcal{U}$ to the Learn module is present in the semi-supervised case only. See algorithm 1.

Table 4.1: **Selection based AL algorithms.** *Learn* column is base learner and *Select* is the selection strategy used by the algorithm. text. Modulation by information density (4.15) is shown as "x Sim" alongside with the base strategy. We have included "Random" as a *Select* type in order to include passive learning in the list of algorithms that are tested. The Type column categorizes each of the algorithms as: A-SL (active supervised), A-SSL (active semi-supervised), P-SL (passive supervised) and P-SSL (passive semi-supervised).

Learn	*Select*	**Algorithm Name**	Type
GMS	Uncertainty	Active-SL	A-SL
GMSS	Uncertainty	Active-SSL	A-SSL
SVM-RBF	Margin	Active-Margin-SL	A-SL
ML	Min Expected Err	MAED	A-SSL
GMS	Uncertainty x Sim	Active-SL-SIM	A-SL
GMSS	Uncertainty x Sim	Active-SSL-SIM	A-SSL
SVM-RBF	Margin x Sim	Active-Margin-SL-SIM	A-SL
GMS	Random	Passive-SL	P-SL
GMSS	Random	Passive-SSL	P-SSL

Disagreement Based AL Algorithms

A generic disagreement based algorithm, sometimes also referred to as *Query-By-Committee (QBC)* ([1, 90, 158, 169]) is shown in 2, a single iteration of which is depicted in Fig 4.4. Here an ensemble of base learners, each of which can be supervised or semi-supervised, are trained. A measure of disagreement between the trained models,*Disagree,* on the set of

Algorithm 1: Generic Active Learning Using a SL (SSL) Base Learner

Given: Labeled set $\mathcal{L} = \{x_j, y_j\}_{j=1}^{L}$, Unlabeled set $\mathcal{U} = \{x_j\}_{j=L+1}^{L+U}$, Batch size B, Total number of active labels K

repeat
- $\theta^* =$ `Learn(`$\mathcal{L}$,$\mathcal{U}$`)` ; `// learn model using current` $\mathcal{L}$,$\mathcal{U}$
- $(x_{i_1}...x_{i_B}) =$ `Select(`θ^*,$\mathcal{U}$`)` ; `// select` B `informative samples`
- $(\ y_{i_1}...y_{i_B}) =$ `Oracle(`$x_{i_1}...x_{i_B}$`)` ; `// query oracle for labels`
- $\mathcal{L} = \mathcal{L} \cup \{(x_{i_1} y_{i_1}) \ldots (x_{i_K}, y_{i_K})\}$; `// update labeled set`
- $\mathcal{U} = \mathcal{U} \setminus (x_{i_1}...x_{i_B}\}$; `// update unlabeled set`

until K *unique active labels obtained*;

return θ^*, $\mathcal{L}$, $\mathcal{U}$;

unlabeled samples is used to determine which samples to actively query. Different flavors of the algorithm are possible, depending upon the type of *ensemble* method, the *base learner*, and the *disagreement* method (Table 4.2). We used only the bagging *ensemble* technique though others such as boosting ([1] and sampling ([90, 158, 169]) are possible. Sometimes when using bagging instead of sampling, this method is termed *Query-By-Bagging (QBB)* instead of *Query-By-Committee (QBC)*.For our data sets, we only test with GMS and GMSS base learners. The *Disagree* methods we tested are count, vote entropy and Kullback-Leibler divergence [199] In addition, the disagreement methods were optionally modulated using information density (4.15,4.16).

Importance Weighted AL Algorithms

An algorithm adapted from Importance Weighted Active Learning (IWAL) ([24, 248]) and Unbiased pool based active learning (UPAL) [95, 94] for GMS and GMSS base learners is shown in 3, a single iteration of which is depicted in Fig 4.5. The *WLearn* method is the base learning method for weighted samples (4.7). The function *QueryProbability* computes a distribution P_t on unlabeled samples $\mathcal{U}$ for the t^{th} iteration as:

$$P_t(x_j) = \frac{1}{Ut^{\kappa}} + (1 - \frac{1}{t^{\kappa}}) \frac{H(\eta_t(x_j)}{\sum_{j=L+1}^{L+U} H(\eta_t(x_j))} \tag{4.17}$$

where $\eta_t(x) = p_{\theta_t}(\widehat{y} = 1|x)$, $p_{\theta_t}(\widehat{y}|x)$ is the likelihood output of the trained base learner θ_t at iteration t, $H(p) = -(p \ln(p) + (1-p) \ln(1-p))$ and $\kappa = \frac{1}{2}$ ([95]). We call the two

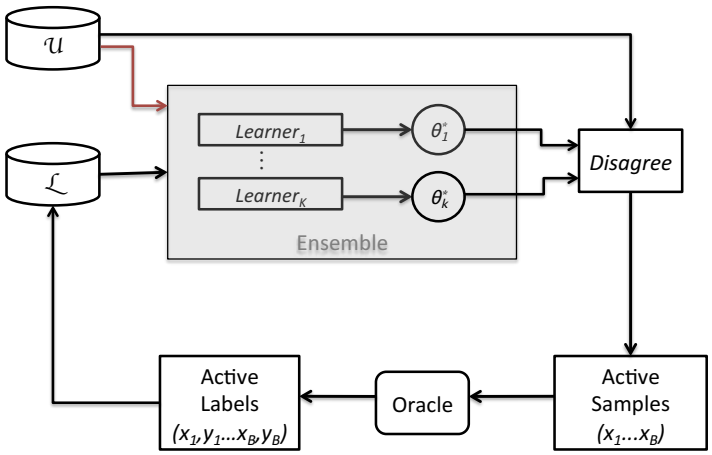


Figure 4.4: Schematic of a single iteration of QBC AL algorithm. $\mathcal{U}$ is the unlabeled set, $\mathcal{L}$ is the labeled set, $Learners_1 \cdots Learners_K$ are base learners in the ensemble and $\theta_1^* \cdots \theta_k^*$. their trained models. The red arrow from $\mathcal{U}$ to the individual learners is present in the SSL case only. See algorithm 2

Table 4.2: **Disagreement-based AL algorithms** *Ensemble* is the ensemble creation method, *Learner* is the base learner type in the ensemble, *Disagree* is the disagreement method. Modulation by information density (4.15) is shown as "x Sim" alongside with the base method. Type column is as in Table 4.5

Ensemble	Learner	Disagree	Name	Type
Bagging	GMS	Count	Active-QBB-SL-Count	A-SL
Bagging	GMSS	Count	Active-QBB-SSL-Count	A-SSL
Bagging	GMS	KL Divergence	Active-QBB-SL-KL	A-SL
Bagging	GMSS	KL Divergence	Active-QBB-SSL-KL	A-SSL
Bagging	GMS	Vote Entropy	Active-QBB-SL-VE	A-SL
Bagging	GMSS	Vote Entropy	Active-QBB-SSL-VE	A-SSL
Bagging	GMS	Count x Sim	Active-QBB-SL-Count-Sim	A-SL
Bagging	GMSS	Count x Sim	Active-QBB-SSL-Count-Sim	A-SSL
Bagging	GMS	KL Divergence x Sim	Active-QBB-SL-KL-Sim	A-SL
Bagging	GMSS	KL Divergence x Sim	Active-QBB-SSL-KL-Sim	A-SSL
Bagging	GMS	Vote Entropy x Sim	Active-QBB-SL-VE-Sim	A-SL
Bagging	GMSS	Vote Entropy x Sim	Active-QBB-SSL-VE-Sim	A-SSL

Algorithm 2: Generic QBC Using SL (SSL) Base Learners

Given: Labeled set $\mathcal{L} = \{x_j, y_j\}_{j=1}^{L}$, Unlabeled set $\mathcal{U} = \{x_j\}_{j=L+1}^{L+U}$, Batch size B, Total number of active labels K

repeat
  $(\texttt{Learner}_1, \ldots, \texttt{Learner}_K) = \texttt{GenerateEnsemble}(\mathcal{L})$;
  for $k =$ **to** K **do**
    $\theta_k^* = \texttt{Learner}_k(\mathcal{L}, \mathcal{U})$; // train using current $\mathcal{L}$,$\mathcal{U}$
  end
  $(x_{i_1} \ldots x_{i_B}) = \texttt{Disagree}(\theta_k^*, \cdots, \theta_k^*, \mathcal{U})$; // get B samples
  $(y_{i_1} \ldots y_{i_B}) = \texttt{Oracle}(x_{i_1} \ldots x_{i_B})$; // query oracle for labels
  $\mathcal{L} = \mathcal{L} \cup \{(x_{i_1} y_{i_1}) \ldots (x_{i_K}, y_{i_K})\}$; // update labeled set
  $\mathcal{U} = \mathcal{U} \setminus (x_{i_1} \ldots x_{i_B})$; // update unlabeled set
until K *unique active labels obtained*;
$\theta^* = \texttt{Learn}(\mathcal{L}, \mathcal{U})$; // train model using final $\mathcal{L}$,$\mathcal{U}$
return θ^*, $\mathcal{L}$, $\mathcal{U}$;

importance weighted algorithms that use GMS and GMSS as base learners **Active-IW-SL** and **Active-IW-SSL** respectively.

4.6 Existing Methods For Ensemble Clustering

The following *combination* and *transformation*-based ensemble clustering methods ([251]) are evaluated.

4.6.1 Combination Based Methods for EC

Given crisp label outputs $h_1(x) \ldots h_K(x) \in \{0, 1\}$ and soft outputs $v_1(x) \ldots v_K(x) \in [0, 1]$ (as described above) from K weak classifiers in the ensemble, these algorithms combine these outputs to generate the classifier $h^*(x)$ in one of several ways:

1. **Crisp Voting Methods:** In this case a decision rule of the form

$$h^*(x) = \begin{cases} 1 & \text{if} \quad F(h_1(x), \ldots, h_K(x)) \geq \frac{1}{2} \\ x & \text{otherwise} \end{cases}$$

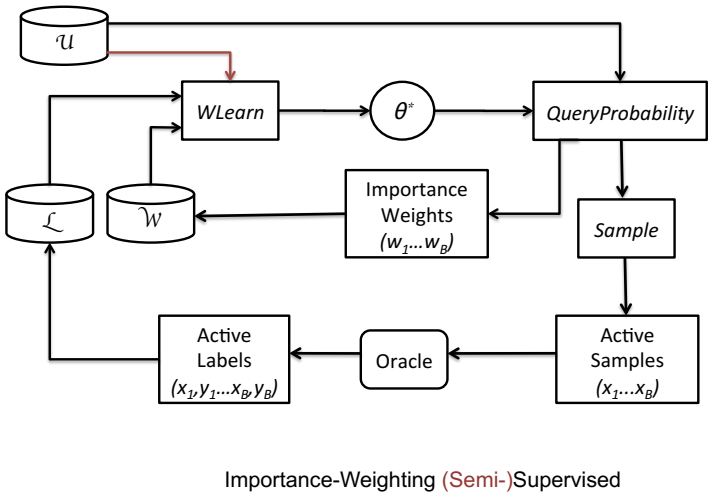


Figure 4.5: **A single iteration of an importance weighting algorithm.** $\mathcal{U}$ is the unlabeled set, $\mathcal{L}$ is the labeled set, $\mathcal{W}$ are the importance weights and θ^* is the trained model from *WLearn* using current labels and weights. *QueryProbability* is used to determine active samples and weights (4.17) The red arrow from $\mathcal{U}$ to the individual learners is present in the SSL case only. See algorithm 3.

is used. In **Majority Voting** (MV) $F(h_1(x),...,h_K(x)) = \frac{1}{K}\sum_{k=1}^{K} h_k(x)$. In **Weighted Voting (WV-1)** $F(h_1(x),...,h_K(x)) = \sum_{k=1}^{K} w_k h_k(x)$ where weights $w_1,..w_K$ are assigned using $w_k = \frac{\alpha_k}{\sum_{k=1}^{K}\alpha_k}$ while in **Weighted Voting (WV-2)** the weights are $w_k \propto \frac{\alpha_k}{1-\alpha_k}$. Here the the accuracy rate α_k of classifier k is determined using all labeled samples $\{x_j, y_j\}_{j=1}^{L}$.

2. **Soft Voting Methods.** Here the soft labels are combined using an averaging function $F(v_1,...v_K)$

$$h^*(x) = \begin{cases} 1 & \text{if} \quad F(v_1(x),...,v_K(x))) \geq \frac{1}{2} \\ x & \text{otherwise} \end{cases}$$

Depending on the averaging function, we get different algorithms. F can be the arithmetic mean **(SVavg)**, median (**SVmed**), maximum (**SVmax**), minimum (**SVmin**), harmonic mean (**SVhar**) and weighted arithmetic mean **(SWVavg)**.

3. **Decision Profile Based Methods.** Two methods, **Decision Template Method (DT)** and **Dempster-Schafer Method (DS)** make use of the *decision profile* $DP(x) = \begin{bmatrix} v_1(x) & v_2(x) & ... & v_K(x) \end{bmatrix}$ for instance x. These methods are reviewed in ([189, 180]) and also described in the Online Supplemental Material. We also used a slightly modi-

Algorithm 3: Importance Weighting for Generative SL (SSL) Base Learners

Given: Labeled set $\mathcal{L} = \{x_j, y_j\}_{j=1}^{L}$, Unlabeled set $\mathcal{U} = \{x_j\}_{j=L+1}^{L+U}$, Batch size B, Total number of active labels K

$t = 1$, $W(x_j) = 1$; `// Initialize counter and weights`

repeat

  $\theta^* =$ `WLearn(`$\mathcal{L}$,$\mathcal{U}$,W`)` ; `// use current` $\mathcal{L}$,$\mathcal{U}$ `and weights`

  $P_t =$ `QueryProbability(`θ^*,$\mathcal{U}$`)` ; `// compute distribution on` $\mathcal{U}$

  $(x_{i_1}...x_{i_B}) =$`Sample(`p_t, B`)` ; `// sample` B `points`

  $(\ y_{i_1}...y_{i_B}) =$ `Oracle(`$x_{i_1}...x_{i_B}$`)` ; `// query oracle for labels`

  for $k =$ **to** B **do**

    $W(x_{i_k}) = 1/P_t(x_{i_k})$; `// update weights`

  end

  $\mathcal{L} = \mathcal{L} \cup \{(x_{i_1}y_{i_1}) ... (x_{i_K}, y_{i_K})\}$; `// update labeled set`

  $\mathcal{U} = \mathcal{U} \setminus (x_{i_1}...x_{i_B})$; `// update unlabeled set`

  $t \leftarrow t + 1$;

until K *unique active labels obtained*;

return θ^*, $\mathcal{L}$, $\mathcal{U}$;

fied Decision Template method (**DT-2**) with additional crisp labels added to the decision profile, $DP(x) = \begin{bmatrix} v_1(x) & v_2(x) & ... & v_K(x) & h_1(x) & h_2(x) & ... & h_K(x) \end{bmatrix}$.

4. **Behavior Knowledge Space Method (BKS).** This is a simple learning rule where the most frequent output y for a particular combination of ensemble outputs $h_1...h_K$ during training is chosen as the final output on a testing sample ([69]).

4.6.2 Transformation Based Methods for EC

In transformation based methodsthe crisp $h_1...h_K$ and/or soft outputs $v_1...v_K$ from classifiers are used as inputs to a meta-classifier. Depending on the meta-classifier and the input type, we get different concrete methods summarized in Table 4.3.

Table 4.3: **Transformation Based EC Methods.** Method names are based upon the combination of meta-classifier (rows) and input types (columns) used. h_k, v_k are the crisp and soft weak classifier outputs, x are the input features. SVM-RBF is support vector machine using radial basis function kernel, LDISCR is linear discriminant method and ML is manifold learning. Other meta-classifiers are as described in the Section 4.4.2. Not all combinations above are possible, for example, as not all classifiers accept all input types.

	$h_1..h_K$	$v_1..v_K$	$h_1..h_K, v_1..v_K$	$h_1..h_K, v_1..v_K, x$
BMM	**BMM-EC**	-	-	-
BMS	**BMS-EC**	-	-	-
BMSS	**BMSS-EC**	-	-	-
GMM	-	**GMM-EC**	-	-
GMS	-	**GMS-EC**	-	-
GMSS	-	**GMSS-EC**	-	-
BGMM	-	-	**BGMM-EC**	**BGMM-EC-F**
BGMS	-	-	**BGMS-EC**	**BGMS-EC-F**
BGMSS	-	-	**BGMSS-EC**	**BGMSS-EC-F**
SVM-RBF	**SVM-I**	**SVM-S**	**SVM-EC**	**SVM-EC-F**
LDISCR	**DISCR-I**	**DISCR-S**	**DISCR-EC**	**DISCR-EC-F**
ML	**-**	**ML-S**	**ML-EC**	**ML-EC-F**

4.7 New Algorithms For Active Learning

4.7.1 Output-based Active Selection (OAS)

The basic building block for OAS based algorithms is shown in Fig 4.6. The original features and outputs - both crisp labels and soft output - from the learnt classifier are used as inputs to a BGMM classifier; the posterior probability of the classified output from this meta-classifier are then used as a measure of sample uncertainty (Algorithm Output based Active Selection) Our algorithm internally uses a BGMM classifier, but in practicality it must match the base learner (classifier) type. Variations where BGMS or BGMSS are used instead are also explored in the Results section.

4.7.2 Output-based Active Learning (OASL and OASSL)

An active learning algorithm based upon OAS is shown in Algorithm 4. A single iteration of this algorithm (Fig 4.7) is very similar to the generic AL algorithm (Fig 4.3) with

the *Select* block replaced by an OAS block. Versions of the algorithm based on GMSS and GMSSL base learners are abbreviated **OASL** and **OASSL** respectively.

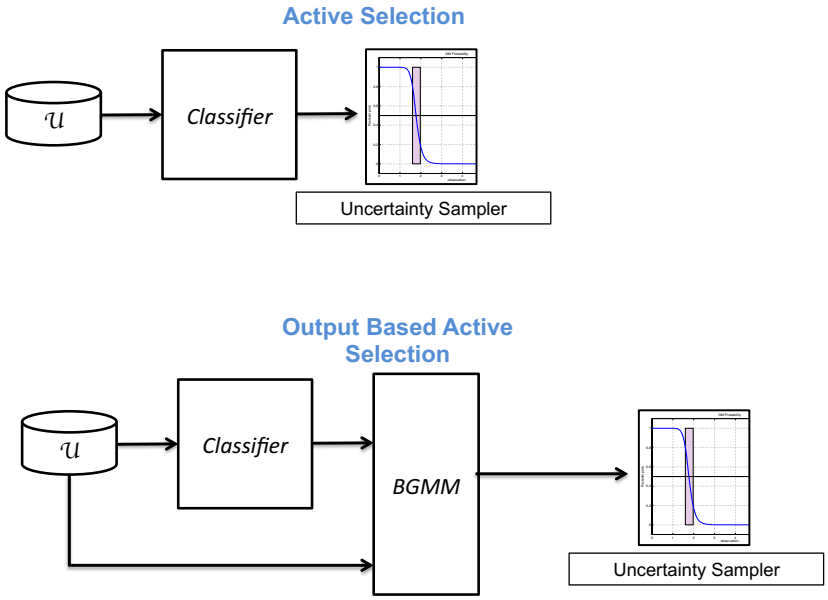


Figure 4.6: **Output based Active Selection (OAS).** Building blocks for standard uncertainty based active selection (US) (top) and OAS (bottom). "Classifier" is a trained model and $\mathcal{U}$ is the unlabeled samples. In US, the classifier's own confidence is used for active selection and in OAS, both output and confidence of the classifier are used as additional features into BGMM classifier whose posterior probability is then used for active selection.

4.7.3 Adaptive Output-based Active Learning (OASL-A and OASSL-A)

A version of OAS that we call OAS-A outputs both confident samples and active samples (Function Adaptive Output based Active Selection). The idea is that when there is no significant improvement in confidence of most confident outputs, OAS adds no value in determining active samples. Algorithms OASL-A and OASSL-A maintain a history of confident samples across iterations and compare it with the confident sample output from OAS-A for the current iteration to determine when to stop using OAS-A to select active samples (Fig 4.8). These algorithms current switches to either standard uncertainty based sampling or random sampling once this condition is reached. Variations where the active learner switches to other methods of selection (such as information density based, random sampling, etc.) are possible but have not been tested at present. OAS-A also differs from OAS in that the active samples are determined out of all labeled as well as labeled samples

Function Output based Active Selection(OAS)

Function `OAS`$(\theta^*, \mathcal{L}, \mathcal{U}, B)$

  Input: Trained classifier θ^*, labeled set $\mathcal{L} = \{x_j, y_j\}_{j=1}^{L}$, unlabeled set $\mathcal{U} = \{x_j\}_{j=L+1}^{L+U}$, batch size B

  Output: Active sample set $\{x_{i_1}...x_{i_B}\}$ of size B

  for *all samples* j **do**

    $(\widehat{y}_j, v_j) = \theta^*(x_j)$; `// get crisp & soft classifier output`

  end

  for *labeled samples* j **do**

    $\widehat{y}_j = y_j$;

  end

  $\theta^*_{BGM} =$ `BGMM`$(\{\widehat{y}_j\}_{j=1}^{N}, \{v_j\}_{j=1}^{N}, \{x_j\}_{j=1}^{N})$; `// train BGMM`

  for *all unlabeled samples* j **do**

    `// get output & posterior probability of output`

    $(z_j, p(z_j)) = \theta^*_{BGM}(\widehat{y}_j, v_j, x_j,)$

  end

  Pick $\{x_{i_1}...x_{i_B}\} \subset \mathcal{U}$ with smallest $\left|p(z_j) - \frac{1}{2}\right|$;

return $(x_{i_1}...x_{i_B})$;

Algorithm 4: Output based Semi-Supervised Active Learning

Given: Labeled set $\mathcal{L} = \{x_j, y_j\}_{j=1}^{L}$, Unlabeled set $\mathcal{U} = \{x_j\}_{j=L+1}^{L+U}$, Batch size B, Total number of active labels K

repeat

  $\theta^* =$ `Learn`$(\mathcal{L}, \mathcal{U})$; `// learn model using current` $\mathcal{L}, \mathcal{U}$

  $\{x_{i_1}...x_{i_B}\} =$ `OAS`$(\theta^*, \mathcal{L}, \mathcal{U}, B)$; `// select` B `informative samples`

  $\{y_{i_1}...y_{i_B}\} =$ `Oracle`$(x_{i_1}...x_{i_B})$; `// query oracle for labels`

  $\mathcal{L} = \mathcal{L} \cup \{(x_{i_1} y_{i_1}) ... (x_{i_K}, y_{i_K})\}$; `// update labeled set`

  $\mathcal{U} = \mathcal{U} \setminus \{x_{i_1}...x_{i_B}\}$; `// update unlabeled set`

until K *unique active labels obtained*;

return θ^*, $\mathcal{L}$, $\mathcal{U}$;

and not just unlabeled samples. The algorithms OASL-A and OASSL-A adaptively change the batch size B and confident output size C requested from OAS-A. Currently these are

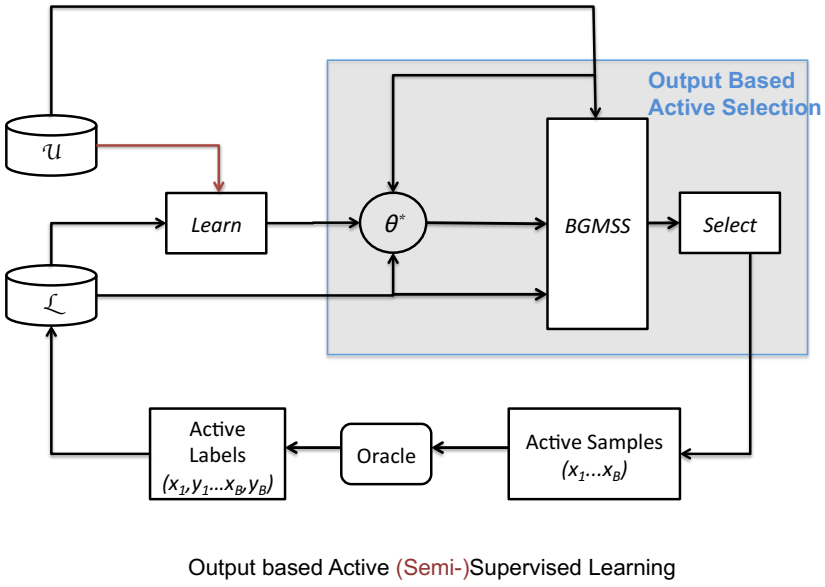


Figure 4.7: **Single iteration of an AL algorithm using OAS**. $\mathcal{U}$ is the unlabeled set, $\mathcal{L}$ is the labeled set, θ^* is the trained model. BGMSS is the E-M based classifier used internally by OAS. *Select* is implemented in OAS by uncertainty sampling on BGMSS output. The red arrow from $\mathcal{U}$ to the *Learn* module is present in the SSL case only. See algorithms Output based Active Selection and 4.

based upon parameters α, β and t_0 of the sigmoidal functions defined below. At iteration t,

$$\begin{aligned} B(t) &= N\left(b_0 + b_1 \frac{e^{\beta(t-t_0)}}{1+e^{\beta(t-t_0)}}\right) \\ C(t) &= N\left(1 - c_0 - c_1 \frac{e^{-\alpha(t-t_0)}}{1+e^{-\alpha(t-t_0)}}\right) \end{aligned} \tag{4.18}$$

where N is the total number of samples, b_0, b_1, c_0, c_1 are constants that are fixed for our algorithms at $b_0 = 0.008, b_1 = 0.012, c_0 = 0.12, c_1 = 0.43$. The constants control rate of convergence, and hence the total number of iterations desired, and were chosen heuristically based upon algorithm accuracy against simulated data sets. The parameters α, β are provided by the user, and the effect of them on the overall performance of the algorithms is discussed in the results. In a future implementation, these will be internally adjusted by the algorithm. The algorithms OASL-A and OASSL-A are fully described in Algorithm 5.

Function Adaptive Output based Active Selection(OAS-A)

Function `OAS-A`$(\theta^*, \mathcal{L}, \mathcal{U}, B)$

Input: Trained classifier θ^*, labeled set $\mathcal{L} = \{x_j, y_j\}_{j=1}^{L}$, unlabeled set $\mathcal{U} = \{x_j\}_{j=L+1}^{L+U}$, batch size B, confident output size C

Output: Active sample set $\mathcal{A}$ of size B,

Confident output set $\mathcal{C}$ of size C

for *each sample* j **do**

  $(\widehat{y}_j, v_j) = \theta^*(x_j)$; `// get crisp & soft classifier output`

end

for *each labeled sample* j **do**

  $\widehat{y}_j = y_j$;

end

$\theta^*_{BGM} =$ `BGMM`$(\{\widehat{y}_j\}_{j=1}^{N}, \{v_j\}_{j=1}^{N}, \{x_j\}_{j=1}^{N})$; `// train BGMM`

for *each samples* j **do**

  `// get output & posterior probability of output`

  $(z_j, p(z_j)) = \theta^*_{BGM}(\widehat{y}_j, v_j, x_j,)$

end

Pick $\{i_1 \cdots i_B\}$ with smallest $\left|p(z_i) - \frac{1}{2}\right|$;

Pick $\{l_1 \cdots l_C\}$ with largest $\left|p(z_l) - \frac{1}{2}\right|$;

return $\mathcal{A} = \{x_{i_1} ... x_{i_B}\}$ *and* $\mathcal{C} = \{(x_{l_1}, \widehat{y}_{l_1}) \cdots (x_{l_C}, \widehat{y}_{l_C})\}$;

4.8 New Algorithms for Ensemble Clustering

4.8.1 OAS-based Active Learning for EC

Our techniques OAS and OAS-A are easily adaptable to ensemble inputs, since the BGMM model takes as input the output of another classifier. So it is possible to augment this input with outputs from other classifiers, including an ensemble. This concept facilitates use of OASL/OASSL and OASL-A/OASSL-A for EC (Fig 4.9). The EC algorithm based upon OASL is depicted in Algorithm 4.9 which we call **OASL-EC** in which supervised ensemble base learners are used. The corresponding algorithm with semi-supervised base learners (used internally by the algorithm) is called **OASSL-EC**. The algorithm shown uses OASL instead of OASLA for simplicity but adaptive versions (**OASLA-EC, OASSLA-EC)** were also implemented. The implementation is free to choose specific base learners that are used

Algorithm 5: Adaptive Output based Active (Semi-)Supervised Learning (OASSL-A)

Given: Labeled set $\mathcal{L} = \{x_j, y_j\}_{j=1}^{L}$, Unlabeled set $\mathcal{U} = \{x_j\}_{j=L+1}^{L+U}$, Total number of active labels K, Parameters α, β

Initialize Confident output $\mathcal{C} = \emptyset$, Active Set $\mathcal{A} = \emptyset$;
$t \leftarrow 0$;
repeat
  $\theta^* =$ `Learn(`$\mathcal{L}$,$\mathcal{U}$`)` ; `// learn model using current` $\mathcal{L}$,$\mathcal{U}$
  Compute B_t, C_t using α, β and (4.18) ;
  $(\{x_{i_1}...x_{i_B}\}, \{(x_{l_1}\widehat{y}_{l_1})...(x_{l_C}, \widehat{y}_{l_1})\}) =$`OAS-A(`$\theta^*$,$\mathcal{U}, \mathcal{L}, B_t, C_t$`)` ;
  $\{\ y_{i_1}...y_{i_B}\} =$ `Oracle(`$x_{i_1}...x_{i_B}$`)` ;
  $\mathcal{A} = \mathcal{A} \cup (x_{i_1}...x_{i_B})$;
  $\mathcal{C} = \mathcal{C} \cup \{(x_{l_1}\widehat{y}_{l_1}) \cdots (x_{l_C}, \widehat{y}_{l_1})\}$;
  $\mathcal{L} = \mathcal{L} \cup \{(x_{i_1}y_{i_1}) \cdots (x_{i_K}, y_{i_K})\}$;
  $\mathcal{U} = \mathcal{U} \setminus \{x_{i_1}...x_{i_B}\}$;
  $t \leftarrow t + 1$;
until $K = |\mathcal{A}|$ *or no change in* $|\mathcal{C}|$;
if $|\mathcal{A}| < K$ **then**
  $\mathcal{U} = \mathcal{U} \setminus \mathcal{C}$;
  $K = K - |\mathcal{A}|$;
  Continue with Random Selection AL for $\mathcal{L}$,$\mathcal{U}, K, B$;
end
return θ^*, $\mathcal{L}$, $\mathcal{U}$,, $\mathcal{C}$;

internally by the algorithm but we chose SVM-RBF, LDISCR and GMS as the specific base learners in our implementation that was tested against real artifact ensemble data. The algorithm makes use of the ensemble version *OASEns* of the single classifier *OAS* version, which takes in multiple trained classifiers $\theta_1^*...\theta_E^*$ as inputs (Function Ensemble Output based Active Selection). The OASL-EC algorithm calls *OASEns* with the trained base learners along with the given ensemble classifiers. Note that the ensemble classifiers are assumed to be not re-trainable, i.e. members $h_1...h_K$ of the ensemble always return the same output for a given input.

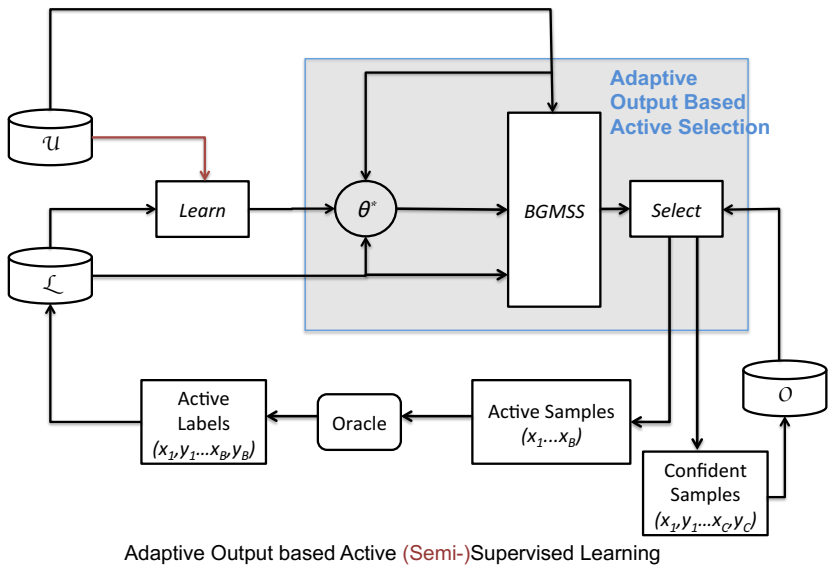


Figure 4.8: **Single iteration of adaptive AL algorithms using OAS-A**. $\mathcal{U}$ is the unlabeled set, $\mathcal{L}$ is the labeled set, θ^* is the trained model. BGMSS is the E-M based model used internally by OAS-A. *Select* also returns most confident samples that are persisted in $\mathcal{O}$ across iterations The red arrow from $\mathcal{U}$ to the *Learn* module is present in the SSL case only. See algorithm 5 .

4.8.2 Uncertainty-based Active Learning for EC

In a manner analogous to OAS based EC algorithms, we also implemented Active-SL and Active-SSL based EC algorithms, that use selection based active sampling. A schematic is shown in Fig 4.10

4.8.3 Disagreement-based Active Learning for EC

In order to compare performance of use of OASL for EC, we formulated active versions of the supervised and semi-supervised transformation based methods listed in Table 4.3. While none of the individual methods for these are novel, to our knowledge AL has not been previously applied to our particular problem of EC. We have implemented active versions of these transformation based methods using disagreement method: samples for which the ensemble members disagree the most (using vote entropy) are chosen as active samples for querying. Based upon each supervised/semi-supervised transformation method of Table 4.3 becomes an active ensemble clustering algorithm, and is called by the transformation algorithm name with an "**A**" suffix for "active". For example, active version of **ML-EC** will

Function Ensemble Output based Active Selection(OAS)

Function `OASEns`$(\theta_1^* \cdots \theta_K^*, \mathcal{L}, \mathcal{U}, B)$

Input: Trained classifiers $\theta_1^* \cdots \theta_K^*$, labeled set $\mathcal{L} = \{x_j, y_j\}_{j=1}^{L}$,
unlabeled set $\mathcal{U} = \{x_j\}_{j=L+1}^{L+U}$, batch size B

Output: Active sample set $\{x_{i_1} ... x_{i_B}\}$ of size B

for *each sample* j **do**
  for *each classifier* k **do**
    $\left(\widehat{y}_j^{(k)}, v_j^k\right) = \theta_k^*(x_j)$; `// get crisp & soft classifier output`
  end
end
for *each labeled sample* j *and each* k **do**
  $\widehat{y}_j^{(k)} = y_j$;
end
Define $\widehat{y}_j := \left(\widehat{y}_j^{(1)} \cdots \widehat{y}_j^{(K)}\right), v_j := \left(v_j^{(1)} \cdots v_j^{(K)}\right)$;
$\theta_{BGM}^* =$ `BGMM`$(\{\widehat{y}_j\}_{j=1}^{N}, \{v_j\}_{j=1}^{N}, \{x_j\}_{j=1}^{N})$; `// train BGMM`
for *each unlabeled samples* j **do**
  `// get output & posterior probability of output`
  $(z_j, p(z_j)) = \theta_{BGM}^*(\widehat{y}_j, v_j, x_j,)$
end
Pick $\{x_{i_1} ... x_{i_B}\} \subset \mathcal{U}$ with smallest $\left|p(z_j) - \frac{1}{2}\right|$;
return $(x_{i_1} ... x_{i_B})$;

be called **ML-EC-A**. Note that while several ensemble active learning algorithms exist in the literature (see [252, 84, 142, 90, 1, 158]) these assume that the ensemble is re-trainable, an assumption that is not valid for our purposes.

In the case of **ML-EC** another active selection method was tested, this was based a method where uncertainty is computed from the posterior probability of a clustering BGMM that takes in both input features and ensemble outputs as inputs. We call this method **ML-EC-A-GM.** Similarly, **ML-EC-F-A-GM** was tested where the same BGMM based active selection. We did not find any improvement when using this active selection on transformation methods other than ML based, so they are not reported.

Algorithm 6: OASL Based Ensemble Clustering (OASL-EC)

Given: Unlabeled feature set $\mathcal{U} = \{x_j\}_{j=1}^{N}$, Fixed (untrainable) ensemble models $(h_1 \cdots h_C)$, Batch size B, Total number of active labels K

Choose: L base learners $\mathsf{Learner}_1 \cdots \mathsf{Learner}_L$

, Initialize $\mathcal{L} = \emptyset$;

repeat

  for $l =$ **to** L **do**

    $\theta_l^* = \mathtt{Learner}_l(\mathcal{L})$; `// train l'th base model`

  end

  $\{x_{i_1} ... x_{i_B}\} = \mathtt{OASEns}(\theta_1^* \cdots \theta_L^*, h_1 \cdots h_C, \mathcal{L}, \mathcal{U}, B)$

  $\{ y_{i_1} ... y_{i_B}\} = \mathtt{Oracle}(x_{i_1} ... x_{i_B})$; `// query oracle for labels`

  $\mathcal{L} = \mathcal{L} \cup \{(x_{i_1} y_{i_1}) ... (x_{i_K}, y_{i_K})\}$; `// update labeled set`

  $\mathcal{U} = \mathcal{U} \setminus \{x_{i_1} ... x_{i_B}\}$; `// update unlabeled set`

until K *unique active labels obtained*;

θ^* =$\mathtt{Learner}_1(\mathcal{L})$; `// train one learner with final` $\mathcal{L}$

;

for *each sample* j **do**

  $\widehat{y}_j = \theta^*(x_j)$;

end

return $\{\widehat{y}_j\}_{j=1}^{N}$;

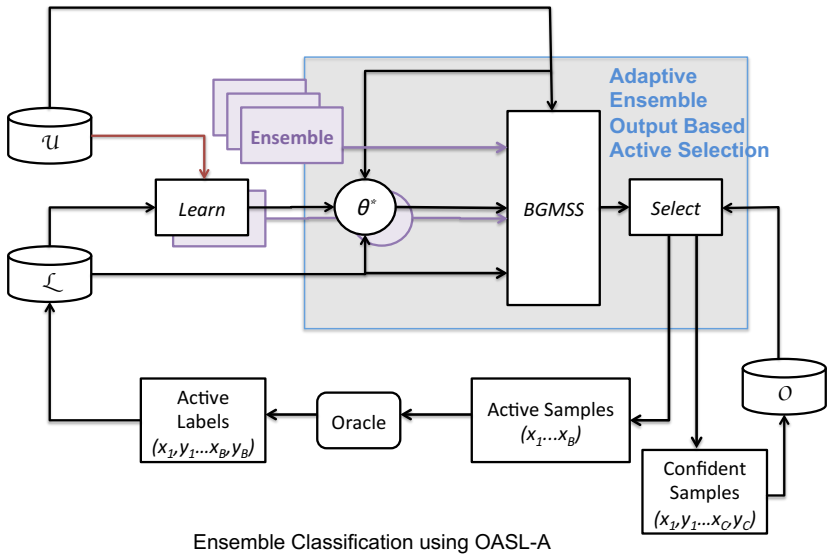


Figure 4.9: **Single iteration of EC algorithm using OASLA**. The algorithm is identical to that in Fig 4.8 with modifications shown in purple. BGMSS takes additional ensemble inputs from both the given input ensemble and internal ensemble chosen by the implementation. The red arrow from $\mathcal{U}$ to the *Learn* module is in the case of SSL base learners only. See algorithm 6.

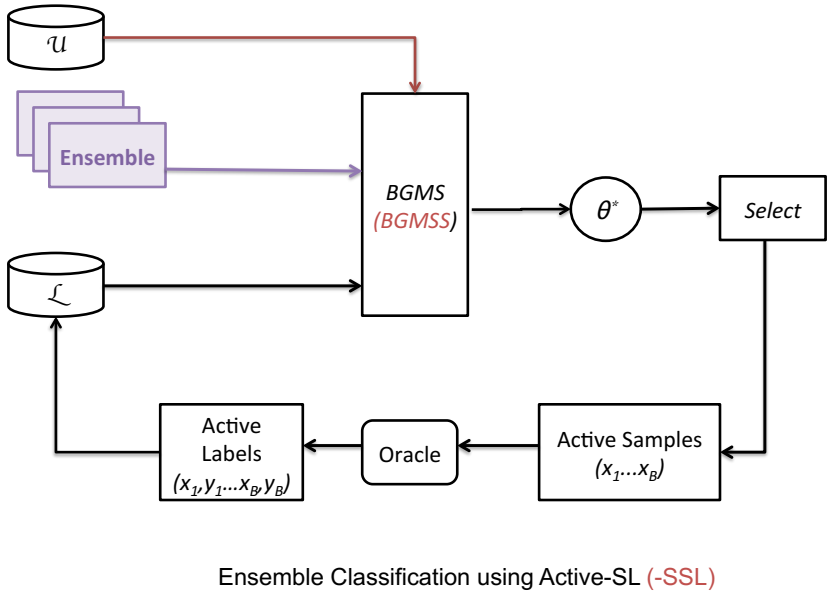


Figure 4.10: **Single iteration of EC algorithm using Active-SL/Active-SSL**. The algorithm is similar to that in Fig 4.3 with BGMS (BGMSS) as the base learner which takes in additional inputs from given input ensemble. The red arrow from $\mathcal{U}$ to the *Learn* module is in the case of BGMSS (SSL learner) only.

