Anomaly Detection and Sequential Filtering with Partial Observations

by

Elizabeth Mary Hou

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering and Computer Science)
in the University of Michigan
2019

Doctoral Committee:

Professor Alfred O. Hero, Chair
Assistant Professor Laura Balzano
Assistant Professor Danai Koutra
Dr. Earl Lawrence, Los Alamos National Laboratory

Elizabeth Mary Hou

emhou@umich.edu

ORCID iD: 0000-0002-8100-6206

A C K N O W L E D G M E N T S

This work was partially supported by the Consortium for Verification Technology under Department of Energy National Nuclear Security Administration award number DE-NA0002534 and partially by the University of Michigan ECE Departmental Fellowship.

Preface

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

ABSTRACT

Anomaly Detection and Sequential Filtering with Partial Observations

by

Elizabeth Hou

Chair: Alfred O. Hero III

With the rise of "big data" where any and all data is collected, comes a series of new challenges involving the computation and analysis of such massive data sets. Nowadays, data is continuously collected leading to questions of at which point should analysis begin and how to incorporate new data into the analysis. And, within the massive amounts of data collected, there can be other complications in addition to the noise. The features of interest may not be directly observable to a user, and thus are modeled as latent variables. There may be only a very small subset of the data with certain properties that are of interest to the user. Or, there could be data that is only partially labeled due to the costs of user labeled data or simply a lack of information.

In this thesis, we develop methods that deal with data containing partial labels, latent variables, and anomalies. Many of the models in our frameworks are extendable to an online or streaming scenario where the data is continuously being collected and discarded. We also illustrate some real world applications of our proposed models using datasets from cyber security, transportation, and weather systems.

The contributions of this thesis are that we have developed:

1. Penalized ensemble Kalman filter that is designed for superior performance in non-linear high dimensional systems.

2. Framework to generate and update regression and classification models, which can be used to build an optimal non-linear filter and also an approximation to it that is computationally efficient.

3. Recursive versions of supervised and semi-supervised maximum margin classifiers.

4. Method for detecting anomalous points that are partially labeled high utility by a domain expert.

5. Framework and probabilistic model for detecting anomalous activity in the traffic rates of sparse networks.

CHAPTER 1

Introduction

Data collected in the real world is not perfect. In addition to being noisy, there can be many other complications that make it hard to analyze. The data collection process may not be finished before analysis starts. The data may only be partially labeled due to the costliness of labeling or simply a lack of information. There may be features of the data that cannot be directly observed. Or the data may contain anomalies, which if interesting, must be identified by the model, as opposed to just robustifying the model against them. In this thesis, we will address these additional issues of non-batch data, partial labels, latent variables, and anomalies.

In particular, many of the problem areas addressed in thesis were motivated by the real world scenario of data collected in the interests of nuclear non-proliferation. A nuclear fuel cycle produces an abundant amount of data; however, like in all realistic scenarios, the data observed or collected is not necessarily the exact information that is desired. For example, materials transported between facilities in the fuel cycle constitutes a traffic network and anomalous traffic in the network could indicated a diversion of nuclear materials. But it may be expensive or impossible to directly track the movement of materials throughout the fuel cycle. Instead if the facilities are monitored, the ingress and egress of materials for each facility would be observed and the traffic of materials can be reconstructed in order to test for anomalies. Or, if we know of some particular types of anomalies that are more likely to indicate a diversion of nuclear materials, we can lower our false alarms by only identifying those anomalies. Also, since a nuclear fuel cycle is continuously producing data, and it is of essence to identify diversions of nuclear materials as soon as possible, we must be able update our model to make predictions in an online fashion. The methods in this thesis, while not directly applied to real nuclear fuel cycle data, are applied to real datasets that could be surrogates for such unobtainable data.

Overall this thesis emphasizes the optimization of probabilistic models with prior information. This builds off work from the past twenty years that have pushed towards solving regularized objective functions. While we do not contribute new optimization algorithms,

much of this work is only possible due to cleverly solving dual formulations that we have proposed or have been recently studied. The probabilistic nature of our models has many advantages such as incorporating latent variables with the Expectation-Maximization algorithm or testing hypothesis for the existence of anomalies. It also allows us to bound the errors of our model due to inaccuracies such as finite sample size or model mismatch. In what remains of this section we will give an overview of the contributions of each chapter and then a list of publications.

Chapter 2 treats the problem of the Kalman filter, which is the Bayesian optimal filter for Gaussian distributed dynamic systems. This means it can incorporate new data by updating the current model in an equivalent way to if the model was learned on the full batch of data. However, when the system is strongly non-linear with potentially unknown gradients, it is no longer optimal. In the second chapter of this thesis, we propose an extension to the ensemble Kalman filter (EnKF) to deal with its collapsing problems in high dimensional systems. The ensemble Kalman filter is a data assimilation technique that uses an ensemble of models, updated with data, to track the time evolution of a non-linear system. It does so by using an empirical approximation to the well-known Kalman filter. Unfortunately, its performance suffers when the ensemble size is smaller than the state space, as is often the case for computationally burdensome models. This scenario means that the empirical estimate of the state covariance is not full rank and possibly quite noisy. To solve this problem in this high dimensional regime, a computationally fast and easy to implement algorithm called the penalized ensemble Kalman filter (PEnKF) is proposed. Under certain conditions, it can be proved that the PEnKF does not require more ensemble members than state dimensions in order to have good performance. Further, the proposed approach does not require special knowledge of the system such as those used by localization methods. These theoretical results are supported with superior performance in simulations of several non-linear and high dimensional systems.

Next we leave the specific Gaussian case and look at sequential filtering more broadly. In the third chapter of this thesis called "Sequential Sparse Maximum Entropy Models for Non-linear Regression", we examine how to use the principle of minimum relative entropy to create sequential filters for other models. We first present an optimal filter for Gaussian linear regression derived from a minimum relative entropy objective, which can be seen as a dual formulation for the recursive least squares filter. Extending upon this, we then consider Gaussian non-linear regression using kernel functions. We present an optimal filter and then an approximation to it that does not require re-visiting all the previous data. Then in chapter 4, entitled "Sequential Maximum Entropy Discrimination with Partial Labels", we use the framework presented in the previous chapter for binary classification in the supervised and

semi-supervised settings. We show how to update maximum margin classifiers to allow for sequential filtering of binary response variables. Our maximum margin classifier admits a kernel representation to represent large numbers of features and can also be regularized with respect to a smooth sub-manifold, allowing it to incorporate unlabeled observations. We compare the performance of our classifier to its non-sequential equivalents in both simulated and real datasets.

Now, instead of standard classification, where many samples are provided for each class, in anomaly detection there are many samples from the nominal class and only very few samples from the anomalous class. Most data-driven anomaly detection approaches formulate the anomaly class as rare events that lie in the tails of the nominal class density. Data-driven anomaly detection methods suffer from the drawback that they do not take account of the practical importance, or utility, of an anomaly to the user. Furthermore, standard classification methods suffer when the difference in class sizes is extremely large, which is bound to occur if one of the classes is anomalous data. In the fifth chapter of this thesis entitled "Maximum Entropy Discrimination with Partial Labels for Anomaly Detection", we address the problem of learning how to detect anomalies when there some of the anomalous instances are labeled, e.g., by a human, as high utility. To this end, we propose a novel method called Latent Laplacian Maximum Entropy Discrimination (LatLapMED) as a potential solution. This method uses the EM algorithm to simultaneously incorporate the Geometric Entropy Minimization principle for identifying statistical anomalies, and the Maximum Entropy Discrimination principle to incorporate utility labels, in order to detect high-utility anomalies. We apply our method in both simulated and real datasets to demonstrate that it has superior performance over existing alternatives that independently pre-process with unsupervised anomaly detection algorithms before classifying.

The previous chapter approaches anomaly detection from a partially supervised scenario where some of the anomalies are known. In the sixth chapter of this thesis, entitled "Anomaly Detection in Partially Observed Traffic Networks", we will address a specific unsupervised scenario, that of detecting anomalous activity in traffic networks where the network is not directly observed. Given knowledge of what the node-to-node traffic in a network should be, any activity that differs significantly from this baseline would be considered anomalous. We propose a Bayesian hierarchical model for estimating the traffic rates and detecting anomalous changes in the network. The probabilistic nature of the model allows us to perform statistical goodness-of-fit tests to detect significant deviations from a baseline network. We show that due to the more defined structure of the hierarchical Bayesian model, such tests perform well even when the empirical models estimated by the EM algorithm are misspecified. We apply our model to both simulated and real datasets to

demonstrate its superior performance over existing alternatives. Finally, in Chapter 7, we summarize the thesis and highlight some future directions of research that we think would be worthwhile.

1.1 Publications

Journals

- "Latent Laplacian Maximum Entropy Discrimination for Detection of High-Utility Anomalies". E. Hou, K. Sricharan, A. O. Hero. IEEE Transactions on Information Forensics and Security (2018).
- "Anomaly Detection in Partially Observed Traffic Networks" E. Hou, Y. Yilmaz, A. O. Hero. IEEE Transactions on Signal Processing (2019).

Conferences

- "Sequential Maximum Margin Classifiers for Partially Labeled Data". E. Hou, A. O. Hero. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing.
- "Online Diversion Detection in Nuclear Fuel Cycles via Multimodal Observations". Y. Yilmaz, E. Hou, A. O. Hero. ANS Advances in Nuclear Nonproliferation Technology and Policy Conference.
- "Diversion Detection in Partially Observed Nuclear Fuel Cycle Networks". E. Hou, Y. Yilmaz, A. O. Hero. ANS Advances in Nuclear Nonproliferation Technology and Policy Conference.

In Preparation

- "Penalized Ensemble Kalman Filters for High Dimensional Non-linear Systems" E. Hou, E. Lawrence, A. O. Hero.
- "Sequential Sparse Maximum Entropy Models for Non-linear Regression" E. Hou, A. O. Hero.

CHAPTER 2

Penalized Ensemble Kalman Filters for High Dimensional Non-linear Systems

2.1 Introduction

The Kalman filter is a well-known technique to track the state of a linear system over time, and many variants based on the extended and ensemble Kalman filters have been proposed to deal with non-linear systems. The ensemble Kalman filter (EnKF) [2, 3] is particularly popular when the non-linear system is extremely complicated and its gradient is infeasible to calculate, which is often the case in geophysical systems. However, these systems are often high dimensional and forecasting each ensemble member forward through the system is computationally expensive. Thus, the filtering often operates in the high dimensional regime where the number of ensemble members, n, is much less than the size of the state, p. It is well known that even when $p/n \to const.$ and the samples are from a Gaussian distribution, the eigenvalues and the eigenvectors of the sample covariance matrix do not converge to their population equivalents, [4, 5]. Since our ensemble is both non-Gaussian and high dimensional ($n << p$), the sample covariance matrix of the forecast ensemble will be extremely noisy. In this paper, we propose a variant of the EnKF specifically designed to handle covariance estimation in this difficult regime, but with weaker assumptions and less prior information than competing approaches.

Other Work

To deal with the sampling errors, many schemes have been developed to de-noise the forecast sample covariance matrix. These schemes "tune" the matrix with variance inflation and localization, [6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]. However, these schemes are often not trivial to implement because they require carefully choosing

the inflation factor and using expert knowledge of the true system to set up the localization. Additionally, the EnKF with perturbed observations introduces additional sampling errors due to the perturbation noise's lack of orthogonality with the ensemble. Methods have been devised that construct perturbation matrices that are orthogonal, [20]; however these methods are computationally expensive, [21]. This has lead to the development of matrix factorization versions of the EnKF such as the square root and transform filters, [22, 23, 24, 21, 25, 26, 27, 28], which do not perturb the observations and are designed to avoid these additional sampling errors.

The ensemble Kalman filter is closely related to the particle filter [29], although it uses a Gaussian approximation of the conditional state distribution in order to get an update that is a closed form expression for the analysis ensemble (as opposed to one that requires numerical integration). While the particle filter does not use this approximation, it also requires an exponential number of particles to avoid filter collapse, [30]. Recently, there has been significant effort to apply the particle filter to larger scale systems using equal weights, [31, 32], and merging it with the ensemble Kalman filter to form hybrid filters, [29, 33, 34, 35, 36]. EnKF is also related to the unscented Kalman filter, [37, 38], which handles nonlinearity by propagating a carefully selected set of "sigma points" (as opposed to the randomly sampled points of the EnKF) through the nonlinear forecast equations. The results are then used to reconstruct the forecasted mean and covariance.

Most similar to our proposed work are [39] and [40], which also propose methods that use sparse inverse covariance matrices. Both methods justify the appropriateness of using the inverse space with large scale simulations or real weather data. The former reports that their computational complexity is polynomial in the state dimension and requires the stronger assumptions of Gaussianity and structural knowledge. The latter algorithm can be implemented in parallel, making it very efficient, however, the paper still makes the *much* stronger assumptions of Gaussianity and conditional independence.

Proposed Method

We propose a penalized ensemble Kalman filter (PEnKF), which uses an estimator of the forecast covariance whose inverse is sparsity regularized. While the localization approaches effectively dampen or zero out entires in the covariance, *our approach zeros out entries in the inverse covariance, resulting in a sparse inverse covariance*. This provides two advantages. First, it makes a weaker assumption about the relationship between state variables. Second, our approach does not require anything like localization's detailed knowledge of which covariances to fix at zero or how much to dampen. Instead, it merely

favors sparsity in the inverse covariance. Additionally, our method is very easy to implement because it just requires using a different estimator for the covariance matrix in the EnKF. We can explicitly show the improvement of our estimator through theoretical guarantees.

Outline

In Section 2, we explain the assumptions in our high-dimensional system and we give background on the EnKF and ℓ_1 penalized inverse covariance matrices. In Section 3, we give details on how to modify the EnKF to our proposed PEnKF and provide theoretical guarantees on the filter. Section 4 contains the simulation results of the classical Lorenz 96 system and a more complicated system based on modified shallow water equations.

2.2 Background

In this paper, we consider the scenario of a noisy, non-linear dynamics model $f(\cdot)$, which evolves a vector of unobserved states vec$x_t \in \mathbb{R}^p$ through time. We observe a noisy vector vec$y_t \in \mathbb{R}^r$, which is a transformation of vecx_t by a function $h(\cdot)$. Both the process noise $\boldsymbol{\omega}_t$ and the observation noise $\boldsymbol{\epsilon}_t$ are independent of the states vecx_t. We assume both noises are zero mean Gaussian distributed with known diagonal covariance matrices, $\mathbf{Q}$ and $\mathbf{R}$. Often, it is assumed that the dynamics model does not have noise making $\boldsymbol{\omega}_t$ a zero vector, but for generality we allow $\boldsymbol{\omega}_t$ to be a random vector.

$$\text{vec}x_t = f(\text{vec}x_{t-1}) + \boldsymbol{\omega}_t \qquad \text{Dynamics Model}$$
$$\text{vec}y_t = h(\text{vec}x_t) + \boldsymbol{\epsilon}_t \qquad \text{Observation Model}$$

As with localization methods, we make an assumption about the correlation structure of the state vector in order to handle the high dimensionality of the state. In particular, we assume that relatively few pairs of state variables have non-zero conditional correlation, $\text{Cov}(x_i, x_j | x_{-(i,j)}) \neq 0$ where $x_{-(i,j)}$ represents all state variables except x_i and x_j. This means that, conditioning on all of the rest of the state, x_i and x_j are uncorrelated. They may have a dependency, meaning that the correlation between them is non-zero, but that correlation is entirely explained by dependence on other parts of the state. A sample example is given by a one-dimensional spatial field with the three locations x_1, x_2, and x_3 where x_1 and x_3 are both connected to x_2, but not each other. In this case, it might be reasonable to model x_1 and x_3 as uncorrelated conditional on x_2 although not necessarily unconditionally uncorrelated. Their simple correlation might not be zero, but their conditional or

partial correlation is zero. This is similar to our assumption, although we do not assume any particular pattern of the conditional dependencies as you might in a spatial field.