4.9 Experiments

4.9.1 Noisy Gaussian Mixture Data Sets

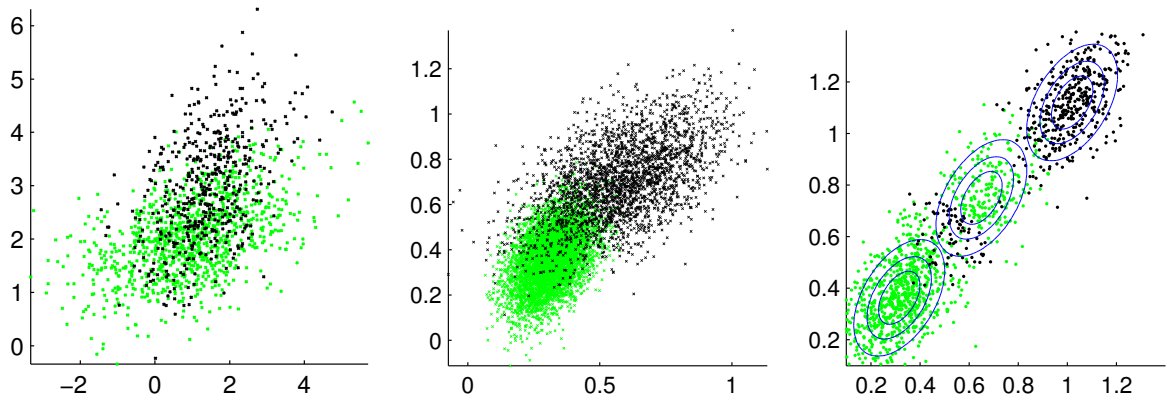
We use the following generated noisy Gaussian mixtures to evaluate how OAS based AL compares with existing methods for binary classification on noisy and non-separable data.

1. **Synthetic Problem I (SP-I):** A mixture of two Gaussian components is generated in 2-D space with $N_0 = 1000$ points sampled from the first component of the mixture ($c = 0$) and $N_1 = 500$ from the second component ($c = 1$) (Fig 4.9.1(a)). This example is non-separable with high degree of non-agnostic noise. The exact values used for mean and covariances for the mixtures appear in the Online Supplemental Material.

2. **Synthetic Problem II (SP-II):** A mixture of two Gaussian components in 2-D space with $N_0 = 5000$ and $N_1 = 2500$ (Fig 4.9.1(b)) is also non-separable but lower noise than SP-I. The exact values used for mean and covariances for the mixtures appear in the Online Supplemental Material.

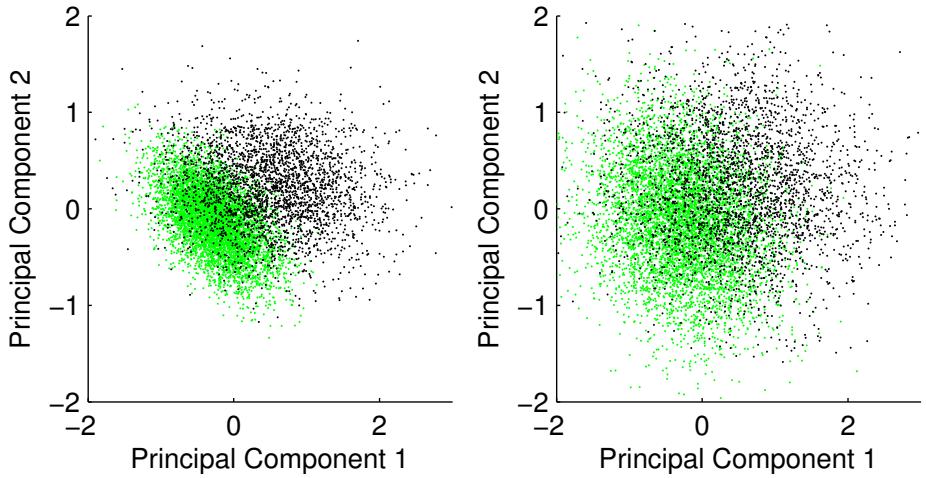
3. **Synthetic Problem III (SP-III):** A mixture of *three* Gaussians (adapted from [240]) was used (Fig 4.9.1(c)). Standard uncertainty based AL algorithms fail to perfom the binary classification correctly on this example. To generate data for this example, $N_1 = 400, N_2 = 100, N_3 = 300$ points are sampled from mixtures $c = 1, 2, 3$ respectively. Points from the mixtures $c = 1, 3$ are assigned labels "0" and "1" respectively, and remaining points are assigned random labels. This is an example containing *both agnostic and non-agnostic noise*. The exact values used for mean and covariances for the mixtures appear in the Online Supplemental Material.

4. **Synthetic Problem IV (SP-IV):** Two multi-variate (31-dimensional) normal distributions were fitted to non-artifactual and artifactual epochs of real EEG data (Section 4.9.2). Then $N_0 = 5800$ samples were sampled from the first distribution and $N_1 = 2900$ from the second. These values correspond roughly to the actual number of non-artifactual and artifactual epochs. Then noise that is uniformly distributed over $[-\eta\ \eta]^d$ $(d = 31)$ was added to each sample of the mixture. Here η corresponds to the noise rate. A scatter plot of the first two principal components of generated data is shown for $\eta = 0$ and $\eta = 0.3$ in Fig 4.9.1.



(a) SP-I (b) (SP-II) (c) (SP-III)

Figure 4.11: Scatter plot of the 2-D synthetic data (two Gaussian mixture) in problems (a) SP-I (b) SP-II and (c) SP-III. In (c) contour plots of the individual Gaussian distributions are shown for visual clarity. The mixture component in the middle contains points from both "0" and "1" labels at random.



(a) SP-IV, $\eta = 0$ (b) SP-IV,$\eta = 0.3$

Figure 4.12: Scatter plot of the first two principal components of the 31 dimensional synthetic data (two Gaussian mixture) in Problem SP-IV. (a) Noise rate $\eta = 0$ (b) Noise rate $\eta = 0.3$

4.9.2 EEG Artifact Data

EEG data were collected from inpatient studies of healthy young individuals . Data from the KDT portions during wake were used. EEG electrodes were placed useing the 10-20 system at the midline locations Fz-Cz-Pz-Oz. 2s epochs marked by an RPSGT as artifacts were discarded, resulting in a total of 41.4h worth of 6-channel data (a total of 74552 epochs

from 9 individuals). Remaining data were analyzed on an epoch by epoch basis and a 31-dimensional feature vector was obtained for each epoch. Six different automatic artifact detection methods were employed as described below. Each method outputted, per epoch, (i) a binary value indicating if the epoch was classified as one containing an artifact, (ii) a non-negative real valued confidence value for the classification. For 8700 out of the 74552 epochs, exact determination of whether the epoch is artifactual or not was done manually by careful visual inspection and consultation with another RPSGT. The automated artifact detection methods comprise an ensemble, and this subset of 8700 epochs was used to test EC algorithms.

Feature Sets. The EEG signal $z(t)$ for an epoch $0 \leq t \leq T$ of $T = 2$s is analyzed using the CSSR algorithm (Chapter 3) with sparsity level of 2 to give the sparse vector $v^{(c)}$ for channel c and sparse representation $\widetilde{z}^{(c)}(t) = Hv^{(c)}$ where H is the dictionary. In addition, the CSSR algorithm also returns a scalar $q^{(c)}$ representing the quality of the representation (a measure of the error in sparse representation). A resulting feature vector comprises of, for each channel c, (i) the two largest components $\left|v_1^{(c)}\right|, \left|v_2^{(c)}\right|$ of $v^{(c)}$ (ii) mean $m^{(c)}$ of $\left|z^{(c)}(t)\right|$ over $0 \leq t \leq T$ (iii) mean of $\left|\widetilde{z}^{(c)}(t)\right|$ over $0 \leq t \leq T$ (iv) $q^{(c)}$, and the correlation coefficient $r$between the Fz and VEOG channels. Using 6 channels, this gives a 31-dimensional feature vector. The gathered feature vectors are then normalized *per KDT episode.*

Ensemble Description The six automated detection methods used that comprise the ensemble are as follows (all k-means clustering methods are done per KDT episode). The choice of classifiers is made heuristically, and no single method works best on all epochs, and no single method works for all artifact types.

C1. k-means clustering using values of dominant sparse components, their quality of match and raw signal values as features.

C2. k-means clustering on first two principal components of the feature set used by **C1**.

C3. k-means clustering using the first two principal components of the feature set used by **C1**, augmented by the correlation value r, and normalized per KDT episode.

C4. Semi-supervised clustering (using Manifold Learning) on all features using most confident samples from **C1** as the labeled examples.

C5. Thresholding based upon values and match quality of dominant sparse components.

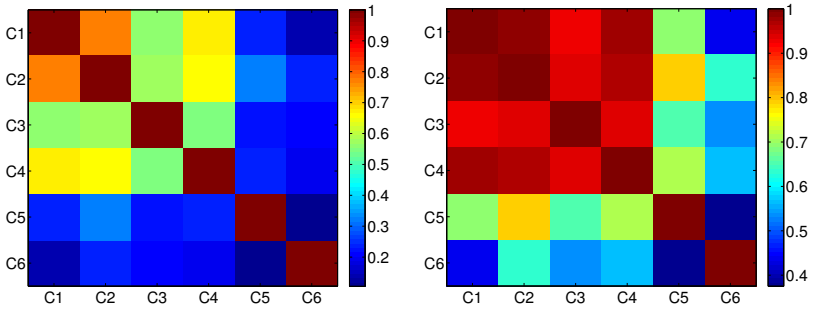
C6. Thresholding based upon z-scored *differential* (Fz-Cz and Pz-Oz) raw signal values.

Ensemble Characteristics For classifiers C1 through C6 on 8700 test epochs, no individual classifier attains over 91% accuracy by itself (Fig 1). Correlation diversity and Q-statistics were computed for the ensemble as measures of diversity ([180, 133, 132]. See Online Supplemental Material for definition. These statistics indicate (Fig 4.13) a good degree of diversity between C5,C6 and rest of the classifiers, and a high degree of dependence between C1,C2. The entropy E ([131]) of the ensemble was computed to be roughly 0.7 ($E = 1$ indicates highest diversity).

Table 4.4: Performance of individual classifiers C1 through C6 in the ensemble for the EEG test data across 8700 epochs.

1.

	Accuracy	FPR	FNR
C1	87.97	6.61	5.43
C2	87.84	8.43	3.74
C3	84.46	8.79	6.75
C4	89.48	6.83	3.69
C5	91.23	5.37	3.40
C6	80.03	16.91	3.06



(a) ρ (b) Q

Figure 4.13: (a) Correlation Diversity, and (b) Q-Statistics across classifiers C1 through C6 shown as matrix plots. Lower values indicate diversity and higher values dependence.

4.9.3 Metrics For Algorithm Performance Evaluation

For simplicity we have only chosen to compare classification (i) accuracy (or its complement, error rate), (ii) false positive rate and (iii) false negative rate when comparing algorithms. Other standard metrics for binary classifiers such as F-1 metric are not reported. Learning algorithms are also evaluated based upon the following characteristics:

1. **Label Complexity:** The number of active labels required for a particular error rate, a particularly characteristic for comparing active learners. An ideal active learner has $O(\ln \frac{1}{\varepsilon})$ label complexity compared to the $O(\frac{1}{\varepsilon})$ complexity of passive learning [105].

2. **Noise Sensitivity:** The error rate as a function of noise rate η for a fixed number of active labels.The lower bound of label complexity of agnostic algorithms is known to be $O(\frac{\eta^2}{\varepsilon^2})$ [121], which implies that for a fixed label size, accuracy decreases at best linearly with noise rate.

4.10 Results

4.10.1 Base Learner Characteristics in Passive Mode

A comparison of label complexity of the base learners GMS, GMSS, SVM-RBF and ML as passive learners (Fig 4.14) for the three data sets SP-I, SP-II and SP-IV (SP-III was evaluated on AL only) indicates that GMS does better than SVM when fewer labels are used, and GMSS does better than ML with fewer labels except for SP-II where ML seem to do better. This seems to imply that ML is a better SSL algorithm when data are low dimensional and have mediocre separability. It is also interesting to note that in the medium separability cases (SP-II, SP-IV) SSL does worse than SL even with few labels, whereas in data with very low separability (SP-I) SSL is a better choice especially with fewer labels.

4.10.2 Performance of AL Algorithms on the SP-I DataSet.

OAS based algorithms - OASL, OASSL and OASSL-A - have better accuracy than other AL algorithms (section 4.5) for a variety of label fractions and a fixed batch size of 25 (Table 4.5). While QBB based on vote entropy has smaller FNR, it has much larger FPR (vote entropy is more conservative than other active selection methods). With 15% labels OAS based algorithms show modest improvement in overall accuracy for both SL and SSL base learners (Fig 4.15). Label complexity characteristics (Figs 4.18, 4.17, 4.18 and 4.19) indicate that OAS outperforms all others at all label fractions, for both SL and SSL base learners; although the advantage is greater at low label fractions and for SSL. This example illustrates that OAS outperfoms other AL algorithms even without agnostic noise when data are highly non-separable.

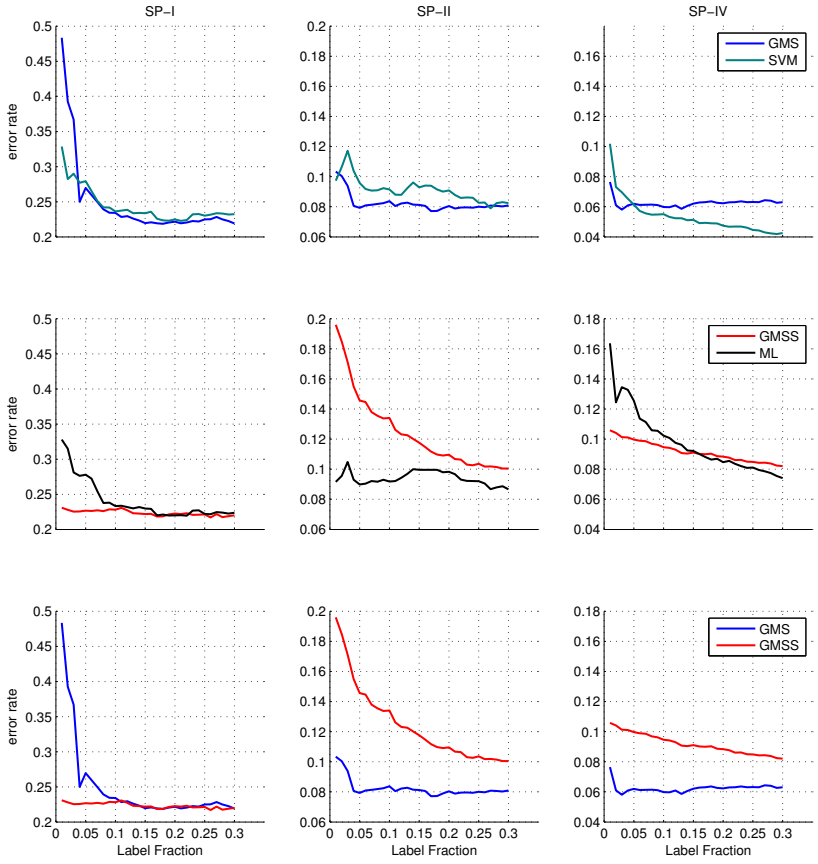


Figure 4.14: Label complexity of passive learners for the data sets SP-I, SP-II and SP-IV. The top panels compare supervised learners (GMS and SVM), the middle ones compare semi-supervised learners (GMSS and ML). The label complexity for GMS and GMSS is re-plotted in the bottom panel to show a comparison of supervised/unsupervised base learners

4.10.3 Performance of AL Algorithms on the SP-II Dataset

Within the class of SL based AL algorithms, OAS offers very modest improvement over the standard uncertainty based algorithm (Active-SL) (Fig 4.20), other AL Active-QBB-SL-Count-Sim and Active-IW-SL do worse than the Active-SL on the SP-II dataset. However, OAS offers significant improvement over standard and other AL algorithms based on SSL(Fig 4.21). It was noted earlier that on SP-II dataset, SSL actually does worse than SL in passive mode (Fig 4.14) but with OAS this difference is greatly reduced (Fig 4.22). Thus OAS can significantly improve SSL based AL especially at low label fractions.

As an insight into why OAS provides improvement over Active-SSL, consider the detected classification boundaries at different iterations in a sample run of OASSL and Active-

Table 4.5: Comparison of accuracy, FPR and FNR of AL algorithms for the SP-I dataset using 5%,10% and 15% active labels. All algorithms use 25 as batch size (or initial batch size, in the case of OASSL-A). The **bold** metrics indicate ones better than others for the same number of labels.

	Accuracy			FPR			FNR		
	5%	*10%*	*15%*	*5%*	*10%*	*15%*	*5%*	*10%*	*15%*
Passive-SL	77.15	78.97	80.65	16.81	16.17	14.69	6.04	4.85	4.66
Passive-SSL	77.79	79.05	80.49	16.77	16.38	14.75	5.44	4.57	4.76
Active-SL	75.93	79.73	79.33	12.47	12.4	12.73	11.6	10.13	7.93
Active-Margin-SL	48.6	61.13	68.13	**3.93**	**7.47**	**9**	47.47	31.4	22.87
Active-SSL	**78.07**	**81.4**	81.47	17.13	17.13	18.2	4.8	3.4	0.33
Active-IW-SL	76.93	78.8	80.27	15.27	15	15.27	7.8	6.2	4.47
Active-IW-SSL	**78.47**	79.87	80.67	17.53	15.27	14.8	4	4.87	4.53
Active-QBB-SL-Count	77.73	79	80.73	16.13	16.53	15	6.13	4.47	4.27
Active-QBB-SL-Count-Sim	53.13	77.67	81.93	**7.6**	19.13	15.93	39.27	3.2	2.13
Active-QBB-SL-KL	76.73	77.73	78.8	14.53	13.53	14.73	8.73	8.73	6.47
Active-QBB-SSL-Count	77.4	78.67	80.53	19.2	17.6	15.53	3.4	3.73	3.93
Active-QBB-SSL-VE	75.87	77.8	80.2	23.67	21.73	19.33	**0.47**	0.47	0.47
Active-QBB-SSL-KL	77.87	79.13	80	15.33	15.47	15.07	6.8	5.4	4.93
Active-SSL-MEAD	77.47	78.53	79.6	15.53	14.47	13.67	7	7	6.73
OASL	76.33	70.53	**82.6**	21.33	28.33	15.67	**2.33**	**1.13**	**1.73**
OASSL	**78.13**	**81.4**	**82.53**	16.2	**14.4**	12.47	5.67	5.33	5
OASSL-A	**78.73**	**81.6**	**83.87**	17.27	15.73	14.47	4	2.67	1.67

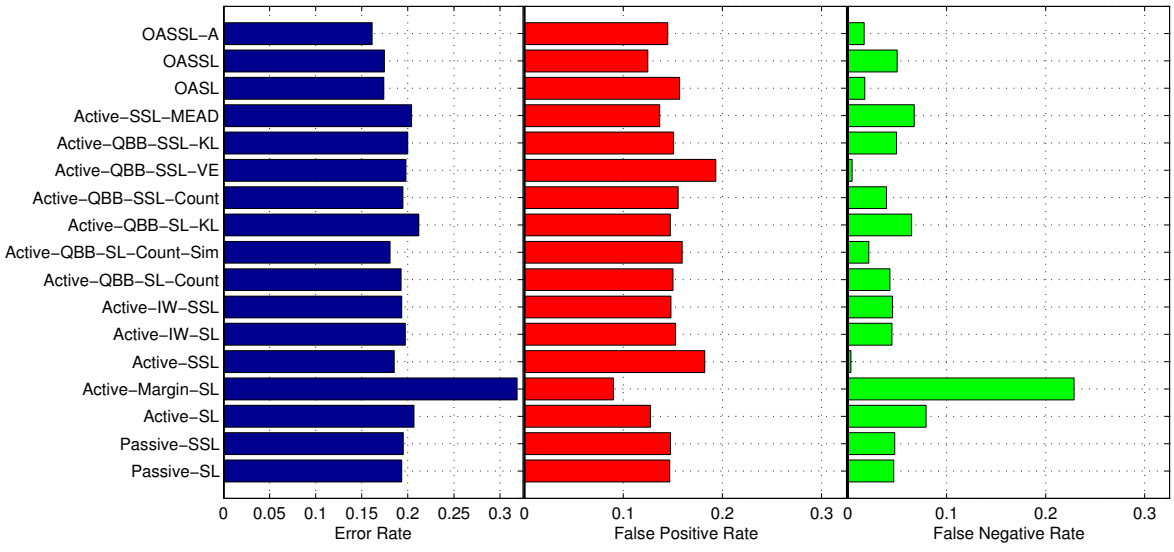


Figure 4.15: Comparison of error rate (1-accuracy), FPR and FNR of AL algorithms for the SP-I dataset when using 15% active labels and a batch size of 25.

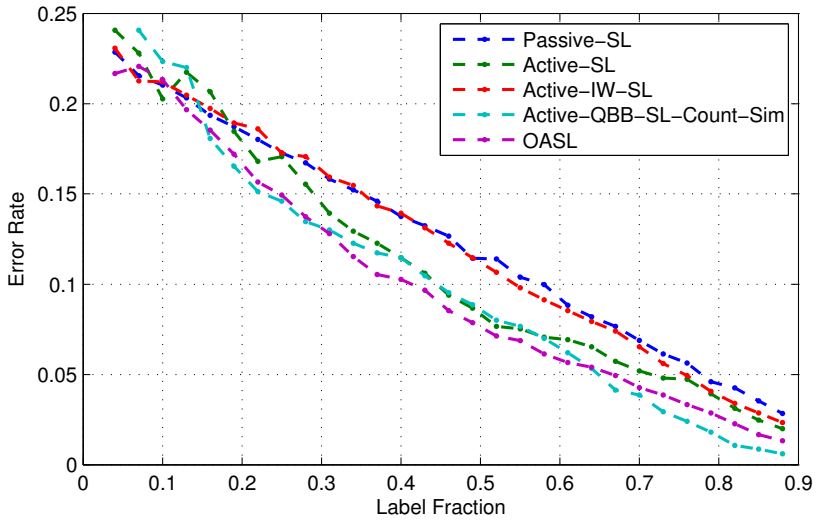


Figure 4.16: Label complexity of some SL based AL for the SP-I dataset. Only algorithms that perform better than the ones shown in Fig 4.18) are shown. Passive SL is included as reference.

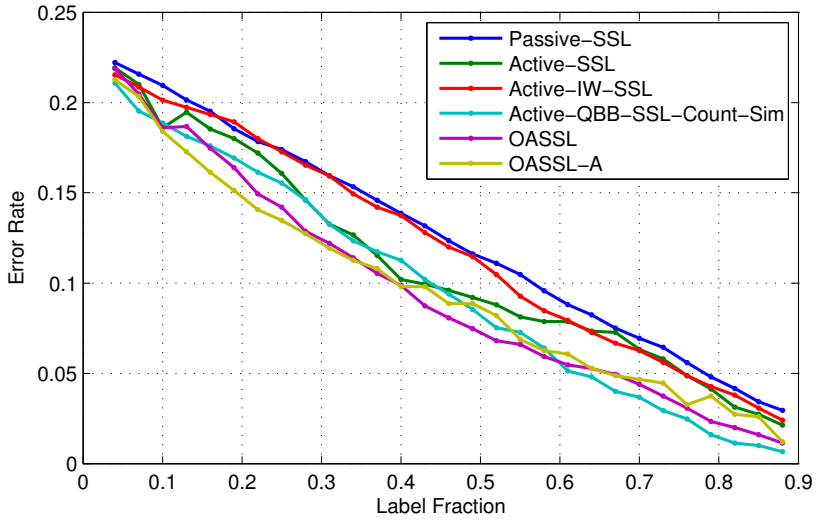


Figure 4.17: Label complexity of some SSL based AL for the SP-I dataset. Only algorithms that perform better than the ones shown in Fig 4.19 are shown. Passive SSL is included as reference.

SSL (Fig 4.23).Unlabeled samples result in a bias of the decision boundary further away from the optimal hypothesis due to the overlap in the clusters. Thus, with few labeled samples, uncertainty based active samples (as in Active-SSL) tend to be biased away from

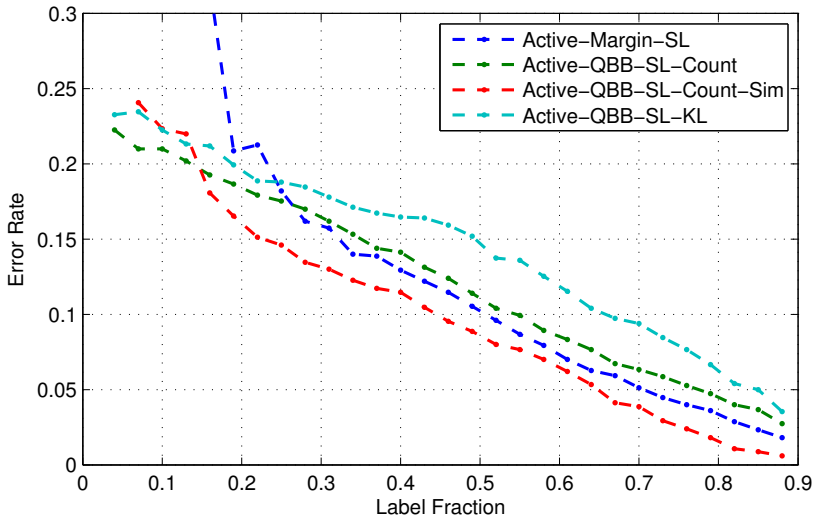


Figure 4.18: Label complexity of some SSL non-OAS based AL algorithms for the SP-I dataset. These algorithms, except the best performing one (Active-QBB-SSL-Count-Sim) are not included in Fig 4.18, and are omitted in subsequent results.

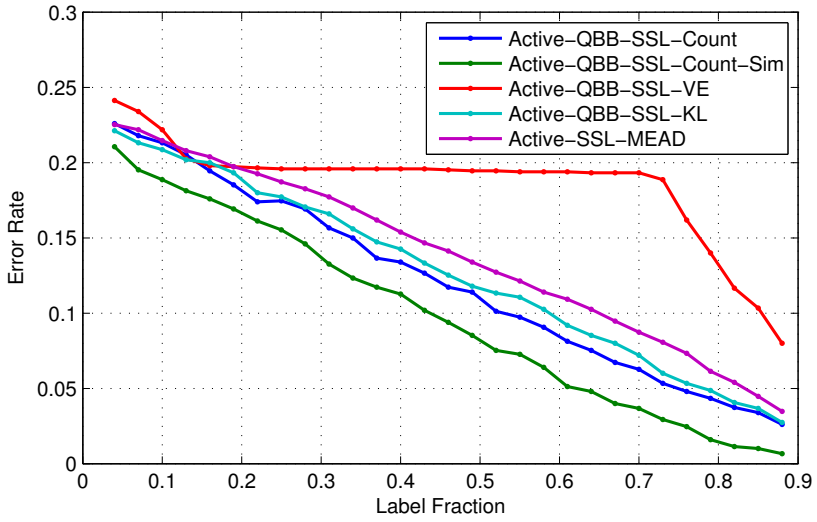


Figure 4.19: Label complexity for some SL non-OAS based AL algorithms for the SP-I dataset. These algorithms, except the best performing one (Active-QBB-SL-Count-Sim) are not included in Fig 4.19, and are omitted in subsequent results.

the decision boundary, which is offset only upon acquisition of several labeled samples after several iterations. With OAS, this bias is offset with a very few labeled samples and thus the classification boundary approaches the optimal hypothesis in fewer iterations.

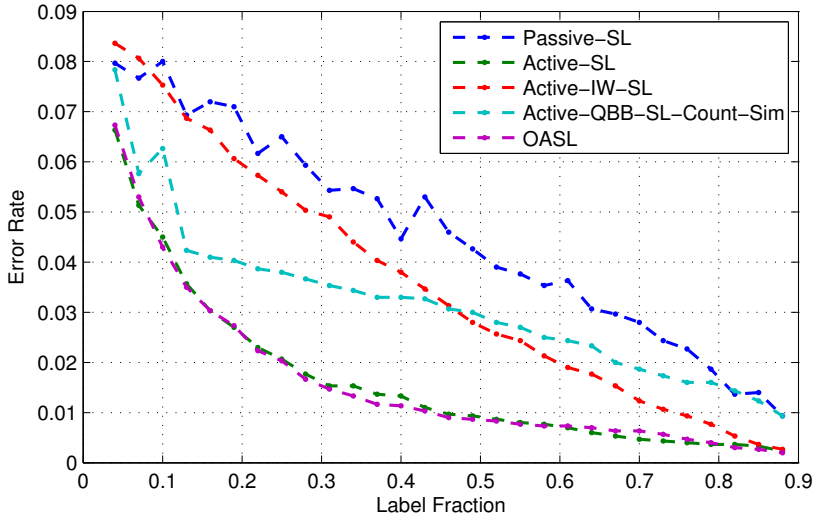


Figure 4.20: Label complexity of SL based AL algorithms for the SP-II dataset. Passive SL is included as reference.

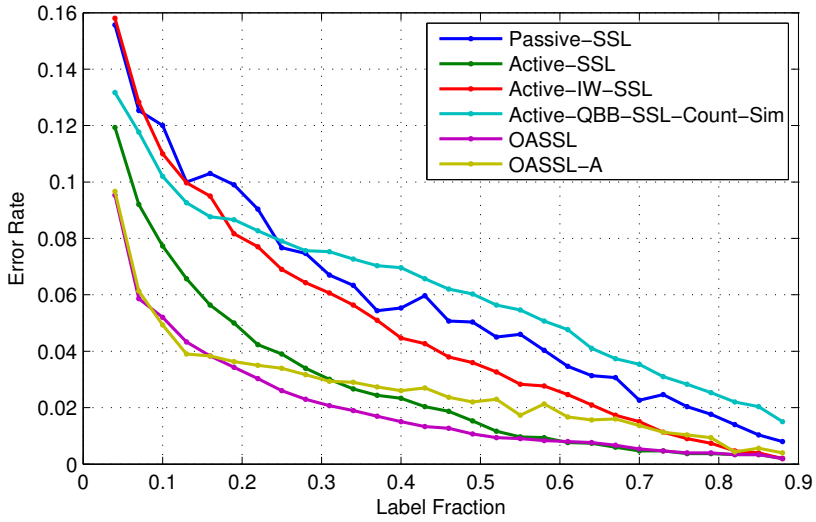


Figure 4.21: Label complexity of SSL based AL algorithms for the SP-II dataset. Passive SSL is included as reference.

4.10.4 Performance of AL Algorithms on the SP-III Dataset

Standard US-based AL such as Active-SL fails to correctly classify the three Gaussian SP-III dataset (Fig 4.24 (b)) when few active labels are used; the performance is worse than that of passive learning (Fig 4.25). Algorithms such Active-IW-SL handle this patholog-

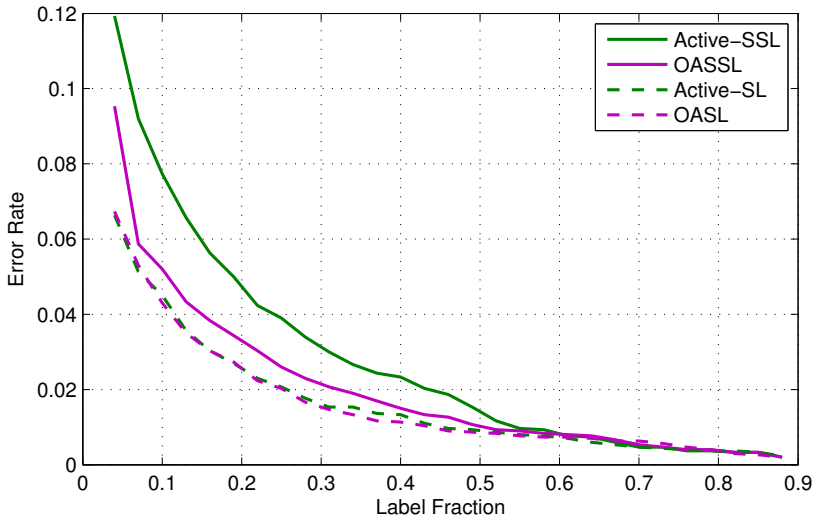


Figure 4.22: **Use of OAS boosts performance of SSL based AL.** Label complexity of standard (UC-based) and OAS-based AL for SL and SSL base learners.

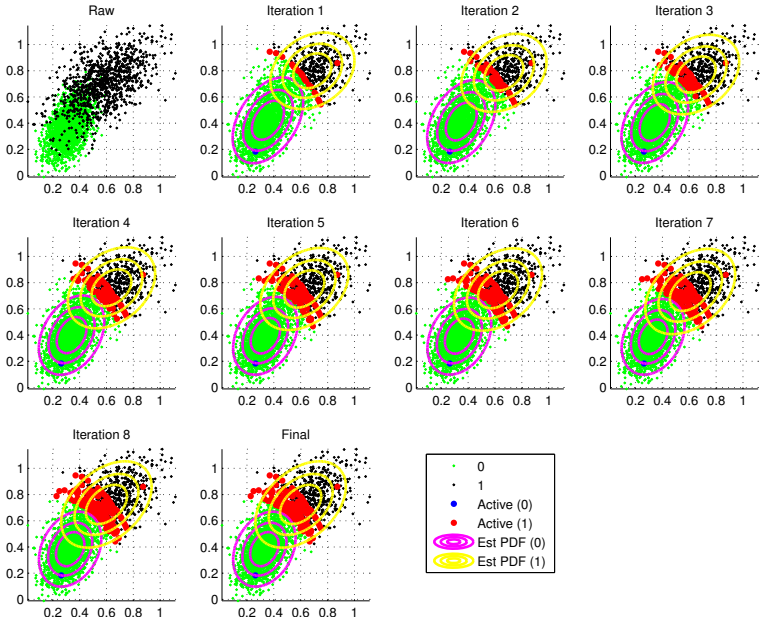
ical case well but OASL performs even better (Fig 4.25) especially at very low label fractions (Fig 4.27). An analysis of the active regions explored indicate that OASL has fewer "wasted" active labels than Active-IW-SL (Fig 4.24 (c),(d)).

As an insight into how OAS works, consider the detected classification boundaries at various iterations in a sample run of OASL and Active-SL. Sampling bias from US results in Active-SL quickly becoming confident about incorrectly classified regions (Fig 4.26(a), Iteration 3) which is subsequently not sampled, resulting in learning a hypothesis with large error even after several iterations. By requesting active samples away the uncertainty zone (Fig 4.26(b), iteration 2), OASL is able to mitigate sampling bias resulting in convergence toward the optimal hypothesis fairly quickly. With SSL base learners, OAS shows modest improvement (Fig 4.28).

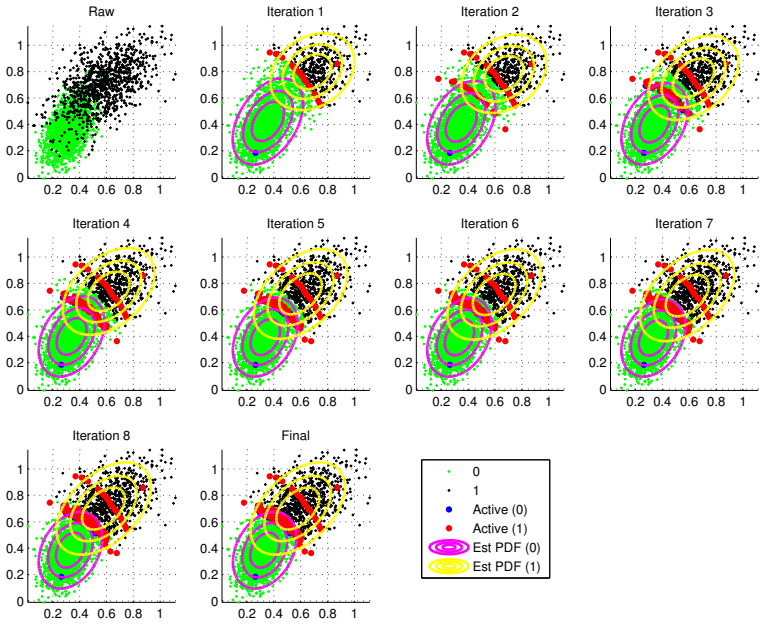
4.10.5 Performance of AL Algorithms on the SP-IV Dataset

With $\eta = 0$ (no agnostic noise), the qualitative behavior of OAS based algorithms when compared to others for the SP-IV dataset is similar to that for the SP-II dataset: OAS helps significantly with active learning for SSL base learners (Fig 4.30) but only very modestly with SL (Fig 4.29). As in the SP-II dataset, SL base learners perform better than SSL base learners (See FIG 4.14) but OAS reduces this difference (Fig 4.10.5(a)).

While same qualitative behavior is observed when agnostic noise is added (Figs 4.31, 4.32), though other state-of-the-art active learning methods approach passive learning when noise is added whereas the relative advantage of OASSL is maintained even with noise. The distinction between SL vs SSL, however, becomes relevant in this context in that at higher noise rates SSL surpasses SL (Figs 4.10.5(a) and (b)). Thus, the OAS based boost is most useful when there is high level of agnostic noise, especially when only few labels can be acquired. This is better illustrated with the noise sensitivity characteristics of SSL and SL based algorithms (Figs 4.10.5(a), (b)). When the best algorithm with each category (OAS-based, non-OAS based, passive) for each noise level is plotted against the noise level, we see that OAS based algorithms do best with the least amount of noise sensitivity (Fig 4.35).



(a)



(b)

Figure 4.23: Eight iterations of (a) Active-SSL (b) OASSL on the SP-II data set (batch size is 20). The raw data (top left) shows true labelings (green and black). Subsequent plots indicate the labels produced by the base learner (green and black) requested active labels (blue and red) along with contours of the estimated ("Est") PDFs (yellow and magenta). (a) Unlabeled samples result in a bias of the uncertain zone away from the optimal boundary (the red band in iteration 1) which grows only slowly toward the optimal boundary in the case of Active-SSL (b) In OASSL, this bias is quickly offset by active sampling from a region that is closer to the optimal boundary (the second smaller red band in iteration 2), resulting in quicker convergence toward the optimal hypothesis.

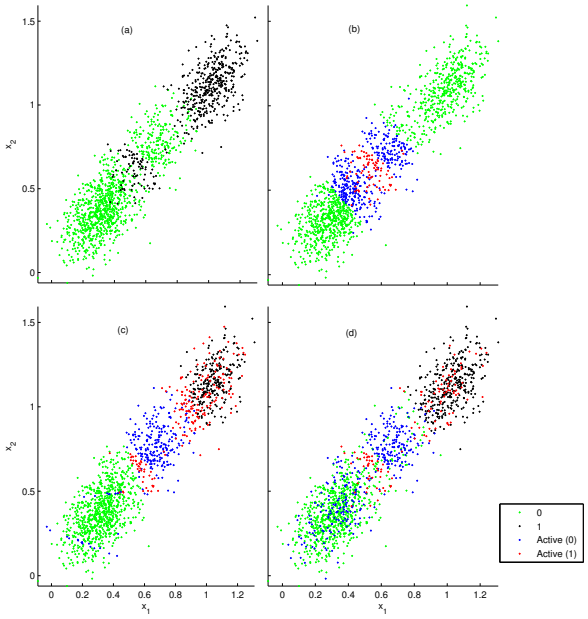


Figure 4.24: **OASL is more efficient in its use of active labels than Active-IW-SL.** Classification regions and active samples for Active-SL (b), OASL (c) and Active-IW-SL (d) in the three Gaussian SP-III dataset (a).

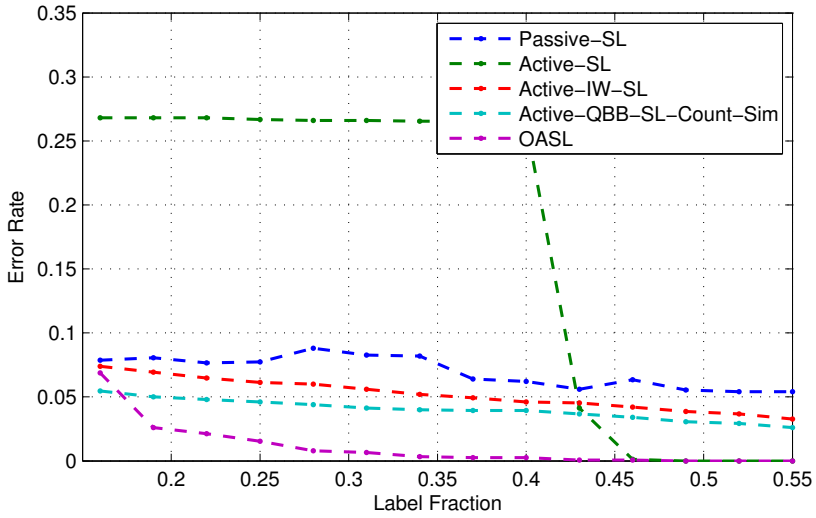
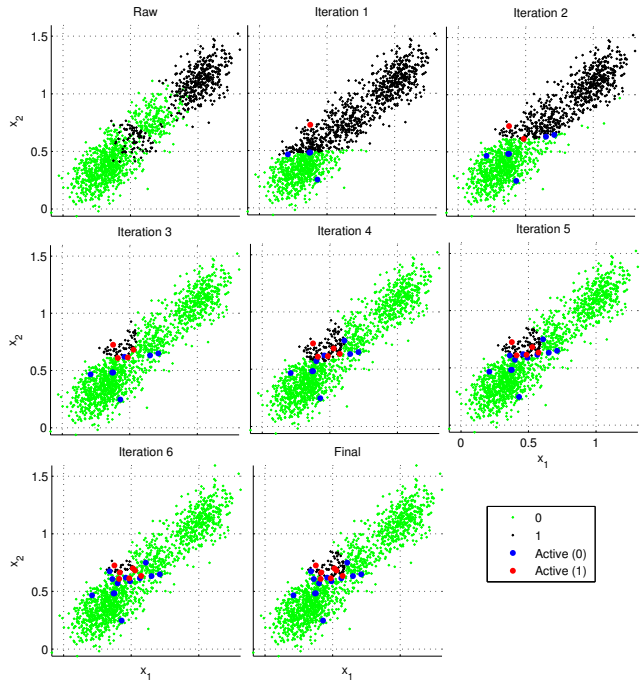
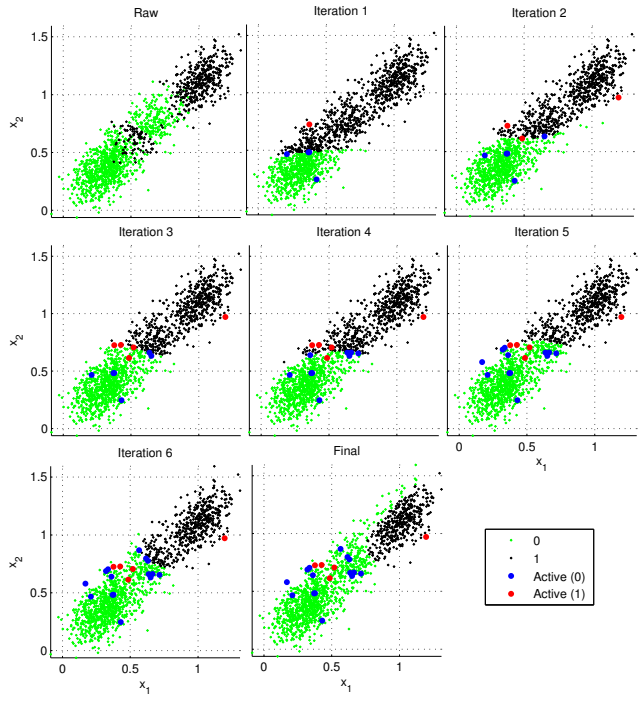


Figure 4.25: Label complexity of SL-based AL algorithms for the SP-III dataset. Passive SL is included as reference.Note classification is almost perfect for Active-SL after 45% active samples since those cover the intermediate Gaussian component entirely (Fig 4.9.1(c)). A batch size of 15 was used for each active algorithm shown.



(a)



(b)

Figure 4.26: Eight iterations of (a) Active-SL (b) OASL for the SP-III data set. The raw data (top left) shows true labelings (green and black). Subsequent plots indicate the labels produced by the base learner (green and black) requested active labels (blue and red). (a) Uncertainty sampling results in requesting active labels from a region biased away from the optimal boundary, resulting in learning a hypothesis with large error. (b) In OASL, requesting a label (the red dot on the top right in Iteration 2) away from the uncertainty zone offsets this bias.

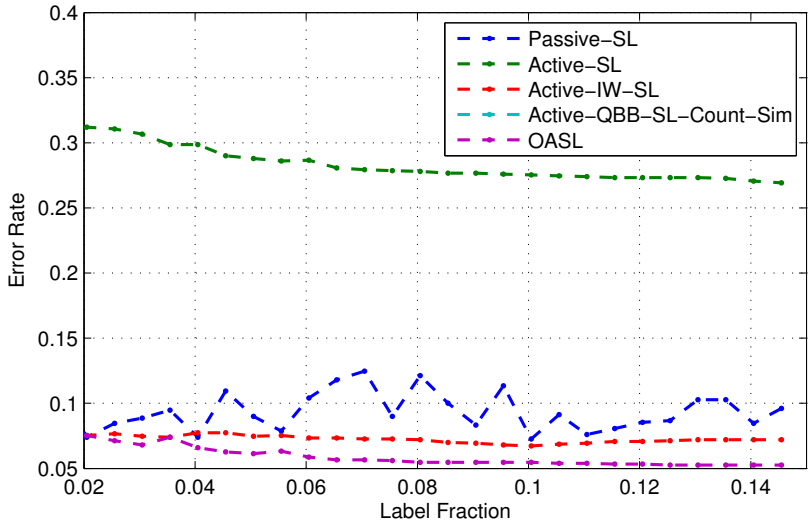


Figure 4.27: Label complexity at *small* label fractions of SL based AL algorithms for the SP-III. A batch size of 2 was used for each active algorithm shown.

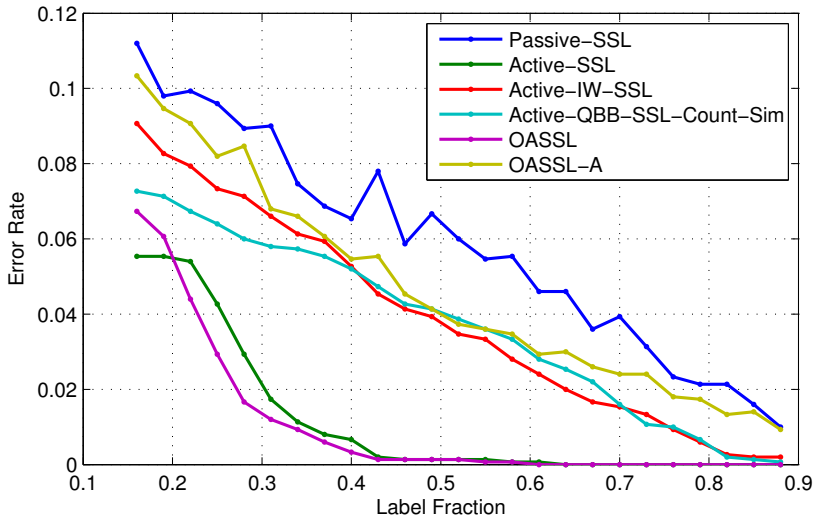


Figure 4.28: Label complexity of SSL based AL algorithms for the SP-III dataset. Passive SSL is included as reference. A batch size of 15 was used for each active algorithm shown.

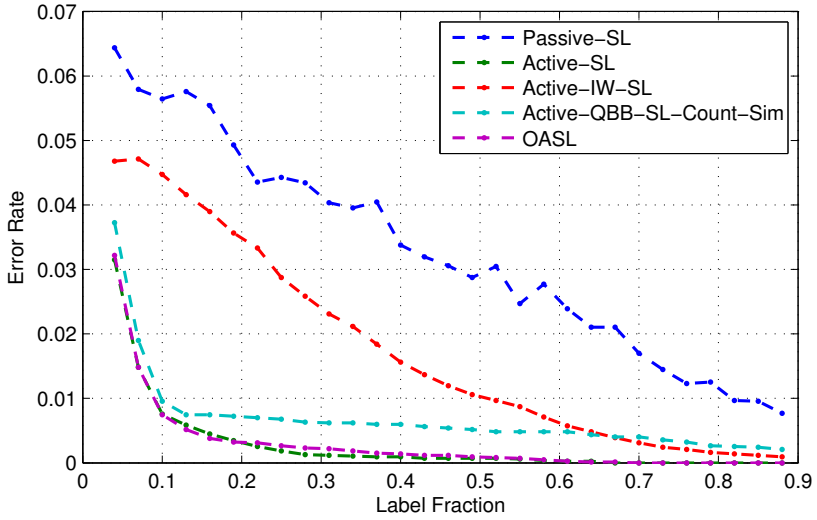


Figure 4.29: Label complexity of SL based AL algorithms for the SP-IV dataset with agnostic noise rate of $\eta = 0$. Passive SL is included as reference. A batch size of 75 was used for each active algorithm shown.

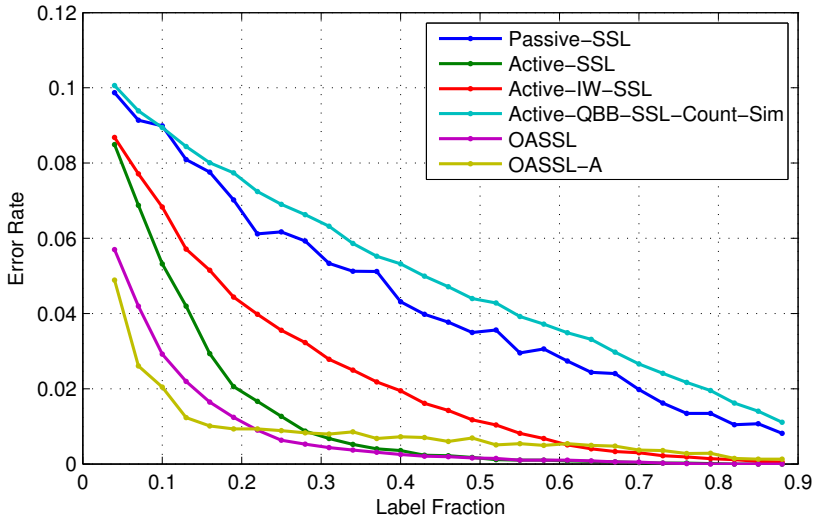


Figure 4.30: Label complexity of SSL based AL algorithms for the SP-IV dataset with an agnostic noise rate of $\eta = 0$. Passive SSL is included as reference. A batch size of 75 was used for all algorithms shown except OASSL-A, where the values of parameters α, β, t_0 determine the batch size at each iteration. The parameters α, β, t_0 were fixed to their optimal values for a label fraction of 0.1.

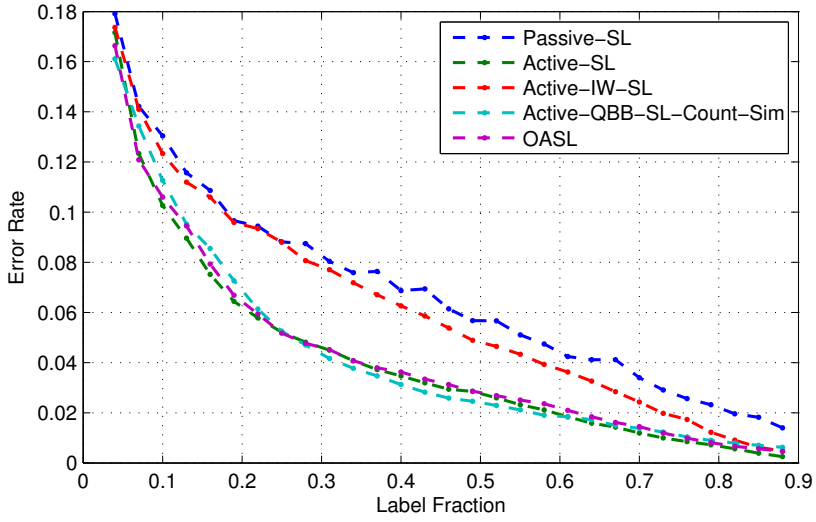


Figure 4.31: Same as Fig 4.29 with $\eta = 0.4$.

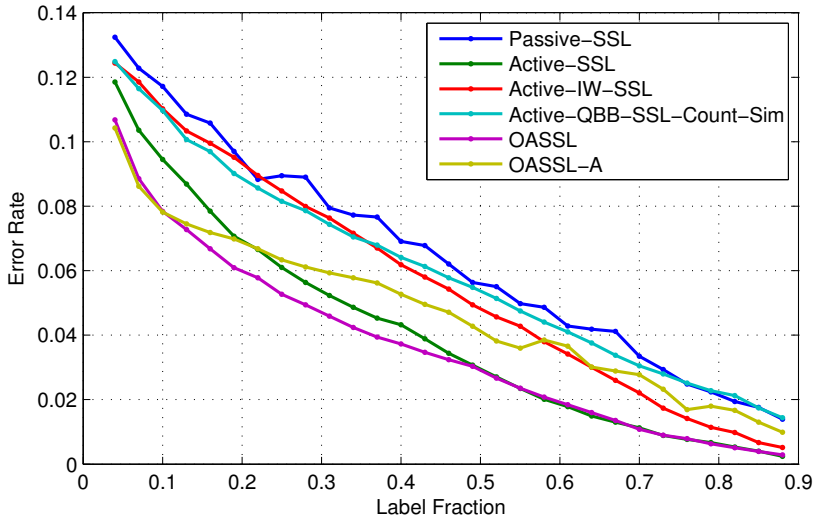
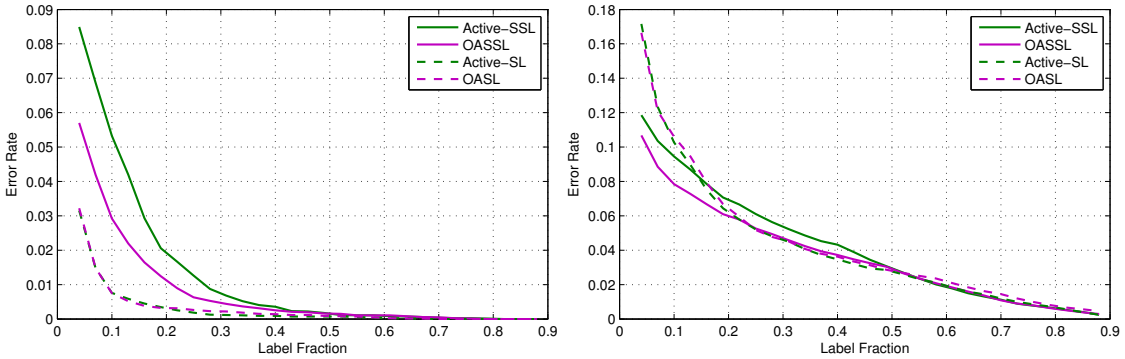
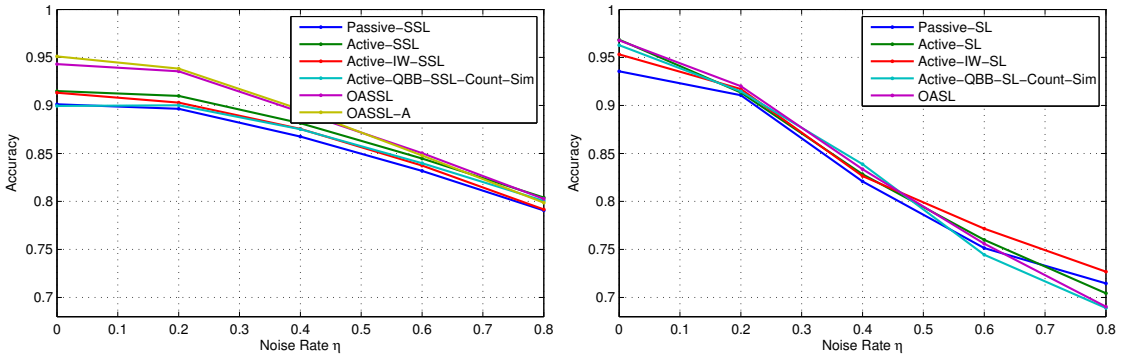


Figure 4.32: Same as Fig 4.30 with $\eta = 0.4$.



(a) $\eta = 0$ (b) $\eta = 0.4$

Figure 4.33: Label complexity of standard US based and OAS based AL for SL and SSL base learners on the SP-IV dataset with an agnostic noise rate of (a) $\eta = 0$ and (b) $\eta = 0.4$. Use of OAS boosts performance of SSL based AL but not SL. However, at the larger noise rate, SSL surpasses SL in performance.



(a) $\eta = 0$ (b) $\eta = 0.4$

Figure 4.34: Noise complexity of standard US based and OAS based AL for (a) SL (b) SSL base learners on the SP-IV dataset with a fixed label fraction of 4%. At higher noise rates SSL based algorithms perform better than SL and the OAS provided boost for SSL learners becomes significant.

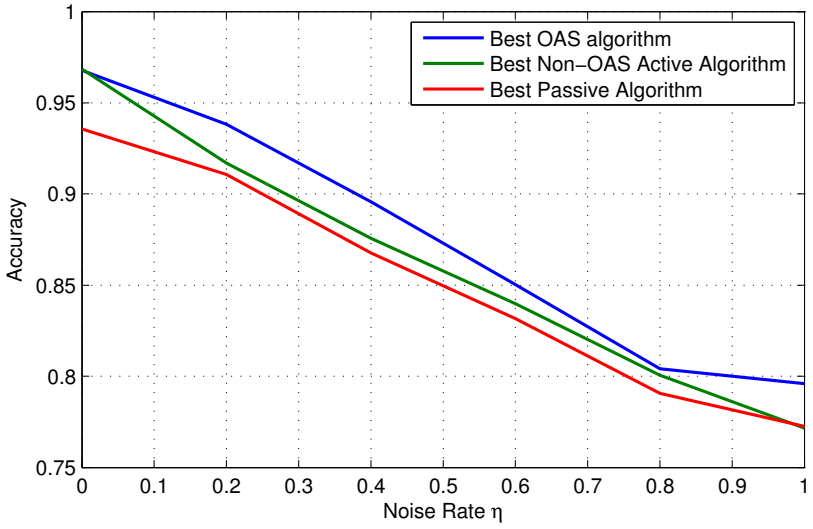


Figure 4.35: Comparison of noise sensitivity (i.e. rate of decline of accuracy with noise rate) for the best algorithms in the OAS-based, non-OAS based and passive categories. OAS based algorithms perform better at all noise levels and have the best sensitivity.

4.10.6 Ensemble Clustering on EEG Artifact Data

Several existing and new algorithms for EC presented in sections 4.6, 4.8 were tested on the EEG artifact and ensemble data (section 4.9.2) for the 8700 manually marked epochs. Results for each category of classifiers are presented below.

1. *Combination based methods for EC:*Table 4.6 shows the accuracy, FPR and FNR from 13 different combination based methods indicating that the best achievable accuracy is 92.75% when using the BKS method.

2. *Transformation based (passive) methods for EC:* Results from several of the unsupervised, SL and SSL passive transformation based methods for various label fractions are shown in Tables 4.7, 4.8 , 4.9 respectively. BGMM-EC-F is the best unsupervised method though it does worse than the best combination method. The best SL method is SVM-EC, with an accuracy of 93.27% which is comparable to the 93.30% accuracy of the best SSL method ML-EC with 10% labels.

3. *Disagreement-based AL for EC:* The algorithms described in the section 4.8.3 were implemented and the results reported separately for SL and SSL base learners Fig

4.10 and 4.11. Note that the ˜100% accuracy when using over ˜40% active labels is reflective of the fact that our ensemble has overall 35% disagreement, that is, 35% of all samples have at least one disagreement amongst members of the ensemble. With 10% active labels, SVM-I-A gives us an accuracy of 94.51%, a modest improvement over the best passive transformation based SL algorithm. In the SSL category, BGMSS-EC-F-A has the best accuracy 93.88% which is only modestly better than best passive SSL algorithm. Thus disagreement based AL methods do not give us much improvement over transformation based passive methods.

4. *Uncertainty-based and OAS-based AL for EC:* Results for the uncertainty-based algorithms Active-SL-EC, Active-SSL-EC and OAS-based algorithms OASL-A-EC, OASSL-EC are in Fig 4.36. With 10% active labels, Active-SL-EC and OASSL-EC have 96% accuracy, while the OASL-A-EC has the best performance, with an accuracy of 97.5%.

Results from the some algorithms in each of the above categories (1-4 above) are summarized in Fig 4.37, showing that overall OASL-A-EC offers the best solution, followed by Active-SL-EC. While the latter has slightly smaller FNR, it has poorer FPR.

Table 4.6: Results from *combination based* methods for ensemble clustering on EEG artifact epoch data. See section 4.6.1.

	Accuracy	FPR	FNR
MV	90.51	7.11	2.38
WV-1	89.36	8.57	2.07
WV-2	90.30	6.94	2.76
SVavg	87.41	11.83	0.76
SWVavg	82.51	17.23	0.26
SVmed	89.09	8.44	2.47
SVmin	75.00	25.00	0.00
SVmax	85.37	2.47	12.16
SVhar	87.36	11.92	0.72
DT	89.49	6.94	3.56
DT-2	90.25	6.44	3.31
DS	89.30	7.25	3.45
BKS	92.75	3.55	3.70

4.10.7 Impact of Parameters on OAS-based algorithms.

For non-adaptive OAS algorithms (and other active learning iterative algorithms), the batch size B has an impact on algorithm performance. In general, a smaller batch size

Table 4.7: Results from **unsupervised** *transformation based* methods for ensemble clustering on EEG artifact epoch data. The best classifier with respect to each of three metrics (accuracy, FNR and FPR) is shown in **bold**.

	Accuracy	FPR	FNR
BMM-EC	89.51	3.62	6.87
GMM-EC	84.84	12.15	**3.01**
BGMM-EC	88.29	7.47	4.24
BGMM-EC-F	**91.01**	**2.91**	6.08

Table 4.8: Results from **supervised** *transformation based* methods for ensemble clustering on EEG artifact epoch data. *Testing* accuracies for 10,30,50 and 70% (randomized) training data are indicated, and *training* accuracy on all of the samples (the shaded columns under the heading 100%). Training accuracy is the accuracy of the trained classifier on trained samples. The best classifier with respect to each of three metrics (accuracy, FNR and FPR) is shown in **bold**.

	Accuracy					FPR					FNR				
	10%	*30%*	*50%*	*70%*	*100%*	*10%*	*30%*	*50%*	*70%*	*100%*	*10%*	*30%*	*50%*	*70%*	*100%*
BMS-EC	9.99	90.34	90.85	89.92	90.32	60.22	3.5	3.29	3.14	3.39	29.8	6.16	5.86	6.93	6.29
GMS-EC	88.84	91.74	91.63	91.34	91.75	6.85	3.55	3.4	3.1	3.03	4.32	4.71	4.97	5.56	5.22
BGMS-EC	89.2	91.76	92.09	91.38	91.8	6.39	3.76	3.2	3.26	3.39	4.42	4.48	4.71	5.36	4.8
BGMS-EC-F	91.39	92.13	92.46	91.38	92.2	2.78	**2.92**	**2.34**	2.64	2.48	5.82	4.94	5.2	5.98	5.32
SVM-I	92.41	92.27	92.62	91.72	92.75	3.58	5.06	4.76	3.91	3.72	4.01	2.68	2.62	4.37	3.53
SVM-S	92.61	93	93.17	92.91	94.17	**2.5**	3.25	2.78	**2.45**	2.14	4.89	3.74	4.05	4.64	3.69
SVM-EC	**93.27**	**93.22**	**93.63**	92.91	94.89	3.36	3.55	3.91	3.75	2.85	3.37	3.23	2.46	3.33	2.26
SVM-EC-F	92.77	93.2	93.43	**93.07**	**97.41**	4.2	4.32	4.3	3.68	**1.32**	3.03	2.48	2.28	3.26	1.26
DISCR-I	90.96	88.18	89.08	87.74	89.03	7.62	11.51	10.46	11.95	10.59	**1.42**	**0.31**	**0.46**	**0.31**	**0.38**
DISCR-S	91.46	91.36	91.49	91	91.47	3.18	4.15	3.66	3.56	3.51	5.36	4.48	4.85	5.44	5.02
DISCR-EC	92.06	90.51	91.22	91.07	91.28	6.48	8.64	7.7	7.78	7.72	1.47	0.85	1.08	1.15	1
DISCR-EC-F	92.54	92.46	92.92	92.34	93.22	4.73	5.47	4.97	4.94	4.45	2.73	2.07	2.11	2.72	2.33

gives better performance though the relationship may not be strictly monotonic. However, smaller B means a longer run-time for all algorithms. We have made the assumption that the qualitative behavior or all algorithms relative to batch size is the same, so in the results presented above we fixed B for all algorithms within each dataset to a value that trades off accuracy vs run-time. To illustrate the impact of batch size, the accuracy of OASL on the SP-IV dataset with $\eta = 0$ is shown in Fig 4.38.

For adaptive OAS algorithms, the parameters α, β, t_0 make a difference in the algorithm convergence and performance. We fixed $t_0 = 5$ as it is a determinant of the total number of iterations which may be a desirable parameter to fix upfront in practical applications. We observed that with $t_0 = 5$ the total number of iterations was between 10 and 25

Table 4.9: Results from **semi-supervised** *transformation based* methods for ensemble clustering on EEG artifact epoch data when using 10,30,50 and 70% (randomized) training data. The best classifier with respect to each of three metrics (accuracy, FNR and FPR) is shown in **bold**

	Accuracy				FPR				FNR			
	10%	*30%*	*50%*	*70%*	*10%*	*30%*	*50%*	*70%*	*10%*	*30%*	*50%*	*70%*
BMSS-EC	9.99	10.95	90.71	89.92	60.22	58.93	3.26	3.14	29.8	30.11	6.02	6.93
GMSS-EC	88.84	89.54	90	91.11	6.85	6.14	5.59	3.52	4.32	4.32	4.41	5.36
BGMSS-EC	89.2	89.61	91.43	91.23	6.39	5.88	3.79	3.41	4.42	4.52	4.78	5.36
BGMSS-EC-F	91.39	91.58	92.18	90.96	2.78	3.12	**2.48**	2.68	5.82	5.3	5.33	6.36
ML-S	92.54	93.17	93.08	92.64	**1.89**	**2.64**	2.57	**2.38**	5.57	4.19	4.34	4.98
ML-EC	**93.3**	93.37	93.77	93.22	2.59	3.32	3.72	3.52	**4.11**	3.32	**2.51**	3.26
ML-EC-F	93.14	**93.92**	**93.91**	**93.33**	2.29	2.89	3.31	3.14	4.57	**3.19**	2.78	**3.52**

Table 4.10: Results from **supervised** *disagreement based AL* methods for ensemble clustering on EEG artifact epoch data when using 10,30,50 and 70% active labels. The best classifier with respect to each of three metrics (accuracy, FNR and FPR) is shown in **bold** for 10% and 30% active labels.

	Accuracy				FPR				FNR			
	10%	*30%*	*50%*	*70%*	*10%*	*30%*	*50%*	*70%*	*10%*	*30%*	*50%*	*70%*
BMS-EC-A	6.55	98	99.49	99.43	65.73	0.38	0.07	0.08	27.71	1.63	0.44	0.5
GMS-EC-A	91.8	94.5	99.36	99.16	4.23	4.3	0.21	0.31	3.97	1.2	0.44	0.54
BGMS-EC-A	93.72	97.93	99.36	99.39	1.48	**0.54**	0.21	0.11	4.8	1.53	0.44	0.5
BGMS-EC-F-A	93.88	97.9	99.4	99.35	**1.46**	0.66	0.18	0.15	4.66	1.44	0.41	0.5
SVM-I-A	**94.51**	97.91	99.49	99.43	3.1	1.31	0.07	0.08	2.39	0.77	0.44	0.5
SVM-S-A	45.68	92.18	99.03	99.31	53.74	7.04	0.6	0.23	0.57	0.77	0.37	0.46
SVM-EC-A	35.98	86.86	99.49	99.43	63.77	12.4	0.07	0.08	0.26	0.74	0.44	0.5
SVM-EC-F-A	35.06	66.88	99.26	99.16	64.74	32.63	0.32	0.34	**0.2**	**0.49**	0.41	0.5
DISCR-I-A	92.76	**97.98**	99.49	99.43	5.95	1.12	0.07	0.08	1.29	0.9	0.44	0.5
DISCR-S-A	43.79	94.01	99.26	99.16	55.42	4.83	0.3	0.31	0.79	1.17	0.44	0.54
DISCR-EC-A	62.01	90.67	99.36	99.27	36.65	8.56	0.21	0.23	1.34	0.77	0.44	0.5
DISCR-EC-F-A	56.92	91.31	98.53	98.77	42.09	7.87	1.1	0.8	0.98	0.82	0.37	0.42

on the SP-IV dataset. We optimized α, β for this value of t_0 by using a grid search on the accuracy outcome of the OASL-A algorithm on the SP-IV dataset, and found the value to be $\alpha = 3.2, \beta = 1.2$.

We noticed that with these optimal values, the total number of active samples requested and confident samples' output saturate to a steady value. As discussed before, the adaptive versions OAS-A switch to random sampling after there is no change in the

Table 4.11: Results from **semi-supervised** *disagreement based AL* methods for ensemble clustering on EEG artifact epoch data when using 10,30,50 and 70% active labels. The best classifier with respect to each of three metrics (accuracy, FNR and FPR) is shown in **bold** for 10% and 30% active labels.

	Accuracy				FPR				FNR			
	10%	*30%*	*50%*	*70%*	*10%*	*30%*	*50%*	*70%*	*10%*	*30%*	*50%*	*70%*
BMSS-EC-A	6.55	2	99.49	99.43	65.73	74.88	0.07	0.08	27.71	23.12	0.44	0.5
GMSS-EC-A	91.8	97.5	99.17	99.08	4.23	1.17	0.39	0.38	3.97	1.33	0.44	0.54
BGMSS-EC-A	93.72	97.93	99.4	99.39	1.48	0.51	0.16	0.11	4.8	1.56	0.44	0.5
BGMSS-EC-F-A	**93.88**	**98.03**	99.4	99.35	**1.46**	**0.49**	0.16	0.15	4.66	1.48	0.44	0.5
ML-S-A	47.19	94.91	99.15	99.35	52.22	4.27	0.48	0.19	**0.59**	**0.82**	0.37	0.46
ML-EC-F-A	52.35	95.67	99.49	99.43	47.06	3.58	0.07	0.08	**0.59**	**0.76**	0.44	0.5
ML-EC-A	75.95	95.34	99.49	99.43	23.14	3.81	0.07	0.08	0.91	0.85	0.44	0.5
ML-EC-F-A-GM	90.55	97.37	-	-	3.65	1.82	-	-	5.8	0.8	-	-
ML-EC-A-GM	90.8	97.21	-	-	3.42	1.9	-	-	5.77	0.89	-	-

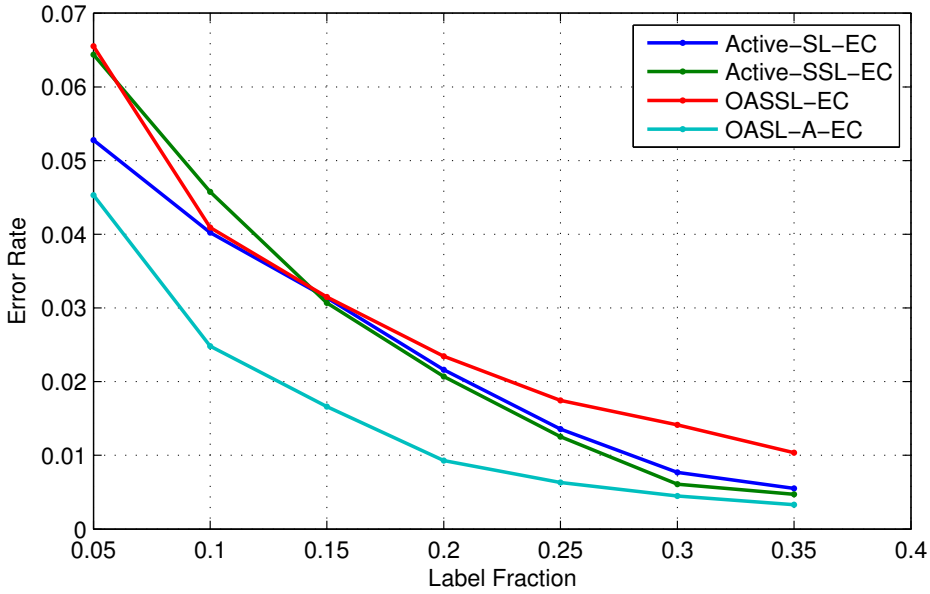


Figure 4.36: Label complexity of *uncertainty and OAS based AL* methods for ensemble clustering on EEG artifact epoch data. Since the ensemble has 35% disagreement rate, labels beyond 35% are not necessary. The fewest possible active labels are preferred; therefore OASL-EC-A offers the best solution.

number of confident samples outputted. The empirical analysis shows that accuracy does not improve beyond this point (Fig 4.39), thereby justifying the idea behind the algorithm. The best possible accuracy at this saturation point is achieved when the total number of

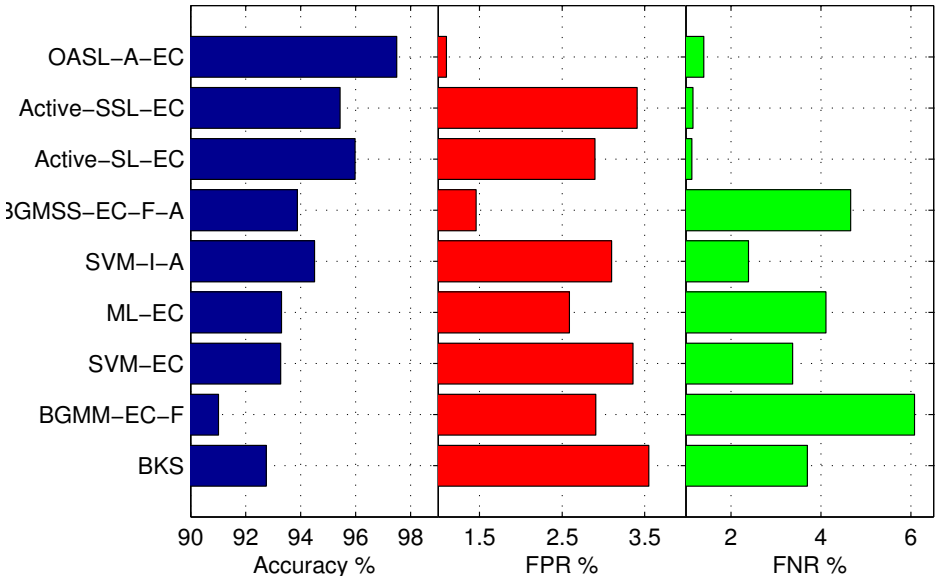


Figure 4.37: Comparison of the some ensemble classification algorithms for the EEG artifact epoch data. Only the best algorithms in their respective categories are shown.

active samples requested equals the maximum number of active requests allowed. Empirical analysis shows that this is attained for the optimal values of α, β above for the SP-IV dataset.

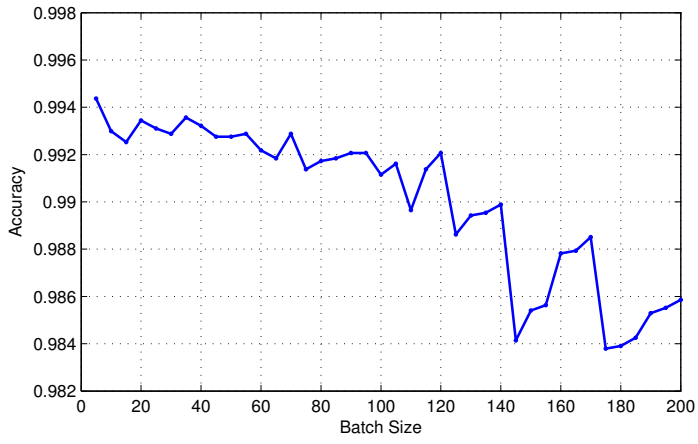


Figure 4.38: Impact of batch size on accuracy of the OASL algorithm on the SP-IV dataset with label fraction = 0.1 and $\eta = 0$.

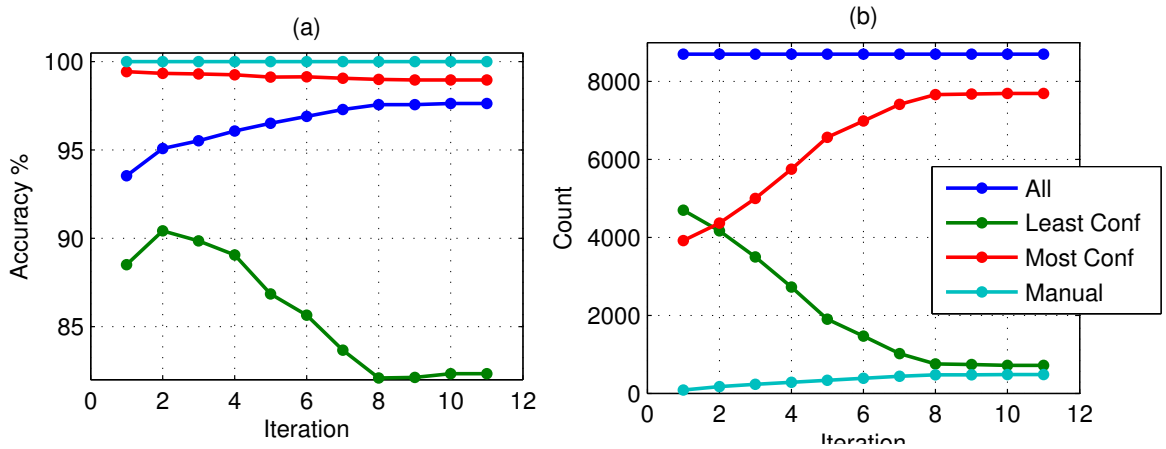


Figure 4.39: (a) Accuracies and (b) Number of all, most confident, least confident and manual (active) samples, for iterations of the OASSL-A algorithm on the SP-IV dataset with $\alpha = 3.2, \beta = 1.2, t_0 = 5$. At a certain point (iteration 9) there is no increase in the number of confident samples that can be learnt and hence no more active samples are requested. For the optimal set of parameters, this limit of active samples equals the maximum number of active samples that can be requested.

4.11 Conclusions and Discussion

We have introduced a new method (OAS) for selection of active samples that provides a computationally simple alternative to existing active selection algorithms that attempt to address pitfalls of uncertainty based sampling in noisy (non-separable) data. We implemented our method using a Bernoulli-Gaussian model based E-M algorithm and tested it on four different simulated Gaussian mixtures using both supervised and semi-supervised Gaussian mixture base learners. Our algorithms perform better than several state-of-the-art algorithms that we adapted specifically for these base learners. Finally, we showed how we can solve the clustering problem in an ensemble with non re-trainable members using adaptive versions of our algorithms and showed that when applied to real EEG artifact data, our algorithm boosts the accuracy of epoch classification by an ensemble of automated classifiers from 92% to 97.5% which was not possible with standard methods.

OAS obviates the problem of sampling bias inherent in uncertainty sampling by incorporating the predicted output of a base learner. This is particularly helpful for supervised learning (SL), as in the case of the three-Gaussian mixture (SP-III), where the presence of localized but agnostic noise causes uncertainty sampling bias to ignore poorly classified regions. We also saw that in the case of non-separable but non-agnostic noise, as in SP-II

and SP-IV, OAS also offsets excess bias resulting from unlabeled examples of semi-supervised learners (SSL) and is thus able to converge to the optimal decision boundary much faster than uncertainty sampling based active learning. When the level of separability is low (as in SP-I) OAS modestly helps both SL and SSL. In cases with non-localized agnostic noise (as in SP-IV), OAS did not give any significant added advantage for SL but did for SSL, a distinction that becomes relevant with added degrees of noise.OAS also consistently demonstrated improved performance across all data sets. In constrast, some algorithms - such as QBC based algorithms - tended to work well on some data sets and break down on others. In some cases active learning can actually perform worse than passive learning past a certain number of labels; our adaptive versions (OAS-A) are able to detect when active learning ceases to be effective and thereby maintain high accuracy at all label fractions.

To summarize, OAS has the following advantages over other active learning algorithms: (i) computational simplicity, which permits real-time implementations (ii) consistent better performance in presence of both agnostic and non-agnostic noise, (iii) ease of adaptivity to ensemble learning since it internally uses an ensemble based classifier, (iv) option of adaptive implementations that can detect when active learning is no longer effective.

In our current testing and implementation, some parameters (such as α, β) were chosen by empirical testing. A future implementation could be able to learn these parameter adaptively perhaps resulting in even higher accuracy. OAS could also be used in conjunction with existing active sampling techniques such as density weighting. In addition, alternate implementations would allow it to be used with non EM based base learners.

Chapter 5

Estimation & Control For Decomposable Markov Chains: Theory & Applications

5.1 Abstract

The subject of this chapter is to explore a class of Markov processes where the transition rates are, in addition, dependent upon the state of another stochastic processes and are thus Markov processes themselves. Our purpose is to describe a broad range problems in which these so-called *cascade Markov processes* (CMP) admit explicit solutions for both hidden state estimation and optimal control [31]. While a cascade Markov process can be equivalently represented on the joint (coupled) state space as a non-cascade, the main purpose of this paper is to investigate solutions on *decomposed* state spaces, in particular, decoupled equations for inference and state estimation in hidden Markov models for the purpose of computational efficiency. By reduction of a partially observable cascade optimal control problem to a lower dimensional non-cascade problem we are able to circumvent the "curse of dimensionality". Our approach of working decomposed representations is generalizable to multi-factor processes, stochastic automata networks [179], and even quantum Markov chains and controls [101],[100].

5.2 Introduction

It is well known that both Markov decision processes (MDPs) and hidden Markov models (HMMs) suffer from the "curse of dimensionality" where the state space grows exponentially in the number of factors or variables under consideration. For example, with 20 boolean valued variables the total state space size is 10^6. In this chapter, we look at "factorization" techniques for problems on large state space problems, where the state space is expressed as the product of sub-spaces. The "decomposition" approach, where the state space is expressed as the direct sum of sub spaces was the subject of chapters 1-3. Our goal is to derive factored *solutions* because factored representations in themselves do not always guarantee efficient solutions. We use the framework of discrete-time Hidden Markov Models (HMMs) for state estimation, and for the particular framework of continuous-time Markov decision processes that closely follows the assumptions and modeling of [31]. In this part of the chapter, a mathematical framework is outlined, and then solutions are derived for hidden state estimation and a class of optimal control problems where the cost function is a the expectation of a functional. In Part II of this chapter, the solutions developed for hidden state estimation are applied to a real-time solution for gait classification and fall detection and demonstrated on actual human data. Toy examples for application demonstrating application of our techniques on Markov decision processes is available in the Online Supplemental Material. While our exposition here is limited to a single cascade with only a single level of dependency, the model is easily extendible to many layers of dependencies, such as Markov decision trees ([116]) and stochastic automata networks ([179]).