We assume that the set of non-zero conditional correlations is sparse. This is equivalent to assuming that the inverse correlation matrix of the model state is sparse. In other words, the inverse covariance matrix will have s non-zero off-diagonal entries. We can also quantify the sparsity level as d, which is the maximum number of non-zero off-diagonals in any row, so $d^2 << p^2$. Note that our assumption is on the conditional *correlation* or lack of it, and we do not make any claims on the independence between states. This is because vecx_t is not Gaussian when $f(\cdot)$ is non-linear so uncorrelation does not imply independence thus the zeros in the inverse covariance matrix do not imply conditional independence. This assumption is weaker than the one made using localization. That assumption is equivalent to assuming that the covariance matrix itself is sparse whereas our assumption admits a dense covariance. Finally, because we do not assume that the state variable interactions are the same for different time points, we allow the set $\mathcal{E}_t$ and its size s_t to change over time.

2.2.1 Ensemble Kalman Filter

The standard EnKF algorithm of [20] is shown in Algorithm 1. At time $t = 0$, n samples are drawn from some distribution, which is often chosen as the standard multivariate normal distribution, if the true initial distribution is unknown, to form an initial ensemble $\mathbf{A} \in \mathbb{R}^{p \times n}$. And, at every time point t, the observations vecy_t are perturbed n times with Gaussian white noise, vec$\eta^j \sim N(\text{vec}0, \mathbf{R})$, to form a perturbed observation matrix $\mathbf{D}_t \in \mathbb{R}^{p \times n}$, where vec$d_t^j = \text{vec}y_t + \text{vec}\eta^j$.

Algorithm 2.1 Ensemble Kalman Filter

Input: $\mathbf{A}, \mathbf{H}, \mathbf{Q}, \mathbf{R}$, and $\mathbf{D}_t$

where $\mathbf{H}$ is the the measurement operator

for $t \in \{1, ..., T\}$ **do**

▷ Evolve each ensemble member forward in time

$\text{vec}a_0^j = f(\text{vec}a^j) + \text{vec}w^j \quad \forall j \in \{1, ..., n\}$

where vec$w^j \sim N(\text{vec}0, \mathbf{Q})$

▷ Correct the ensemble with the observations

$\mathbf{A} = \mathbf{A}_0 + \hat{\mathbf{K}}(\mathbf{D}_t - \mathbf{H}\mathbf{A}_0)$

where $\hat{\mathbf{K}} = \hat{\mathbf{P}}^f \mathbf{H}^T (\mathbf{H}\hat{\mathbf{P}}^f \mathbf{H}^T + \mathbf{R})^{-1}$

▷ Predict using the analysis ensemble mean

$\hat{\text{vec}}x_t = \frac{1}{n} \sum_{j=1}^{n} \text{vec}a^j$

end for

Output: $\hat{\text{vec}}x_t$

The forecast covariance estimator $\hat{\mathbf{P}}^f$ is typically the sample covariance of the forecast ensemble, defined as $\widehat{\text{Cov}}(\mathbf{A}_0) = \frac{1}{n-1}(\mathbf{A}_0 - \bar{\mathbf{A}}_0)(\mathbf{A}_0 - \bar{\mathbf{A}}_0)^T$, where $\bar{\mathbf{A}}_0$ is the sample mean vector, but it can be another estimator such as a localized estimator (one that is localized with a taper matrix), or a penalized estimator as proposed in this paper.

2.2.2 Bregman Divergence and the Penalty

Below, we give a brief overview of the ℓ_1 penalized log-determinant Bregman divergence and some properties of its minimizing estimator, as described in [41]. We denote $\mathbf{S}$ to be any arbitrary sample covariance matrix, and $\mathbf{\Sigma} = \text{E}(\mathbf{S})$ to be its true covariance matrix, where $\text{E}(\cdot)$ is the expectation function.

The Bregman divergence is a very general method to measure the difference between two functions. Here the functions to be compared are covariance matrices. Since we are interested in finding a sparse positive definite estimator for the inverse covariance matrix, a natural choice of Bregman function is $-\log\det(\cdot)$, which has a domain restricted to positive definite matrices. Thus $\mathbf{\Theta}$, our optimal estimator for the inverse covariance matrix $\mathbf{\Sigma}^{-1}$, will minimize

$$\underset{\mathbf{\Theta}\in\mathbb{S}_{++}^{p\times p}}{\arg\min} \; -\log\det(\mathbf{\Theta}) - \log\det(\mathbf{\Sigma}) + \text{tr}\left(\mathbf{\Sigma}(\mathbf{\Theta} - \mathbf{\Sigma}^{-1})\right)$$

where $\mathbb{S}_{++}^{p\times p}$ is the set of all symmetric positive definite $p \times p$ matrices. This loss function requires the covariance matrix $\mathbf{\Sigma}$ to be known, but it can approximated by an empirical loss, which replaces $\mathbf{\Sigma}$ with its empirical equivalent $\mathbf{S}$ and adds a penalty term to ensure strict convexity.

The empirical Bregman divergence with function $-\log\det(\cdot)$ and an ℓ_1 penalty term essentially reduces (by dropping the constants) to

$$\underset{\mathbf{\Theta}\in\mathbb{S}_{++}^{p\times p}}{\arg\min} \; -\log\det(\mathbf{\Theta}) + \text{tr}(\mathbf{\Theta}\mathbf{S}) + \lambda||\Theta||_1 \tag{2.1}$$

where $\lambda \geq 0$ is a penalty parameter, and $||\cdot||_1$ denotes an element-wise ℓ_1 norm. This can be generalized so that each entry of $\mathbf{\Theta}$ can be penalized differently if λ is a matrix and using a element-wise product with the norm.

This objective has a unique solution, $\mathbf{\Theta} = (\tilde{\mathbf{S}})^{-1}$, which satisfies

$$\frac{\partial}{\partial\mathbf{\Theta}}\text{B}^{\lambda}(\mathbf{\Theta}||\mathbf{S}^{-1}) = \mathbf{S} - \mathbf{\Theta}^{-1} + \lambda\partial||\mathbf{\Theta}||_1 = 0$$

where $\partial||\mathbf{\Theta}||_1$ is a subdifferential of the ℓ_1 norm defined in (2.2) in the appendix. The solution $(\tilde{\mathbf{S}})^{-1}$ is a sparse positive definite estimator of the inverse covariance matrix $\mathbf{\Sigma}^{-1}$, and we can write its inverse explicitly as $\tilde{\mathbf{S}} = \mathbf{S} + \lambda\tilde{\mathbf{Z}}$, where $\tilde{\mathbf{Z}}$ is the unique subdifferential matrix that makes the gradient zero. See [41] or [42] for a more thorough explanation of $\tilde{\mathbf{Z}}$.

[41] show that for well-conditioned covariances and certain minimum sample sizes, the estimator $(\tilde{\mathbf{S}})^{-1}$ has many nice properties including having, with high probability, the correct zero and signed non-zero entries and a sum of squared error that converges to 0 as $n, p, s \to \infty$. These properties will allow our method, described in the next section, to attain superior performance over the EnKF.

2.3 Penalized Ensemble Kalman Filter

Our penalized ensemble Kalman filter modifies the EnKF, by using a penalized forecast covariance estimator $\tilde{\mathbf{P}}^f$. This penalized estimator is derived from its inverse, which is the minimizer of (2.1). Thus from Section 2.2.2.2, it can be explicitly written as $\tilde{\mathbf{P}}^f = \hat{\mathbf{P}}^f + \lambda\tilde{\mathbf{Z}}$, implying that we essentially learn a matrix $\tilde{\mathbf{Z}}$, and use it to modify our sample covariance $\hat{\mathbf{P}}^f$ so that $(\hat{\mathbf{P}}^f + \lambda\tilde{\mathbf{Z}})^{-1}$ is sparse. From this, our modified Kalman gain matrix is

$$\tilde{\mathbf{K}} = (\hat{\mathbf{P}}^f + \lambda\tilde{\mathbf{Z}})\mathbf{H}^T \left(\mathbf{H}(\hat{\mathbf{P}}^f + \lambda\tilde{\mathbf{Z}})\mathbf{H}^T + \mathbf{R}\right)^{-1}.$$

The intuition behind this estimator is that since only a small number of the state variables in the state vector vecx_t are conditionally correlated with each other, *the forecast inverse covariance matrix* $(\mathbf{P}^f)^{-1}$ *will be sparse with many zeros in the off-diagonal entries.* Furthermore, since minimizing (2.1) gives a sparse estimator for $(\mathbf{P}^f)^{-1}$, this sparse estimator will accurately capture the conditional correlations and uncorrelations of the state variables. Thus $\tilde{\mathbf{P}}^f$ will be a much better estimator of the true forecast covariance matrix $\mathbf{P}^f$ because the ℓ_1 penalty will depress spurious noise in order to make $(\tilde{\mathbf{P}}^f)^{-1}$ sparse, while the inverse of the sample forecast covariance $(\hat{\mathbf{P}}^f)^{-1}$, when it exists, will be non-sparse. As in most penalized estimators, the $\tilde{\mathbf{P}}^f$ is a biased estimator of the forecast covariance, while the sample forecast covariance is not. But because the forecast distribution is corrected for in the analysis step, it is acceptable to take this bias as a trade-off for less variance (sampling errors). A more in-depth study of the consequences of this bias in ℓ_1 penalized inverse covariance matrices and their inverses is described in [42]. Additionally, this bias due to penalization in the inverse covariance behaves in a similar way as variance inflation because the bias on the diagonal of $(\tilde{\mathbf{P}}^f)^{-1}$ is due to it being increased by λ, so having a biased estimator is not necessarily disadvantageous. And finally, since we do not assume

the state variables interact in the same way over all time, we re-learn the matrix $\tilde{\mathbf{Z}}$ every time the ensemble is evolved forward.