5.3 Continuous-time Cascade Markov Chains

We use the framework of [31] for continuous-time finite-state (FSCT) Markov processes. We assume a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and right-continuous stochastic processes adapted to a filtration $\mathbb{F} = (\mathcal{F}_t)_{t \in T}$ on this space., the set of n standard basis vectors in $\mathbb{R}^n$, has the following sample path (Itô) description: descriptions:

$$dx = \sum_{i=1}^{m} G_i x dN_i \tag{5.1}$$

where $G_i \in \mathbb{G}^n$ are *distinct*[1], $\mathbb{G}^n$ being the space of square $n-$matrices of the form $F_{kl} - F_{ll}$ where F_{ij} is the matrix of all zeros except for one in the $i'th$ row and $j'th$ column, and N_i are Poisson counters with rates λ_i. The resulting *infinitesimal generator* that governs the transition probabilities of the process is $P \in \mathbb{P}^n$, the space of all stochastic $n-$matrices and is given by:

$$P = \sum_{i=1}^{m} G_i \lambda_i \tag{5.2}$$

For continuous-time formulation, we define a stochastic matrix P with non-negative entries and each column has zero sum.

For purposes of this chapter, we will be interested in the case where transition rates of $x_t \in \{e_i\}_{i=1}^n$ are themselves stochastic: specifically, they depend on the state of another Markov process, say, $z_t \in \{e_i\}_{i=1}^r$. We will call such a pair to form a (continuous-time) **Cascade Markov chain (**CT-CMC) In general, various levels of interactions between two processes x_t and z_t defines a joint Markov process $y_t = z_t \otimes x_t$ that evolves on the product space $\{e_i\}_{i=1}^n \times \{e_i\}_{i=1}^r$ but we are specifically interested in CT-CMCs where sample paths of z_t and x_t have the following have the following Ito description

$$dz = \sum_{i=1}^{s} H_i z dM_i \tag{5.3}$$

$$dx(z) = \sum_{i=1}^{m} G_i(z) x dN_i(z) \tag{5.4}$$

where $H_i \in \mathbb{G}^r,\ G_i(z) \in \mathbb{G}^n$ and the rates of Poisson counters M_i and N_i are ν_i and λ_i with λ_i depending on the state of z_t. Thus the infinitesimal generators P and C of x_t and z_t (P depends on z_t and $P(z)$ propagates the *conditional* probabilities of x_t given z) are

$$P(z) = \sum_{i=1}^{m} G_i \lambda_i(z) \tag{5.5}$$

$$C = \sum_{i=1}^{s} H_i \nu_i \tag{5.6}$$

5.3.1 Markov Processes on Product State Spaces

We describe representations of a Markov Process y_t that evolves on the product state space $\{e_i\}_{i=1}^r \times \{e_i\}_{i=1}^n$. The sample path $y(t)$ can be written as the tuple $(z(t), x(t))$

[1]If the $G_i's$ are not distinct, then one can combine the Poisson counters corresponding to identical $G_i's$ to get a set of distinct $G_i's$. For example, $G_1 y dN_1 + G_1 y dN_2$ can be replaced by $G_1 y dN$ where $dN = dN_1 + dN_2$, a Poisson counter with rate equal sum of the rates of the counters N_1, N_2

where $z(t) \in \{e_i\}_{i=1}^r$ and $x(t) \in \{e_i\}_{i=1}^n$. The corresponding stochastic processes z_t and $x_t$are the **components** of y_t. The transition matrix for x_t may depend on $z(t)$ and hence describes the propagation of the *conditional probability* distribution $p_{x|z}$: The dynamics of component *marginal* probabilities are not necessarily governed by a single stochastic matrix. Different degrees of coupling between x_t and y_t leads to a possible categorization of the joint Markov Process y_t (see notation defined in Section 5.6).

- For an **Uncoupled Markov Process** on $\{e_i\}_{i=1}^r \times \{e_i\}_{i=1}^n$ the transition probability from state (e_i, e_j) to (e_i, e_k) does not depend on i, for all i, j, k where $1 \leq i \leq r$ and $1 \leq j, k \leq n$. In this case, the infinitesimal generator P of y_t can be written in the form

$$P = I_r \otimes A + C \otimes I_n$$

  where $A \in \widehat{P}_n$ and $C \in \widehat{P}_r$.

- In a **Cascade Markov process**[2]**,** transition probability from state (e_i, e_j) to (e_l, e_j) does not depend on j, for all i, l, j where $1 \leq i, l \leq r$ and $1 \leq j \leq n$. In this case, the infinitesimal generator P of y_t can be written in the form

$$P = \sum_{i=1}^{p} E_i^r \otimes B_i^n + C \otimes I_n$$

  where $C \in \widehat{P}_r$, where $B_i^n$are matrices such that $\sum_{i=1}^{p_1} B_i^n \in \widehat{P}_n$

- In a **Weakly-coupled** or **decomposable Markov Process,** all non-zero transition probabilities are between states of the form (e_i, e_j) to (e_i, e_k), or (e_i, e_j) to (e_l, e_j) for $i \neq k$ and $j \neq l$. In this case, the infinitesimal generator P of y_t can be written in the form

$$P = \sum_{i=1}^{p_1} E_i^r \otimes B_i^n + \sum_{i=1}^{p_2} B_i^r \otimes E_i^n$$

  where B_i^n, B_i^r are matrices such that $\sum_{i=1}^{p_1} B_i^n \in \widehat{P}_n$ and $\sum_{i=1}^{p_2} B_i^r \in \widehat{P}_r$.

- In the most general **Non-Decomposable Markov Process,** there exist states (e_i, e_j) and (e_k, e_l) with $i \neq k$ and $j \neq l$ having non-zero transition probability. In this case, the infiinitesmal generator P of y_t can be written in the form

$$P = \sum_{i=1}^{p_1} E_i^r \otimes B_i^n + \sum_{i=1}^{p_2} B_i^r \otimes E_i^n$$

[2]In this paper we mainly focus on Cascade Markov processes, and they are closely related to Markov-modulated Poisson processes (MMPPs) which have vast applications in traffic control, operations research and electronics and communications.

where B_i^n, B_i^r are matrices such that $\sum_{i=1}^{p_1} B_i^n \in \widehat{P}_n$ and $\sum_{i=1}^{p_2} B_i^r \in \widehat{P}_r$.

The first three cases above have what is known as *functional* transition rates, that is, the transition rates are state dependent but do not have any synchronous transitions. Non-decomposable Markov chains exhibit *synchronous transitions*: that is, transitions amongst states of x_t and z_t can occur simultaneously.

5.3.2 Cascade Markov Decision Processes (CMDP)

For the problem of optimal control we will use the above framework of CT-CMCs. For the Markov chain described by (5.1) and (5.2) if we let the transition rates are allowed to depend on $\mathcal{F}_t-$progressively measurable control processes $u = (u_{1,}u_2...u_p)$ in an affine accordance with[3]:

$$\lambda_i = \lambda_{i0} + \sum_{j=1}^{p} \mu_{ij} u_j$$

then the resulting process is called a *Markov Decision Process*. The infinitesimal generator can then be written as:

$$P(u) = \sum_{i=1}^{m} G_i \left(\lambda_{i0} + \sum_{j=1}^{p} \mu_{ij} u_j \right)$$

In a **Cascade Markov decision process (CMDP)**, we assume the rates λ_i of counters N_i are allowed to additionally depend on $\mathcal{F}_t-$progressively measurable control processes $u = (u_{1,}u_2...u_p)$ in accordance with [4]

$$\lambda_i(z) = \lambda_{i0}^0 + \lambda_{i0}(z) + \sum_{j=1}^{p} \mu_{ij}(z) u_j$$

so that the conditional probability vector $p(z,u)$ [5] of x_t given z evolves as

$$\dot{p}(z,u) = \sum_{i=1}^{m} G_i \left(\lambda_{i0}^0 + \lambda_{i0}(z) + \sum_{j=1}^{p} \mu_{ij}(z) u_j \right) p(z,u)$$

which will be abbreviated as

$$P(z,u) = A_0 + A(z) + \sum_{j=1}^{p} u_j B_j(z) \tag{5.7}$$

$$\dot{p}(z,u) = P(z,u) p(z,u) \tag{5.8}$$

[3] that is, we assume an affine dependence on controls

[4] Each term is, in additional, a function of time t but for clarity explicit dependence on t will not be specified in notation.

[5] same as above.

The CMDP model is completely specified by (A_0, A, B_j). The requirements on $P(z,u)$ to be an infinitesimal generator *for each* z put constraints on the matrices A_0, A, B_j and impose admissibility constraints on the controls u_j. We will require A_0 and A to be infinitesimal generators themselves (for each t and z) and the B_j to be matrices whose columns sum to zero (for each t and z). We also allow the controls to be dependent on z and x which will define the set of admissible controls $\mathcal{U}$ as the set of measurable functions mapping the space $\{e_i\}_{i=1}^r \times \{e_i\}_{i=1}^n$ to the space of controls $\mathbb{R}^p$ such that the matrix with j^{th} column

$$f_j = A_0 e_j + A(e_k)e_j + \sum_{i=1}^{p} u_i(e_k, e_j)B_j(e_k)$$

for $j = 1..n,\ k = 1..r$ is an infinitesimal generator.

5.4 Discrete-time Cascade Markov Chains

In a manner similar to that of CTMCs, we define a finite-state discrete-time Markov chain (DTMC) as the discrete-time process $\{x_t\}_{t\in\mathbb{Z}^+}$ with each $x_t \in \{e_i\}_{i=1}^n$ with the Markov property, if $\mathcal{F}_t$ is the filtration generated by the sigma field $\sigma\{x_1, ...x_t\}$ then

$$\Pr(x_{t+1} = e_j|\mathcal{F}_t) = \Pr(x_{t+1} = e_j|x_t)$$

Further, we assume the Markov chain is time-homogeneous, that is, the above probability is independent of t. In that case the dynamics of the process $\{x_t\}$ are completely determined by the *transition matrix* $A \in \mathbb{R}^{n\times n}$ where $A_{ij} = \Pr(x_{t+1} = e_i|x_t = e_j)$. If p_t is the vector with $p_{t,i} = \Pr(x_t = e_i)$ then it is easy to see from the Markov property that

$$p_{t+1} = Ap_t$$

Note that each column of a transition matrix sums to one, and the space of transition matrices is the convex hull of permuation matrices.

In a manner similar to the continuous-time case, we will be interested in the case where the transition matrix of x_t is itself stochastic, depending on the state of another DTMC $z_t \in \{e_i\}_{i=1}^r$. We will call this pair a discrete-time Cascade Markov chain (DT-CMC) and denote the transition matrix as the function $A(z_t)$ to emphasize this dependence. Just like in the continuous-time case, various levels of coupling between x_t and z_t can be defined, but we will be specifically interested in this one-way coupling, where the transition matrix C of z_t does not depend on x_t. Analogous to the definitions in the continuous case, the different levels

of couplings between the two processes allows us to write the representations for dynamics of the Markov process $y_t = z_t \otimes x_t$ on the product space $\{e_i\}_{i=1}^r \times \{e_i\}_{i=1}^n$. However, unlike the case of continuous-time, we will not ignore the case of synchronous transitions. Specifically, if P is the transition matrix of y_t then

- The process $y_t = z_t \otimes x_t$ is called **Decoupled** the transition matrix P of y_t is of the form

$$P = C \otimes A$$

  In literature, sometimes this is also called a **Factorial** Markov chain.

- The process $y_t = z_t \otimes x_t$ is a **Cascade** if the transition matrix P of y_t is of the form

$$P = \sum_{j=1}^{J} E_j C \otimes B_j$$

  where $C \in \mathbb{R}^{r\times r}$ and $B_j \in \mathbb{R}^{n\times n}$ are transition matrices, and $E_j \in \mathbb{R}^{r\times r}$ are rank one matrices with only $0s$ and $1s$, and $J \leq r$.

- In the most general or **Non-decomposable** case, P can be written as

$$P = \sum_{j=1}^{J}\sum_{k=1}^{K} E_j C_k \otimes B_j F_k$$

  where $C_i \in \mathbb{R}^{r\times r}$, $B_j \in \mathbb{R}^{n\times n}$ are transition matrices, $E_j \in \mathbb{R}^{r\times r}$,$F_k \in \mathbb{R}^{n\times n}$ are rank one matrices with only $0s$ and $1s$, and $J \leq r, K \leq n$. Note that alternate representation in terms of a normalized product of tensor products is also possible ([30]).

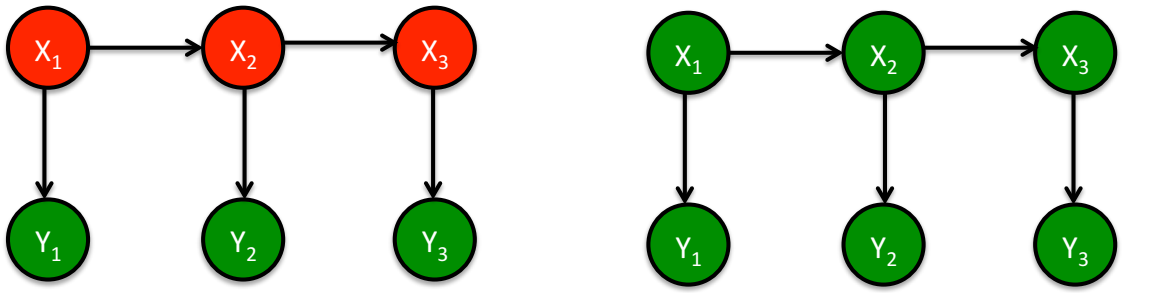
5.4.1 Cascade Hidden Markov Models (CHMM)

For the problem of state estimation, we will use the above framework of DT-CMCs. In particular, we will be using the framework of Hidden Markov Models (HMM). In the (non-cascade) HMM model, the DTMC is not observed directly, but is via an observation process $\{y_t\}_{t\in\mathbb{Z}^+}$ where the state space of y_t can be either continuous or discrete. For purposes of our analysis, we will assume that the state space of y_t is *continuous* i.e. $y_t \in \mathbb{R}^d$. The observation y_t depends only the current state x_t, that is,

$$\Pr(y_t|x_1, x_2...x_t, y_1...y_{t-1}) = \Pr(y_t|x_t)$$

The above model is sometimes represented via a dynamic Bayesian network (DBN) (Fig 5.1). In an HMM, the state x_t is unobservable and the problem is often that of estimating the state x_t given the observations.

In a cascade hidden Markov model (CHMM), the Markov chain x_t is replaced with a DT-CMC (z_t, x_t) as described above. However, the observations y_t are dependent on *both* x_t and z_t (Fig 5.2). In a fully hidden CHMM, both z_t, x_t are unobservable. This model is a simplification of the hidden Markov decision tree model of [117]. The coupling of x_t, z_t via the output y_t is what makes this problem difficult to solve. Even if the Markov chain were decomposable, the coupling via the output can make this problem intractable and difficult to solve. As a simplification, we will be also interested in the case where only x_t is unobservable but z_t is not, which we call a partially observable cascade HMM (PO-CHMM). Note that in the latter model, the process z_t is also a Markov process, and it is slightly differen than the Input-Output HMM of [23, 161, 194]. For purposes of our particular application discussed in Part B of this chapter, we will be also interested in a further simplification, where z_t is the same for all t. That is, z_t is simply a random variable rather than a stochastic process (Fig 5.3).



(a) Hidden Markov Model (b) Fully Observable Markov Model

Figure 5.1: A dynamic Bayesian network (DBN) for a (a) Hidden Markov model and (b) fully observable Markov model, with three time steps. X_1, X_2, X_3 are the state variables at the three time steps, and Y_1, Y_2, Y_3 are the corresponding observations. A red circle indicates a variable that is hidden (unobservable), and green circle indicates a variable that is observable.

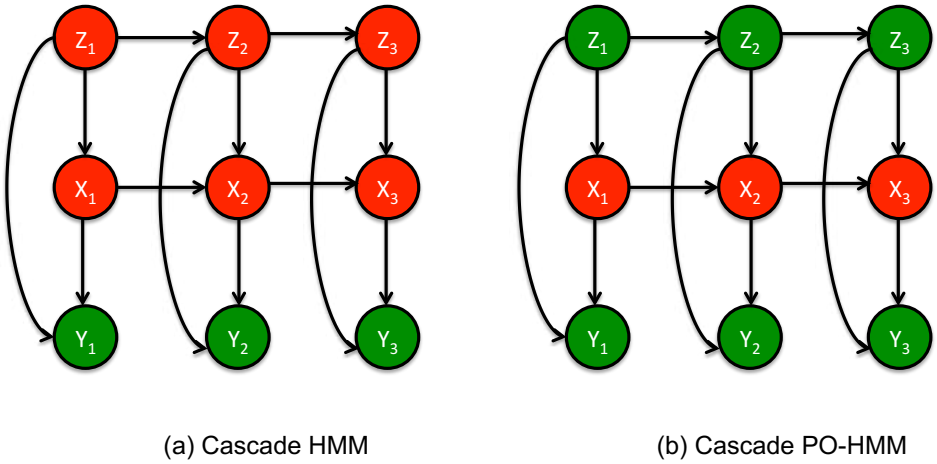


Figure 5.2: Dynamic Bayesian network (DBN) for a (a) Cascade Hidden Markov Model and (b) Partially Observable Cascade Hidden Markov Model, with three time steps. (X_i, Z_i) for $i = 1, 2, 3$ denote the state variables with one-way dependence amongst them, and Y_i are the corresponding observations. Red circles denote unobservable variables, adn green circles denote observable variables.

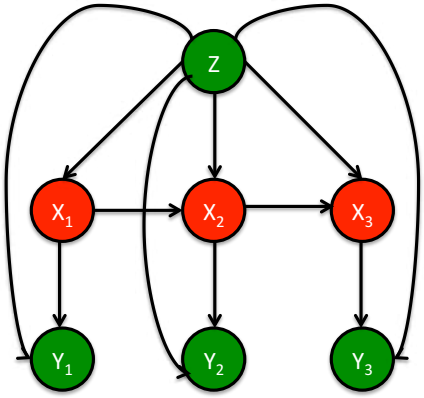


Figure 5.3: Special case of a PO-CHMM when z_t is assumed stationary over a training period $t = 1..T$. In this case we replace the process z_t by a single random variable Z. This is one model in the application discussed in Chapter 5, part B.

5.5 Algorithms for State Estimation In CHMMs

The problem of inference of model parameters and estimation of hidden state given observations $y_1...y_T$ in standard (non-cascade) HMMs is briefly reviewed first before we extend

the exposition to cascade HMMs.

5.5.1 Review of (non-cascade) HMMs

A brief discussion of algorithms of HMMs is provided below, with more details in the Online Supplemental Material. We consider the CTMC $\{x_t\}_{i=1,2,...}$ where $x_t \in \{e_i\}_{i=1}^n$ with transition matrix A i.e. $A_{ij} = \Pr(x_t = e_i | x_{t-1} = e_j)$ and initial probability vector π, i.e. $\pi_i = \Pr(x_1 = e_i)$. It is easy to see that in our notation, $\Pr(x_t) = \mathbb{E}(x_t)$ and $\Pr(x_t = e_i) = \mathbb{E}(x_{t,i})$, where $\mathbb{E}$ is the expectation operator, and the the i^{th} component of the vector x_t is $x_{t,i}$. We assume a Gaussian output model, that is, $y_t | x_t \sim N(\mu_t, \Sigma)$ i.e conditional on x_t the process y_t is Gaussian with covariance $\Sigma > 0$ and mean μ_t that depends on the value of x_t. In our notation, we can write $\mu_t = W x_t$ where $W \in \mathbb{R}^{d \times n}$ thus,

$$\Pr(y_t | x_t) = (2\pi)^{-\frac{d}{2}} |\Sigma|^{-\frac{1}{2}} \exp\left\{ -\frac{1}{2} (y_t - W x_t)^T \Sigma^{-1} (y_t - W x_t) \right\}$$

Three well-known algorithms for three problems of interest on HMMs are described below. We use the compact notation $y_{1:T}$ to denote the sequence $y_1, y_2 ... y_T$ etc.

1. Parameter Learning

Here we are interested in *learning the parameters* $\theta = (\pi, A, W, \Sigma)$ *given observations* $y_{1:T}$. We assume x_t is unobservable. This can be done via the E-M algorithm by forming the log likelihood of complete data comprising of observed $y_{1:T}$ and the hidden $x_{1:T}$. Writing $\langle \cdot \rangle$ for the conditional expectation $\mathbb{E}\left(\cdot | y_{1:T}, \theta^{(k)} \right)$ given observations $y_{1:T}$ and existing model $\theta^{(k)}$ it can be shown that the update rules using the M-step for the k^{th} iteration are

$$\pi \longleftarrow \left\langle x_1^T \right\rangle \tag{5.9}$$

$$A_{ij} \longleftarrow \frac{\sum_{t=2}^{T} \langle x_{t,i} x_{t-1,j} \rangle}{\sum_{t=2}^{T} \langle x_{t-1} \rangle_j} \tag{5.10}$$

$$W \longleftarrow \left(\sum_{t=1}^{T} y_t \langle x_t \rangle^T \right) \left(\sum_{t=1}^{T} \left\langle x_t x_t^T \right\rangle \right)^+ \tag{5.11}$$

$$\Sigma \longleftarrow \frac{1}{T} \sum_{t=1}^{T} y_t y_t^T - \frac{1}{T} \sum_{t=1}^{T} W \langle x_t \rangle y_t^T \tag{5.12}$$

$$\log p(y_{1:T}, x_{1:T}) = \log p(x_1) + \sum_{t=1}^{T} \log p(y_t|x_t) + \sum_{t=2}^{T} \log p(x_t|x_{t-1}) \tag{5.13}$$

where $()^+$ is the pseudo-inverse. Note that $\left\langle x_t x_t^T \right\rangle = diag \left\langle x_t \right\rangle$ so we only need compute the expectations $\langle x_t \rangle$ and $\left\langle x_t x_{t-1}^T \right\rangle$ using the current $\theta^{(k)}$ using the E-step, which is described next. The algorithm above that iterates between the E-step and M-step is also known as the **Baum-Welch algorithm**.

2. Inference of Forward/Backward Variables

Computation of t $\langle x_t \rangle$ and$\left\langle x_t x_{t-1}^T \right\rangle$ defined above can be done efficiently by defining the *forward* and *backward* variables defined as

$$\begin{aligned} \alpha_t &= p(x_t, y_{1:t}) \\ \beta_t &= p(y_{t+1:T}|x_t) \end{aligned}$$

Then the quantities $\gamma_t = \langle x_t \rangle$ and $\xi_t = \left\langle x_t x_{t-1}^T \right\rangle$ are computed using

$$\begin{aligned} \langle x_t \rangle_i &= \gamma_{t,i} = \frac{\alpha_{t,i}\beta_{t,i}}{\sum_{i=1}^{n} \alpha_{t,i}\beta_{t,i}} \\ \langle x_{t,i} x_{t-1,j} \rangle &= \xi_{t,i,j} = \frac{\alpha_{t-1,j} p(x_{t,i}|x_{t-1,j}) p(y_t|x_{t,i}) \beta_{t,i}}{\sum_{ij} \alpha_{t-1,j} p(x_{t,i}|x_{t-1,j}) p(y_t|x_{t,i}) \beta_{t,i}} = \frac{\alpha_{t-1,j} A_{ij} N(y_t; W_i, \Sigma) \beta_{t,i}}{\sum_{ij} \alpha_{t-1,j} A_{ij} N(y_t; W_i, \Sigma) \beta_{t,i}} \end{aligned}$$

The values α_t, β_t are computed using the **Forward-Backward Algorithm** which make use of the recursive relationships

$$\begin{aligned} \alpha_{t,i} &= P(y_t|x_{t,i}) \sum_{j=1}^{n} \alpha_{t-1,j} A_{ij} = N(y_t; W_i, \Sigma) \sum_{j=1}^{n} \alpha_{t-1,j} A_{ij} \\ \beta_{t,i} &= \sum_{j=1}^{n} P(y_{t+1}|x_{t+1,j}) \beta_{t+1,j} A_{ji} = \sum_{j=1}^{n} N(y_{t+1}; W_j, \Sigma) \beta_{t+1,j} A_{ji} \end{aligned}$$

The above formulation also allows to *estimate the likelihood of an output sequence* using $p(y_{1:T}) = \sum_{i=1}^{n} \alpha_{T,i}$.

3. Hidden State Estimation

The most likely state (MAP estimate) at time t given the observations $y_{1:T}$ given by $x_t^* = \arg\max_{x_t} p(x_t|y_{1:T})$ is solved as $x_t^* = e_{i_t^*}$ where

$$i_t^* = \arg\max_i \gamma_{t,i}$$

However, since this can give implausible transitions (it does not give the most likely *sequence*), the MAP estimate of the *sequence* given by

$$x^*_{1:T} = \arg\max_{x_{1:T}} p(x_{1:T}|y_{1:T}) \tag{5.14}$$

can be computed using the dynamic programming principle (DPP). If we define

$$\delta_t = \max_{x_1,...x_{t-1}} p(x_1, ..x_t, y_{1:t})$$

then the DPP tells us that

$$\delta_{t,i} = \left[\max_j (\delta_{t-1,j} A_{ij})\right] P(y_t|x_{t,i}) \tag{5.15}$$

so that the problem can then be solved by backtracking the maximizers above for the optimal state sequence. The resulting algorithm is sometimes also known as the **Viterbi Algorithm**.

Special Case when x_t is Observable During Training

If x_t are observable during training, then the Baum-Welch algorithm reduces to simple MLE for estimating π, A and the standard E-M iterations for Gaussian mixture model estimation for W, Σ. That is equations (5.9), (5.10), (5.11) and 5.12) become

$$\pi \longleftarrow x_1^T \tag{5.16}$$

$$A_{ij} \longleftarrow \frac{\sum_{t=2}^{T} x_{t,i} x_{t-1,j}}{\sum_{t=2}^{T} x_{t-1j}} \tag{5.17}$$

$$W \longleftarrow \left(\sum_{t=1}^{T} y_t x_t^T\right)\left(\sum_{t=1}^{T} x_t x_t^T\right)^+ \tag{5.18}$$

$$\Sigma \longleftarrow \frac{1}{T}\sum_{t=1}^{T} y_t y_t^T - \frac{1}{T}\sum_{t=1}^{T} W x_t y_t^T \tag{5.19}$$

5.5.2 Completely Hidden Cascade HMM (CHMM)

We will use the following tensor notation. A 3rd order tensor will be written as the multidimensional array (A_{ijk}) with components A_{ijk}. The model is defined in terms of the parameters: (1) Transition matrix C of z_t where $C_{ij} = \Pr(z_t = e_i|z_{t-1} = e_j)$, (2) Initial probability vector ω of z_1 i.e. $\omega_i = \Pr(\omega_1 = e_i)$ (3) the conditional transition matrices of x_t represented by the tensor (A_{ijk}) where $A_{ijk} = \Pr(x_t = e_i|x_{t-1} = e_j, z_t = e_k)$ (3) The intiial

probability matrix π of x_1 i.e. $\pi_{ik} = \Pr(x_1 = e_i | z_1 = e_k)$. In addition, we assume a Gaussian output model, where the effect of x_t and z_t on y_t is coupled (i.e. can not be decomposed), so we write this in fully coupled form as

$$\Pr(y_t|x_t, z_t) \quad = \quad (2\pi)^{-\frac{d}{2}} |\Sigma|^{-\frac{1}{2}} \exp\left\{ -\frac{1}{2} \left(y_t - W\left(z_t \otimes x_t\right)\right)^T \Sigma^{-1} \left(y_t - W\left(z_t \otimes x_t\right)\right) \right\}$$

where $W \in \mathbb{R}^{d \times nr}$ and $\Sigma \in \mathbb{R}^{d \times d} > 0$. Note one can think of W as a block matrix with concatenated matrices $[W_1\ W_2\ ...W_r]$ where each $W_k \in \mathbb{R}^{d \times n}$. We are interested in the following problems on CHMM:

1. Estimate parameters C, ω, A, π, W given observations $y_{1:T}$
2. Estimate the likelihood of a particular output sequence $y_{1:T}$ i.e. $p(y_{1:T})$
3. Estimate the most likely hidden states x_t, z_t, given output sequence $y_{1:T}$, i.e.

$$\arg\max_{(x_t, z_t)} p(x_t, z_t | y_{1:T}).$$

1. Parameter Estimation

Parameters can be estimated using E-M to derive an algorithm similar to Baum-Welch, where the complete data comprises of the hidden data $\{x_{1:T}, z_{1:T}\}$ and observed data $y_{1:T}$. Writing $\langle \cdot \rangle$ for the conditional expectation $\mathbb{E}\left(\cdot | y_{1:T}, \theta^{(k)}\right)$ given observations $y_{1:T}$ and existing model $\theta^{(k)}$ the M-step gives the update rules at k^{th} iteration

$$\begin{aligned}
\omega &\longleftarrow \left\langle z_1^T \right\rangle \\
\pi_{ij} &\longleftarrow \frac{\left\langle x_{1,i} z_{1,j} \right\rangle}{\left\langle z_1 \right\rangle_j} \\
C_{ij} &\longleftarrow \frac{\sum_{t=2}^{T} \left\langle z_{t,i} z_{t-1,j} \right\rangle}{\sum_{t=2}^{T} \left\langle z_{t-1} \right\rangle_j} \\
A_{ijk} &\longleftarrow \frac{\sum_{t=2}^{T} \left\langle x_{t,i} x_{t-1,j} z_{t,k} \right\rangle}{\sum_{t=2}^{T} \left\langle x_{t-1,j} z_{t,k} \right\rangle}
\end{aligned} \tag{5.20}$$

$$\begin{aligned}
W &\longleftarrow \left(\sum_{t=1}^{T} y_t \left\langle z_t \otimes x_t \right\rangle^T \right) \left(\sum_{t=1}^{T} \left\langle z_t z_t^T \otimes x_t x_t^T \right\rangle \right)^+ \\
\Sigma &\longleftarrow \frac{1}{T} \sum_{t=1}^{T} y_t y_t^T - \frac{1}{T} \sum_{t=1}^{T} W \left\langle z_t \otimes x_t \right\rangle y_t^T
\end{aligned} \tag{5.21}$$

where $()^+$ is the pseudo-inverse. The expectations in the above expressions are computed in the E-step as described next. Note that $\langle z_t \otimes x_t \rangle$ is the vectorized version of $\left\langle z_t x_t^T \right\rangle$ and $\left\langle z_t z_t^T \otimes x_t x_t^T \right\rangle$ is a diagonal matrix with $vec(\left\langle z_t x_t^T \right\rangle)$ as its diagonal. So the expectations needed at the E-step are $\left\langle z_t x_t^T \right\rangle, \left\langle z_t z_{t-1}^T \right\rangle, \left\langle z_t^T \right\rangle, \left\langle x_{t-1} z_t^T \right\rangle$ and the tensor $\langle x_{t,i} x_{t-1,j} z_{t,k} \rangle$.

2. Inference of Forward/Backward Variables and Likelihood Estimation

While the M-step above was straightforward and even extendible trivially to a cascade of multiple layers, computing the expectations in the E-step become untractable with the number of cascade layers. The compute these for the two-layer cascade in our formulation. We define the following forward and backward variables (note that these are now matrices and not vectors)

$$\begin{aligned} \alpha_t &= p(z_t, x_t, y_{1:t}) \\ \beta_t &= p(y_{t+1:T} | x_t, z_t) \end{aligned}$$

From which one can compute the tensors $\gamma_t = \left\langle x_t z_t^T \right\rangle$ and $\xi_t = (\langle x_{t,i} x_{t-1,j} z_{t,k} z_{t-1,l} \rangle)$ as

$$\begin{aligned} \left\langle x_{t,i} z_{t,k}^T \right\rangle &= \gamma_{t,ik} = \frac{\alpha_{t,ik} \beta_{t,ik}}{\sum_{i=1}^n \sum_{k=1}^r \alpha_{t,ik} \beta_{t,ik}} \\ \langle x_{t,i} x_{t-1,j} z_{t,k} z_{t-1,l} \rangle &= (\xi_t)_{ijkl} = \frac{\alpha_{t-1,jl} A_{ijk} C_{kl} P(y_t | x_{t,i}, z_{t,k}) \beta_{t,ik}}{\sum_{i,j=1}^n \sum_{k,l=1}^r \alpha_{t-1,jl} A_{ijk} C_{kl} P(y_t | x_{t,i}, z_{t,k}) \beta_{t,ik}} \end{aligned} \tag{5.22}$$

Then the expectations required in (5.20) and (5.21) can be expressed in terms of $\gamma_t, \xi_t, \sigma_t$ as

$$\begin{aligned} \left\langle z_t x_t^T \right\rangle &= \gamma_t \\ \left\langle x_{t-1} z_t^T \right\rangle &= \sum_{x_t, z_t} \xi_t \\ (\langle x_{t,i} x_{t-1,j} z_{t,k} \rangle) &= \sum_{z_{t-1}} \xi_t \\ \left\langle z_{t-1} z_t^T \right\rangle &= \sum_{x_t, x_{t-1}} \xi_t \\ \langle z_t \rangle &= \sum_{x_t} \gamma_t \\ \langle x_t \rangle &= \sum_{z_t} \gamma_t \end{aligned} \tag{5.23}$$

The variables α_t, β_t are computed using the recursion relations (**Forward-Backward Algorithm for CHMM)** in component form:

$$\begin{aligned} \alpha_{t,ik} &= P(y_t|x_{t,i}, z_{t,k}) \sum_{j=1}^{n} \sum_{l=1}^{r} A_{ijk} C_{kl} \alpha_{t-1,lj} \\ \beta_{t,ik} &= \sum_{j=1}^{n} \sum_{l=1}^{r} \beta_{t+1,jl} P(y_{t+1}|z_{t+1,l}, x_{t+1,j}) C_{lk} A_{jil} \end{aligned} \tag{5.24}$$

The forward-backward algorithms for the two-level cascade described above are of $O(Tn^2r^2)$and if the cascade has M levels, each with state space of size K then $O(TK^{2M})$ and thus it quickly becomes very inefficient for larger cascades. Careful book-keeping has been shown to reduce this complexity to $O(Tn^2r)$ ([98]) but that's still pretty ineffcient.

3. State Estimation

If we want to estimate both z_t, x_t then most likely state (MAP estimate) at time t given the observations $y_{1:T}$ is simply given by

$$(z_t^*, x_t^*) = \arg\max_{x_t, z_t} p(z_t, x_t|y_{1:T})$$

where $p(z_t, x_t|y_{1:T})$ was computed using γ_t as above. If we are only intereseted in estimate of the state z_t then the MAP estimate of each state at time t is given by

$$z_t^* = \arg\max_{z_t} p(z_t|y_{1:T})$$

where $p(z_{t,k}|y_{1:T})$ can be computed as $p(z_{t,k}|y_{1:T}) = \sum_i \gamma_{t,ik}$.However, just like the non-cascade case if we are interested, instead, in the most likely *sequence,* which is given by

$$(z_{1:T}^*, x_{1:T}^*) = \arg\max_{x_{1:T}, z_{1:T}} p(z_{1:T}, x_{1:T}|y_{1:T})$$

then a cascade analog of the **Viterbi Algorithm** based upon DPP can be used. The recursion formula is similar to that of α_t with summation replaced by maximization. That is, if we define

$$\delta_t = \max_{x_{1:t-1}, z_{1:t-1}} p(x_{1:t}, z_{1:t}, y_{1:t})$$

then from the DPP we have

$$\delta_{t,ik} = \left[\max_{j,l} (\delta_{t-1,jl} A_{ijk} C_{kl})\right] P(y_t|x_{t,i}, z_{t,k})$$

so that the problem can then be solved by backtracking the maximizers above for the optimal state sequence.

5.5.3 Partially Observable Cascade HMM (PO-CHMM)

As mentioned in the last section, the forward-backward algorithm can not be decoupled even if the underlying Markov model was fully decoupled, due to the coupling via the output. Approximations are used, such as variational methods and mean-field ([98, 117]). However, for our two-level cascade model these are not applicable. We instead, consider a slight variation of the original CHMM model which will be used in our application demonstrated in Part II of this chapter. In this case, during the inference phase, we are able to observe the z_t process. This can be thought of as a supervised learning model, where we are able to learn model parameters given partial observations in training sequences. The task is then to estimate state z_t on test sequences. The model parameters are still C, ω, A, π, W as defined in the CHMM case. The problems we consider are:

1. Estimate parameters C, ω, A, π, W given observations $y_{1:T}$ and state $z_{1:T}$

2. Estimate the likelihood of a particular output sequence $y_{1:T}$ i.e. $p(y_{1:T})$

3. Estimate the most likely state z_t given a particular output sequence, i.e.

$$\arg\max_{(z_t)} p(z_t|y_{1:T})$$

1. Parameter Estimation

We can still use E-M except that now the hidden variables are only $x_{1:T}$ and $y_{1:T}, z_{1:T}$ are observed variables. Denoting by $\langle\cdot\rangle$ the expectation $\mathbb{E}\left(\cdot|y_{1:T}, z_{1:T}, \theta^{(k)}\right)$ given $y_{1:T}, z_{1:T}$ and existing model $\theta^{(k)}$, the M-step thus becomes, using (5.20) and (5.21), using the fact

that $\langle z_t \rangle = z_t$ and $\langle z_t X \rangle = z_t \langle X \rangle$ for any random variable X.

$$\begin{aligned}
\omega &\longleftarrow z_1^T && (5.25)\\
\pi_{ij} &\longleftarrow \langle x_{1,i} \rangle \text{ for } j \text{ s.t. } z_1 = e_j \\
C_{ij} &\longleftarrow \frac{\sum_{t=2}^{T} z_{t,i} z_{t-1,j}}{\sum_{t=2}^{T} z_{t-1,j}} \\
A_{ijk} &\longleftarrow \frac{\sum_{t=2}^{T} \langle x_{t,i} x_{t-1,j} \rangle z_{t,k}}{\sum_{t=2}^{T} \langle x_{t-1,j} \rangle z_{t,k}} \\
W &\longleftarrow \left(\sum_{t=1}^{T} y_t \left(z_t \otimes \langle x_t \rangle \right)^T \right) \left(\sum_{t=1}^{T} \left(z_t z_t^T \otimes \left\langle x_t x_t^T \right\rangle \right) \right)^+ && (5.26)\\
\Sigma &\longleftarrow \frac{1}{T} \sum_{t=1}^{T} y_t y_t^T - \frac{1}{T} \sum_{t=1}^{T} W \left(z_t \otimes \langle x_t \rangle \right) y_t^T && (5.27)
\end{aligned}$$

The only expectations that need to be computed to evaluate the above are $\langle x_t \rangle$ and $\left\langle x_t x_{t-1}^T \right\rangle$. Note that as before $\left\langle x_t x_t^T \right\rangle = diag(\langle x_t \rangle)$.

2. Inference of Forward/Backward Variables

The values of the forward and backward variables α_t, β_t and other variables γ_t, ξ_t used to compute the expecations $\langle x_t \rangle$ and $\left\langle x_t x_{t-1}^T \right\rangle$ above are defined as:

$$\begin{aligned}
\alpha_t &= p(x_t, y_{1:t} | z_{1:t}) \\
\beta_t &= p(y_{t+1:T} | x_t, z_{t+1:T})
\end{aligned}$$

The variables α_t, β_t are computed using the recursion relations (**Forward-Backward Algorithm for PO-CHMM)** in component form:

$$\begin{aligned}
\alpha_{t,i} &= \sum_{k=1}^{r} z_{t,k} \left(p(y_t | x_{t,i}, z_{t,k}) \sum_{j=1}^{n} A_{ijk} \alpha_{t-1,j} \right) && (5.28)\\
\beta_{t,i} &= \sum_{k=1}^{r} z_{t+1,k} \left(\sum_{j=1}^{n} \beta_{t+1,j} p(y_{t+1} | z_{t+1,k}, x_{t+1,j}) A_{jik} \right)
\end{aligned}$$

The outer summation over index k in the above formulae with a premulitplication by $z_{t,k}$ or $z_{t+1,k}$ implies that the only k for which the summand is non-zero is the k such that $z_t = e_k$.

To avoid confusion, **we will introduce a new notation**, where we write $\sum_{k=1}^{r} z_{t,k} A_{ijk}$ as $A_{ij}(z_t)$ and $\sum_{k=1}^{r} z_{t,k} p(y_t|z_{t,k}, x_{t,i}) = p(y_t|z_t, x_{t,i})$ which simply means that the transition matrices A_{ij} and emission probabilities $p(y_t|x_t)$ depend on z_t. Using this simplified notation, we have the forward-backward recursion formulae

$$\begin{aligned} \alpha_{t,i} &= p(y_t|x_{t,i}, z_t) \sum_{j=1}^{n} A_{ij}(z_t) \alpha_{t-1,j} \\ \beta_{t,i} &= \sum_{j=1}^{n} \beta_{t+1,j} p(y_{t+1}|z_{t+1}, x_{t+1,j}) A_{ji}(z_{t+1}) \end{aligned} \tag{5.29}$$

Then $\gamma_t = \langle x_t \rangle$ and $\xi_t = \left\langle x_t x_{t-1}^T \right\rangle$ are computed as

$$\begin{aligned} \gamma_{t,i} &= \langle x_{t,i} \rangle = \frac{\alpha_{t,i} \beta_{t,i}}{\sum_{i=1}^{n} \alpha_{t,i} \beta_{t,i}} \\ \xi_{t,ij} &= \langle x_{t,i} x_{t-1,j} \rangle = \frac{\alpha_{t-1,j} A_{ij}(z_t) P(y_t|x_{t,i}, z_t) \beta_{t,i}}{\sum_{i=1}^{n} \alpha_{t-1,j} A_{ij}(z_t) P(y_t|x_{t,i}, z_t) \beta_{t,i}} \end{aligned} \tag{5.30}$$

which is all that is required for the expectations $\langle x_t \rangle, \left\langle x_t x_{t-1}^T \right\rangle$ in (5.25). Note that the complexity of the forward backward algorithm for this PO-CHMM is the same as that of a non-cascade HMM ie. $O(Tn^2)$.

3. State Estimation

In this problem, once the parameters have been estimated using test sequences as above, one is interested in estimating the best z_t given observations $y_{1:T}$. That is,

$$z_t^* = \arg\max_{z_t} p(z_t|y_{1:T}) \tag{5.31}$$

To do so, we can't use the γ_t evaluated from (5.30) since those were estimated assuming $z_{1:T}$ were known. Hence we use the forward-backward equations for the fully hidden CHMM, i.e. equations 5.24) and (5.22) *using the parameters estimated from the PO-CHMM*, and then use $p(z_{t,k}|y_{1:T}) = \sum_i \gamma_{t,ik}$ to evaluate the MAP state (5.31).

5.5.4 Special Case of PO-CHMM With Quasi-Stationary z_t

The equations for estimation and inference as derived above for a PO-CHMM are still not decoupled: while each individual iteration in the M step, equation (5.25) and each pass of each iteration of the E-step, equation (5.29), computes the parameters and expectations for a single value of z_t (i.e. a single k value in those equations), these values are

dependent on previous iterations which may use different values of k. Hence across iterations, the computations are not completely decoupled. We found it useful to make the following approximation, especially in our application described in Part II. If we assume that the z_t is quasi-stationary, in that the value of z_t does not change from $t = 1$ to $t = T$. This is equivalent to having its transition matrix C as the identity, or representing z_t by a single random variable $Z \in \{e_k\}_{k=1}^r$ rather than a stochastic process (Fig 5.3). The model parameters are of the parameters: (1) the conditional transition matrices of x_t represented by the tensor (A_{ijk}) where $A_{ijk} = \Pr(x_t = e_i | x_{t-1} = e_j, Z = e_k)$ (3) The initial probability matrix π of x_1 i.e. $\pi_{ik} = \Pr(x_1 = e_i | Z = e_k)$. In addition, we assume a Gaussian output model, where the effect of x_t and Z on y_t is coupled:

$$\Pr(y_t|x_t, Z) = (2\pi)^{-\frac{d}{2}} |\Sigma|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\left(y_t - W\left(Z \otimes x_t\right)\right)^T \Sigma^{-1} \left(y_t - W\left(Z \otimes x_t\right)\right)\right\}$$

where $W \in \mathbb{R}^{d \times nr}$ and $\Sigma \in \mathbb{R}^{d \times d} > 0$. In addition, we will also assume that there are *multiple training sequences* available, each of length T. We are interested in the following problems :

1. Estimate parameters A, π, W given P sequences of independent *training* observations $y_{1:T}^{1:P}$, and $Z^{1:P}$ (where notation y_t^p refers to the value of y at time t for sequence p, and the notation $y^{1:P}$ is the set $\{y^1, y^2, ...y^P\}$ and $y_{1:T}$ the set $\{y_1, y_2, ...y_T\}$.

2. Estimate the likelihood of a particular output sequence $y_{1:T}$ i.e. $p(y_{1:T})$

3. Estimate the most likely state Z given a particular output *testing* sequence, i.e.

$$\arg\max_Z p(Z|y_{1:T}).$$

1. Parameter Estimation

This is a special case of a PO-HMM with $z_t = Z$ but with the addition of sequences $1...P$. For the E-step, we take expectations $\mathbb{E}\left(\cdot|y_{1:T}^{1:P}, Z^{1:P}, \theta^{(k)}\right)$ given $y_{1:T}^{1:P}, Z^{1:P}$ and existing model $\theta^{(k)}$ denoted by the operator $\langle\cdot\rangle$. Similar to the PO-HMM case, M-step update rules

are

$$\omega_k \longleftarrow \frac{1}{P}\sum_{p=1}^{P} Z_j^p \tag{5.32}$$

$$\pi_{ik} \longleftarrow \frac{\sum_{p=1}^{P}\left\langle x_{1,i}^p\right\rangle Z_k^p}{\sum_{p=1}^{P} Z_k^p} \tag{5.33}$$

$$A_{ijk} \longleftarrow \frac{\sum_{p=1}^{P}\sum_{t=2}^{T}\left\langle x_{t,i}^p x_{t-1,j}^p\right\rangle Z_k^p}{\sum_{p=1}^{P}\sum_{t=2}^{T}\left\langle x_{t-1,j}^p\right\rangle Z_k^p}$$

$$W \longleftarrow \left(\sum_{p=1}^{P}\sum_{t=1}^{T} y_t^p \left(Z^p \otimes \langle x_t^p\rangle\right)^T\right)\left(\sum_{p=1}^{P}\sum_{t=1}^{T}\left(Z^p Z^{pT} \otimes \left\langle x_t^p x_t^{pT}\right\rangle\right)\right)^+ \tag{5.34}$$

$$\Sigma \longleftarrow \frac{1}{PT}\sum_{p=1}^{P}\sum_{t=1}^{T} y_t^p y_t^{pT} - \frac{1}{PT}\sum_{p=1}^{P}\sum_{t=1}^{T} W\left(Z^P \otimes \langle x_t^p\rangle\right) y_t^{pT} \tag{5.35}$$

To perform the above iteration, we only need the values of $\langle x_t^p\rangle$ and $\left\langle x_t^p x_{t-1}^{pT}\right\rangle$ for sequence p from the E-step.

2. Inference of Forward/Backward Variables

The values of the forward and backward variables α_t, β_t and other variables γ_t, ξ_t used to compute the expecations $\langle x_t^p\rangle$ and $\left\langle x_t^p x_{t-1}^{pT}\right\rangle$ for each sequence p above are defined as:

$$\begin{aligned}\alpha_t^p &= p(x_t^p, y_{1:t}^p | Z^p)\\ \beta_t^p &= p(y_{t+1:T}^p | x_t^p, Z^p)\end{aligned}$$

The variables α_t^p, β_t^p are computed using the Forward-Backward Algorithm for PO-CHMM except this is done on a per sequence basis:

$$\alpha_{t,i}^p = p(y_t^p | x_{t,i}, Z^p)\sum_{j=1}^{n} A_{ij}(Z^p)\alpha_{t-1,j}^p \tag{5.36}$$

$$\beta_{t,i}^p = \sum_{j=1}^{n} \beta_{t+1,j}^p p(y_{t+1}^p | Z^p, x_{t+1,j}) A_{ji}(Z^p)$$

Then the variables $\gamma_t^p = \langle x_t^p \rangle$ and $\xi_t^p = \left\langle x_t^p x_{t-1}^{pT} \right\rangle$ are computed as

$$\begin{aligned} \gamma_{t,i}^p &= \left\langle x_{t,i}^p \right\rangle = \frac{\alpha_{t,i}^p \beta_{t,i}^p}{\sum_{i=1}^n \alpha_{t,i}^p \beta_{t,i}^p} \\ \xi_{t,ij}^p &= \left\langle x_{t,i}^p x_{t-1,j}^p \right\rangle = \frac{\alpha_{t-1,j}^p A_{ij}(Z^p) P(y_t^p | x_{t,i}, Z^p) \beta_{t,i}^p}{\sum_{i=1}^n \alpha_{t-1,j}^p A_{ij}(Z^p) P(y_t^p | x_{t,i}, Z^p) \beta_{t,i}^p} \end{aligned} \tag{5.37}$$

3. Decoupled Estimation & Inference Algorithm

The E-M iteration steps as written above still do not decoupled. However, if we make a slight change in the output model, i.e. assume Σ also depends on Z, which we explicitly write as follows:

$$\Pr(y_t | x_t, Z) = (2\pi)^{-\frac{d}{2}} |\Sigma Z|^{-\frac{1}{2}} \exp\left\{ -\frac{1}{2} (y_t - W(Z \otimes x_t))^T \Sigma^{-1} (y_t - W(Z \otimes x_t)) \right\}$$

where now Σ is a block matrix of the form $[\Sigma_1\ \Sigma_2\ ...\ \Sigma_r]$ with each Σ_i a $d \times d$ covariance matrix corresponding to $Z = e_r$. Since each sequence is independent of each other, we can collect sequences with the same value of Z and decouple each update step in (5.32) to obtain an efficient fully **Decoupled PO-CHMM Estimation & Inference Algorithm** to estimate the parameters $\theta = (\omega, \pi, A, W, \Sigma)$ given training sequences $y_{1:T}^{1:P}, Z^{1:P}$ which can be summarized as: For each $k = 1...r$

1. Collect all sequences p with $Z^p = e_k$. Re-index all sequences so that sequences $1...P_k$ are sequences from this subset.

2. Set $\omega_k = \frac{P_k}{P}$

3. Iterate till convergence: (here we have written W_k, Σ_k as the k^{th} block of W, Σ respectively. Recall that each of W, Σ are concatenated matrices of the form $[W_1\ W_2\ ...\ W_r]$. Similarly, W_{ik} is the i^{th} column of the k^{th} block of W.)

(a) **M-step:**

$$\pi_{ik} \longleftarrow \frac{1}{P}\sum_{p=1}^{P_k}\left\langle x_{1,i}^p \right\rangle \tag{5.38}$$

$$A_{ijk} \longleftarrow \frac{\sum_{p=1}^{P_k}\sum_{t=2}^{T}\left\langle x_{t,i}^p x_{t-1,j}^p \right\rangle}{\sum_{p=1}^{P_k}\sum_{t=2}^{T}\left\langle x_{t-1,j}^p \right\rangle}$$

$$W_k \longleftarrow \left(\sum_{p=1}^{P_k}\sum_{t=1}^{T} y_t^p \left\langle x_t^p \right\rangle^T\right)\left(\sum_{p=1}^{P_k}\sum_{t=1}^{T}\left\langle x_t^p x_t^{pT}\right\rangle\right)^+ \tag{5.39}$$

$$\Sigma_k \longleftarrow \frac{1}{P_k T}\sum_{p=1}^{P_k}\sum_{t=1}^{T} y_t^p y_t^{pT} - \frac{1}{P_k T}\sum_{p=1}^{P_k}\sum_{t=1}^{T} W_k y_t^{pT} \tag{5.40}$$

(b) **E-step:**

$$\left\langle x_{t,i}^p \right\rangle = \frac{\alpha_{t,i}^p \beta_{t,i}^p}{\sum_{i=1}^n \alpha_{t,i}^p \beta_{t,i}^p} \tag{5.41}$$

$$\left\langle x_{t,i}^p x_{t-1,j}^p \right\rangle = \frac{\alpha_{t-1,j}^p A_{ijk} N(y_t^p; W_{ik}, \Sigma_k)\beta_{t,i}^p}{\sum_{i=1}^n \alpha_{t-1,j}^p A_{ijk} N(y_t^p; W_{ik}, \Sigma_k)\beta_{t,i}^p}$$

where α, β defined as

$$\alpha_t^p = p(x_t^p, y_{1:t}^p | Z = e_k)$$
$$\beta_t^p = p(y_{t+1:T}^p | x_t^p, Z = e_k)$$

are evaluated using the forward/backward recursion equations

$$\alpha_{t,i}^p = N(y_t^p; W_{ik}, \Sigma_k)\sum_{j=1}^n A_{ijk}\alpha_{t-1,j}^p \tag{5.42}$$

$$\beta_{t,i}^p = \sum_{j=1}^n \beta_{t+1,j}^p N(y_{t+1}^p; W_{jk}, \Sigma_k) A_{jik}$$

The above algorithm computes parameters separately for each k and hence fully decouples x and z.

4. State Estimation

To solve the problem of estimation of state Z given a testing sequence $y_{1:T}$ of observations, we need to find $\arg\max_Z p(Z|y_{1:T})$. The algorithm, using the decoupled equations above, is as follows.

1. For each $k = 1...r$

   (a) With $p = 1$, using the estimated values of (π, A, W, Σ) compute α_T^1 using the forward-backward recursions (5.42) with $y_{1:T}^1 = y_{1:T}$.

   (b) Compute the likelihood

$$L_k = l(y_{1:T}|Z = e_k) = \sum_{i=1}^{n} \alpha_{T,i}^1 \tag{5.43}$$

2. Compute the MAP estimate of Z as $Z^* = e_{k^*}$ where

$$k^* = \arg\max_k L_k \omega_k \tag{5.44}$$

5.6 Algorithms for Optimal Control On CMDPs

In this section, we solve an *expected utility maximization problem,* i.e. where the performance measure is the expectation of a functional, for a fully observable CMDP on finite time-horizon case is solved using the framework of Section 5.3.2.

Problem Definition

Fix a finite time horizon T on the cascade MDP (z_t, x_t) defined in Section 5.3.2. and define the cost function η,

$$\eta(u) = \mathbb{E} \int_0^T (z^T(\sigma)\mathbf{L}^T(\sigma)x(\sigma) + \psi(u(\sigma))d\sigma + z^T(T)\Phi^T(T)x(T) \tag{5.45}$$

where c, ϕ are real-valued functions on the space $\mathbb{R}^+ \times \{e_i\}_{i=1}^r \times \{e_i\}_{i=1}^n$, that are represented by the real matrices $\mathbf{L}(t)$ and $\mathbf{\Phi}(t)$ as $c(t, z, x) = z^T\mathbf{L}(t)x$ and $\phi(t, z, x) = z^T\mathbf{\Phi}(t)x$; and ψ a (Borel) measurable function $\mathbb{R}^p \to \mathbb{R}$. If c is bounded the problem of finding the solution to

$$\eta^* = \min_{u \in \mathcal{U}} \eta(u) \tag{5.46}$$

is well-defined and will be subsequently referred to as Problem (**OCP-I**). The corresponding optimal control is given by

$$u^* = \arg\min_{u \in \mathcal{U}} \eta(u) \tag{5.47}$$

Solution Using Dynamic Programming Principle

Theorem 1 *Let (z_t, x_t) be a cascade MDP as defined in Section 5.3.2 with C, A_0, A and B_i as defined thereof. Let $T > 0$, and $\mathcal{U}, \psi$,$\Phi$and η be as defined in section 5.6. Then there exists a unique solution to the equation (on the space of $n \times r$ matrices)*

$$\begin{aligned} \dot{K} &= -KC - L - A_0^T K - A^T(z)K - \min_{u(z,x)\in\mathcal{U}}(\sum_{i=1}^{p} u_i z^T K^T B_i(z)x + \psi(u)) \quad (5.48) \\ K(T) &= \Phi(T) \quad (5.49) \end{aligned}$$

*on the interval $[0, T]$, where $A^T(z)K$ denotes the matrix whose $j'th$ column is $A(e_j)K^T e_j^T$ (which can be more explicitly written as $\sum_z A^T(z)Kzz^T$, that is, the matrix representation of the functional $x^T A^T(z)Kz$). Furthermore, if $K(t)$ is the solution to 5.48 then the optimal control problem **OCP-I** defined in (5.46) has the solution*

$$\begin{aligned} \eta^* &= \mathbb{E}z^T(0)K^T(0)x(0) \quad (5.50) \\ u^* &= \arg\min_{u(z,x)\in\mathcal{U}}(\sum_{i=1}^{p} z^T K^T u_i B_i(z)x + \psi(u_i)) \quad (5.51) \end{aligned}$$

Proof. With z,x, η as defined above let the minimum return function be $k(t, z, x) = z^T K^T(t)x$,where $K(t)$ is an $n\times r$ matrix, so that $k(0, z(0), x(0) = \eta^*$. Using Ito rule for $z^T K^T x$

$$d(z^T K^T x) = \sum_{i=1}^{s} z^T H_i^T K^T x dM_i + z^T \dot{K}^T x + \sum_{i=1}^{n} z^T K^T G_i x dN_i$$

Since the process $dN_i - (\lambda_{i0}^0 + \lambda_{i0}(z) + \sum_{j=1}^{p} \mu_{ij}(z)u_j)dt$ is a martingale equating the expectation to zero gives

$$\begin{aligned} \mathbb{E}(\sum_{i=1}^{n} z^T K^T G_i x dN_i) &= \mathbb{E}(g(t, x, z, u)dt) \\ \mathbb{E}(\sum_{i=1}^{s} z^T H_i^T K^T x dM_i) &= \mathbb{E}(z^T C^T K^T x) \end{aligned}$$

with $g(t, x, z, u) = z^T K^T A_0 x + z^T K^T A(z) + \sum_{i=1}^{p} z^T K^T u_i B_i(z)x$. Writing $c(t, z, x) + \psi(u) = f(t, z, x, u)$ and $z^T C^T K^T x + g(t, x, z, u) = \xi(t, x, z, u)$, a simple application of the stochastic dynamic programming principle shows that

$$z(t)^T \dot{K}(t)^T x(t) + \min_u(\xi(t, x, z, u) + f(t, z, x, u)) \geq 0$$

The minimum value of 0 is actually achieved by u^*so that the inequality above must be an equality. Introducing notation $A^T(z)K$, we get precisely (5.48). Proof of uniqueness is identical to that in [31] Theorem 1. ■

Note that the Bellman equation (5.48) is very similar to that of a single (non cascade) MDP with the additional term $-KC$ representing the backward (adjoint) equation for the process $z(t)$ and the appearance of z in the term for minimization which permits feedback of the optimal control u^* on z in addition to x. The matrix K above is also known as the **Minimum Return Function.** The above solution is a single point boundary value problem instead of two-point. For small KC, the above decouples one column at a time. This form is readily generalizable to multifactor MDPs as well.

Corollary 2 *(*Quadratic Cost of Control*) Under the hypothesis of the above theorem, if* $\psi(u_i) = u_i^2$ *then if* $u_i(t,z,x) = \frac{-1}{2} z^T(t) K^T(t) B_i(z) x(t)$ *lies in the interior of* $\mathcal{U}$ *then it is the optimal control. Otherwise the optimal control is on the boundary of* $\mathcal{U}$*. If the former is the case at every* $t \in [0,T]$*, then equation (5.48) defining the optimal solution becomes (where the notation* $M^{\cdot 2}$ *for a matrix is element-wise squared matrix):*

$$\dot{K} = -KC - L - A_0^T K - A^T(z)K + \frac{1}{4}\sum_{i=1}^{p} (B_i^T(z)K)^{\cdot 2}$$

Corollary 3 *(*No Cost of Control*) Under the hypothesis of the above theorem, if* $\psi(u_i) = 0$ *then the optimal control is at the boundary of* $\mathcal{U}$*. If* $\mathcal{U}$ *is defined as the set* $\{ -a_i \le |u_i| \le a_i\}$ *the optimal control is the bang-bang control* $u_i(t,z,x) = -a_i \operatorname{sgn}(z^T K^T(t) B_i(z) x)$ *and equation (5.48) defining the optimal solution becomes*

$$\dot{K} = -KC - L - A_0^T K - A^T(z)K + \sum_{i=1}^{p} a_i \left| B_i^T(z)K \right| ;$$

Note that the stochastic control problem **OCP-I** can be formulated as a deterministic optimization problem (and hence also an open-loop optimization problem) using probability densities permitting the application of variational techniques. Derivation of the solution in Theorem 1 using the maximum principle is available in the Online Supplemental Material.

Summary of Notations and Symbols

A stochastic basis $(\Omega, \mathcal{F}, \mathbb{F}, \mathbb{P})$ is assumed where $(\Omega, \mathcal{F}, \mathbb{P})$ is a probability space and $\mathbb{F}$ a filtration $(\mathcal{F}_t)_{t\in T}$ on this space for a totally ordered index set T ($\subseteq \mathbb{R}^+$in our case). All

stochastic processes are assumed to be right continuous and adapted to $\mathbb{F}$.

$\mathbb{F}$	A filtration $(\mathcal{F}_t)_{t\in T}$ on $(\Omega, \mathcal{F}, \mathbb{P})$ where T is a totally ordered index set
$\mathbb{G}^n$	The space of square matrices of dimension n of the form $F_{kl} - F_{ll}$ where F_{ij} is the matrix of all zeros except for one in the $i'th$ row and $j'th$ column
$\mathbb{E}^n$	The space of diagonal $n \times n$ matrices with only 1's or 0's
I_n	$n \times n$ identity matrix, $I_n \in \mathbb{E}^n$
$\mathbb{P}^n$	The space of all stochastic matrices of dimension n
$\{e_i\}_{i=1}^n$	The set of n standard basis vectors in $\mathbb{R}^n$
$\boldsymbol{\phi}(t)$	A real-valued function ϕ on $\mathbb{R}^+ \times \{e_i\}_{i=1}^n$ will be written as the vector $\boldsymbol{\phi}(t) \in \mathbb{R}^n$ as $\phi(t,x) = \boldsymbol{\phi}^T(t)x$ where $x \in \{e_i\}_{i=1}^n$
$\boldsymbol{\Phi}(t)$	A real-valued function ϕ on $\mathbb{R}^+ \times \{e_i\}_{i=1}^r \times \{e_i\}_{i=1}^n$ is written as the $r \times n$ real matrix $\boldsymbol{\Phi}(t)$ as $\phi(t,z,x) = z^T\Phi(t)x$ where $z \in \{e_i\}_{i=1}^r$ and $x \in \{e_i\}_{i=1}^n$
$A^T(z)K$	Denotes the matrix whose $j'th$ column is $A(e_j)K^Te_j^T$ which can be more explicitly written as $\sum_z A^T(z)Kzz^T$
$\lvert M\rvert$	For a matrix M represents the element-by-element absolute value of a matrix
$M^{\cdot 2}$	For a matrix M represents the element-by-element squared
$\mathbf{e}_r$	The $r-$vector $[1\ 1...1]^T$

Chapter 6

Estimation & Control On Decomposable Markov Chains: Application to Gait & Fall Detection

Abstract

Real-time gait analysis using minimally invasive technology is a valuable addition to a tele-health platform. Our prototype Smart Slipper uses inexpensive and generically designed ready-to-wear shoe insoles and continuously streams pressure and accelerometer data via a lightweight RF/Zigbee protocol to a real-time analytic engine. Data-driven algorithms that exploit the Karhunen-Loève (KL) transform, Hidden Markov Models (HMM) and Bayesian analysis are able to detect arbitrary activity modes using relevant feature sets from gait data. We demonstrated accuracy up to 99% for real-time detection of sitting, walking, jumping and running. The online algorithm can also perform clinically important event detection such as a falling with up to 97% accuracy.

6.1 Introduction

Among elderly adults over the age of 65, fall-related injuries are the leading cause of emergency room visits (10% of all visits in 2006 [173]) and the primary cause of accidental

deaths in persons over the age of 65 (56% of all unintentional injury deaths in 2013 [86]). According to the Center for Disease Control and Prevention (CDC), an older adult dies from a fall in the United States every 20 minutes, resulting in about $34 billion in medical costs annually [86]. The World Health Organization [172] estimates that 28-35% of people aged 65 or over fall each year, this number increasing substantially with age. 51% of these falls result in severe fracture, only 50% of these are able to resume an ambulatory lifestyle, and 25% of them die within one year [53]. Some of the adverse consequences of a fall can be mitigated by automatic fall detection systems which can provision rapid assistance [212, 191, 155] and lower fear of fall which in turn improves safety [91]. Wireless streaming body sensor networks promise to revolutionize health care by allowing inexpensive remote health monitoring in realtime [162]; with detection of gait anomalies they have the potential to considerably improve the quality of the life of elderly living at home or in assisted living facilities and drastically reduce medical costs.

Most current fall detection systems [111] require the use of special devices that are not part of usual dailywear, such as cameras providing video signals, magnetic motion capture systems providing joint angle data [166], body wearable sensor networks [182] or ear-worn sensors [152]. Some systems use customized shoes [17] or those requiring require gyroscopes [44], magnetometers [26], several body-worn biaxial accelerometers [156] or goniometric joint measurements [45] - all of which are too cumbersome to be worn continuously. We have developed and tested a prototype "SmartSlipper", a generic, ready-to-wear, "one size fits all", lightweight and completely unobstrusive footwear system that permits continuous streaming of plantar pressure and accelerometer data for realtime fall and gait detection. By quantifying gait metrics in addition to detecting falling events, this device can be used to monitor disease progression, dementia onset [167, 227], exercise decline and routine abnormality in the elderly as well as for those in sports or other physical rehabilitation and gait anomaly in soldiers from "march fractures" [233]. Gait can even be used for identification of individuals [207].

We implement two *probabilistic* methods: (i) a *frame-based* method that uses inference from a Gaussian mixture model (GMM), and a (ii) *sequential* method that uses inference using coupled Hidden Markov Models (HMM). Both methods use feature reduction via Karhunen-Loève (KL) transform. Our methods achieve accuracy exceeding or comparable with state-of-the-art activity and posture detection systems reported in [156] with small initial training data. Existing sequential ([26], [210, 156]), linear ([17]) and non-linear [126, 29] methods require use of high dimensional feature sets or introduce significant detection latency

making their use limited where falls must be detected in real-time. Some *geometric* classification techniques([44, 207]) (e.g. SVM, k-NN, ANN) require careful construction of feature boundaries in training phase; and *thresholding* or *template-matching* techniques require careful setting of thresholds and sensor placement. Our algorithms, on the other hand, probe for distinguishing features rather than momentary time-frequency characteristics and are able to detect activity mode accurately with low computational power, permitting fully online and real-time implementation without requiring subject specific training data sets. Furthermore, being completely data-driven, our algorithms are easily extendable to incorporate additional sensor data which would allow distinguishing more complex gait patterns. Flexible segmentation and outlier policies allow for customization of our algorithms to specified latency and robustness requriements.

Our first GMM based algorithm was based upon a similar methodology that was previously applied to successful detection of sleep spindles [13]. However, while the algorithm showed up to 98% accuracy in detection of single activity modes such as walking, jumping, etc. it showed up to 30% false positive rate (FPR) and 8% false negative rate (FNR) when tested for falling. Our second HMM based algorithm, dramatically improved upon this accuracy, with false 8% FPR and 3% FNR though involves estimation of a larger number of parameters and hence requires a larger training data set size than the first. We demonstrated real-time implementations of both algorithms on live data.

6.2 Hardware: Sensor/Gateway Design

A generic and cost-effective shoe called the SmartSlipper^{TM} contains an insole with embedded four sensors measuring presure at the heel, ball, inside arch and outside arch of the foot respectively (Fig 6.1) The analog signal from these sensors is wired to a low-powered unit placed on the inside strap of the shoe that includes a micro-controller procesor unit, a micro electro mechanical system (MEMS) based 3-axis accelerometer and a wireless transducer. The output from the pressure sensors and the accelerometer are sampled with an ADC to feed data into a 2.4 GHz RF/ZigBee wireless transceiver. The wireless device sends data reports, each comprising of four pressure sensor readings and three accelerometer readings (one in each direction), at the rate of about 18 reports per second via the lightweight IEEE 802.15.4 ZigBee protocol. The data are received by a gateway (CC2351, Texas Instruments) that is able to transmit these packets to the host computer via a serial COM interface. Each frame comprises of 6 reports (i.e. of data batched over a period of 330 ms) along with

an identifier encoding sensor location. Heartbeat information is also transmitted periodically (more details on the protocol are on the Online Supplemental Material. The collector software on the host computer decodes the reports and annotates approximate timestamps to convert the data back into time series data. All sensor signals are then normalized: pressure sensor data are normalized to a scale of 0-1 and acceleration data are normalized to a scale of -0.5 to 0.5.

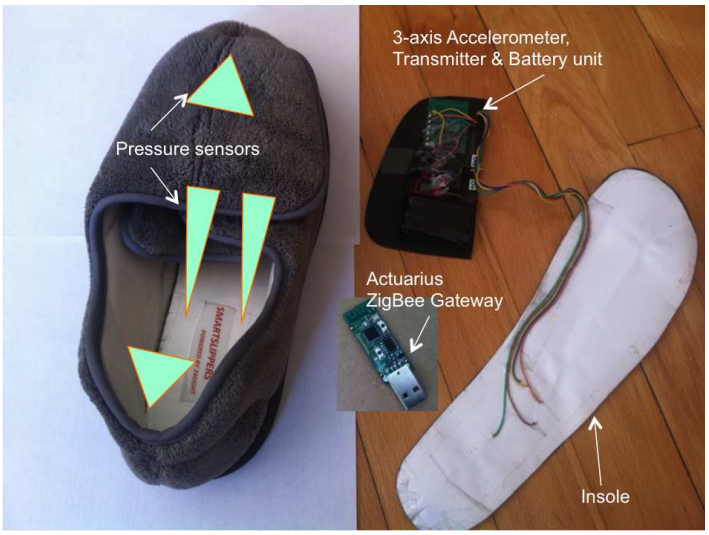


Figure 6.1: SmartSlipper^{TM}: Shoes, insoles and associated hardware. *(Left)* The shoe and location of four pressure sensors on the shoe. *(Right)* The insole containing the pressure sensors and a powered unit that goes in the inside strap of the shoe. The powered unit contains the ZigBee wireless transmitter as well as the accelerometion sensors. *(Center)* USB dongle that acts as the gateway for ZigBee data on the receiving end.

6.3 Theoretical Framework

6.3.1 Spectral Decomposition of the Motion Process

Different gait activities including falling can be considered to be generated by a random *motion process* suppported on the time interval $[0, T]$. Denoting by $\mu(t) := \mathcal{E}\{y(t)\}$ the *mean* process, and by $\mathcal{K}(s,t) := \mathcal{E}\{(y(t) - \mu(t))(y(s) - \mu(s))\}$ the *covariance kernel* for $t, s \in [0, T])$, Mercer's theorem asserts that the eigenfunctions of the Fredholm operator T_K with kernel $\mathcal{K}$ and eigenvalues λ_k, i.e. solutions to

$$\int_0^T \mathcal{K}(s,t)\psi_k(t)ds = \lambda_k\psi_k(t), \quad k = 1, 2, \cdots \tag{6.1}$$

form a complete orthonormal basis of $L^2[0,T]$. Then according to the *Kahrunen-Loeve (K-L)* theorem, $y(t)$ can be respresented in terms of these basis functions $\{\psi_k(t)\}_{k=1}^{\infty}$, i.e.

$$y(t) = \lim_{N\to\infty} \sum_{k=1}^{N} z_k \psi_k(t), 0 \le t \le T \tag{6.2}$$

where convergence is in L^2 and is uniform [226], and the coefficients $\{z_k\}_{k=1}^{\infty}$ given by

$$z_k = \int_0^T y(t)\psi_k(t)dt, \quad k = 1, 2, \cdots \tag{6.3}$$

are *mutually uncorrelated* with variance λ_k. That is,

$$\mathcal{E}\{(z_i - m_i)(z_j - m_j)\} = \lambda_i \delta_{ij}, \quad i, j = 1, 2, \cdots \tag{6.4}$$

where $m_k = \mathcal{E}\{z_k\}$ and δ_{ij} is the Kronecker delta function. The expansion (6.2) of a random process $y(t)$ in terms of basis functions $\psi_k(t)$ satisfying (6.4) is also known as the *Kahrunen-Loeve* (K-L) expansion and the corresponding transform (6.3) is known as the *K-L* or *Hotelling* transform.

6.3.2 Discrete-time Adapation: Empirical Motion Transform

Let $N := [f_s T]$ denote the number of samples of a signal digitally recorded at sampling frequency f_s during a fixed duration T. The discrete-time analogue to (6.1) is given by:

$$K\boldsymbol{\psi}_k = \lambda_k \boldsymbol{\psi}_k, \qquad k = 1, 2, \ldots, N \tag{6.5}$$

where $K \in \mathbb{R}^{N\times N}$ is the covariance matrix with elements $K_{i,j} := \mathcal{K}(\frac{i}{f_s}, \frac{j}{f_s})$ for $i, j = 1, 2, \ldots, N$ and $\boldsymbol{\psi}_k \in \mathbb{R}^N$ for $i = 1, 2, \ldots, N$ are the eigenvectors of K.To adapt this to gait sensory data, suppose we are monitoring a window of incoming sensory data of length W from a total of L sensory streaming observations $\{s_1[n], \cdots, s_L[n]\}_{n=1}^{W}$. Then for any $n \in [1, W]$ we can form an observation vector $\mathbf{y}$ of length $N = LW$ defined as

$$\mathbf{y} = (s_1[n-W], s_1[n-W+1], \cdots, s_1[n], \cdots, s_L[n-W], s_L[n-W+1], \cdots, s_L[n])^T \tag{6.6}$$

This observation encoding format is used to simultaneously analyze the data of different sensors while maintaining the correlation information of different sensors. We can estimate the statistical characteristics of the motion process using a reasonably large sample pool (say, of size M) of observation vectors (corresponding to different times and experiments)

$\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \cdots, \mathbf{y}^{(M)}$. The *empirical mean vector* $\widehat{\boldsymbol{\mu}}$ (of dimension N) and *empirical covariance matrix* $\widehat{K}$ (of dimension $N \times N$) are the corresponding unbiased estimates

$$\begin{aligned} \widehat{\boldsymbol{\mu}} &= \frac{1}{M}\sum_{j=1}^{M}\mathbf{y}^{(j)} \\ \widehat{K} &= \frac{1}{M-1}\sum_{j=1}^{M}(\mathbf{y}^{(j)}-\widehat{\boldsymbol{\mu}})(\mathbf{y}^{(j)}-\widehat{\boldsymbol{\mu}})^T \end{aligned} \tag{6.7}$$

Eigenvectors $\{\widehat{\boldsymbol{\psi}}_k\}_{k=1}^{N}$ of $\widehat{K}$ then form the *empirical motion eigenfunctions.* Let $\widehat{\Psi} := (\widehat{\boldsymbol{\psi}}_1, \widehat{\boldsymbol{\psi}}_2, \cdots, .\widehat{\boldsymbol{\psi}}_N) \in \mathbb{R}^{N\times N}$. The *empirical motion transform* of an arbitrary observation vector $\mathbf{w}$ of length N (corresponding to W samples of L sensory data) is then

$$\widehat{\mathbf{w}} = \widehat{\Psi}^T\mathbf{w} \tag{6.8}$$

which is a discrete-time version of the K-L transform of $\mathbf{w}$, sometimes also known as the *Principal Components Transform,* that represents the expansion coefficients of an arbitrary signal in terms of the special basis $\widehat{\Psi}$ designed to ensure that the transform coefficients $\widehat{w}(i)$ are uncorrelated. Furthermore, the vectors $\widehat{\boldsymbol{\psi}}_i$ in this basis are sorted in order of decreasing variance (eigenvalues), so that the first d principal components give the best $d-$dimensional approximation to the N-dimensional observation vector $\mathbf{w}$ in the sense that they capture the most important gait dynamics.

6.3.3 GMM-Based Bayesian Gait Mode Recognition & Fall Detection

The above principal components (PCs) are by themselves not adequate for classification based on observed data. Firstly, being completely unsupervised, it does not take into account class labels of feature vectors. Secondly, being purely data-driven, it is not possible to have a generative or probabilistic model for the observed data, which is often the case when there is an underlying structure in the input factors. Instead we use the PCs to generate a Gaussian mixture model (GMM) and estimate its parameters from data. Then based upon class labels in training data we use a Bayesian classifier that maximizes the class conditional probability of observed data. Similar approaches have been used in other domains such as in [239] and [9] but we are not aware of its application in sensor based gait analysis.

We assume that the observations corresponding to activity mode A_c for $c = 1, 2..C$, where C is the number of classes (such as Jumping, Walking, Running, Standing, Falling) lead to principal coefficients $\widehat{\mathbf{w}} = (\widehat{w}_1, \widehat{w}_2, \ldots \widehat{w}_d)$ distributed according to a joint probability

density $p_c(\widehat{w}_1, \cdots, \widehat{w}_d)$ that can be approximated as a Gaussian mixture:

$$p_c(\widehat{\mathbf{w}}) = \sum_{m=1}^{M_c} \pi_{m,c} \mathcal{N}(\widehat{\mathbf{w}}; \boldsymbol{\mu}_{m,c}, \boldsymbol{\Sigma}_{m,c}) \tag{6.9}$$

where $\sum_{m=1}^{M} \pi_{m,c} = 1$ and $\pi_{m,c} > 0$ and $\mathcal{N}(\mathbf{x};\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes is a multivariate normal distribution on $\mathbf{x}$ with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Parameters of this model, $\pi_{m,c}, \boldsymbol{\Sigma}_{m,c}, \boldsymbol{\mu}_{m,c}$ for $m = 1 \ldots M$ and number of mixtures M_c for $c = 1 \ldots C$ are estimated from training data using the Expectation-Maximization (E-M) algorithm ([68]). Once the distributions above are known, given an unknown signal $\mathbf{w}$ with principal components $\widehat{\mathbf{w}} = (\widehat{w}_1, \widehat{w}_2, \ldots \widehat{w}_d)$ we can use Bayes' theorem to produce the posterior probability that $\mathbf{w}$ was generated by activity mode A_c:

$$\mathcal{P}(A_c | \widehat{w}_1, \cdots, \widehat{w}_d) = \frac{p_c(\widehat{w}_1, \cdots, \widehat{w}_d)\mathcal{P}(A_c)}{\sum_{c=1}^{C} p_c(\widehat{w}_1, \cdots, \widehat{w}_d)\mathcal{P}(A_c)} \tag{6.10}$$

This can then be used to determine the most probable activity mode for an unknown waveform. The prior probabilities $\mathcal{P}(A_c)$ can be determined based upon the relative sizes of the per class training sets.

Different flavors of GMM based upon varying constraints (see Table 6.1) are explored in this study. In addition, we will also use a variant of GMM, what we call a *Heterogeneous GMM* (HGMM) model where the components $w_1, w_2, \cdots, w_d$ of $\mathbf{w}$ are uncorrelated, and for each $k = 1, ..d$, w_k is a mixture of M_k (univariate) normally distributed random variables. That is,

$$p(\mathbf{w}) = \prod_{k=1}^{d} p_k(w_k) \tag{6.11}$$

where each $p_k(w_k)$ is a Gaussian mixture of M_k components with component m of w_k having a normal distribution of mean μ_{mk} and variance σ_{mk}. A HGMM allows choosing different mixtures for different components and may thus reduce the total number of parameters needed to be estimated.

Table 6.1: Gaussian Mixture Models

Type	Abbrev	Constraints	#Params
Full Covariance	FC	none	$m(m-1)$
Diagonal Covariance	DC	$\Sigma_m = diag(v_m)$	m
Spherical Covariance	SC	$\Sigma_m = \sigma_m I$	1

6.3.4 HMM Based Gait Mode Recognition & Fall Detection

In the above formulation, if write the transformed observation vector $(\widehat{w}_1, \cdots, \widehat{w}_d)$ during time segment t as y_t then the Bayesian classification rule (6.10) can be thought of as way of estimating the activity *state* $z_t \in \{A_1, A_2 \cdots A_C\}$ at time t given observations y_t. Since the above rule ignores observations corresponding to any other time segments, the above classifier is *frame-based*, and by ignoring any history of y_t implicitly assumes that $\{z_t, t = 1, 2, \cdots\}$ are iid. However, it is possible that if the history of y_t were taken into account, as in a *sequence-based* classifier, we could get a better estimate of z_t. Making use of temporal dependence across activity modes is particularly helpful if y_t is very noisy of if there is high variability across subjects and may also help delineate similar activities (e.g. an immediate past "run" mode is indicative that a "stand" mode is more likely than a "sit" mode). We employed the following two Markov sequential models for gait classification.

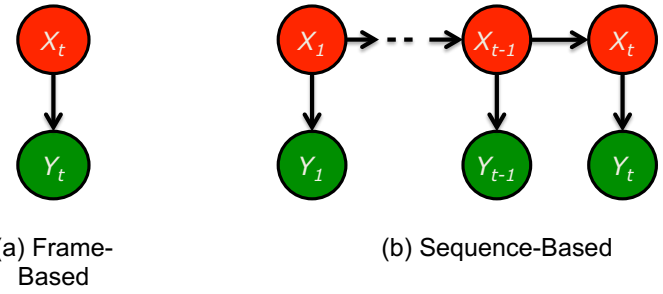


Figure 6.2: A frame-based vs. sequence-based activity classifier. X_t is the activity mode at frame t, which needs to be estimated. Y_t are the sensor observations at frame t. Estimation using frame based approach seeks to maximize $\Pr(X_t|Y_t)$ whereas the sequence-based approach maximizes $\Pr(X_t|Y_t, Y_{t-1}, ..Y_1)$.

1. *Simple Markov Model:* We assume that activity mode z_t at frame t depends only on activity at frame $t-1$, and thus can be modeled as a discrete-time Markov chain (DTMC) with transition matrix Q (Fig 6.3). Sensor observations y_t for frame t depend on x_t only with a Gaussian mixture output model, i.e.

$$\Pr(y_t|z_t = A_c) = \sum_{j=1}^{M} a_{cj} N(\mu_{cj}, \Sigma_{cj}) \tag{6.12}$$

   where M is the number of mixtures, and A_c is the activity mode.

2. *Cascade Markov Model:* The simple Markov model is able to capture temporal dependencies across frames (˜110s in our implementation), i.e. between different modes.