We can choose the penalty parameter λ in a systematic fashion by calculating a regularization path, solving (2.1) for a list of decreasing λs, and evaluating each solution with an information criterion such as an extended or generalized Akaike information criterion (AIC) or Bayesian information criterion (BIC), [43, 44]. Additionally, if we have knowledge or make assumptions about the moments of the ensemble's distribution, we know the optimal proportionality of the penalty parameter (see proof of Theorem 2.3.1. Thus, we can refine the penalty parameter by calculating a regularization path for the constant of the optimal order. In Section 2.4, we describe a practical approach to choosing λ using a free forecast model run like in [36] and the BIC.

2.3.1 Implications on the Kalman Gain Matrix

The only estimated randomness in the EnKF occurs in the ensemble update step, which is a linear function of the Kalman gain matrix. So, having an accurate estimator of the true Kalman gain matrix $\mathbf{K}$ will ensure that the algorithm performs well. And, because the true Kalman gain matrix inherits many of the properties of the forecast covariance matrix $\mathbf{P}^f$, our modified Kalman gain matrix $\tilde{\mathbf{K}}$ will benefit from many of the nice properties of our forecast covariance estimator $\tilde{\mathbf{P}}^f$.

How good of an estimator we can get for the forecast covariance matrix $\mathbf{P}^f$ will of course depend on its structure. If it is close to singular or contains lots of entries with magnitudes smaller than the noise level, it will be always be difficult to estimate. So for the following theorem, we assume that the forecast covariance matrix is well-behaved. This means that it satisfies the standard regularity conditions (incoherence, bounded eigenvalue, sparsity, sign consistency and monotonicity of the tail function) found in many places including [41, 42] and also defined in the appendix.

Theorem 2.3.1. *Under regularity conditions and for the system described in Section 2.2, when* $\lambda \asymp \sqrt{3\log(p)/n}$ *for sub-Gaussian ensembles and* $\lambda \asymp \sqrt{p^{3/m}/n}$ *for ensembles with bounded* $4m^{\text{th}}$ *moments,*

$$\textit{Sum of Squared Errors of } \tilde{\mathbf{K}} \lesssim \textit{ Sum of Squared Errors of } \hat{\mathbf{K}}$$

and as long as the sample size is at least $o(n) = 3d^2\log(p)$ *for sub-Gaussian ensembles and* $o(n) = d^2p^{3/m}$ *for ensembles with bounded* $4m^{\text{th}}$ *moments,*

$$\textit{Sum of Squared Errors of } \tilde{\mathbf{K}} \to 0 \textit{ with high probability as } n, p, s \to \infty.$$

The above theorem gives us a sense of the performance of the modified Kalman gain matrix in comparison to the sample Kalman gain matrix. It shows that with high probability, the modified Kalman gain matrix will have an asymptotically smaller sum of squared error (SSE) than a Kalman gain matrix formed using the sample forecast covariance matrix. Also, for a given number of states p, the theorem tell us the minimum ensemble size n required for our modified Kalman gain matrix to be a good estimate of the true Kalman gain matrix. The sub-Gaussian criterion, where all moments are bounded, is actually very broad and includes any state vectors with a strictly log-concave density and any finite mixture of sub-Gaussian distributions. However even if not all moments are bounded, the larger the number of bounded fourth-order moments m, the smaller the necessary sample size. In comparison, the sample Kalman gain matrix requires $o(n) = p^2$ samples in the sub-Gaussian case, and also significantly more in the other case (see appendix for exact details). When the minimum sample size for a estimator is not met, good performance cannot be guaranteed because the asymptotic error will diverge to infinity instead of converge to zero. This is why when the number of ensembles n is smaller than the number of states p, just using the sample forecast covariance matrix is not sufficient.

2.3.2 Implications on the Analysis Ensemble

It is well known that due to the additional stochastic noise used to perturb the observations, the covariance of the EnKF's analysis ensemble, $\widehat{\mathrm{Cov}}(\mathbf{A})$ is not equivalent to its analysis covariance calculated by the Gaussian update $\hat{\mathbf{P}}^a = (\mathbf{I} - \hat{\mathbf{K}}\mathbf{H})\hat{\mathbf{P}}^f$. This has led to the development of deterministic variants such as the square root and transform filters, which do have $\widehat{\mathrm{Cov}}(\mathbf{A}) = \hat{\mathbf{P}}^a$. However, in a non-linear system, this update is sub-optimal because it uses a Gaussian approximation of $\Pr(\mathrm{vec}x_t|\mathrm{vec}y_{t-1})$, the actual distribution of forecast ensemble $\mathbf{A}_0$. Thus let us denote $\mathbf{P}^a$ as the true analysis covariance defined as

$$\int (\mathrm{vec}x_t)^2 \Pr(\mathrm{vec}x_t|\mathrm{vec}y_t) d\mathrm{vec}x_t - \left(\int \mathrm{vec}x_t \Pr(\mathrm{vec}x_t|\mathrm{vec}y_t)\, d\mathrm{vec}x_t \right)^2$$

where $\Pr(\mathrm{vec}x_t|\mathrm{vec}y_t) = \Pr(\mathrm{vec}y_t|\mathrm{vec}x_t) \Pr(\mathrm{vec}x_t|\mathrm{vec}y_{t-1}) / \Pr(\mathrm{vec}y_t)$ is not Gaussian. Then, $\mathrm{E}(\hat{\mathbf{P}}^a) \neq \mathbf{P}^a$ and there will always be this analysis spread error regardless of whether $\widehat{\mathrm{Cov}}(\mathbf{A}) = \hat{\mathbf{P}}^a$ or not.

As also mentioned in [33], actually none of the analysis moments of the EnKF are consistent with the true moments including the analysis mean. However this analysis error is present in all methods that do not introduce particle filter properties to the EnKF, and thus is not the focus of our paper. We are primarily concerned with the sampling errors in

high-dimensional systems and simply wanted to address that the lack of equivalence to the Gaussian update is irrelevant in our case of a non-linear system.

2.3.3 Computational Time and Storage Issues

The computational complexity of solving for the minimizer of (2.1) with the GLASSO algorithm from [45] is $O(sp^2)$ because it is a coordinate descent algorithm. Although the final estimator $(\tilde{\mathbf{P}^f})^{-1}$ is sparse and only requires storing $s+p$ values, the algorithm requires storing $p \times p$ matrices in memory. However, by using iterative quadratic approximations to (2.1), block coordinate descent, and parallelization, the BIGQUIC algorithm of [46] has computational complexity $O(s(p/k))$ and only requires storing $(p/k) \times (p/k)$ matrices, where k is the number of parallel subproblems or blocks.

The matrix operations for the analysis update $\mathbf{A} = \mathbf{A}_0 + up$ can also be linear in p if $\mathbf{R}$ is diagonal and $\mathbf{H}$ is sparse (like in banded interpolation matrices) with at most $h << q$ non-zero entries in a row. Then $((\tilde{\mathbf{P}^f})^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})$ has at most $(s + p + qh^2) << p^2$ non-zero entries and can be computed with $O(s + p + qh^2)$ matrix operations. And, solving for up only takes $O(n(s+p+qh^2))$ matrix operations because it is made from the solutions to the sparse linear systems $((\tilde{\mathbf{P}^f})^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})up = \mathbf{H}^T\mathbf{R}^{-1}(\mathbf{D}_t - \mathbf{H}\mathbf{A}_0)$ where the right-hand side takes $O(pq^2 + qpn)$ matrix operations to form.

2.4 Simulations

In all simulations, we compare to an ensemble Kalman filter where the forecast covariance matrix is localized with a taper matrix generated from equation (4.10) in [47]. The taper matrix parameter c is chosen using the true interactions of the system, so the localization should be close to optimal for simple systems. We use this TAPER-EnKF as the baseline because if the PEnKF can do as well as this filter, it implies that the PEnKF can learn a close to optimal covariance matrix, even without the need to impose a known neighborhood structure. If the PEnKF can do better than this filter, it implies that the PEnKF is learning some structure that is not captured by localization with a taper matrix.

In order to choose the penalty parameter for PEnKF, we assume that the state variables in our examples are sub-Gaussian. In this case, we can set $\lambda = c_\lambda \sqrt{R \log(p)/n}$ for some appropriate choice of c_λ (see the proof of Theorem 2.3.1), where R is the observation noise's variance. To estimate c_λ, we generate a representative ensemble (which may also be our initial ensemble) using a free forecast run like in [36] in which a state vector is drawn at random (e.g. from $N(\text{vec}0, \mathbf{I})$) and evolved forward. The representative ensemble

is produced by taking a set of equally spaced points (e.g. every 100th state vector) from the evolution. This ensemble is used to choose c_λ from some predefined interval by minimizing the extended Bayesian information criterion (eBIC) of [43] if $p > n$ or the BIC of [48] if $p < n$.

Of course (2.1) is not a likelihood unless the states are Gaussian. So, we have a misspecified model where we are treating the states as having a Gaussian likelihood when evaluating a potential penalty parameter using an information criterion. In this case, we should correct our information criterion for the misspecification as in [44]. However, this can be quite difficult and we leave an in-depth exploration of this problem for future work. In the meantime, we assume the misspecified information criterion is close to the correct information criterion, and it does seem to perform well despite its lack of optimality.

We define the root mean squared error (RMSE) used to evaluate a filters performance by

$$\text{RMSE}_t = \sqrt{(||\hat{\text{vec}}x_t - \text{vec}x_t||_2)^2/p}$$

where RMSE_t is an element of a vector indicating the RMSE at time point t, $\text{vec}x_t$ is a vector of the true hidden state variables, $\hat{\text{vec}}x_t$ is a filter's estimators for the true state vector, and $||\cdot||_2$ is the ℓ_2 norm. We will refer to quantiles such as the mean or median RMSE to be the mean or median of the elements of the RMSE vector.

2.4.1 Lorenz 96 System

The 40-state Lorenz 96 model is one of the most common systems used to evaluate ensemble Kalman filters. The state variables are governed by the following differential equations

$$\frac{dx_t^i}{dt} = \left(x_t^{i+1} - x_t^{i-2}\right) x_t^{i-1} - x_t^i + 8 \qquad \forall i = 1, \ldots, 40$$

where $x_t^{41} = x_t^1, x_t^0 = x_t^{40},$ and $x_t^{-1} = x_t^{39}$.

We use the following simulation settings. We have observations for the odd state variables, so $\text{vec}y_t = \mathbf{H}\text{vec}x_t + \boldsymbol{\epsilon}_t$ where $\mathbf{H}$ is a 20×40 matrix with ones at entries $\{i, j = 2i-1\}$ and zeros everywhere else and $\boldsymbol{\epsilon}_t$ is a 20×1 vector drawn from a $N(\text{vec}0, 0.5\,\mathbf{I})$. We initialize the true state vector from a $N(\text{vec}0, \mathbf{I})$ and we assimilate at every $0.4t$ time steps, where $t = 1, \ldots, 2000$. The system is numerically integrated with a 4th order Runge-Kutta method and a step size of 0.01. The main difficulties of this system are the large assimilation time step of 0.4, which makes it significantly non-linear, and the lack of observations for the even state variables.

Since the exact equations of the Lorenz 96 model are fairly simple, it is clear how the state variables interact with each other. This makes it possible to localize with a taper matrix that is almost optimal by using the Lorenz 96 equations to choose a half-length parameter c. However, we do not incorporate this information in the PEnKF algorithm, which instead learns interactions by essentially extracting it from the sample covariance matrix. We set the penalty parameter $\lambda = c_\lambda \sqrt{0.5 \log(p)/n}$ by using an offline free forecast run to search for the constant c_λ in the range $[0.1, 10]$ as described at the beginning of this section.

We average the PEnKF estimator of the forecast inverse covariance matrix at the time points 500, 1000, 1500, and 2000 for 50 trials with 25 ensembles members, and we compare it to the "true" inverse covariance matrix, which is calculated by moving an ensemble of size 2000 through time. In Figure 2.1, each line represents the averaged normalized rows of an inverse covariance matrix and the lines are centered at the diagonal. The penalized inverse covariance matrix does a qualitatively good job of capturing the neighborhood information and successfully identifies that any state variables far away from state variable i, do not interact with it.

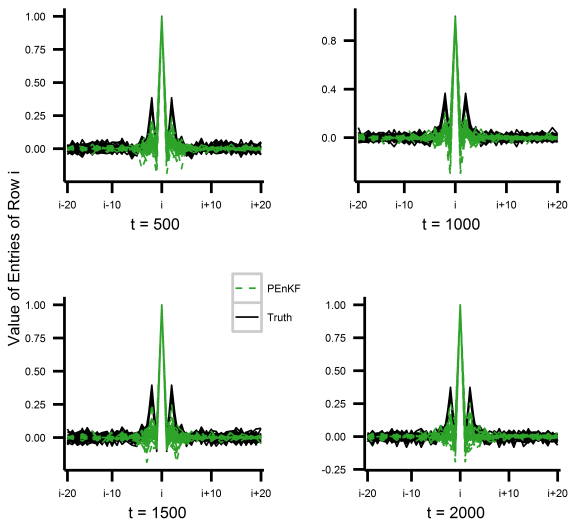


Figure 2.1: Each line represents the normalized values of entries of row i of the inverse covariance matrix, ordered from $i - 20$ to $i + 20$, where the values are averaged over 50 trials. The PEnKF algorithm is successful at identifying that the state variables far away from variable i have no effect on it, even though there are fewer ensemble members than state variables.

Because the PEnKF is successful at estimating the structure of the inverse covariance matrix and thus the forecast covariance matrix, we expect it will have good performance for estimating the true state variables. We compare the PEnKF to the TAPER-EnKF and other

estimators from [49, 33, 34, 50] by looking at statistics of the RMSE. Note that in order to have comparable statistics to as many other papers as possible, we do not add variance inflation to the TAPER-EnKF (like in [49, 34, 50] and unlike in [33]). Also, like in those papers, we initialize the ensemble from a $N(\text{vec}0, \mathbf{I})$, and we use this ensemble to start the filters. Note that in this case, the initial ensemble is different than the offline ensemble that we use to estimate the PEnKF's penalty parameter. This is because the initial ensemble is not representative of the system and its sample covariance is an estimator for the identity matrix. The TAPER-EnKF, which is simply called the EnKF in the other papers, is localize by applying a taper matrix where $c = 10$ to the sample covariance matrix.

We show the mean, median, 10%, and 90% quantiles of the RMSE averaged over 50 independent trials for ensembles of size 400, 100, 25, and 10 in Table 2.1. For 400 ensemble members, the PEnKF does considerably better than the TAPER-EnKF and its relative improvement is larger than that of the XEnKF reported in [49] and similar to those of the NLEAF, EnKPF, and XEnKF reported in [33, 34, 50] respectively. For 100 ensemble members, the PEnKF does do worse than the TAPER-EnKF and EnKPF of [34]; this we suspect may be do to the bias-variance trade-off when estimating the forecast covariance matrix. The PEnKF has the most significant improvement over the TAPER-EnKF in the most realistic regime where we have fewer ensemble members than state variables. For both 25 and 10 ensemble members, the PEnKF does considerably better than the TAPER-EnKF and it does not suffer from filter divergence, which [34] report occurs for the EnKPF at 50 ensemble members.