However each gait cycle can be considered a complex interaction of physiological states and has dynamics at time scales finer than the gait cycle length. For example, it is well known that the walk and run gait cycles comprises of the sub-states of: right/left swing/stance, with how they occur relative to each otherdiffering in each case (Fig 6.3.5). Hence it is more natural to model the gait process as a *Cascade Markov Model* (Fig 6.3(b)) as described in Part A, with state $z_t \in \{A_1, A_2 \cdots A_C\}$ representing activity, $x_t \in \{S_1, S_2 \cdots S_M\}$ representing intermediate sub-states during the activity and y_t the sensor observations. In this case the time scale t is much shorter than the gait cycle (we have used about 10s in our implementation). We assume a Gaussian output model, with outputs y_t dependent on internal state x_t and activity z_t.

$$\Pr(y_t|x_t = S_m, z_t = A_c) = N(\mu_{cm}, \Sigma_{cm}) \tag{6.13}$$

6.3.5 Activity Classification using Markov Models

We assume we are given training sequences of observations for each activity as well as combined activities, and we wish to detect activity for a test observation sequence. For both of the above Markov models, this can be considered to be the problem of state estimation in hidden Markov Models (HMM). For the simple Markov model, parameters (π, Q) are estimated using simple maximum likelihood estimation, where π is the initial probability of z_t and Q is the transition matrix of $z_{t.}$ Since the state z_t is fully observable on training sequences, the MLE can be done in closed form without having to use E-M. The parameters $a_j, \mu_{cj}, \Sigma_{cj}$ of (6.12) are estimated using E-M for Gaussian mixtures similar ot that in teh non-sequential case. During testing the state z_t is considered hidden, and the most likely hidden state sequence is estimated as

$$z^*_{0:t} = \arg\max_{z_{1:t}} \Pr(z_{1:t}|y_{1:t})$$

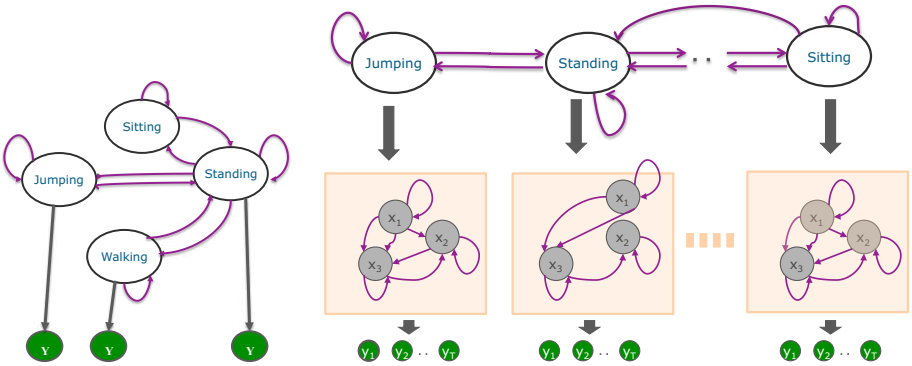
which can be done using the Viterbi algorithm.

For the cascade Markov model, since activity modes z_t are observable during the training phase, the problem of estimation of paramters can be considered one of partially observable cascade HMM (PO-CHMM). The E-M based algorithm for PO-CHMM (5.25) is used to estimate the parameters $(\omega, \pi, Q, A, \mu, \Sigma)$ where ω is the initial probability of z_t, π is the initial conditional probability $x_t|z_t$, Q is the transition matrix of z_t, A is the conditional transition matrix of x_t, and μ, Σ are the parameters in (6.13). During testing, the most likely

state sequence z_t

$$z^*_{0:t} = \arg\max_{z_{1:t}} \Pr(z_{1:t}|y_{1:t})$$

is estimated using the Viterbi algorithm for PO-CHMM (5.5.3). Since the time scale for change in the process z_t is much larger than that of x_t we used the stationary approximation for PO-CHMM in our implementation (see Section 5.5.4).



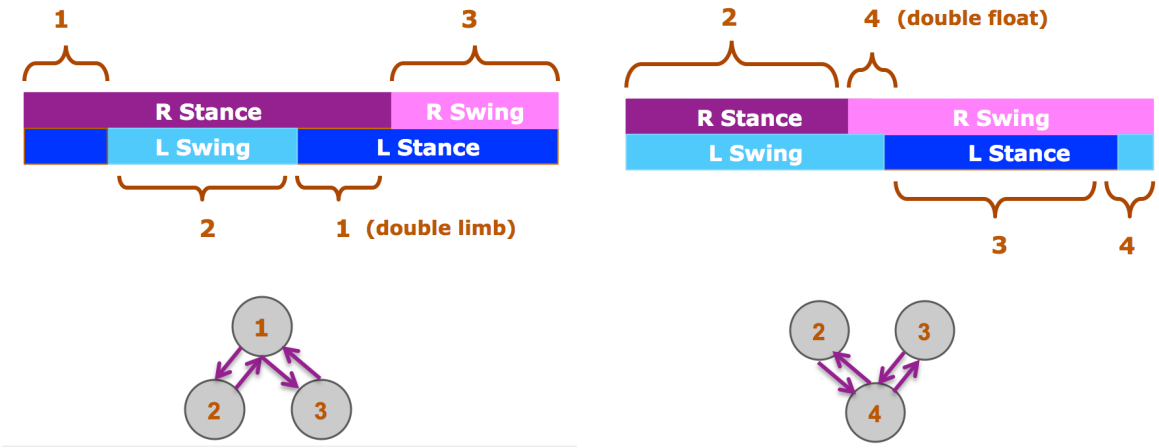
(a) (b)

Figure 6.3: (a) Simple Markov and (b) Cascade Markov models for the gait process. Each activity corresponds to a state of z_t. In (a) sensor observations y_t at frame t are directly dependent on z_t. In (b) sensor observations y_t are dependent on an intermediate process x_t representing physiological states during a gait cycle, whose transitions in turn are influenced by z_t. The joint process (z_t, x_t) forms a *cascade* Markov process.

6.4 Methods

6.4.1 Experiment Design

We created an offline database of sensory data for different modes of motion as performed by 20 volunteers varying in gender, age, height and gait stability. About 60s of per mode (Sitting, Standing, Jumping, Walking, Running) and combination activity data were collected per subject. Subjects were also asked to fall in four different ways (front/back/left/right); approximately 10-15s worth of data starting just prior to the subject's fall to after the fall was recorded. The type of activity was labelled based upon directions to and observations of the subject. Calibration data of about 60s was recorded prior to each experiment. Recorded data was visualized as it was recorded and experiments were repeated if necessary to ensure data quality using a custom graphical user interface written in Java



(a) Walking

(b) Running

Figure 6.4: The various sub-cycles in (a) walking and (b) running gait cycles. The corresponding state transitions allowed are shown at the bottom. These gait cycles can be considered as transitions between three states, "1" : Right Stance/Left Stance (also known as double limb), "2": Right Stance/Left Swing, "3": Right Swing/Left Stance, "4": Right Swing/Left Swing (also known as double float). In Walking, there is no transition to double float, and in Running there is no transition to double limb.

(more details in Online Supplemental Material. Collected data samples from right and left shoe sources were reconciled and segmented into contiguous and consistent sets by analyzing timestamp gaps and packet sequence inconsistencies.

Collected data segments of 60s each were thus either single-mode (containing a single activity) or multi-mode (containing multiple activities or events). Segments containing falling data were always multi-mode, from which single-mode fall data segments were visually extracted. A random selection of approximately 42% of the single-mode data segments (corresponding to about 45000 data points per mode) were designated as the *training data set*. The remaining single-mode and all multi-mode data segments (corresponding to 47000-60000 data points per mode) were assigned to the *testing data set*. In addition, varying lengths of single-mode data were randomly ordered and concatenated to synthesize multi-mode data in order to be able to test for mode transition latency. A subset of training data (approximately 5% of total single-mode data segments, or roughly a total of 22000 data points across all modes), visually inspected to ensure data quality, were assigned to an *exemplary data set* used for eigen function generation during the training phase (see below). Each data point in all segments was labelled with the mode type. All data were preprocessed using calibration

data collected prior to each experiment (data collection cycle per subject). Calibration data is smoothed and filtered using a standard deviation based outlier filter and then subtracted from incoming sensor data.

6.4.2 Algorithm I: Using GMM/Bayesian Model

A schematic of an analytic engine based upon methods discussed in section 6.3 is shown in Fig. 6.5. The engine comprises of an offline *training* phase, and an online *detection* algorithm.. The training phase produces basis functions and estimated statistical parameters from segments of testing and exemplary data, which are applied by the detection algorithm to consecutive segments of streaming input data to determine instantaneous gait activity or occurence of a fall. The latter is used for both continuous classification of incoming sensory data as well as algorithm accuracy evaluation against recorded test data. Details on the testing and training phases follow.

Data Encoding and Segmentation. The input data comprises of continuous streaming data at sampling rate f_s from four pressure sensors and three accelerometers from each of left and righ t insoles. Thus data at each time point is $L = 14$-dimensional. We segment the data using a fixed window size W and overlap O. We thus get a sequence of WL dimensional input vectors, thereby reducing the sampling frequency from f_s to $\left\lfloor \frac{N_s}{W-O} \right\rfloor$ where $\lfloor \cdot \rfloor$ is the floor operator. To generate training samples we do not use any overlap, i.e. we set $O = 0$ and retain class labels on observation vectors. The impact of O on the accuracy and latency of classification of test data is discussed in the Results section.

Offline Training Phase. Given the training data set $\mathcal{S}$ and exemplary data set $\mathcal{O} \subset \mathcal{S}$, the offline training phase outputs the d eigenfunctions $\widehat{\Psi_d} = \{\widehat{\boldsymbol{\psi}}_k\}_{k=1}^{d}$ and the estimated HMM parameters (GMM parameters $\Theta_c^* = \{M_c, \pi_{m,c}, \boldsymbol{\Sigma}_{m,c}, \boldsymbol{\mu}_{m,c}, m = 1 \ldots M\}$ for $c = 1 \ldots C$.The encoded WL-dimensional observation vectors from the exemplary data set $\mathcal{O}$ used to compute the empirical eigenfunctions $\widehat{\Psi}_d$ using 6.7. To avoid singularities in the covariance matrix we do not subtract the mean and thus ignore the largest eigenvalue (the corresponding eigenvector is the sample mean). For each class c, the reduced feature vectors $\widehat{\mathbf{w}}$ (using 6.8) are derived for all encoded WL-dimensional observation vectors $\mathbf{w} \in \mathcal{S}$ with class label c, and all such reduced vectors for class c form the training pool Y_c. For each class c, an initial estimate $\Theta_c^{(0)}$ of the GMM parameters is obtained by using $k-$means on the pool Y_c. These are then used to determine the maximum likelihood estimate (MLE) Θ_c^*

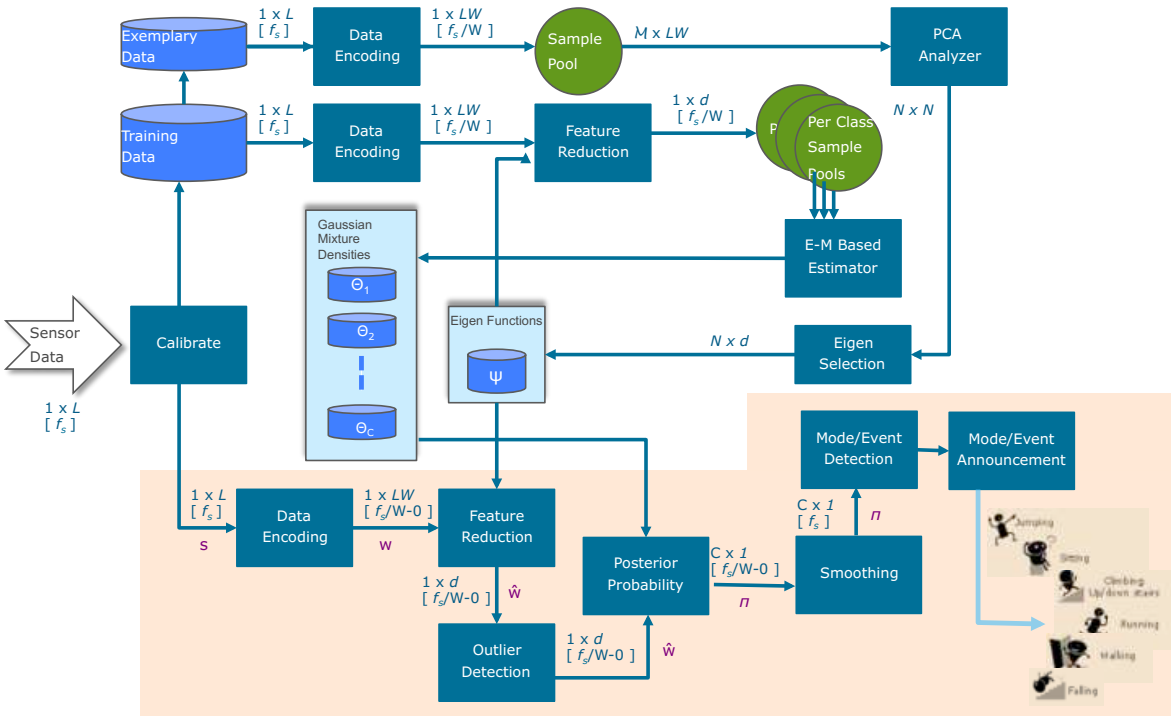


Figure 6.5: Schematic of GMM-based Bayesian analytic engine for activity classification. Each block represents a processing step of the training (unshaded) or detection (shaded) phase of the algorithm with signal length of inputs and outputs specified next to arrows in and out of the block. Sampling frequency for a signal where relevant appears in square brackets. L is the number of sensors, f_s is the sampling rate, W is the window size in samples, d is the dimensionality of the reduced feature space, O is the overlap size, C is the number of classes, M is the sample pool size and $N = WL$.

via the E-M algorithm with termination condition based upon threshold of log likelihood on successive iterations.

Online Detection Algorithm. A realtime algorithm is used for online activity detection, presented in Algorithm 7 and depicted schematically in Fig 6.5. The input to the online detection algorithm at time t_i is a length T segment of sensory data centered at t_i, encoded using a window length $W = f_s T$ and overlap size O to form the input vector $\mathbf{w}_{t_i}$. The output of the online detection algorithm is the classified mode $c^*_{t_i}$ at time t_i. Adaptation for testing samples is done by simply augmenting inputs with class label information and using that to compute performance metrics, as described the Section 6.4.5. Outlier detection described in step (2) of the algorithm is done using a variance based threshold on a moving window per

dimension. A moving average filter $MovAvg$ using window S_O is used in step (6).

Our choice of the following parameters is discussed in section 6.5: segmentation frame window size W, overlap O, number of principal components d, GMM model type, outlier thresholds α_k, smoothing window size S_0, that appear in the algorithm above.

Algorithm 7: Real-time Gait Detection Using PCA

Input: $\mathbf{w}_{t_i}$

Given: $\widehat{\Psi}_d$, Θ^*_c, $\mathcal{P}(A_c)$ for $c = 1 \dots C$

Parameters: W, O, d, S_O, $\alpha_k, k = 1 \dots d$

Output: $c^*_{t_i}$

1 Compute $\widehat{\mathbf{w}}_{t_i} = \widehat{\Psi}_d^T \mathbf{w}$;

2 Replace if $\widehat{\mathbf{w}}_{t_i}$ is an outlier using thresholds α_k ;

3 Compute $p_c(t_i)$ using Θ^*_c for $c = 1 \dots C$ using (6.9) ;

4 Compute $\mathcal{P}(A_c|\widehat{\mathbf{w}}_{t_i})$ using $p_c(t_i), \mathcal{P}(\mathcal{A}_\rfloor)$ and (6.10) ;

5 Compute class likelihood sequence $\tilde{p_c}(n)$ where $\tilde{p_c}(nf_s + W/2) := \mathcal{P}(A_c|\widehat{\mathbf{w}}_{t_i})$;

6 Compute smoothed likelihood $\tilde{\pi} := \texttt{MovAvg}(S_O, \tilde{p_c})$;

7 Set $c^*_{t_i} = \arg\max_c \tilde{\pi}_c(t_n)$

8 **return** $c^*_{t_i}$;

6.4.3 Algorithm II: Using Simple Markov Model

A schematic of an analytic engine based upon methods discussed in section 6.3 is shown in Fig. 6.6. Similar to the PCA based engine, there is an offline *training* phase, and an online *detection* algorithm. Data encoding and segmentation is identical to that for the PCA based algorithm.

Offline Training Phase. Given the training data set $\mathcal{S}$ the exemplary data set $\mathcal{O} \subset \mathcal{S}$ is used to produce the d eigenfunctions $\widehat{\Psi_d} = \{\widehat{\psi}_k\}_{k=1}^d$ identical to that in the PCA based case. Multi-mode training sequences of reduced feature vectors $\widehat{\mathbf{w}}$ obtained from encoded WL-dimensional observation vectors $\mathbf{w} \in \mathcal{S}$ are first used to estimate the transition matrix Q and initial probabilties π. Then subsequences of single mode training sets are concatenated in accordance with the above Q, π to synthesize a much larger training data set from which the parameters $\Theta^* = \{Q, \pi_c, a_{ck}, \mu_{ck}, \Sigma_{ck}, m = 1 \dots M,\ c = 1 \dots C\}$are estimated using MLE and E-M. See equations (6.12) and 5.16.

Online Detection Algorithm. Similar to the Bayesian algorithm, the algorithm for real-time activity detection based upon the simple Markov model (8) takes as input at time t_i a length T segment of sensory data centered at t_i, encoded using a window length $W = f_s T$ and overlap size O to form the input vector $\mathbf{w}_{t_i}$ which is then feature reduced to $\widehat{\mathbf{w}}_{t_i}$ and checked for outliers. In addition the algorithm keeps history of H_a inputs i.e. $\widehat{\mathbf{w}}_{t_j}$ for $j = i-1 \cdots i-H_a$. In addition, the algorithm can use future inputs of length H_b i.e. $\widehat{\mathbf{w}}_{t_j}$ for $j = i+1, \cdots, i+H_b$. The parameters H_a, or *pre-observation window* and H_b, *post-observation window* are tunable. Then Viterbi algorithm (5.14) is applied to the sequence $\left\{c^*_{t_j}\right\}_{j=i-H_a}^{i+H_b}$ using trained Θ^* values, and the value $c^*_{t_i}$ is used as the classified mode $c^*_{t_i}$ at time t_i.

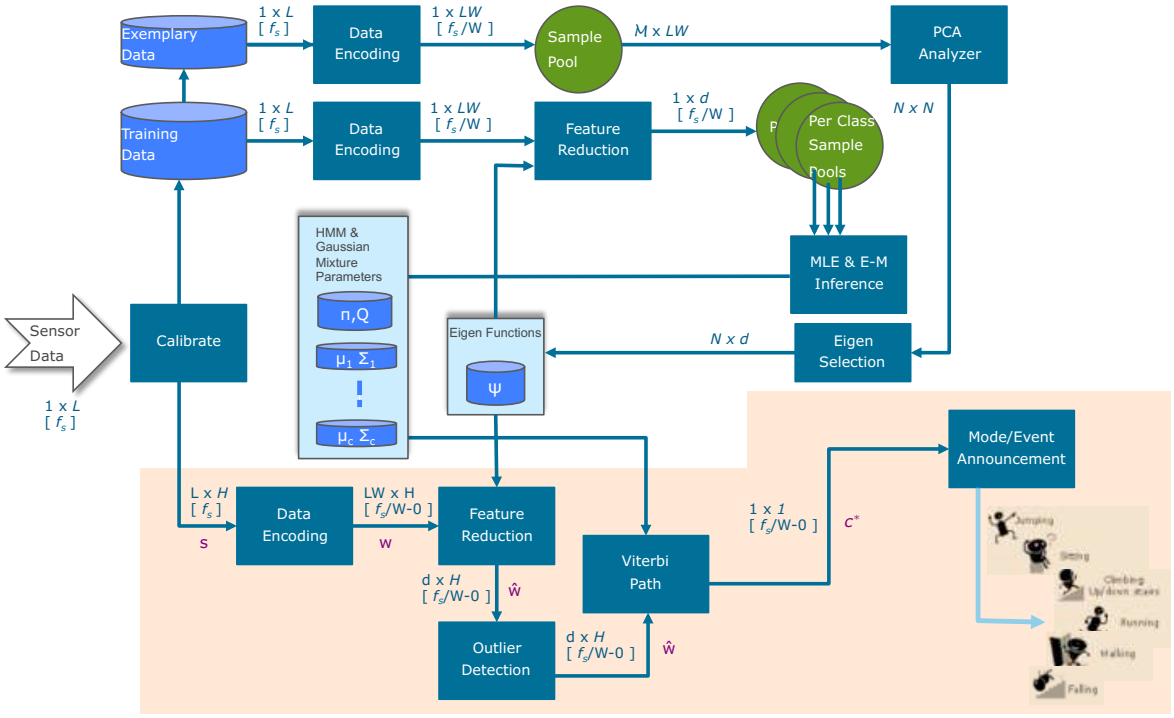


Figure 6.6: Schematic of simple Markov model based analytic engine for activity classification. Description as in Fig 6.5. $H = H_a + H_b$ is the observation window length used for the Viterbi algorithm.

6.4.4 Algorithm III: Using Cascade Markov Model

The CHMM discussed in Chapter 5, Part A used for this model with z_t being the acivity mode and x_t being the hidden sub-states. Since the timescale of dynamics of z_t is larger than that of x_t we use the quasi-stationary approximation to PO-CHMM model

Algorithm 8: Real-time Gait Detection Using Simple Markov Model

Input: $\mathbf{w}_{t_i}$

Given: $\widehat{\Psi}_d$, $\Theta^* = \{Q, \pi_c, a_{ck}, \mu_{ck}, \Sigma_{ck}\}_{c=1..C,k=1..M}$

Parameters: H_a, H_b, $\alpha_k, k = 1 \ldots d$

Require: $\mathbf{w}_{t_j}$ for $j \in [i - H_a : i + H_b]$

Output: $c^*_{t_i}$

1 Compute $\widehat{\mathbf{w}}_{t_j} = \widehat{\Psi}_d^T \mathbf{w}_{t_j}$ for $j \in [i - H_a : i + H_b]$;

2 Replace if $\widehat{\mathbf{w}}_{t_j}$ is an outlier using thresholds α_k. ;

3 Compute $\left\{c^*_{t_j}\right\}_{j=i-H_a}^{i+H_b}$ from $\left\{\widehat{\mathbf{w}}_{t_j}\right\}_{j=i-H_a}^{i+H_b}$, Θ^* using Viterbi (5.14);

4 **return** $c^*_{t_i}$

(section 5.5.4). A schematic of an analytic engine based upon this model is shown in Fig. 6.7. Similar to the PCA based engine, there is an offline *training* phase, and an online *detection* algorithm. Data encoding and segmentation is identical to that for the PCA based algorithm, except that the window length W and overlap size O are much smaller to capture the much faster dynamics of the sub-states x_t.

Offline Training Phase. Given the training data set $\mathcal{S}$ the exemplary data set $\mathcal{O} \subset \mathcal{S}$ is used to produce the d eigenfunctions $\widehat{\Psi_d} = \{\widehat{\boldsymbol{\psi}}_k\}_{k=1}^d$ identical to that in the PCA based case. Training sequences of reduced feature vectors $\widehat{\mathbf{w}}$ obtained from encoded WL-dimensional observation vectors $\mathbf{w} \in \mathcal{S}$ per class c are then used, as described in the "Decoupled PO-CHMM Estimation & Inference Algorithm" of section 5.5.4 are used to estimate the parameters $\Theta^* = \{\omega_c, \pi_c, A_c, \mu_c, \Sigma_c,\ c = 1 \ldots C\}$ (see equations (6.13)).

Online Detection Algorithm. Similar to the case based upon the simple Markov model, the algorithm for realtime activity detection based upon the the cascade Markov model (9) takes as input the vector $\mathbf{w}_{t_i}$ which is then feature reduced to $\widehat{\mathbf{w}}_{t_i}$ and checked for outliers, requires inputs from the pre-and post-observation windows i.e $\left\{\widehat{\mathbf{w}}_{t_j}\right\}_{j=i-H_a}^{i+H_b}$, computes the likelihood of the observation sequence is computed for each class c (5.43) and uses the class with maximum likelihood (5.44) is used as the output classified mode $c^*_{t_i}$ at time t_i.

All three algorithms described above were implemented in MATLAB for offline training and testing. Versions for each were adapted to MATLAB/Simulink for online detection permitting generation of embedded code for realtime computing environment.

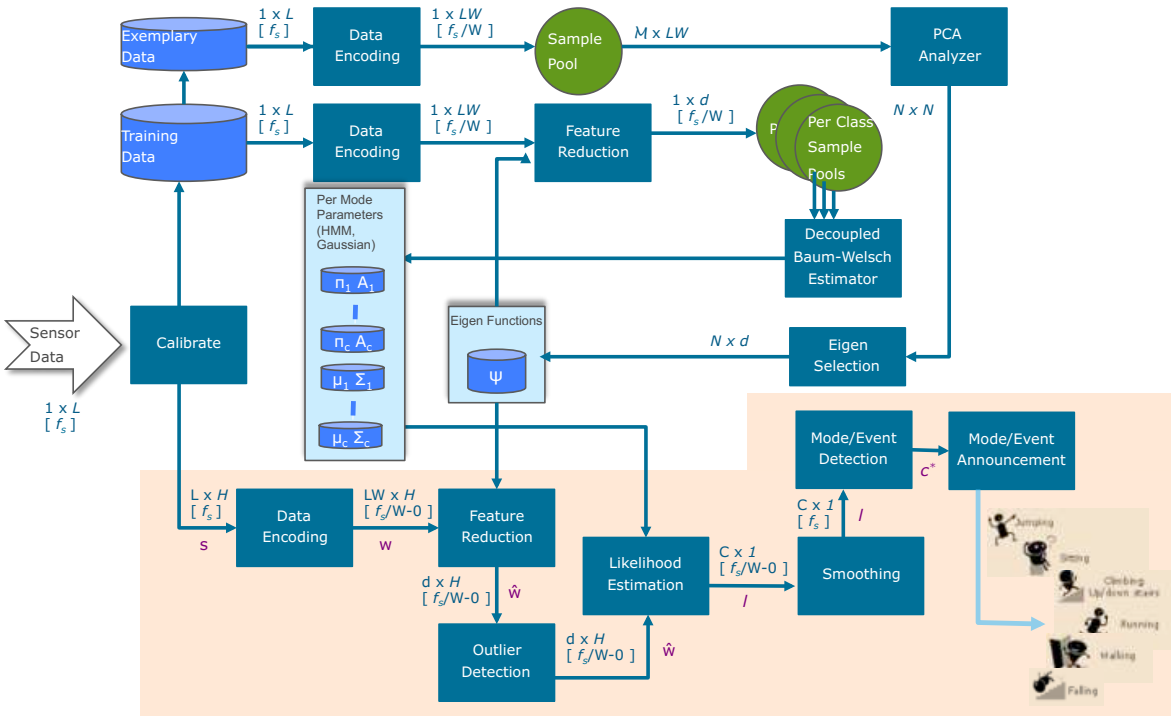


Figure 6.7: Schematic of cascade Markov model based analytic engine for activity classification. Description as in Fig 6.5. $H = H_a + H_b$ is the observation window length used for likelihood estimation (5.43). The block labeled "Decoupled Baum Welsch Estimator" refers to the steps in the algorithm described in 5.5.4.

Algorithm 9: Real-time Gait Detection Using Cascade Markov Model

Input: $\mathbf{w}_{t_i}$

Given: $\widehat{\Psi}_d$, $\Theta_c^* = (\omega_c, \pi_c, A_c, \mu_c, \Sigma_c)$ for $c = 1 \dots C, k = 1 \cdots M$

Parameters: H_a, H_b, $\alpha_k, k = 1 \dots d$

Require: $\mathbf{w}_{t_j}$ for $j \in [i - H_a : i + H_b]$

Output: $c_{t_i}^*$

Compute $\widehat{\mathbf{w}}_{t_j} = \widehat{\Psi}_d^T \mathbf{w}_{t_j}$ for $j \in [i - H_a : i + H_b]$;

Replace if $\widehat{\mathbf{w}}_{t_j}$ is an outlier using thresholds α_k ;

Compute L_c from $\left\{\widehat{\mathbf{w}}_{t_j}\right\}_{j=i-H_a}^{i+H_b}$ and Θ_c^* using (5.43) ;

Set $c_{t_i}^* = \arg\max_c L_c \omega_c$;

return $c_{t_i}^*$

6.4.5 Performance Metrics

We used the following performance metrics to quantify the accuracy of the detection algorithms against labelled single-mode and multi-mode test data as described in section 6.4.1. *Misclassification rate (MR)* is the percentage of data points where the classification does not match its class label. For single-mode test data, this is an indicator of false negative (FN) rate and for multi-mode data is it indicative of both FPR and FNR. FP and FN rates are evaluated separately for multi-mode data containing falls only. *Boundary Latency (BL)* is the aggregate delay in detection of mode transitions (in a multi-mode data set), expressed as a percentage of the total test runtime. We are able to accurately assess it for synthesized multi-mode data only.

6.5 Results

6.5.1 Empirical Eigenfunctions

The eigenvectors $\{\hat{\boldsymbol{\psi}}_j\}_{j=1}^{N}$ of the empirical covariance matrix $\hat{K}$ estimated using the exemplary training samples form the empirical motion basis functions, as in (6.5) and (6.7) . Preprocessed gait sensor waveforms can be then projected onto these eigenfunctions. The first ten eigenfunctions in decreasing order of eigenvalues (Fig 6.8) capture distinctive characteristics of various gait activities. Most of the energy of gait patterns is captured in the first seven coefficients (Fig 6.9), and accordingly we set $d = 7$, dimensionality of the reduced feature set.

6.5.2 Transform Coefficient Distributions & Gaussian Mixtures

Examination of the empirical marginal and scatter distributions of the first seven principal components $w_1..w_7$ of the training samples in each activity class allows us to determine what mixture model to use per activity class for the PCA based algorithm (Figs 6.10 and 6.11). For example, if using the heterogeneous Gaussian mixture model (HGMM), the empirical marginal distributions would suggest using three component Gaussian mixtures each for w_2,w_3 and single Gaussian for the rest in the WLK class, but using three component Gaussian mixtures for w_4, w_6, two-component Gaussian mixture for w_1 and single Gaussians for the rest in the RUN class. It is this differentiation amongst modes that allows us to detect them using our algorithm. Based upon this empirical analysis, four different Gaussian mixture models were explored for our implementation: **MIX-1** and **MIX-2** are *heterogeneous*

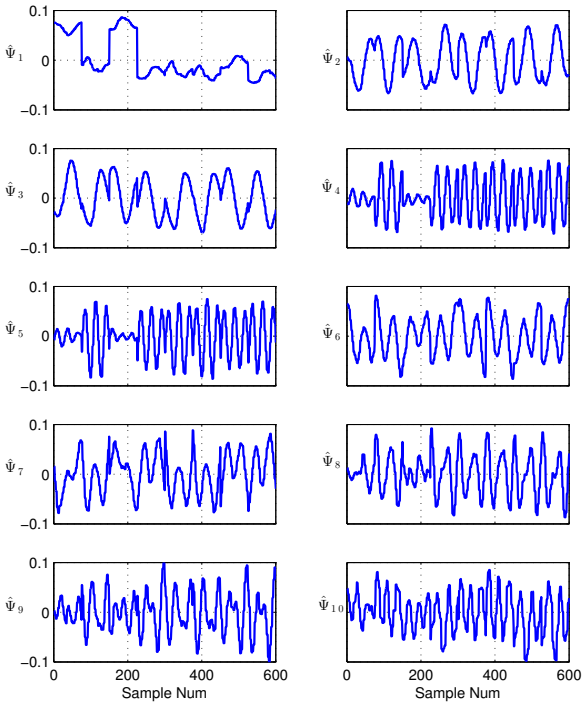


Figure 6.8: First ten eigenfunctions of the gait basis used to represent waveforms over segmentation period $W = 75$ samples collected from 8 gait sensors. These are derived from the K-L transform of exemplary samples. The x-axis represents the sample number corresponding to $W \times 8 = 600$ samples in each waveform.

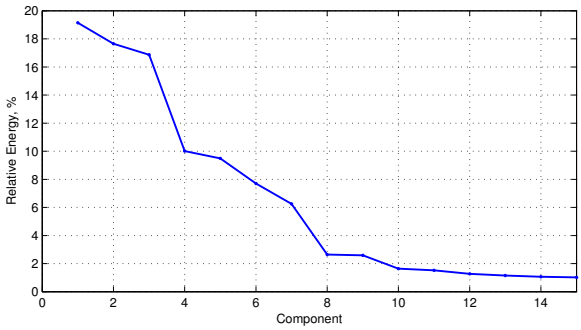


Figure 6.9: 15 largest eigenvalues of the empirical covariance matrix of the samples in the exemplary pool.

GMMs (6.11) with varying number of mixtures per component shown in Tables 6.2(a) and 6.2(b) respectively. **MIX-3** and **MIX-4** are *homogeneous* GMMs (6.9) with $M = 4$ mixtures and *spherical* or *diagonal* covariance matrix respectively..

Table 6.2: Number of Mixtures M_k Per Component k For the (a) MIX-1 (b) MIX-2 model

Activity	M_1	M_2	M_3	M_4	M_5	M_6	M_7
WLK	1	3	3	1	1	1	1
RUN	1	1	1	2	2	1	1
JMP	1	1	1	1	1	2	2
STD	1	1	1	1	1	1	1

(a)

Activity	M_1	M_2	M_3	M_4	M_5	M_6	M_7
WLK	2	3	3	1	1	1	1
RUN	2	1	1	3	1	3	1
JMP	2	1	1	1	3	1	3
STD	2	1	1	1	1	1	1

(b)

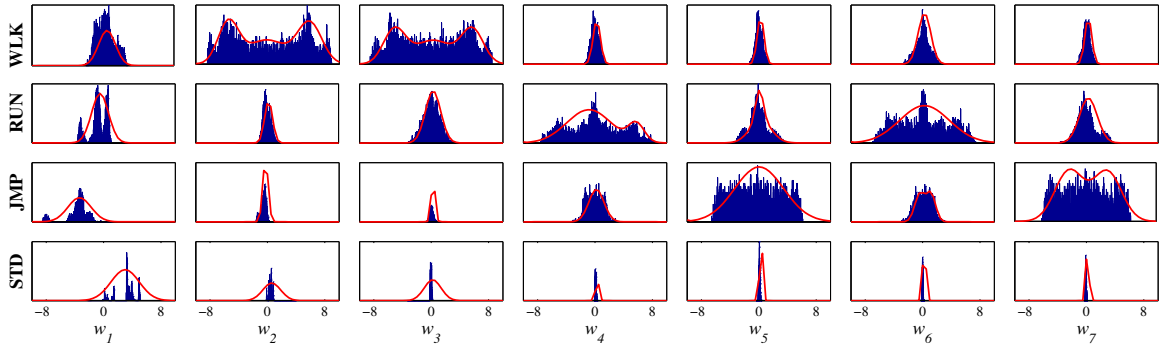


Figure 6.10: Empirical marginal distributions of the first seven principal components $w_i, i = 1..7$ of training samples corresponding to walking (WLK), running (RUN), jumping (JMP) and standing (STD). The estimated densities when using the **MIX-1** model are shown in red.

6.5.3 Activity Classification

Application of the PCA-based algorithm and CHMM-based algorithm on individual test data sets showed close to perfect classification accuracy when single mode segments are used for testing (example in Fig. 6.12, Fig. 6.13), and when tested on multi-mode data that were concatenated using random single mode subsegments, the PCA-based algorithm showed more lag in classification especially around boundaries whereas CHMM-based algorithm had almost no latency. An interesting observation (Fig 6.13, bottom right panel) is that transitions amongst the hidden substates x_t during running is similar to that expected amongst swing/stance related sub-phases of the gait cycle (Fig 6.3.5). A comparison of the classification accuracy across all test data for both multi-mode data and single mode data for the PCA-based, simple Markov model based and CHMM based algorithms, when each used their determined set of optimal parameters as discussed in the section 6.5.5, shows that

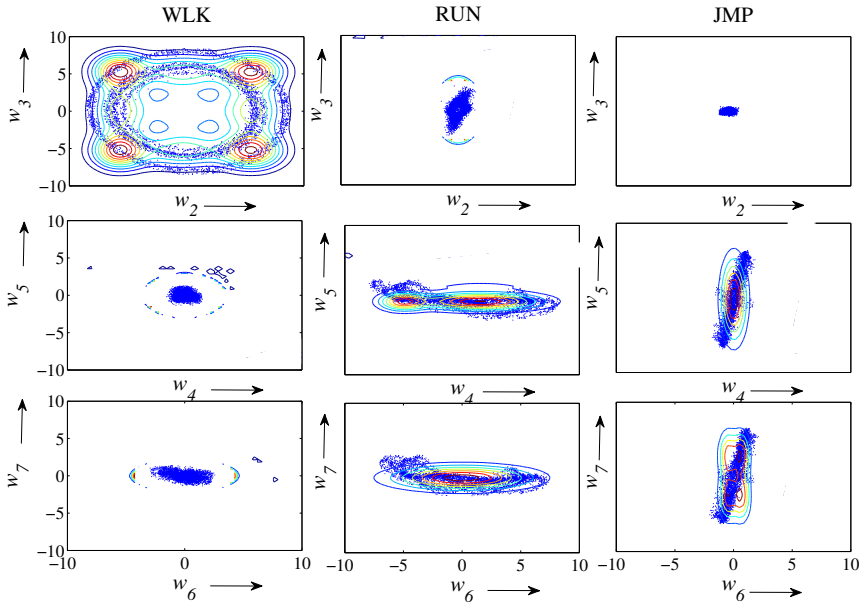


Figure 6.11: Some empirical pairwise joint distribution of the first sevel principal components $w_1, i = 1..7$ of training samples corresponding to walking (WLK), running (RUN), jumping (JMP) and standing (STD). Only four pairs are shown. Contour plots (solid color lines) are the estimated joint densities with the **MIX-1** model.

the CHMM based classifier has the least error rate when detecting multi-mode data, and higher error rate when classifying single activity JMP and RUN data. All algorithms are nearly perfect when classifying STD and WLK modes, and the performance of PCA-based and simple Markov model-based algorithms is comparable across all activities (Fig 6.15).

6.5.4 Fall Detection

For detecting falls, empirical eigenfunctions were recomputed with including fall exemplary data sets using the procedure described in section 6.4.2. Both PCA based and CHMM based algorithms were applied to test data segments, and the PCA based approach shows higher latency in fall detection than the CHMM-based method (Figs 6.16, 6.17). The Markov model based methods also exhibit lower FPR and FNR than the GMM based method (Fig 6.18).

A demonstration video of the real-time implementation of the CHMM based algorithm, in particular, detection of falling is available on Online Supplemental Material.

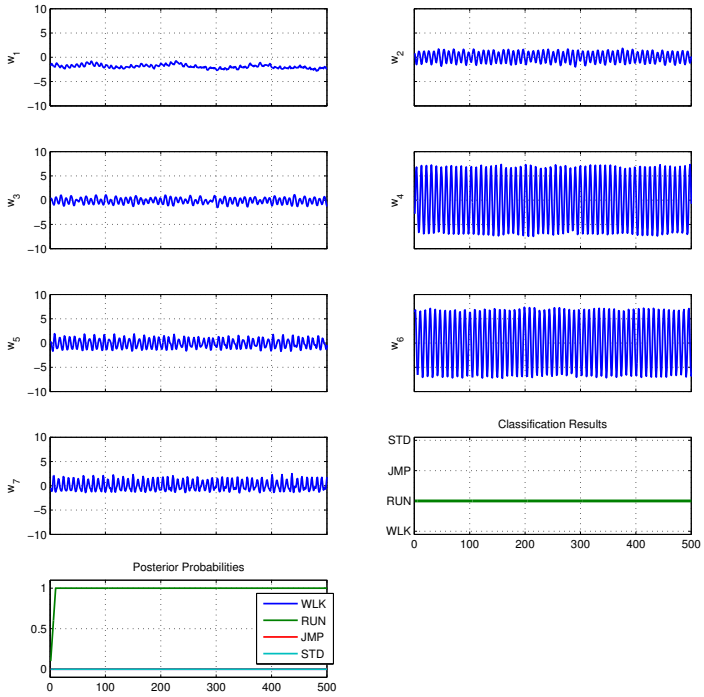


Figure 6.12: Principal values $w_1..w_7$, posterior class probabilities and classification result for a RUN segment of about 5s using the PCA-based algorithm with window $W = 75$ and overlap $O = 75$. The x-axis on each of the plots is the sample number.

6.5.5 Impact of Algorithm Parameters

Impact of Parameters on Algorithm I (GMM-based). Variation of algorithm parameters such as W, O, S_O and α impact the performance of the PCA-based algorithm differently in single activity and multi-activity detection (Fig 6.5.5 (a),(b) and (c)). No single segmentation policy gave optimal results for single-mode, multi-mode as well as falling data. Multi-mode or fast varying data is detected with higher accuracy using smaller window sizes whereas slowly varying single-mode data performs better with larger window sizes. A choice of $W = 110, O = 105$ yields overall 98% accuracy for single activity classification and 94% in multi-activity classification. For falls, this choice of W gives 90% fall detection rate but a 30% FPR. Decreasing the frame size improves fall detection accuracy, but decreases activity mode detection accuracy. A small choice for the probability filtering window $S_0 = 10$ suffices and longer window sizes increase boundary latency. The ideal choice for outlier detection threshold $\alpha = 3$ (the same value is used for all components). The impact of various mixture GMM models on performance (Fig 6.20) indicates that **MIX3** is overall the best model.