While it is clear the PEnKF does well even when there are fewer ensemble members than state variables, 40 variables is not enough for the problem to be considered truly high-dimensional. We now consider simulation settings where we increase the dimension of the state space p while holding the number of ensemble members n constant. We initialize the ensemble from the free forecast run and set λ and the taper matrix in the same way as in the previous simulations. We examine the mean RMSE averaged over 50 trials and its approximate 95% confidence intervals in the Figure 2.2. The mean RMSE of the PEnKF is significantly smaller than the mean RMSE of the TAPER-EnKF for all p. Additionally the confidence intervals of the mean RMSE are much narrower than the ones for the TAPER-EnKF. This suggest that there is little variability in the PEnKF's performance, while the TAPER-EnKF's performance is more dependent on the trial, with some trials being "easier" for the TAPER-EnKF than others.

Table 2.1: Mean, median, 10% and 90% quantile of RMSE averaged over 50 trials. The number in the parentheses is the summary statistics' corresponding standard deviation.

$n = 400$	10%	50 %	Mean	90%
TAPER-EnKF	0.580 (.01)	0.815 (.01)	0.878 (.02)	1.240 (.03)
PEnKF	0.538 (.02)	0.757 (.03)	0.827 (.03)	1.180 (.05)

$n = 100$	10%	50 %	Mean	90%
TAPER-EnKF	0.582 (.01)	0.839 (.02)	0.937 (.03)	1.390 (.06)
PEnKF	0.717 (.04)	0.988 (.04)	1.067 (.04)	1.508 (.05)

$n = 25$	10%	50 %	Mean	90%
TAPER-EnKF	0.769 (.04)	1.668 (.13)	1.882 (.09)	3.315 (.11)
PEnKF	0.971 (.03)	1.361 (.03)	1.442 (.03)	2.026 (.04)

$n = 10$	10%	50 %	Mean	90%
TAPER-EnKF	2.659 (.07)	3.909 (.06)	3.961 (.05)	5.312 (.06)
PEnKF	1.147 (.02)	1.656 (.02)	1.735 (.02)	2.437 (.04)

2.4.2 Modified Shallow Water Equations System

While the Lorenz 96 system shows that the PEnKF has strong performance because it is successful at reducing the sampling errors and capable of learning the interactions between state variables, the system is not very realistic in that all state variables are identical and the relationship between state variables is very simplistic. We now consider a system based on the modified shallow water equations of [51], which models cloud convection with fluid dynamics equations, but is substantially computationally less expensive than actual numerical weather prediction models. The system has three types of state variables: fluid height, rain content, and horizontal wind speed.

To generate this system we use the R package "modifiedSWEQ" created by [52], and the same simulation settings as in [36]. So we always observe the rain content, but wind speed is only observed at locations where it is raining and fluid height is never observed. Explicitly for the R function $generate.xy()$, we use $h_c = 90.02, h_r = 90.4$ for the cloud and rainwater thresholds, a 0.005 rain threshold, $\sigma_r = 0.1, \sigma_u = 0.0025$ to be the standard deviation of the observation noise for rain and wind respectively, and $\mathbf{R} = \text{diag}([R_r^2 = 0.025^2 \; R_u^2 = \sigma_u^2])$ to be the estimated diagonal noise covariance matrix. All other parameters are just the default ones in the function. The initial ensemble is drawn from a free forecast run with 10000/60 time-steps between each ensemble member. We give a snapshot of the system at a random time point in Figure 2.3. There are $p = 300$ state variables

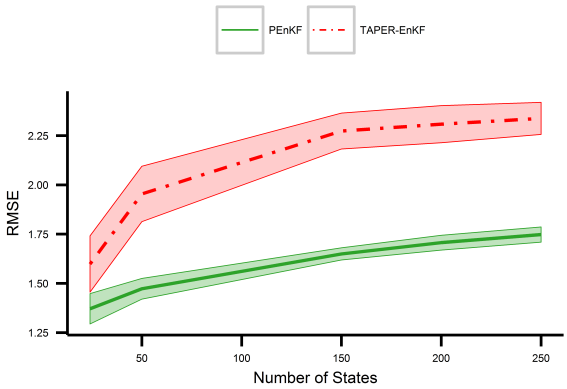


Figure 2.2: The RMSE of the TAPER-EnKF and PEnKF over 50 trials. The darker lines of each linetype are the mean and the colored areas are the 95% confidence intervals. There is clear separation between the RMSE of the two filters with the PEnKF's error as significantly smaller.

for each type, making the state space have 900 dimensions and we assimilate the system every 5 seconds for a total time period of 6 hours. Like in [36], we choose to use only 50 ensemble members and we do not perturb rain observations that are 0, because at these points there is no measurement noise.

The TAPER-EnKF uses a $3p \times 3p$ taper matrix with $c = 5$, however the entries off the $p \times p$ block diagonals are depressed (they are multiplied by 0.9). The NAIVE-LEnKPF uses the same settings as in [36], so a localization parameter of 5km, which gives the same taper matrix as the one used in the TAPER-EnKF, and an adaptive γ parameter. For the PEnKF, we set the penalty parameter to be a $3p \times 3p$ matrix, $\mathbf{\Lambda} = c_\lambda \sqrt{\boldsymbol{\lambda}_R \boldsymbol{\lambda}_R^T \log(3p)/n}$, where the first p entries of the vector $\boldsymbol{\lambda}_R$ are reference units and the rest are to scale for the perturbation noise of the different state types. So the first p are 1 (reference) for fluid height, the second p are R_u for wind, and the last p are R_r for rain. We choose the constant c_λ with eBIC like before and search in the range $[.005, 1]$.

Figure 2.4 shows the mean and approximate 95% confidence intervals of the RMSE for fluid height, wind speed, and rain content over 6 hours of time using 50 trials. The mean RMSE for all three filters are well within each others' confidence intervals for the fluid height and wind variables. For the rain variables, the mean RMSE of neither the TAPER-EnKF nor the NAIVE-LEnKPF are in the PEnKF's confidence intervals and the mean RMSE of the PEnKF is on the boundary of the other two models' confidence intervals. This strongly suggests that the PEnKF's rain error is statistically smaller than the rain errors of the other two filters. Since this simulation is not as simple as the previous ones,

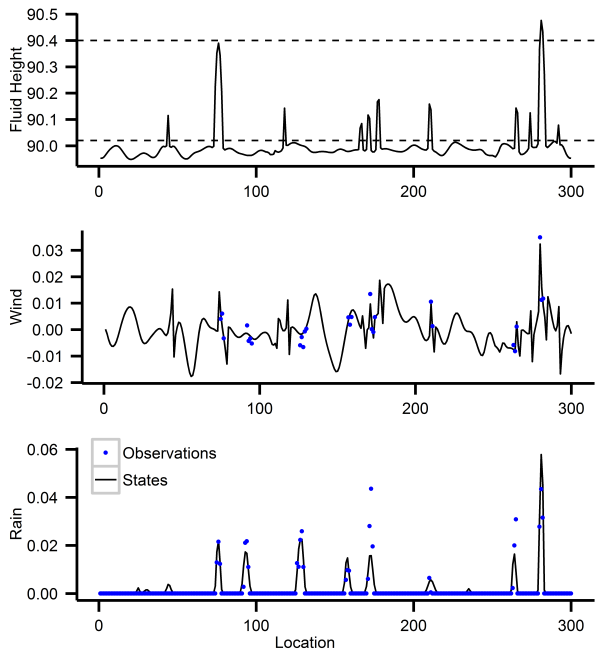


Figure 2.3: Fluid height, rain, and wind at 300 different locations at an instance of time. The blue dots are observations; rain is always observed, wind is only observed when the rain is non-zero, fluid height is never observed. The dashed lines in fluid height are the cloud and rainwater thresholds.

the interactions between the state variables are most likely not as effectively captured by the taper matrix or other localization methods, and the results from this simulation suggest that the PEnKF is learning more accurate interactions for the rain variables. We do not show the results of the BLOCK-LEnKPF of [36] because the algorithm suffered from filter divergence in 27 of the 50 trials, and in the trials where it did not fail, it performed very similar to the NAIVE-LEnKPF.

2.5 Discussion

We propose a new algorithm based on the ensemble Kalman filter that is designed for superior performance in non-linear high dimensional systems. This algorithm we call the penalized ensemble Kalman filter because it uses the popular statistical concept of penalization/regularization in order to make the problem of estimating the forecast covariance matrix well-defined (strictly convex). This in turn both decreases the sampling errors in the forecast covariance estimator by trading it off for bias and prevents filter divergence by ensuring that the estimator is positive definite. The PEnKF is computationally efficient in that it is not significantly slower than the standard EnKF algorithms and easy to implement

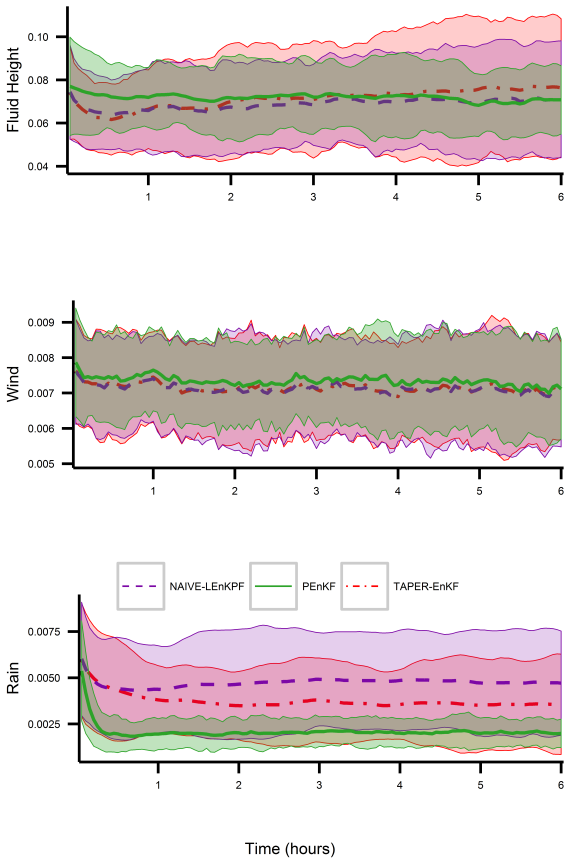


Figure 2.4: The RMSE of the TAPER-EnKF, NAIVE-LEnKPF, and PEnKF over 50 trials. The darker lines of each linetype are the mean and the colored areas are the 95% confidence intervals. All three filters are pretty indistinguishable except for the PEnKF's rain error, which is statistically smaller than the others.

since it only adds one additional step, and this step uses the well-established GLASSO algorithm available in almost any scientific computing language. We give theoretical results that prove that the Kalman gain matrix constructed from this estimator will converge to the population Kalman gain matrix under the non-simplistic asymptotic case of high-dimensional scaling, where the sample size and the dimensionality increase to infinity.

Through simulations, we show that the PEnKF can do at least as well as, and sometimes better than, localized filters that use much more prior information. We emphasize that by doing just as well as the TAPER-EnKF which has a close to optimal taper matrix, the PEnKF is effectively correctly learning the structure of interactions between the state variables. In a non-simulation setting where there is no ground-truth knowledge of the interactions between state variables, correct localization is much more difficult, making any localized filter's performance likely sub-optimal. In contrast, since the PEnKF does not use

any of this "oracle" information, its performance will not differ in this way between simulations and real-life situations. The more complicated simulation, based on the modified shallow water equations, highlights this advantage of the PEnKF through its substantial superior performance in estimating the hidden states of the rain variables. Another feature of the approach is that it seems to require less variance inflation. None was applied to any algorithm in our comparison, but the PEnKF approach never collapsed. The penalization of the inverse covariance actually produces a slight inflation on the diagonal of the covariance, which seems to help in this regard.

While we display a very naive way of searching for a good penalty parameter for the PEnKF in the simulations, it is theoretically incorrect and thus not a way to chose the truly optimal penalty parameter. We do believe deriving a specific information criterion for our PEnKF with correct theoretical properties is very important since the PEnKF can be sensitive to the penalty parameter. However, this model selection in misspecified models problem is not trivial to solve and an active topic in current statistical research. Therefore, we will leave deriving a theoretically correct information criterion for future work.

Definition.

(D1) $\sigma_{\min}(\mathbf{A})$ *and* $\sigma_{\max}(\mathbf{A})$ *denote the minimum and maximum singular values of any matrix* $\mathbf{A}$.

(D2) The spectral $||\cdot||_2$ *and Frobenius* $||\cdot||_F$ *norms are submultiplicative* $\|\mathbf{AB}\| \leq \|\mathbf{A}\|\|\mathbf{B}\|$ *and unitary invariant* $\|\mathbf{AU}\| = \|\mathbf{U}^T\mathbf{A}\| = \|\mathbf{A}^T\|$ *where* $\mathbf{U}\mathbf{U}^T = \mathbf{I}$. *So* $\|\mathbf{AB}\|_F = \|\mathbf{AUDV}^T||_F = \|\mathbf{AUD}\|_F \leq \|\mathbf{AU}\sigma_{\max}(\mathbf{D})\|_F = \|\mathbf{AU}\|_F\|\mathbf{D}\|_2 = \|\mathbf{A}||_F\|\mathbf{B}\|_2$

(D3) $||\mathbf{A}^{-1}||_2 = \sigma_{\max}(\mathbf{A}^{-1}) = 1/\sigma_{\min}(\mathbf{A})$

(D4) $\mathbf{K}$ *and* $\tilde{\mathbf{K}}$ *can be decomposed like*

$$\begin{aligned}
\mathbf{K} &= \mathbf{P}^f\mathbf{H}^T(\mathbf{H}\mathbf{P}^f\mathbf{H}^T + \mathbf{R})^{-1} \\
&= \mathbf{P}^f\mathbf{H}^T\left(\mathbf{R}^{-1} - \mathbf{R}^{-1}\mathbf{H}((\mathbf{P}^f)^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{R}^{-1}\right) \\
&= \left(\mathbf{P}^f - \mathbf{P}^f\left((\mathbf{P}^f)^{-1}(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1} + \mathbf{I}\right)^{-1}\right)\mathbf{H}^T\mathbf{R}^{-1} \\
&= \left(\mathbf{P}^f - \mathbf{P}^f\left((\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1} + \mathbf{P}^f\right)^{-1}\mathbf{P}^f\right)\mathbf{H}^T\mathbf{R}^{-1} \\
&= \left((\mathbf{P}^f)^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{R}^{-1}.
\end{aligned}$$

(D5) $\mathcal{E}$ *is the edge set corresponding to non-zeros in* $(\mathbf{P}^f)^{-1}$ *and* $\mathcal{E}^c$ *is its complement.* Γ *is the Hessian of* (2.1).

(D6) $\partial\|\boldsymbol{\Theta}\|_1$ *can be any number between -1 and 1 where* $\boldsymbol{\Theta}$ *is 0 because the derivative of an absolute value is undefined at zero. Thus, it is the set of all matrices* $\mathbf{Z} \in \mathbb{S}^{p\times p}$ *such that*

$$Z_{ij} = \begin{cases} \textit{sign}(\Theta_{ij}) & \textit{if} \quad \Theta_{ij} \neq 0 \\ \in [-1, 1] & \textit{if} \quad \Theta_{ij} = 0\,. \end{cases} \tag{2.2}$$

(D7) A bounded $4m^{\text{th}}$ *moment is the highest fourth-order moment of a random variable that is finite, where* m *is the number of fourth-order moments.*

Lemma 2..1. *Because* $\mathbf{H}$ *is a constant matrix, it does not affect the asymptotic magnitude of the modified or sample Kalman gain matricies under any norm.*

Proof of Lemma 2..1. $||\tilde{\mathbf{K}}|| \asymp ||\mathbf{H}||$ $||\tilde{\mathbf{K}}|| \asymp ||\mathbf{H}\tilde{\mathbf{K}}||$ under any norm where $\mathbf{H}\tilde{\mathbf{K}}$

$$\begin{aligned} &= \mathbf{H}\tilde{\mathbf{P}}^f\mathbf{H}^T(\mathbf{H}\tilde{\mathbf{P}}^f\mathbf{H}^T + \mathbf{R})^{-1} = \left(\mathbf{I} + \mathbf{R}(\mathbf{H}\tilde{\mathbf{P}}^f\mathbf{H}^T)^{-1}\right)^{-1} \\ &= \mathbf{R}\mathbf{R}^{-1}\left(\mathbf{R}^{-1} + (\mathbf{H}\tilde{\mathbf{P}}^f\mathbf{H}^T)^{-1}\right)^{-1}\mathbf{R}^{-1} \\ &= \mathbf{I} - \mathbf{R}(\mathbf{H}\tilde{\mathbf{P}}^f\mathbf{H}^T + \mathbf{R})^{-1} \end{aligned}$$

The same argument holds for $\hat{\mathbf{K}}$, where $(\mathbf{H}\hat{\mathbf{P}}^f\mathbf{H}^T)^{-1}$ is the pseudoinverse if the inverse does not exist. □

Assumptions. *The following assumptions are necessary for the minimizer of* (2.1) *to have good theoretical properties, [41]. Thus we assume they are true for the theorem.*

(A1) There exists some $\alpha \in (0, 1]$ *such that* $\max_{e\in\mathcal{E}^c} ||\Gamma_{e\mathcal{E}}(\Gamma_{\mathcal{E}\mathcal{E}})^{-1}||_1 \leq (1 - \alpha)$.