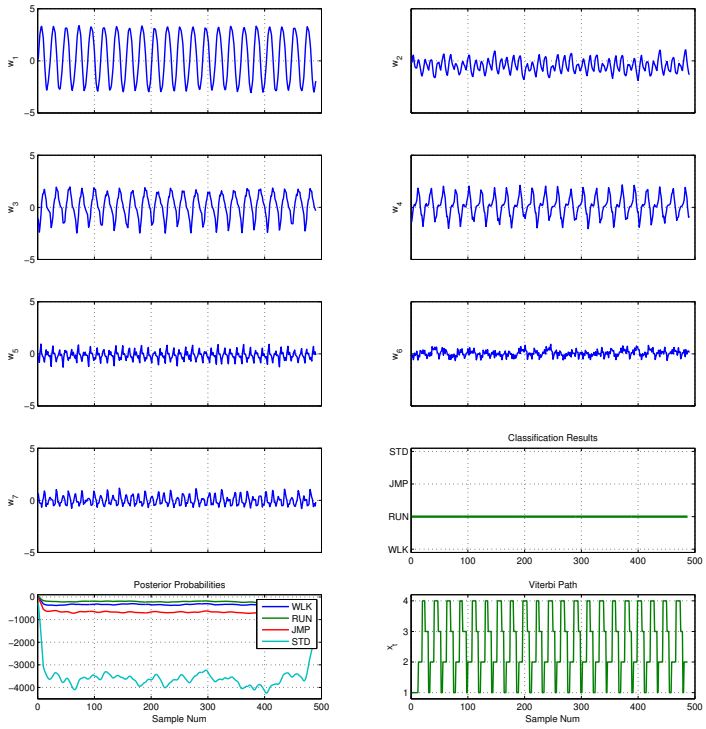


Figure 6.13: Principal values $w_1..w_7$, posterior class probabilities and classification result for a RUN segment of about 7s using the HMM-based algorithm with window $W = 10$ and overlap $O = 8$. The x-axis on each of the plots is the sample number. The bottom right panel also shows the Viterbi path for the observed sequence, i.e. the most likely sequence of hidden substates x_t. The relatively stable pattern of transitions suggests that these hidden states could be interpreted as stance/swing related sub-states within a gait cycle.

Impact of Parameters on Algorithm-II (simple Markov model based). Impact of W, O, α are similar on the simple Markov based algorithm to that of the PCA-based algorithm and are not shown. The impact of observation window H and its components H_a, H_b (Fig 6.21(a)) on multi-mode and single-mode data imply that smaller values of both H_a, H_b are better. Three different initial transition matrices Q were evaluated, as follows: (i) Q_1, with each inter-class transition probability being 0.1, (ii) Q_2, with each inter-class transition probability being 0.01, and (iii) Q_3 with all transitions equally likely. We observed that the choice of Q_1 or Q_2 does not make a difference in the performance, and Q_3 gives poorer detection accuracy (specifics omitted).

Impact of Parameters on Algorithm-III (CHMM-based). Impact of W, O, α are similar on the CHMM-based algorithm is similar to that on PCA-based algorithm and are

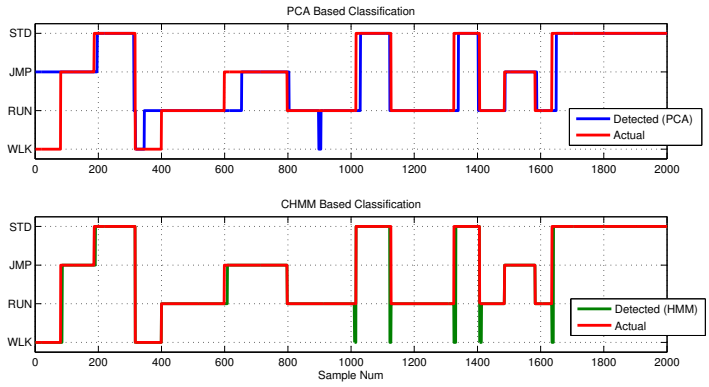


Figure 6.14: Classification results for testing on a ˜30s segment of multi-mode data that was generated using random subsequences of single mode data from the same subject using PCA-based algorithm (top) and CHMM-based algorithm (bottom). The x-axis indicates sample number.

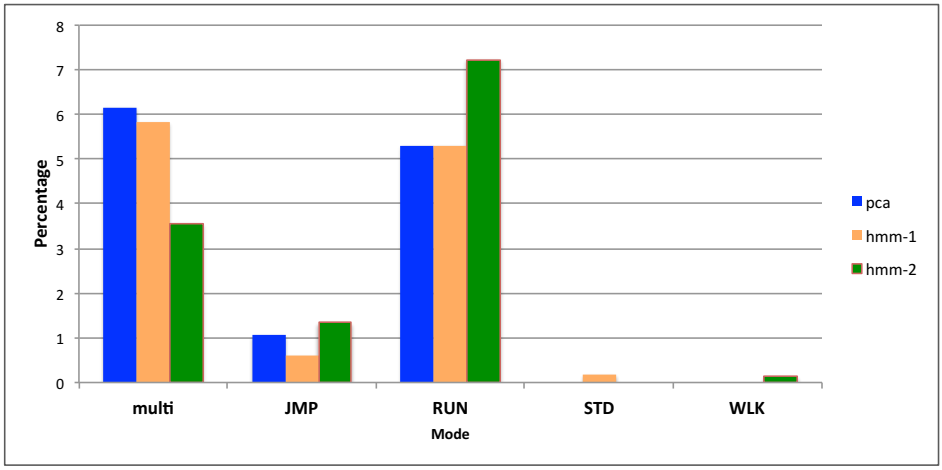


Figure 6.15: Comparison of misclassification rate across all test data for single activity test segments - jumping (JMP), walking (WLK), running (RUN) and standing (STD) - as well as multi-mode data (multi), for PCA-based algorithm (pca), simple Markov model based algorithm (hmm-1) and cascade Markov model based algorithm (hmm-2).

not shown. A study of the lag/lead observation windows parameters H_a, H_b parameters (Fig 6.21(b)) the observation window H and its components H_a, H_b indicates that a large observation window improves single-mode detection accuracy but a small window improves latency and hence multi-mode detection accuracy. We used the values $H_b = 5$, $H_a = 20$ that provide a reasonable tradeoff between accuracy and latency for the classification results presented previously.

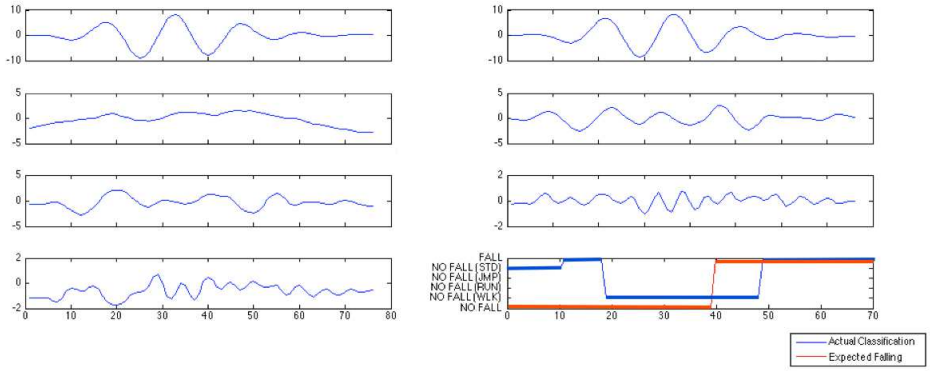


Figure 6.16: First seven principal values $w_1 \cdots w_7$ and detected mode for a FALL test segment (˜3.5s) using the PCA based algorithm. The x-axis denotes scaled sample numbers (actual values are three times the values indicated). The actual data was only labeled as "no fall" vs "falling" whereas the detection is for all modes and falls.

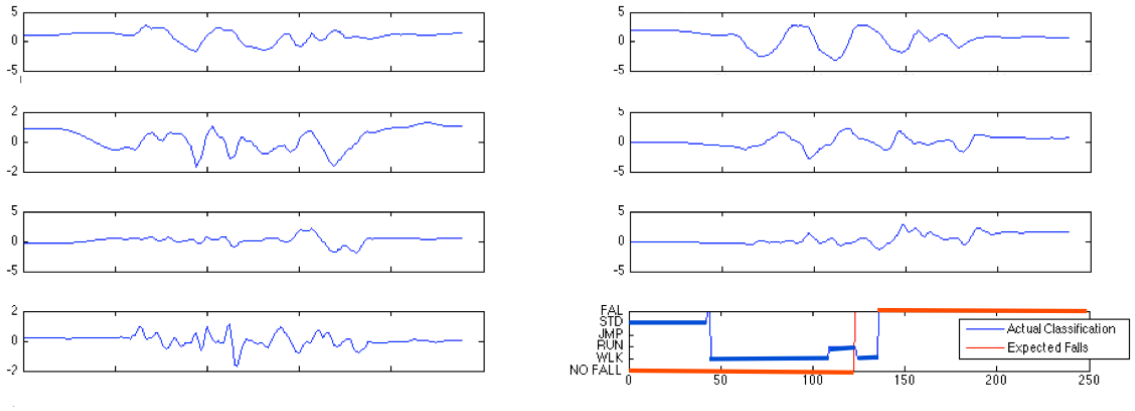


Figure 6.17: First seven principal values $w_1 \cdots w_7$ and detected mode for a FALL test segment (˜3.5s) using the CHMM based algorithm. The actual data was only labeled as "no fall" vs "falling" whereas the detection is for all modes and falls.

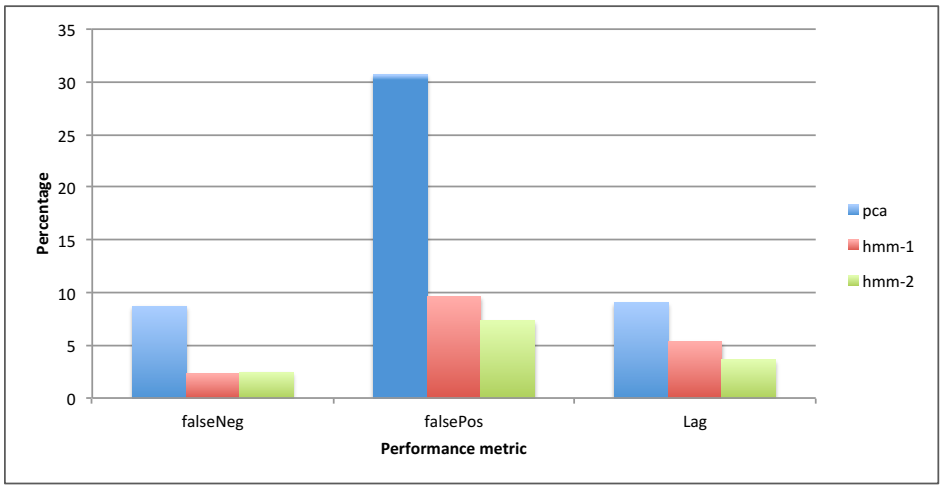
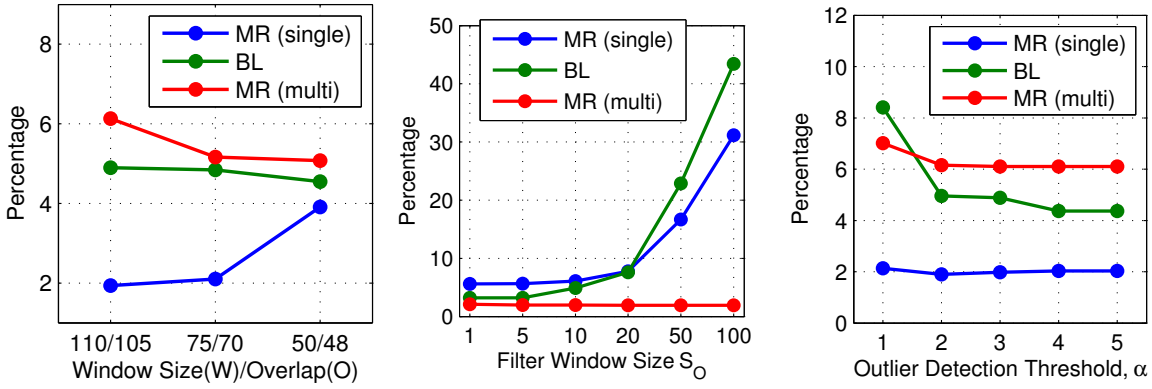


Figure 6.18: Comparison of FNR (falseNeg), FNR (falsePos) and boundary lag (lag) rate, expressed as percentage, across all test data for segments with fall activity for PCA-based algorithm (pca), simple Markov model based algorithm (hmm-1) and cascade Markov model based algorithm (hmm-2).



(a) (b) (c)

Figure 6.19: Impact of (a) data segmentation (W, O) (b) filter window size (S_O) and (c) outlier detection threshold (α) on the performance metrics misclassification rate (MR, single) for single activity detection , boundary detection lag (BL) and misclassification rate (MR, multi) for multi-activity detection on the GMM-based algorithm. All metrics are expressed as percentage. In (a) the choice of O is chosen to be the one that corresponds to maximum accuracy for the given W. Increasing O beyond a certain fraction of W increases complexity but not accuracy, hence W, O are not varied independently.

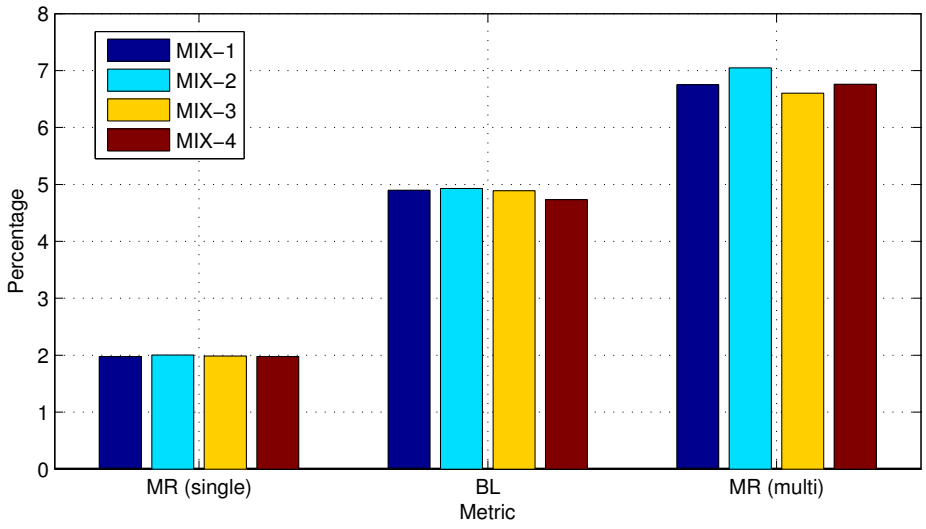
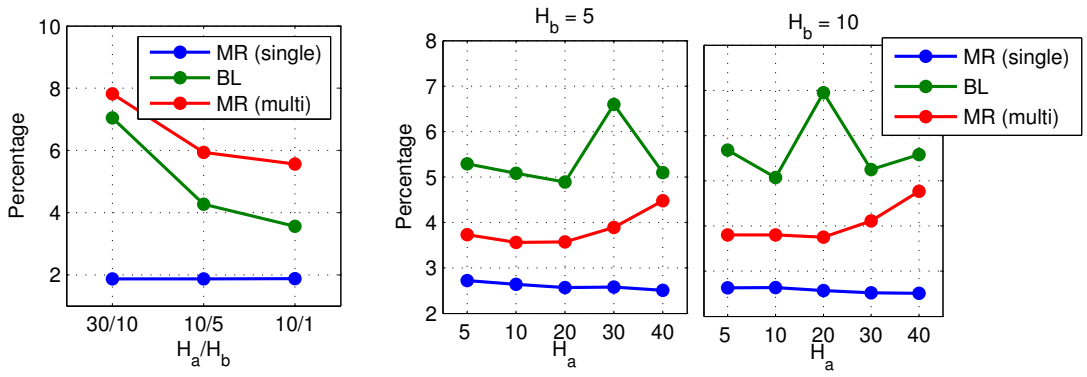


Figure 6.20: Impact of choice of GMM model type on the performance metrics misclassification rate (MR, single) for single activity detection , boundary detection lag (BL) and misclassification rate (MR, multi) for multi-activity detection on the PCA-based algorithm.



(a) (b)

Figure 6.21: Impact of the values of the lead and lag observation windows H_a, H_b on misclassification rate for single activity detection (MR, single), boundary detection lag (BL) and misclassification rate (MR, multi) for multi-activity detection. (a) For the simple Markov model, H_a, H_b impact Viterbi state sequence estimation. (b) For CHMM based algorithm, H_a, H_b impact output sequence likelihood computation.

6.6 Conclusions & Discussion

We have presented a prototype SmartSlipper and data-driven algorithms that are able to distinguish activity modes such as standing, running, jumping and walking with up to 98% accuracy based upon plantar pressure data, and detect falls with up to 97% accuracy by incorporating additional acceleration data. Our method achieves accuracy exceeding or comparable with current state-of-the-art activity and posture detection systems reported in [156] even with a rather small set of initial training data. Our technology has several distinctive advantages. Sensory data collection is via minimally invasive, generic, ready-to-wear, 'one size fits all' devices. Being completely data-driven, our algorithm is easily able to detect arbitrary activity modes using relevant feature sets from gait data, and does not depend on subject specific training data. Flexible segmentation and outlier detection policies allow for customization of the algorithm to specified application latency and robustness requirements. Our Markov model based implementation is able to capture dynamics within a gait cycle, which can be exploited to capture gait metrics such as cadence, stride length, step length, etc. Furthermore, the relatively low computational complexity of the algorithm makes is suitable for online implementation, and we have implemented and successfully tested this for real-time gait and fall detection.

Our first algorithm, that uses a PCA based feature reduction method to generate a GMM based Bayesian classifier, is a simple f*rame-based* classifier and performs amazingly well for slow changing activities and requires minimal training data. Using this we were able to achieve 98% accuracy in single activity test data, 94% accuracy in real-time and multi-activity test data and 90% fall detection accuracy though with a relatively high (30%) false positive rate. We implemented two *sequential* models, one based on a simple Markov model, that is able to better detect similar looking activities by making use of prior activity history, and achieves slightly improved accuracy. The second model, based upon a decoupled algorithm for cascade hidden Markov models that was developed in Chapter 5, Part A, is able to capture correlations in data much smaller than the gait cycle and is able to detect falls with up to 97% accuracy with less than 7% false positive rate and also able to delineate transitions between activities with smaller latency by modeling dynamics within each gait cycle. However, compared to the PCA based algorithm, it requires more training data with more parameters that need estimation. We have implemented embedded real-time verison of both algorithms and demonstrated real-time activity classification and fall detection on arbitrary subjects.

Several enhancements are possible to our algorithm. Spurious data rejection, such as one based upon relative class conditional likelihoods, could yield better robustness. Additional preprocessing such as AR on the time-domain signal prior to segmentation can also help alleviate noise (such approaches has been used for EEG or speech data, see [10, 63]). A a variable window/overlap for data segmentation such as one that maximizes the classification confidence could be used to provide optimal detection performance of both fast changing (multi-mode/falling) activities as well as static (single-mode) activity. Other variants such as threshold-based clustering or binary-tree based classification (as in [9]) can be concurrently used to improve classification accuracy. A variant of the algorithm that simultaneously diagonalizes the inter-class and intra-class covariance matrices when computing the principal component transform (the "Webb/Fukunaga" method) has been also known to peform better than the more traditional PCA method. Future revisions of the algorithm may add features such as cumulative sums to approximate distance/velocity in addition to the currently used pressure and acceleration inputs. Extension of our CHMM to multi-factor, factorial or decision tree HMMs would allow one to model complex gait dynamics thus allow us to detect finer gait modes or even distinguish different kinds of falls.

Chapter 7

Conclusion

This dissertation demonstrated the use of two techniques for near real-time biological signal processing for the purposes of physiological state estimation and classification in high dimensional systems: state *decomposition* and state *factorization.* Algorithms developed based upon the above techniques were applied to real problems in health care, medicine and public safety. In this final chapter, we will review the main research contributions of our work and its application scope, as well as discuss limitations of the current work and possible directions for future research.

We started out with the problem of predicting attention lapses and fatigue for an individual based upon their sleep history, circadian phase and amount of time awake; this is an unsolved problem in sleep medicine. A naive application of EEG spectral data as augmented input did not improve predictions any further than is possible with use of physiology-inspired models of fatigue (such as the Two Process Model). Specifically, additional EEG measurements make no statistical difference in the mean squared error of a linear prediction model based on an individual's data set. EEG data that was used for this analysis was previously cleaned of artifacts manually, where 2s epochs containing artifacts were completely removed. The *variable* number of artifacts from trial to trial causes a variable number of epochs when the above manual cleaning procedure is used; this makes it difficult to compare individual trials using statistical estimates that use ensemble averaging. Changes in EEG from fatigue can be rather subtle and easily masked by the effects of a variable number of epochs. This motivated us to develop new techniques for artifact identification and subtraction that would not require epochs to be removed since existing artifact elimination methods that preserve epochs often result in data corruption. Based upon the observation that artifact extraction

can be considered a problem of state space decomposition, we developed a structured sparse recovery method called Correlated Sparse Signal Recovery (CSSR) that outperforms existing structured sparse recovery methods in its ability to denoise EEG data with minimal corruption. We demonstrated and applied this to real EEG data to identify and eliminate blink artifacts, which comprise about 70-90% of all artifacts in our wake EEG recordings. To further eliminate remaining artifacts, we developed an ensemble learning technique that makes use of a new active learning algorithm called Output-based Active Selection (OAS). We were able to boost the artifact detection accuracy from approximately 91% to 97.5% using this algorithm.

A second set of problems where the state space can be factored rather than decomposed was addressed in the second half of the thesis. A framework for studying the class of problems where the dynamics of a finite state Markov chain are dependent on an external stochastic process was developed. We then derived lower dimensional (decoupled) solutions to the problems of hidden state estimation and optimal control on these so called Cascade Markov processes. Using one of these algorithms, we were able to detect gait activity with 99% accuracy and falls in real-time with up to 97% accuracy.

Some research contributions of this dissertation include the following:

1. A new structured sparse recovery algorithm, Correlated Sparse Signal Recovery (CSSR), that can model *statistical* rather than *fixed* structure, without the assumption of a common sparsity profile, was developed and tested (Chapter 2). The algorithm uses a Bayesian framework in which structure is modeled using a prior correlation matrix that represents statistical structure amongst coefficients. The algorithm was shown to successfully identify and remove eye blink artifacts in actual EEG recordings in near real-time. The approach of sparse recovery for artifact identification and removal (i) allows preservation of the EEG instead of discarding data, (ii) permits fully on-line implementation, (iii) works with few (4-6) channels and (iv) requires no manual intervention. Our approach is thus superior to several popular artifact removal techniques used currently.

2. Output based active selection (OAS) is a new active learning paradigm that addresses several pitfalls of traditional methods (Chapter 4). In particular, OAS does not suffer from selection bias inherent in traditional uncertainty based sampling, and yet does not compromise computational simplicity. In OAS, selection is based upon predicted output of unlabeled samples in addition to the uncertainty in their classification. OAS

based Active Learning (AL) when applied to some examples of non-separable data demonstrated that OAS achieves the same level of classification accuracy as other state-of-the-art algorithms with fewer active labels. Furthermore, since OAS internally uses an ensemble learner, AL implementation based upon OAS is trivially adaptable to the task of ensemble clustering.

3. We developed a novel skew Gaussian dictionary that is able to model a myriad of eye blink artifact shapes using just 1-3 elements from the dictionary (Chapter 3). When used with the CSSR algorithm, this dictionary successfully matches eye blink artifacts in several EEG recordings that results in denoising of EEG with minimal distortion.

4. An iterative and adaptive AL algorithm that is able to automatically switch to passive selection when active selection is no longer beneficial was developed (Chapter 4). The algorithm achieves this by keeping track of the improvement in number of confident examples across iterations. This algorithm was implemented and tested using OAS based active selection but is applicable to any active selection scheme.

5. Efficient non-linear time series prediction algorithms (Chapter 1) that use an underlying model such as the Two-Process Model were developed. They were shown to predict attention lapses using very few baseline measurements on an individual level with accuracy that is better than existing prediction models for the same problem ([224, 184]). Our predictor was shown to be optimal in a minimum variance sense by comparing it to a non-linear Kalman filter.

6. A decoupled version of Baum-Welch algorithm for coupled Hidden Markov Models for the special case when the state is partially observable and quasi-stationary was implemented (Chapter 5). This algorithm, which has much lower computational complexity compared to the fully coupled version, was successfully applied to detection of gait activities and falls in real-time (Chapter 6).

7. Decoupled matrix differential equations as solutions to a variety of fully observable optimal control problems on a particular class of coupled Markov decision processes were developed (Chapter 5). This class of problems involves optimization of the expectation of the utility of a functional on two coupled processes where the transition rates are stochastic and depend on the other. Our solution requires solving a one-point instead of two-point boundary value problem. The fully coupled counterpart is more

computationally complex by a factor proportional to state space size.

Our exposition has been mainly algorithmic with demonstration on real life examples. Future work can develop theoretical and rigorous analysis of these algorithms. While our algorithms were developed with low computational complexity in mind, our implementations can be further refined for run-time optimization especially in mobile environments.

The prediction algorithms for attention lapses based upon baseline measurements on an individual level can be easily extended to use more sophisticated models such as the Three-Process Model ([112]). These algorithms can be also be implemented using particle filtering, scented or extended Kalman filtering and those based upon alternate prior parameter distributions such as lognormal. The algorithm outlined in Chapter 1 that incorporates augmented EEG spectral information in making predictions can be extended to use alternate measures such as EEG bispectrum or entropy.

Our CSSR algorithm (Chapters 2,3) is based upon expectation-maximization (E-M). While convergence to a local minimum is guaranteed from E-M, the convergence can be slow and one can be stuck in a local minimum. Alternate methods such as stochastic E-M or stochastic gradient based optimization can possibly be an alternate implementation. Our current approach requires the correlation matrix R to be provided a priori based upon heuristic physiological information. However, this choice can be difficult, and as shown in Section 2.5, counter-intuitive. However, since the CSSR algorithm is typically applied on streaming data such as real-time continuous EEG, the structure of R can be learned over multiple samples of data by, for example, modeling it as a sum of parameterized rank one matrices and then estimating these parameters using E-M over multiple epochs. An alternative would be to use a Markov model to better represent the relationships amongst coefficients. The Bayesian method we use facilitates incorporation of such dynamics into the prior model. Another possible extension would be use hierarchical priors on the parameters that are estimated in our algorithm, which can perhaps lead to a more efficient algorithm by requiring estimation of a fewer number of parameters. We demonstrated the algorithm using a skew Gaussian dictionary on eye blink artifacts, but extension to other structured artifacts such as ECG is straightforward. CSSR provides a generic way for recovering structural information from EEG which can to predict sleepiness, for example. A decomposition of the form $x = x_\alpha \oplus x_\beta \oplus x_\gamma ...$ where $x_\alpha, x_\beta, x_\gamma$ represent the alpha, beta and gamma EEG activity respectively, using structural correlations provides an alternative to standard spectral analysis based decomposition. Standard spectral analysis discards phase information and also relies

upon assumptions of stationarity which can often result in loss of pertinent information.

CSSR can be applied to structured sparse recovery in many other domains. Some examples include (i) Recognition of color images where there is correlation amongst color channels, (ii) fluorescence diffuse optimal tomography where anatomy dictates prior correlation (iii) MRI imaging, where prior anatomical knowledge can be used to determine a particular relationship structure amongst voxels (iv) network construction, where biological/social interactions motivate modeling of cliques via correlations, and (v) modeling of earthquakes.

OAS based AL was demonstrated on E-M based base learners, but OAS is a generic active selection paradigm, and hence applicable to a wider class of base learners such as margin-based (support vector machines) or decision trees. Our selection method can be combined with other selection methods such as density weighting and importance weighting to produce even better active selection. In our current implementation of the adaptive algorithms, we have fixed the values of the parameters such as α, β used by the algorithm at optimal values for a particular label fraction. A possible extension of the algorithm is to learn these parameters adaptively, which will presumably yield better label complexity and noise sensitivity characteristics. We applied our approach of using AL for ensemble clustering (for non-retrainable ensemble members) to the particular case of classification of epochs into artifactual or not, but this approach has more general applicability in medical decision making where several classifiers are available which not be re-trained. Determination of sleep stages is such an example, where one needs to learn a classification rule from several criteria, such as automated EEG and actigraphy data, and where having an expert to manually annotate records is expensive. Use of our algorithm can drastically reduce the cost of research projects where sleep scoring is an essential but expensive step. Our approach can be also applied to real-time automated medical diagnosis that needs to be made from various diagnostic sources and even for automatic processing of health insurance claims.

Decoupled learning methods for factored HMMs (Chapter 5) have applicability in medicine beyond gait and fall detection. For example (i) a coupled HMM for multi-channel EEG data can be used to decipher sleep stages, or (ii) a coupled model of respiration and EEG data can be used to detect sleep apnea. The methods implemented pertained to a single cascade level i.e. the process $z \otimes x$ where x depends on z but not vice-versa, but can be easily extended to multiple levels ,i.e. for a process of the form $z_1 \otimes z_2 \otimes \ ... \ \otimes z_K$ where z_k depends only on z_j for $j < k$.

Our particular implementation of the algorithm on cascade HMMs for gait and fall

detection (Chapter 6) can also be enhanced in several ways: (i) spurious data rejection, such as one based upon relative class conditional likelihoods, can yield better robustness, (ii) additional pre-processing such as AR on the time-domain signal prior to segmentation can help alleviate noise, and (iii) concurrent use of clustering or binary-tree based classification can improve classification accuracy.

Our solution for the problem of optimal control on coupled Markov chains obviates the "curse of dimensionality" inherent in HJB equations thereby facilitating real-time solution. The algorithm may have application to medicine, such as finding optimal schedules of light exposure for correction of circadian misalignment and optimal schedules for drug intervention in patients.

The thesis demonstrates new computationally simple methods in biological signal processing with applications to medicine and public safety.

Bibliography

[1] Naoki Abe and Hiroshi Mamitsuka. Query learning strategies using boosting and bagging. In *Proceedings of the 25th International Conference on Machine learning*, volume 388, pages 1–9, 1998.

[2] Yotam Abramson and Yoav Yoav Freund. Active learning for visual object recognition. Technical report, University of California, San Diego, 2003.

[3] Daniel Aeschbach, Jeffery R. Matthews, Teodor T. Postolache, Michael a. Jackson, Holly a. Giesen, and Thomas a. Wehr. Dynamics of the human EEG during prolonged wakefulness: Evidence for frequency-specific circadian and homeostatic influences. *Neuroscience Letters*, 239(2-3):121–124, jan 1997.

[4] Charu C Aggarwal and Philip S Yu. Active Learning : A Survey. In *Data Classification: Algorithms and Applications*, pages 571–605. Taylor & Francis Group, 2014.

[5] T Akerstedt and M Gillberg. Subjective and objective sleepiness in the active individual. *The International journal of neuroscience*, 52(1-2):29–37, 1990.

[6] T Akerstedt, L Torsvall, and M Gillberg. Sleepiness in shiftwork. A review with emphasis on continuous monitoring of EEG and EOG. *Chronobiology international*, 4(2):129–140, 1987.

[7] Torbjörn Akerstedt. Altered sleep/wake patterns and mental performance. *Physiology & behavior*, 90(2-3):209–18, feb 2007.

[8] Mehmet Akin, Muhammed B. Kurt, Necmettin Sezgin, and Muhittin Bayram. Estimating vigilance level by using EEG and EMG signals. *Neural Computing and Applications*, 17(3):227–236, 2008.

[9] Gazi N Ali, Pei-ju Chiang, Aravind K Mikkilineni, George T Chiu, Edward J Delp, Jan P Allebach, and West Lafayette. Application of Principal Components Analysis and Gaussian Mixture Models to Printer Identification. *International Conference on Digital Printing Technologies*, 20(0219893):301–305, jan 2004.

[10] A. Ö Argunah and Müjdat Çetin. AR-PCA-HMM approach for sensorimotor task classification in EEG-based brain-computer interfaces. In *Proceedings - International Conference on Pattern Recognition*, volume 7, pages 113–116. Ieee, jan 2010.

[11] P Artaud, S Planque, C Lavergne, H Cara, C Tarriere, B Gueguen, et al. An on-board system for detecting lapses of alertness in car driving. In *Proceedings: International Technical Conference on the Enhanced Safety of Vehicles*, volume 1995, pages 350–359. National Highway Traffic Safety Administration, 1995.

[12] Behtash Babadi and Emery N Brown. A review of multitaper spectral analysis. *IEEE transactions on bio-medical engineering*, 61(5):1555–64, 2014.

[13] Behtash Babadi, Scott M. McKinney, Vahid Tarokh, and Jeffrey M. Ellenbogen. DiBa: a data-driven Bayesian algorithm for sleep spindle detection. *IEEE transactions on bio-medical engineering*, 59(2):483–493, jan 2012.

[14] Guha Balakrishnan and Zeeshan Syed. Scalable personalized medicine with active learning. *Proceedings of the ACM international conference on Health informatics - IHI '10*, page 83, 2010.

[15] Maria-Florina Balcan and Ruth Urner. Active Learning–Modern Learning Theory. pages 1–6, 2015.

[16] Thomas J. Balkin, William J. Horrey, R. Curtis Graeber, Charles A. Czeisler, and David F. Dinges. The challenges and opportunities of technological approaches to fatigue management. *Accident Analysis & Prevention*, 43(2):565–572, 2011.

[17] S. Bamberg, a.Y. Benbasat, D.M. Scarborough, D.E. Krebs, and J.a. Paradiso. Gait Analysis Using a Shoe-Integrated Wireless Sensor System. *IEEE Transactions on Information Technology in Biomedicine*, 12(4):413–423, jan 2008.

[18] Richard G Baraniuk, Volkan Cevher, and Marco F Duarte. Model-Based Compressive Sensing. 56(4):1982–2001, 2010.

[19] Laura K. Barger, Najib T. Ayas, Brian E. Cade, John W. Cronin, Bernard Rosner, Frank E. Speizer, and Charles A. Czeisler. Impact of extended-duration shifts on medical errors, adverse events, and attentional failures. *PLoS Medicine*, 3(12):2440–2448, 2006.

[20] John Barnard, Robert McCulloch, and Xiao-Li Meng. Modeling Covariance Matrices in Terms of Standard Deviations and Correlations, With Application To Shrinkage. *Statistica Sinica*, 10:1281–1311, 2000.

[21] Q. Barthélemy, C. Gouy-Pailler, Y. Isaac, a. Souloumiac, a. Larue, and J. I. Mars. Multivariate temporal dictionary learning for EEG. *Journal of Neuroscience Methods*, 215:19–28, 2013.

[22] A Belyavin and N A Wright. Changes in electrical activity of the brain with vigilance. *Electroencephalography and clinical neurophysiology*, 66(2):137–144, 1987.

[23] Yoshua Bengio and Paolo Frasconi. An Input Output HMM Architecture. *Neural Information Processing Systems*, pages 427–434, 1995.

[24] Alina Beygelzimer, Sanjoy Dasgupta, and John Langford. Importance Weighted Active Learning. *Proceedings of the 26th Annual International Conference on Machine Learning ICML 09*, abs/0812.4(ii):1–8, 2008.

[25] Katharina Blatter and Christian Cajochen. Circadian rhythms in cognitive performance: methodological constraints, protocols, theoretical underpinnings. *Physiology & behavior*, 90(2-3):196–208, feb 2007.

[26] Stephane Bonnet and Pierre Jallon. Hidden Markov Models Applied Onto Gait Classification. *18th European Signal Processing Conference*, 2(2):929–933, jan 2010.

[27] Naiyana Boonnak, Suwatchai Kamonsantiroj, and Luepol Pipanmaekaporn. Wavelet Transform Enhancement for Drowsiness Classification in EEG Records Using Energy Coefficient Distribution and Neural Network. 5(4), 2015.

[28] A A Borbely. A two process model of sleep regulation. *Human Neurobiology*, 1(3):195–204, 1982.

[29] C V Bouten, K T Koekkoek, M Verduin, R Kodde, and J D Janssen. A triaxial accelerometer and portable data processing unit for the assessment of daily physical activity. *IEEE transactions on bio-medical engineering*, 44(3):136–147, jan 1997.

[30] Matthew Brand. Coupled hidden Markov models for modeling interacting processes. *Submitted to Neural Computation*, 405(405):1–28, jan 1996.

[31] Roger W. Brockett. Optimal control of observable continuous time markov chains. In *Conference on Decision and Control, Proceedings of the 2008*, pages 4269–4274, 2008.

[32] J Buckelmüller, HP Landolt, HH Stassen, and P Achermann. Trait-like individual differences in the human sleep electroencephalogram. *Neuroscience*, 7(10):e44439, jan 2006.

[33] Naoto Burioka, Masanori Miyata, Germaine Cornélissen, Franz Halberg, Takao Takeshima, Daniel T Kaplan, Hisashi Suyama, Masanori Endo, Yoshihiro Maegaki, Takashi Nomura, Yutaka Tomita, Kenji Nakashima, and Eiji Shimizu. Approximate entropy in the electroencephalogram during wake and sleep. *Clinical EEG and neuroscience : official journal of the EEG and Clinical Neuroscience Society (ENCS)*, 36(1):21–24, 2005.

[34] Deng Cai and Xiaofei He. Manifold adaptive experimental design for text categorization. *IEEE Transactions on Knowledge and Data Engineering*, 24(4):707–719, 2012.

[35] C Cajochen, S B Khalsa, J K Wyatt, C a Czeisler, and D J Dijk. EEG and ocular correlates of circadian melatonin phase and human performance decrements during sleep loss. *The American journal of physiology*, 277(3 Pt 2):R640–9, sep 1999.

[36] C Cajochen, JK Wyatt, CA Czeisler, and DJ Dijk. Separation of circadian and wake duration-dependent modulation of EEG activation during wakefulness. *Neuroscience*, 7(10):e44439, jan 2002.

[37] Christian Cajochen, Katharina Blatter, and Dieter Wallach. Circadian and sleep-wake dependent impact on neurobehavioral function. *Psychologica Belgica*, 44-1/2:59–80, 2004.

[38] Lei Cao, Jie Li, Yaoru Sun, Huaping Zhu, and Chungang Yan. EEG-based vigilance analysis by using fisher score and PCA algorithm. *Proceedings of the 2010 IEEE International Conference on Progress in Informatics and Computing, PIC 2010*, 1:175–179, 2010.

[39] R Cassani, TH Falk, and FJ Fraga. The effects of automated artifact removal algorithms on electroencephalography-based Alzheimer's disease diagnosis. *Frontiers in aging ...*, 7(10):e44439, jan 2014.

[40] NP Castellanos and VA Makarov. Recovering EEG brain signals: artifact suppression with wavelet enhanced independent component analysis. *Journal of neuroscience methods*, 7(10):e44439, jan 2006.

[41] Orfeu M. Buxton Charles A. Czeisler. The Human Circadian Timing System and Sleep-Wake Regulation. *PloS one*, 7(10):e44439, jan 2012.

[42] Lan-lan Chen, Yu Zhao, Jian Zhang, and Jun-zhong Zou. Automatic detection of alertness/drowsiness from physiological signals using wavelet-based nonlinear features and machine learning. *Expert Systems with Applications*, 42(21):7344–7355, 2015.

[43] Lisheng Chen, Erbo Zhao, Dahui Wang, Zhangang Han, Shouwen Zhang, and Cuiping Xu. Feature extraction of EEG signals from epilepsy patients based on Gabor Transform and EMD decomposition. *Proceedings - 2010 6th International Conference on Natural Computation, ICNC 2010*, 3(Icnc):1243–1247, 2010.

[44] M Chen, B Huang, and Y Xu. Intelligent shoes for abnormal gait detection. *Robotics and Automation, 2008. ...*, 7(10):e44439, jan 2008.

[45] HJ Chizeck. Fuzzy model identification for classification of gait events in paraplegics. *Fuzzy Systems, IEEE Transactions on*, 5(4):536–544, 1997.

[46] Ioanna Chouvarda, Christos Papadelis, Chrysoula Kourtidou-Papadeli, Panagiotis D Bamidis, Dimitris Koufogiannis, Evaggelos Bekiaris, and Nikos Maglaveras. Non-linear analysis for the sleepy drivers problem. *Studies in health technology and informatics*, 129(Pt 2):1294–1298, 2007.

[47] E. C.-P. Chua, S.-C. Yeo, I. T.-G. Lee, L.-C. Tan, P. Lau, S. S. Tan, I. Ho Mien, and J. J. Gooley. Individual differences in physiologic measures are stable across repeated exposures to total sleep deprivation. *Physiological Reports*, 2:e12129–e12129, 2014.

[48] E.C. Chua, G. McDarby, and C. Heneghan. Combined electrocardiogram and photoplethysmogram measurements as an indicator of objective sleepiness. *Physiological measurement*, 29(8):857–868, 2008.

[49] Daniel A Cohen, Wei Wang, James K Wyatt, Richard E Kronauer, Derk-Jan Dijk, Charles A Czeisler, and Elizabeth B Klerman. Uncovering residual effects of chronic sleep loss on human performance. *Science translational medicine*, 2(14):14ra3, jan 2010.

[50] Alexander Colic, Oge Marques, and Borko Furht. *Driver Drowsiness Detection Systems and Solutions*. 2014.

[51] Facundo Costa, Hadj Batatia, Lotfi Chaari, and Jean-Yves Tourneret. Sparse EEG Source Localization using Bernoulli Laplacian Priors. *IEEE Transactions on Biomedical Engineering*, 9294(c):1–1, 2015.

[52] Madalena Costa, Ary L Goldberger, and C-K Peng. Multiscale entropy analysis of complex physiologic time series. *Physical review letters*, 89(6):068102, 2002.

[53] Evandro Silva Freire Coutinho, Katia Vergetti Bloch, and Claudia Medina Coeli. One-year mortality among elderly people after hospitalization due to fall-related fractures: comparison with a control group of matched elderly. *Cadernos De Saúde Pública*, 28:801–805, 2012.

[54] C. A. Czeisler and J. J. Gooley. Sleep and circadian rhythms in humans. *Cold Spring Harbor Symposia on Quantitative Biology*, 72(10):579–597, jan 2007.

[55] Charles A. Czeisler. Impact of sleepiness and sleep deficiency on public health - Utility of biomarkers. *Journal of Clinical Sleep Medicine*, 7(5), 2011.

[56] Charles a Czeisler. Impact of sleepiness and sleep deficiency on public health–utility of biomarkers. *Journal of clinical sleep medicine : JCSM : official publication of the American Academy of Sleep Medicine*, 7(5 Suppl):S6–8, oct 2011.

[57] Charles A Czeisler and Orfeu M Buxton. The Human Circadian Timing System and Sleep-Wake Regulation. In *Chronobiology*, page 401ff.

[58] Charles a Czeisler and Charles a Czeisler. Sleep Deficit: The Performance Killer. *Harvard Business Review*, (october), 2006.

[59] Charles A CA Czeisler. MEDICAL AND GENETIC DIFFERENCES IN THE ADVERSE IMPACT OF SLEEP LOSS ON PERFORMANCE : ETHICAL CONSIDERATIONS FOR THE MEDICAL PROFESSION. *Transactions of the American Clinical and Climatological Association*, 120:249–285, 2009.

[60] Sanjoy Dasgupta. Two faces of active learning. *Theoretical Computer Science*, 412(19):1767–1781, 2011.

[61] Sanjoy Dasgupta, Daniel Hsu, and Claire Monteleoni. A general agnostic active learning algorithm. Technical report, Department of CSE, University of California, San Diego, 2007.

[62] D Dawson and K Reid. Fatigue, alcohol and performance impairment. *Nature*, 388(6639):235, 1997.

[63] C.B. de Lima, A. Alcaim, and J.A. Apolinario. On the use of PCA in GMM and AR-vector models for text independent speaker verification. In *2002 14th International Conference on Digital Signal Processing Proceedings. DSP 2002 (Cat. No.02TH8628)*, volume 2, pages 595–598. IEEE, 2002.

[64] Dennis A Dean, Adam Fletcher, Steven R Hursh, and Elizabeth B Klerman. Developing mathematical models of neurobehavioral performance for the "real world". *Journal of biological rhythms*, 22(3):246–258, 2007.

[65] A Delorme and S Makeig. EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *Journal of neuroscience methods*, 7(10):e44439, jan 2004.

[66] A Delorme, T Sejnowski, and S Makeig. Enhanced detection of artifacts in EEG data using higher-order statistics and independent component analysis. *Neuroimage*, 7(10):e44439, jan 2007.

[67] Arnaud Delorme, Scott Makeig, and Tj Sejnowski. Automatic artifact rejection for EEG data using high-order statistics and independent component analysis. *International workshop on ICA*, 7(10):457–462, jan 2001.

[68] A.P. AP Dempster, N.M. NM Laird, and DB Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B Methodological*, 39(1):1–38, jan 1977.

[69] T.G. Dietterich. Ensemble Learning. 2007(April 2007):1–16, 2002.

[70] Thomas G. Dietterich. Ensemble Methods in Machine Learning. *Multiple Classifier Systems*, 1857:1–15, 2000.

[71] Thomas G. Dietterich. *Multiple classifier systems*, volume 1857. 2000.

[72] David F Dinges. An overview of sleepiness and accidents. *Journal of sleep research*, 4:4–14, 1995.

[73] David F. Dinges. Critical research issues in development of biomathematical models of fatigue and performance. *Aviation Space and Environmental Medicine*, 75(SUPPL.1), 2004.

[74] Nicolas Dobigeon, Alfred O Hero, Jean-yves Tourneret, and Senior Member. Bayesian Hierarchical Image Reconstruction With Application to MRFM. *Image (Rochester, N.Y.)*, 18(9):2059–2070, 2009.

[75] Hans P A Van Dongen. Comparison of Mathematical Model Predictions to Experimental Data of Fatigue and Performance. *Aviation, Space, and Environmental Medicine*, 75(Supplement 1):A15–A36, 2004.

[76] D P Dos Santos and A.C.P.L.F. De Carvalho. Comparison of active learning strategies and proposal of a multiclass hypothesis space search. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8480 LNAI:618–629, 2014.

[77] Marco F. Duarte and Yonina C. Eldar. Structured compressed sensing: From theory to applications. *IEEE Transactions on Signal Processing*, 59(9):4053–4085, 2011.

[78] Jeanne F Duffy and Derk-Jan Dijk. Getting through to circadian oscillators: why use constant routines? *Journal of biological rhythms*, 17(1):4–13, 2002.

[79] Jeanne F Duffy, Kirsi-Marja Zitting, and Charles A Czeisler. The Case for Addressing Operator Fatigue. *Reviews of Human Factors and Ergonomics*, 10(1):29–78, jun 2015.

[80] Grandjean E. *Fitting the task to the man: an ergonomic approach*. Taylor and Francis, London, 3 edition, 1980.

[81] Ivo Erkens and Gary Garcia Molina. Artifact detection and correction in Neurofeedback and BCI applications. *Electronics*, 2008.

[82] CW Erwin. Al.(1973). In *Psychophysiologic indices of drowsiness. Detroit, Mich: International Automotive Engineering Congress*.

[83] Xiaoli Fan, Qianxiang Zhou, Zhongqi Liu, and Fang Xie. Electroencephalogram assessment of mental fatigue in visual search. *Bio-Medical Materials and Engineering*, 26(s1):S1455–S1463, 2015.

[84] Mohamed Farouk, Abdel Hady, and Friedhelm Schwenker. Combining Committee-Based Semi-Supervised Learning and Active. 25(July):681–698, 2010.

[85] Rosa L Figueroa, Qing Zeng-Treitler, Long H Ngo, Sergey Goryachev, and Eduardo P Wiechmann. Active learning for clinical text classification: is it better than random sampling? *Journal of the American Medical Informatics Association : JAMIA*, 19(5):809–16, 2012.

[86] Center for Disease Control and Prevention. Older adult falls. Technical report, 2013.

[87] The Royal Society for the Prevention of Accidents. Driver fatigue and road accidents: A literature review and position paper. Technical report, Birmingham, U.K, 2001.

[88] D. B. Forger, M. E. Jewett, and R. E. Kronauer. A Simpler Model of the Human Circadian Pacemaker. *Journal of Biological Rhythms*, 14(6):533–538, dec 1999.

[89] National Sleep Foundation. Facts and stats, 2012.

[90] Yoav Freund, H. Sebastian Seung, E. Shamir, and N. Tishby. Selective Sampling Using the Query by Committee Algorithm. *Machine Learning*, 168(1997):133–168, 1997.

[91] Susan M Friedman, Beatriz Munoz, Sheila K West, Gary S Ruben, and Linda P Fried. Falls and Fear of Falling : Which Comes First ? A Longitudinal Secondary Prevention. *Journal of the American Geriatrics Society*, 50(8):1329–1335, 2002.

[92] Guglielmo Frigo and Claudio Narduzzi. EEG Gradient Artifact Removal by Compressive Sensing and Taylor-Fourier Transform. pages 0–5, 2014.

[93] Traci L Galinsky, Roger R Rosa, Joel S Warm, and William N Dember. Psychophysical determinants of stress in sustained attention. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 35(4):603–614, 1993.

[94] Ravi Ganti and Ag Gray. Building bridges: Viewing active learning from the multi-armed bandit lens. *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence (UAI2013)*, 2013.

[95] Ravi Ganti and Alexander Gray. UPAL: Unbiased pool based active learning. *Aistats*, (422):1–20, 2012.

[96] Jing Gao, Wei Fan, and Jiawei Han. On the Power of Ensemble : Supervised and Unsupervised Methods Reconciled - An overview of ensemble methods. *Sdm2010*, 2010.

[97] Agustina Garcés Correa, Lorena Orosco, and Eric Laciar. Automatic detection of drowsiness in EEG records based on multimodal analysis. *Medical Engineering and Physics*, 36(2):244–249, 2014.

[98] Z Ghahramani. Factorial hidden Markov models. *Advances in Neural Information Processing Systems*, 8:472 – 478, 1996.

[99] Namni Goel, Mathias Basner, Hengyi Rao, and David F Dinges. *Circadian rhythms, sleep deprivation, and human performance.*, volume 119. Elsevier Inc., 1 edition, jan 2013.

[100] J Gough, V P Belavkin, and O G Smolyanov. Hamiltonjacobibellman equations for quantum optimal feedback control. *Journal of Optics B: Quantum and Semiclassical Optics*, 7(10):S237, 2005.

[101] Stanley Gudder. Quantum Markov chains. *JOURNAL OF MATHEMATICAL PHYSICS*, 49(7), JUL 2008.

[102] a. Gundel, K. Marsalek, and C. ten Thoren. A critical review of existing mathematical models for alertness. *Somnologie - Schlafforschung und Schlafmedizin*, 11(3):148–156, jul 2007.

[103] Y. Guo and R. Greiner. Optimistic active learning using mutual information. In *Proceedings of International Joint Conference in Artificial Intelligence (IJ-CAI)*, pages 823–829. AAAI Press, 2007.

[104] Nikita Gurudath and H. Bryan Riley. Drowsy Driving Detection by EEG Analysis Using Wavelet Transform and K-means Clustering. *Procedia Computer Science*, 34:400–409, 2014.

[105] Steve Hanneke. Rates of Convergence in Active Learning. *The Annals of Statistics*, 39(1):333–361, 2011.

[106] William Harris. Fatigue, circadian rhythm, and truck accidents. In *Vigilance*, pages 133–146. 1977.

[107] S C H Hoi, Rong Jin, Jianke Zhu, and M R Lyu. Semi-supervised SVM batch mode active learning for image retrieval. *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–7, 2008.

[108] Shuyan Hu and Gangtie Zheng. Driver drowsiness detection with eyelid related parameters by Support Vector Machine. *Expert Systems with Applications*, 36(4):7651–7658, 2009.

[109] Junzhou Huang and Tong Zhang. The Benefit of Group Sparsity. *Statistics*, pages 1–22.

[110] Sheng-jun Huang, Rong Jin, and Zhi-hua Zhou. Active Learning by Querying Informative and Representative Examples. *Advances in Neural Information Processing Systems 23*, 36(10):892–900, 2010.

[111] Raul Igual, Carlos Medrano, and Inmaculada Plaza. Challenges, issues and trends in fall detection systems. *Biomedical engineering online*, 12(1):66, 2013.

[112] Michael Ingre, Wessel Van Leeuwen, Tomas Klemets, Christer Ullvetter, Stephen Hough, Göran Kecklund, David Karlsson, and Torbjörn Åkerstedt. Validating and extending the three process model of alertness in airline operations. *PloS one*, 9(10):e108679, jan 2014.

[113] Budi Thomas Jap, Sara Lal, Peter Fischer, and Evangelos Bekiaris. Using EEG spectral components to assess algorithms for detecting fatigue. *Expert Systems with Applications*, 36(2 PART 1):2352–2359, 2009.

[114] M. E. Jewett and R. E. Kronauer. Interactive Mathematical Models of Subjective Alertness and Cognitive Throughput in Humans. *Journal of Biological Rhythms*, 14(6):588–597, dec 1999.

[115] Shihao Ji, Ya Xue, and Lawrence Carin. Bayesian compressive sensing. *IEEE Transactions on Signal Processing*, 56(6):2346–2356, 2008.

[116] M Jordan and J Kleinberg. *Bishop - Pattern Recognition and Machine Learning*.

[117] Michael I Jordan, Zoubin Ghahramani, and Lawrence K Saul. Hidden Markov Decision Trees. *Advances in Neural Information Processing Systems*, 9:501–507, 1997.

[118] Carrie a. Joyce, Irina F. Gorodnitsky, and Marta Kutas. Automatic removal of eye movement and blink artifacts from EEG data using blind component separation. *Psychophysiology*, 41(2):313–325, 2004.

[119] Tzyy-Ping Jung and S Makeig. Estimating Alertness from the EEG Power Spectrum. *IEEE Transactions on Biomedical Engineering*, 9294(97), 1997.

[120] Tzyy-Ping P Jung, Scott Makeig, Colin Humphries, Te-Won W Lee, Martin J McKeown, Vicente Iragui, and Terrence J Sejnowski. Removing electroencephalographic artifacts by blind source separation. Technical Report 2, jan 2000.

[121] Matti Kääriäinen. Active Learning in the Non-realizable Case. *Alt*, pages 63–77, 2006.

[122] Kosuke Kaida, Masaya Takahashi, Torbjörn Akerstedt, Akinori Nakata, Yasumasa Otsuka, Takashi Haratani, and Kenji Fukasawa. Validation of the Karolinska sleepiness scale against performance and EEG variables. *Clinical neurophysiology : official journal of the International Federation of Clinical Neurophysiology*, 117(7):1574–81, jul 2006.

[123] N. Kannathal, U. Rajendra Acharya, C. M. Lim, and P. K. Sadasivan. Characterization of EEG - A comparative study. *Computer Methods and Programs in Biomedicine*, 80(1):17–23, 2005.

[124] Priyanka Khatwani and Archana Tiwari. A survey on different noise removal techniques of EEG signals. *Interntional Journal of Advanced Research in Computer and Communication Engineering (IJARCCE)*, 2(2):1091–1095, jan 2013.

[125] Rami N Khushaba, Sarath Kodagoda, Sara Lal, and Gamini Dissanayake. Driver Drowsiness Classification Using Fuzzy Wavelet-Packet-Based Feature-Extraction Algorithm. *Ieee Transactions on Biomedical Engineering*, 58(1):121–131, 2011.

[126] K Kiani, CJ Snijders, and ES Gelsema. Computerized analysis of daily life motor activity for ambulatory monitoring. *Technology and Health Care*, 7(10):e44439, jan 1997.

[127] Elizabeth B EB Klerman, Melissa St Hilaire, and MS Hilaire. Review: On mathematical modeling of circadian rhythms, performance, and alertness. *Journal of Biological Rhythms*, 7(10):e44439, jan 2007.

[128] Christine Korner and Stefan Wrobel. Multi-class Ensemble-Based Active Learning. *International Conference on Machine Learning*, 4212:687–694, 2006.

[129] Jan Kremer, Kim Steenstrup Pedersen, and Christian Igel. Active learning with support vector machines. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(4):313–326, 2014.

[130] Richard E Kronauer, Glenn Gunzelmann, Hans P A Van Dongen, Francis J Doyle, and Elizabeth B Klerman. Uncovering physiologic mechanisms of circadian rhythms and sleep/wake regulation through mathematical modeling. *Journal of biological rhythms*, 22(3):233–245, 2007.

[131] L.I. Kuncheva and C.J. Whitaker. Ten measures of diversity in classifier ensembles: Limits for two classifiers. *Proc. Inst. Elect. Eng. Workshop Intell. Sens. Process.*, pages 10–1–10–6, 2001.

[132] Ludmila I. Kuncheva. Using measures of similarity and inclusion for multiple classifier fusion by decision templates. *Fuzzy Sets and Systems*, 122(3):401–407, 2001.

[133] Ludmila I. Kuncheva. Using diversity measures for generating error-correcting output codes in classifier ensembles. *Pattern Recognition Letters*, 26(1):83–90, 2005.

[134] Muhammed B. Kurt, Necmettin Sezgin, Mehmet Akin, Gokhan Kirbas, and Muhittin Bayram. The ANN-based computing of drowsy level. *Expert Systems with Applications*, 36(2 PART 1):2534–2542, 2009.

[135] Rafa Kuś, Piotr Tadeusz Róaski, and Piotr Jerzy Durka. Multivariate matching pursuit in optimal Gabor dictionaries: theory and software with interface for EEG/MEG via Svarog. *Biomedical engineering online*, 12(1):94, jan 2013.

[136] Prakash Lakshmi, Prasad. Survey on EEG Signal Processing Methods. *International Journal of Advanced Research in Computer Science and Software Engineering*, 4(1), jan 2014.

[137] Saroj K L Lal and Ashley Craig. A critical review of the psychophysiology of driver fatigue. *Biological Psychology*, 55(3):173–194, 2001.

[138] Saroj K L Lal, Ashley Craig, Peter Boord, Les Kirkup, and Hung Nguyen. Development of an algorithm for an EEG-based driver fatigue countermeasure. *Journal of Safety Research*, 34(3):321–328, 2003.

[139] Vernon Lawhern, David Slayback, Dongrui Wu, Senior Member, Brent J Lance, and Senior Member. Efficient Labeling of EEG Signal Artifacts using Active Learning. In *Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on*, pages 3217–3222, oct 2015.

[140] Boon-Giin Lee, Boon-Leng Lee, and Wan-Young Chung. Mobile Healthcare for Automatic Driving Sleep-Onset Detection Using Wavelet-Based EEG and Respiration Signals. *Sensors*, 14(10):17915–17936, 2014.

[141] Rachel Leproult, Egidio F Colecchia, Anna Maria Berardi, Robert Stickgold, Stephen M Kosslyn, and Eve Van Cauter. Individual differences in subjective and objective alertness during sleep deprivation are stable and unrelated. Technical Report 2, 2003.

[142] Kun Li. *Advanced Signal Processing Techniques for Single Trial Electroencephalography Signal Classification for Brain Computer Interface Applications*. PhD thesis, 2010.

[143] Wei Li, Qi Chang He, Xiu Min Fan, and Zhi Min Fei. Evaluation of driver fatigue on two channels of EEG data. *Neuroscience Letters*, 506(2):235–239, 2012.

[144] Yaliang Li, Jing Gao, Qi Li, and Wei Fan. Ensemble Learning. In *Data Classification: Algorithms and Applications*, chapter 19, pages 483–504. Taylor & Francis Group, 2015.

[145] S F Liang, C T Lin, R C Wu, Y C Chen, T Y Huang, and T P Jung. Monitoring driver's alertness based on the driving performance estimation and the EEG power spectrum analysis. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society.*, volume 6, pages 5738–41, jan 2005.

[146] Chin Teng Lin, Che Jui Chang, Bor Shyh Lin, Shao Hang Hung, Chih Feng Chao, and I. Jan Wang. A real-time wireless brain-computer interface system for drowsiness detection. *IEEE Transactions on Biomedical Circuits and Systems*, 4(4):214–222, 2010.

[147] Chin-teng Lin, Yu-Chieh Chen, Teng-Yi Huang, Tien-Ting Chiu, Li-Wei Ko, Sheng-Fu Liang, Hung-Yi Hsieh, Shang-Hwa Hsu, and Jeng-Ren Duann. Development of Wireless Brain Computer Interface With Embedded Multitask Scheduling and Its Application on Real-Time Driver's Drowsiness Detection and Warning. *IEEE Transactions on Biomedical Engineering*, 55(5):1582–, 2008.

[148] Chin Teng Lin, Li W. Ko, I. Fang Chung, Teng Y. Huang, Yu Chieh Chen, Tzyy Ping Jung, and Sheng F. Liang. Adaptive EEG-based alertness estimation system by using ICA-based fuzzy neural networks. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 53(11):2469–2476, 2006.

[149] Chin-teng Lin, Ruei-cheng Wu, Sheng-fu Liang, Wen-hung Chao, Yu-jie Chen, and Tzyy-ping Jung. EEG-based drowsiness estimation for safety driving using independent component analysis. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 52(12):2726–2738, dec 2005.

[150] Fu Chang Lin, Li Wei Ko, Chun Hsiang Chuang, Tung Ping Su, and Chin Teng Lin. Generalized EEG-based drowsiness prediction system by using a self-organizing neural fuzzy system. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 59(9):2044–2055, 2012.

[151] Jianping Liu, Chong Zhang, and Chongxun Zheng. EEG-based estimation of mental fatigue by using KPCA-HMM and complexity parameters. *Biomedical Signal Processing and Control*, 5(2):124–130, 2010.

[152] Benny Lo, Julien Pansiot, and Guang Zhong Yang. Bayesian analysis of sub-plantar ground reaction force with BSN. *Proceedings - 2009 6th International Workshop on Wearable and Implantable Body Sensor Networks, BSN 2009*, 7(10):133–137, jan 2009.

[153] Steven W Lockley, Laura K Barger, Najib T Ayas, Jeffrey M Rothschild, Charles A Czeisler, and Christopher P Landrigan. Effects of health care provider work hours and sleep deprivation on safety and performance. *Joint Commission journal on quality and patient safety / Joint Commission Resources*, 33(11 Suppl):7–18, 2007.

[154] Steven W Lockley, Christopher P Landrigan, Laura K Barger, and Charles A Czeisler. When policy meets physiology: the challenge of reducing resident work hours. *Clinical orthopaedics and related research*, 449:116–127, 2006.

[155] Stephen R Lord, Catherine Sherrington, Hylton B Menz, and Jacqueline C T Close. *Falls in older people: risk factors and strategies for prevention*. Cambridge University Press, 2007.

[156] A Mannini and AM Sabatini. Machine learning methods for classifying human physical activity from on-body accelerometers. *Sensors*, 7(10):e44439, jan 2010.

[157] Zahra Mardi, Seyedeh Naghmeh Miri Ashtiani, and Mohammad Mikaili. EEG-based Drowsiness Detection for Safe Driving Using Chaotic Features and Statistical Tests. *Journal of medical signals and sensors*, 1(2):130–7, 2011.

[158] Andrew Kachites McCallum and Kamal Nigam. Employing EM and pool-based active learning for text classification. *Learning*, pages 350–358, 1998.

[159] Peter McCauley, Leonid V Kalachev, Daniel J Mollicone, Siobhan Banks, David F Dinges, and Hans P a Van Dongen. Dynamic circadian modulation in a biomathematical model for the effects of sleep and sleep loss on waking neurobehavioral performance. *Sleep*, 36(12):1987–97, 2013.

[160] R.A. McMurray. Safety recommendation. 2009. Technical report.

[161] Marina Meila and Michael I Jordan. Learning Fine Motion by Markov Mixtures of Experts. *Advances in Neural Information Processing Systems 8*, (1567):1003–1009, 1996.

[162] Aleksandar Milenković, Chris Otto, and Emil Jovanov. Wireless sensor networks for personal health monitoring: Issues and an implementation. *Computer Communications*, 29(13-14):2521–2533, 2006.

[163] Mir Mohsina and Angshul Majumdar. Gabor based analysis prior formulation for EEG signal reconstruction. *Biomedical Signal Processing and Control*, 8(6):951–955, 2013.

[164] Leslie D. Montgomery, Richard W. Montgomery, and Raul Guisado. Rheoencephalographic and electroencephalographic measures of cognitive workload: Analytical procedures. *Biological Psychology*, 40(1-2):143–159, 1995.

[165] Jose Miguel Morales, Leandro Luigi Di Stasi, and Samuel Romero. Real-time Monitoring of Biomedical SIgnals to Improve Road Safety. In *IWANN 2015, LNCS 9094*, volume 6692, pages 89–97. 2015.

[166] Taketoshi Mori, Yu Nejigane, Masamichi Shimosaka, Yushi Segawa, Tatsuya Harada, and Tomomasa Sato. Online recognition and segmentation for time-series motion with HMM and conceptual relation of actions. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, volume 7, pages 2568–2574. Ieee, jan 2005.

[167] SW Muir, M Speechley, J Wells, and M Borrie. Gait assessment in mild cognitive impairment and Alzheimer's disease: The effect of dual-task challenges across the cognitive spectrum. *Gait & posture*, 7(10):e44439, jan 2012.

[168] Ion Muslea, Steven Minton, and Craig a Knoblock. Active + Semi-Supervised Learning = Robust Multi-View Learning. *Proceedings of the Nineteenth International Conference on Machine Learning*, (1998):435–442, 2002.

[169] Kamal Nigam, Andrew McCallum, Sebastian Thrun, and Tom Mitchell. Text Classification from Labeled and Unlabeled Documents using {EM}. *Machine Learning*, 39(2-3):103–134, 2000.

[170] James F. O'Hanlon and Gene R. Kelley. Comparison of Performance and Physiological Changes Between Drivers who Perform Well and Poorly During Prolonged Vehicular Operation. In *NATO Conference Series, Volume 3: Vigilance*, pages 87–109. 1977.

[171] Fredrik Olsson. A literature survey of active machine learning in the context of natural language processing. *Swedish Institute of Computer Science*, pages 134–138, 2009.

[172] World Health Organization. Who global report on falls prevention in older age. Technical report, 2007.

[173] Pamela L. Owens, C. Allison Russo, William Spector, and Ryan Mutter. Emergency department visits for injurious falls among the elderly, 2006. Technical report, 2009.

[174] Nikhil R Pal, Chien-Yao Chuang, Li-Wei Ko, Chih-Feng Chao, Tzyy-Ping Jung, Sheng-Fu Liang, and Chin-Teng Lin. EEG-Based Subject- and Session-independent Drowsiness Detection: An Unsupervised Approach. *EURASIP Journal on Advances in Signal Processing*, 2008(1):519480, 2008.

[175] Christos Papadelis, Zhe Chen, Chrysoula Kourtidou-Papadeli, Panagiotis D. Bamidis, Ioanna Chouvarda, Evangelos Bekiaris, and Nikos Maglaveras. Monitoring sleepiness with on-board electrophysiological recordings for preventing sleep-deprived traffic accidents. *Clinical Neurophysiology*, 118(9):1906–1922, 2007.

[176] Christos Papadelis, Chrysoula Kourtidou-Papadeli, Panagiotis D. Bamidis, Ioanna Chouvarda, D. Koufogiannis, E. Bekiaris, and Nikos Maglaveras. Indicators of Sleepiness in an ambulatory EEG study of night driving. *Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings*, (0):6201–6204, 2006.

[177] Amiya Patanaik, Vitali Zagorodnov, Chee Keong Kwoh, and Michael W L Chee. Predicting vulnerability to sleep deprivation using diffusion model parameters. *Journal of sleep research*, pages 1–9, 2014.

[178] Ingrid Philibert. Sleep loss and performance in residents and nonphysicians: a meta-analytic examination. *Sleep*, 28(11):1392–1402, 2005.

[179] Brigitte Plateau and Karim Atif. Statistical Automata Network For Modeling Parallel Systems. 17(10), 1991.

[180] Robi Polikar. Ensemble based systems in decision making. *Circuits and Systems Magazine*, pages 21–45, 2006.

[181] R. Quian Quiroga, S. Blanco, O.A. Rosso, H. Garcia, and A. Rabinowicz. Searching for hidden information with Gabor Transform in generalized tonic-clonic seizures. *Electroencephalography and Clinical Neurophysiology*, 103(4):434–439, 1997.

[182] Muhannad Quwaider and Subir Biswas. Body posture identification using hidden Markov model with a wearable sensor network. *Proceedings of the ICST 3rd international conference on Body area networks*, pages 19:1—-19:8, jan 2008.

[183] S. Rajaraman, A. V. Gribok, N. J. Wesensten, T. J. Balkin, and J. Reifman. Individualized performance prediction of sleep-deprived individuals with the two-process model. *Journal of Applied Physiology*, 104(2):459–468, jan 2007.

[184] Srinivasan Rajaraman, Andrei V Gribok, Nancy J Wesensten, Thomas J Balkin, and Jaques Reifman. An improved methodology for individualized performance prediction of sleep-deprived individuals with the two-process model. *Sleep*, 32(10):1377–1392, 2009.

[185] Thomas G. Raslear, Steven R. Hursh, and Hans P a Van Dongen. *Predicting cognitive impairment and accident risk*, volume 190. Elsevier B.V., 1 edition, 2011.

[186] Paul Stephen Rau and (NHTSA). Drowsy Driver Detection and Warning System for Commercial Vehicle Drivers : Field Operational Test Design , Data Analyses , and Progress . Technical report, National Highway Traffic Safety Administration, 1996.

[187] R REILLY and H NOLAN. FASTER: Fully Automated Statistical Thresholding for EEG artifact Rejection. *Journal of neuroscience methods*, 192:152–162, jan 2010.

[188] Author Alexander Roederer. *Active Learning for Classification of Medical Signals*. PhD thesis, University of Pennsylvania, 2012.

[189] Lior Rokach. Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1-2):1–39, 2010.

[190] Dawn Royal, F Street, and N W Suite. National Survey of Distracted and Drowsy Driving Attitudes and Behavior : 2002. Technical report, National Highway Traffic Safety Administration, 2002.

[191] Laurence Z. Rubenstein and Karen R. Josephson. The epidemiology of falls and syncope. *Clinics in Geriatric Medicine*, 18(2):141–158, 2002.

[192] Arun Sahayadhas, Kenneth Sundaraj, and Murugappan Murugappan. Detecting driver drowsiness based on sensors: A review. *Sensors (Switzerland)*, 12(12):16937–16953, 2012.

[193] Joan Santamaria and Keith H Chiappa. The eeg of drowsiness in normal adults. *Journal of clinical Neurophysiology*, 4(4):327–382, 1987.

[194] Lawrence K Saul. Mixed Memory Markov Models : Decomposing Complex Stochastic Processes as Mixtures of Simpler Ones. 87:75–87, 1999.

[195] Daniela Schachinger, Kaspar Schindler, and Tilmann Kluge. Automatic reduction of artifacts in EEG-signals. *2007 15th International Conference on Digital Signal Processing, DSP 2007*, 7(10):143–146, jan 2007.

[196] Andrew I. Schein and Lyle H. Ungar. Active learning for logistic regression: An evaluation. *Machine Learning*, 68(3):235–265, 2007.

[197] H V Semlitsch, P Anderer, P Schuster, and O Presslich. A solution for reliable and valid reduction of ocular artifacts, applied to the P300 ERP. *Psychophysiology*, 23(6):695–703, 1986.

[198] Burr Settles. Active Learning Literature Survey. *Machine Learning*, 15(2):201–221, 2010.

[199] Burr Settles. *Active Learning*. Number 1. Morgan and Claypool, 2012.

[200] Burr Settles and Mark Craven. An analysis of active learning strategies for sequence labeling tasks. *Proceedings of the Conference on Empirical Methods in Natural Language Processing EMNLP 08*, (October):1070, 2008.

[201] Kai Quan Shen, Xiao Ping Li, Chong Jin Ong, Shi Yun Shao, and Einar P V Wilder-Smith. EEG-based mental fatigue measurement using multi-class support vector machines with confidence estimate. *Clinical Neurophysiology*, 119(7):1524–1533, 2008.

[202] T Shi, D Tang, L Xu, and T Moscibroda. Correlated compressive sensing for networked data. *Uncertainty in Artificial Intelligence - Proceedings of the 30th Conference, UAI 2014*, pages 722–731, 2014.

[203] Michael Simon, Eike A. Schmidt, Wilhelm E. Kincses, Martin Fritzsche, Andreas Bruns, Claus Aufmuth, Martin Bogdan, Wolfgang Rosenstiel, and Michael Schrauf. Eeg alpha spindle measures as indicators of driver fatigue under real traffic conditions. *Clinical Neurophysiology*, 122(6):1168–1178, 2011.

[204] Anne C. Skeldon, Derk-Jan Dijk, and Gianne Derks. Mathematical Models for Sleep-Wake Dynamics: Comparison of the Two-Process Model and a Mutual Inhibition Neuronal Model. *PLoS ONE*, 9(8):e103877, 2014.

[205] Julie H Skipper and Walter W Wierwille. Drowsy driver detection using discriminant analysis. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 28(5):527–540, 1986.

[206] Christy Spangler and Alice Park. Loss of control on approach colgan air, inc., operating as continental connection flight 3407 bombardier dhc-8-400, n200wq clarence center, new york february 12, 2009. In *ACM SIGGRAPH 2010 Dailies*, SIGGRAPH '10, pages 7:1–7:1, New York, NY, USA, 2010. ACM.

[207] S Sprager and D Zazula. A cumulant-based method for gait identification using accelerometer data with principal component analysis and support vector machine. *WSEAS Transactions on Signal Processing*, 7(10):e44439, jan 2009.

[208] Claudio Stampi, Polly Stone, and Akihiro Michimori. A new quantitative method for assessing sleepiness: The alpha attenuation test. *Work & Stress*, 9(2-3):368–376, 1995.

[209] a Subasi. Automatic recognition of alertness level from EEG by using neural network and wavelet coefficients. *Expert Systems with Applications*, 28(4):701–711, may 2005.

[210] Shin'ichi Takeuchi, Satoshi Tamura, and Satoru Hayamizu. Human action recognition using acceleration information based on hidden markov model. In *Asia-Pacific Signal and Information Processing Association, 2009 Annual Summit and Conference.*, jan 2009.

[211] Brian C Tefft. Asleep at the wheel: the prevalence and impact of drowsy driving. *Technical Report, American Automobile Association Foundation for Traffic Safety*, 2010.

[212] M E Tinetti, W L Liu, and E B Claus. Predictors and prognosis of inability to get up after falls among elderly persons. *JAMA : the journal of the American Medical Association*, 269(1):65–70, 1993.

[213] Michael Tipping. Sparse Bayesian Learning and the Relevance Vector Mach. *Journal of Machine Learning Research*, 1:211–244, 2001.

[214] Katrin Tomanek and Udo Hahn. Semi-supervised active learning for sequence labeling. *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*, (August):1039–1047, 2009.

[215] Shanbao Tong and Nitish V. Thakor. *Quantitative EEG Analysis Methods and Clinical Applications*.

[216] Simon Tong. *Active learning: theory and applications*. PhD thesis, Stanford University, 2001.

[217] L Torsvall and T Akerstedt. Sleepiness on the job: continuously measured EEG changes in train drivers. *Electroencephalography and clinical neurophysiology*, 66(6):502–511, 1987.

[218] Gokhan Tur, Dilek Hakkani-Tür, and Robert E. Schapire. Combining active and semi-supervised learning for spoken language understanding. *Speech Communication*, 45(2):171–186, 2005.

[219] H P a Van Dongen, G Maislin, J M Mullington, and D F Dinges. The cumulative cost of additional wakefulness, 2003.

[220] Hans P A Van Dongen, Maurice D Baynard, Greg Maislin, and David F Dinges. Systematic interindividual differences in neurobehavioral impairment from sleep loss: evidence of trait-like differential vulnerability. Technical Report 3, 2004.

[221] Hans P a Van Dongen, Amy M. Bender, and David F. Dinges. Systematic individual differences in sleep homeostatic and circadian rhythm contributions to neurobehavioral impairment during sleep deprivation. *Accident Analysis and Prevention*, 45(SUPPL.):11–16, 2012.

[222] Hans P A Van Dongen, John A. Caldwell, and J. Lynn Caldwell. *Individual differences in cognitive vulnerability to fatigue in the laboratory and in the workplace*, volume 190. 2011.

[223] Hans P A Van Dongen, Greg Maislin, and David F. Dinges. Dealing with inter-individual differences in the temporal dynamics of fatigue and performance: Importance and techniques. *Aviation Space and Environmental Medicine*, 75(SUPPL.1), 2004.

[224] Hans P a Van Dongen, Christopher G Mott, Jen-Kuang Huang, Daniel J Mollicone, Frederic D McKenzie, and David F Dinges. Optimization of biomathematical model predictions for cognitive performance impairment in individuals: accounting for unknown traits and uncertain states in homeostatic and circadian processes. *Sleep*, 30(9):1129–1143, 2007.

[225] Hans P A Van Dongen, Kristen M Vitellaro, and David F Dinges. Individual differences in adult human sleep and wakefulness: Leitmotif for a research agenda. *Sleep*, 28(4):479–496, 2005.

[226] Harry L. Van Trees. *Detection, Estimation, and Modulation Theory, Part I*, volume 0. 2001.

[227] Joe Verghese, Richard B Lipton, Charles B Hall, Gail Kuslansky, Mindy J Katz, and Herman Buschke. Abnormality of gait as a predictor of non-Alzheimer's dementia. *The New England journal of medicine*, 347(22):1761–1768, 2002.

[228] M. R. Volow and C. W. Erwin. The heart rate variability correlates of spontaneous drowsiness onset. In *SAE Technical Paper Series*. SAE International, jan 1973.

[229] Aleksandra Vuckovic, Vlada Radivojevic, Andrew C N Chen, and Dejan Popovic. Automatic recognition of alertness and drowsiness from EEG by an artificial neural network. *Medical engineering & physics*, 24(5):349–60, jun 2002.

[230] Wei Wang. On Multi-View Active Learning and the Combination with Semi-Supervised Learning. pages 1152–1159, 2008.

[231] Yu Wang, David Wipf, Jeong-min Yun, Wei Chen, and Ian Wassell. Clustered Sparse Bayesian Learning. *Conference on Uncertainty*, 2015.

[232] Earl L Wiener. *Sustained attention in human performance*, chapter Vigilance and inspection, pages 207–246. John Wiley, New York, 1984.

[233] Wikipedia. March fracture.

[234] D.P. Wipf, D.P. Wipf, B.D. Rao, and B.D. Rao. Sparse bayesian learning for basis selection. *IEEE Trans. on Signal Processing, Special Issue on Machine Learning Methods in Signal Processing*, 52(8):2153–2164, 2004.

[235] Dongrui Wu, B. Lance, and V. Lawhern. Transfer learning and active transfer learning for reducing calibration data in single-trial classification of visually-evoked potentials. In *Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on*, pages 2801–2807, oct 2014.

[236] Guosheng Yang, Yingzi Lin, and Prabir Bhattacharya. A driver fatigue recognition model based on information fusion and dynamic Bayesian network. *Information Sciences*, 180(10):1942–1954, 2010.

[237] Mervyn V M Yeo, Xiaoping Li, Kaiquan Shen, and Einar P V Wilder-Smith. Can SVM be used for automatic EEG detection of drowsiness during car driving? *Safety Science*, 47(1):115–124, 2009.

[238] Xinyi Yong, Rabab K. Ward, and Gary E. Birch. Generalized Morphological Component Analysis for EEG source separation and artifact removal. *2009 4th International IEEE/EMBS Conference on Neural Engineering*, 7(10):343–346, jan 2009.

[239] J Yu. Fault detection using principal components-based Gaussian mixture model for semiconductor manufacturing processes. *Semiconductor Manufacturing, IEEE Transactions on*, 7(10):e44439, jan 2011.

[240] Kai Yu, Jinbo Bi, and Volker Tresp. Active learning via transductive experimental design. *Proceedings of the 23rd international conference on Machine learning ICML 06*, 148(6):1081–1088, 2006.

[241] L. Yu, H. Sun, J. P. Barbot, and G. Zheng. Bayesian compressive sensing for cluster structured sparse signals. *Signal Processing*, 92(1):259–269, 2012.

[242] Chi Zhang, Hong Wang, and Rongrong Fu. Automated detection of driver fatigue based on entropy and complexity measures. *IEEE Transactions on Intelligent Transportation Systems*, 15(1):168–177, 2014.

[243] Chong Zhang, Chong-Xun Zheng, and Xiao-Lin Yu. Automatic recognition of cognitive fatigue from physiological indices by using wavelet packet transform and kernel learning algorithms. *Expert Systems with Applications*, 36(3):4664–4671, 2009.

[244] Jiaxiang Zhang, Andrew Bierman, John T. Wen, Agung Julius, and Mariana Figueiro. Circadian system modeling and phase control. *49th IEEE Conference on Decision and Control (CDC)*, pages 6058–6063, dec 2010.

[245] Zhilin Zhang and Bhaskar D. Rao. Sparse Signal Recovery With Temporally Correlated Source Vectors Using Sparse Bayesian Learning. *IEEE Journal of Selected Topics in Signal Processing*, 5(5):912–926, 2011.

[246] Zhilin Zhang and Bhaskar D. Rao. Extension of SBL algorithms for the recovery of block sparse signals with intra-block correlation. *IEEE Transactions on Signal Processing*, 61(8):2009–2015, 2013.

[247] Chunlin Zhao, Chongxun Zheng, Min Zhao, Jianping Liu, and Yaling Tu. Automatic classification of driving mental fatigue with EEG by wavelet packet energy and KPCA-SVM. *International Journal of Innovative Computing, Information and Control*, 7(3):1157–1168, 2011.

[248] Liyue Zhao, G Sukthankar, and R Sukthankar. Importance-weighted label prediction for active learning with noisy annotations. *Pattern Recognition (ICPR), 2012 21st International Conference on*, (Icpr):3476–3479, 2012.

[249] Zhang Zhilin and B D Rao. Recovery of block sparse signals using the framework of block sparse Bayesian learning. *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 3345–3348, 2012.

[250] Shang Ming Zhou, John Q. Gan, and Francisco Sepulveda. Classifying mental tasks based on features of higher-order statistics from EEG signals in brain-computer interface. *Information Sciences*, 178(6):1629–1640, 2008.

[251] Zhi-Hua Zhou. *Ensemble Methods: Foundations and Algorithms*. 2012.

[252] Zhi-hua Zhou, Ke-jia Chen, and Hong-bin Dai. Enhancing Relevance Feedback in Image Retrieval Using Unlabeled Data. 24(2):219–244, 2006.

[253] Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. *Proceedings of the ICML-2003 Workshop on The Continuum from Labeled to Unlabeled Data*, 20(2):912–919, 2003.

[254] Y. Zou, V. Nathan, and R. Jafari. Automatic identification of artifact-related independent components for artifact removal in eeg recordings. *IEEE Journal of Biomedical and Health Informatics*, 20(1):73–81, Jan 2016.