(A2) The ratio between the maximum and minimum eigenvalues of $\mathbf{P}^f$ *is bounded.*

(A3) The maximum ℓ_1 *norms of the rows of* $\mathbf{P}^f$ *and* $(\Gamma_{\mathcal{E}\mathcal{E}})^{-1}$ *are bounded.*

(A4) The minimum non-zero value of $(\mathbf{P}^f)^{-1}$ *is* $\Omega(\sqrt{\log(p)/n})$ *for a sub-Gaussian state vector and* $\Omega(\sqrt{p^{3/m}/n})$ *for state vectors with bounded* $4m^{\text{th}}$ *moments.*

Our assumptions are stronger than necessary, and it is common to allow the error rates to depend on the bounding constants above, but for simplicity we give the error rates only as a function of the dimensionality n, p *and sparsity* s, d *parameters.*

Proof of Theorem 2.3.1. From [53] and [41], we know that for sub-Gaussian random variables and those with bounded $4m^{\text{th}}$ moments respectively, the SSE of the sample covariance matrix are

$$\begin{cases} O\left(p^2/n\right) \\ O\left(\left(\log_2 \log_2(p)\right)^4 p(p/n)^{1-1/m}\right) \end{cases} \tag{2.3}$$

and with high probability and the SSE of $(\tilde{\mathbf{P}}^f)^{-1}$ are

$$\begin{cases} O\left(3(s+p)\log(p)/n\right) \text{ for } \quad \lambda \asymp \sqrt{3\log(p)/n} \\ O\left((s+p)p^{3/m}/n\right) \text{ for } \quad \lambda \asymp \sqrt{p^{3/m}/n} \end{cases} \tag{2.4}$$

with probability 1 - 1/p.

$$\begin{aligned} &\|\mathbf{H}\hat{\mathbf{K}} - \mathbf{H}\mathbf{K}\|_F^2 = \|\mathbf{R}(\mathbf{H}\mathbf{P}^f\mathbf{H}^T + \mathbf{R})^{-1} - \mathbf{R}(\mathbf{H}\hat{\mathbf{P}}^f\mathbf{H}^T + \mathbf{R})^{-1}\|_F^2 \\ &= \|\mathbf{R}\left((\mathbf{H}\hat{\mathbf{P}}^f\mathbf{H}^T + \mathbf{R})^{-1}\left((\mathbf{H}\hat{\mathbf{P}}^f\mathbf{H}^T + \mathbf{R}) - (\mathbf{H}\mathbf{P}^f\mathbf{H}^T + \mathbf{R})\right)(\mathbf{H}\mathbf{P}^f\mathbf{H}^T + \mathbf{R})^{-1}\right)\|_F^2 \\ &= \|\mathbf{R}(\mathbf{H}\hat{\mathbf{P}}^f\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{H}(\hat{\mathbf{P}}^f - \mathbf{P}^f)\mathbf{H}^T(\mathbf{H}\mathbf{P}^f\mathbf{H}^T + \mathbf{R})^{-1}\|_F^2 \\ &\leq \|\mathbf{R}(\mathbf{H}\hat{\mathbf{P}}^f\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{H}\|_2^2\|(\hat{\mathbf{P}}^f - \mathbf{P}^f)\|_F^2\|\mathbf{H}^T(\mathbf{H}\mathbf{P}^f\mathbf{H}^T + \mathbf{R})^{-1}\|_F^2 \end{aligned}$$

So, the second term has the rates in (2.3) and the final term is a constant. The first term is also a constant because

$$\begin{aligned} &\|\mathbf{R}(\mathbf{H}\hat{\mathbf{P}}^f\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{H}\|_2^2 \leq \|(\mathbf{H}\hat{\mathbf{P}}^f\mathbf{H}^T\mathbf{R}^{-1} + \mathbf{I})^{-1}\|_2^2\|\mathbf{H}\|_2^2 \\ &= \|\mathbf{H}\|_2^2/(\sigma_{\min}(\mathbf{H}\hat{\mathbf{P}}^f\mathbf{H}^T\mathbf{R}^{-1} + \mathbf{I}))^2 \leq \|\mathbf{H}\|_2^2. \end{aligned}$$

Thus $\|\mathbf{H}\hat{\mathbf{K}} - \mathbf{H}\mathbf{K}\|_F^2$ also has the rates in (2.3) and from Lemma 2..1, $\|\hat{\mathbf{K}} - \mathbf{K}\|_F^2$ does too.

$$\begin{aligned} &\|\tilde{\mathbf{K}} - \mathbf{K}\|_F^2 \\ &= \|\Big(((\tilde{\mathbf{P}}^f)^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1} - ((\mathbf{P}^f)^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\Big)\mathbf{H}^T\mathbf{R}^{-1}\|_F^2 \\ &= \|\Big(((\mathbf{P}^f)^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\Big(((\mathbf{P}^f)^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}) \\ &\quad -((\tilde{\mathbf{P}}^f)^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})\Big)((\tilde{\mathbf{P}}^f)^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\Big)\mathbf{H}^T\mathbf{R}^{-1}\|_F^2 \\ &\leq \|((\mathbf{P}^f)^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\|_F^2\|(\mathbf{P}^f)^{-1} - (\tilde{\mathbf{P}}^f)^{-1}\|_F^2\|\tilde{\mathbf{K}}\|_2^2 \end{aligned}$$

The first term is a constant and the second term has the rates in (2.4). The final term is also

a constant because $||\tilde{\mathbf{K}}||_2^2 \asymp ||\mathbf{H}\tilde{\mathbf{K}}||_2^2 = 1/\sigma_{\min}(\mathbf{I} + \mathbf{R}(\mathbf{H}\tilde{\mathbf{P}}^f\mathbf{H}^T)^{-1}) \leq 1$. Thus $\|\tilde{\mathbf{K}} - \mathbf{K}\|_F^2$ also has the rates in (2.4). □

CHAPTER 3

Sequential Sparse Maximum Entropy Models for Non-linear Regression

In many real-world applications, data is not collected as one batch, but sequentially over time, and often it is not possible or desirable to wait until the data is completely gathered before analyzing it. Additionally the data generating process may not be stationary, making it undesirable to use data that is too "old" in time. Thus, we propose a framework to sequentially update a model by projecting it with the principle of minimum relative entropy. Our framework allows for non-linear models using a kernel representation to represent large numbers of features and can be seen as a generalization of many popular algorithms. We show the performance and flexibility of our framework of models in both simulated and real datasets.

3.1 Introduction

With the rise of big data, where any and all data is collected, it has become increasingly important to develop sequential models that are able to continuously incorporate new data into a pre-existing model i.e. update a model. These online models are particularly crucial in applications that require real-time responses such as wearable devices that continuously give feedback to the user. To this end, we propose a sequential framework to update the probabilistic regression model built from Jayne's principle of maximum entropy.

This paper discusses how the problem of minimizing the constrained relative entropy for a given model can be cast as recursive Bayesian estimation where the likelihood function is a log-linear model formed from a series of constraints and weighted by Lagrange multipliers. For regression problems with Gaussian noise, the optimized model shares similarities with online regression algorithms, which have been previously studied in []. This framework is built on concepts from the maximum entropy discrimination framework of

[54], which has been extended for sequential data in [55] where they present an online maximum margin classifier specifically for supervised and semi-supervised binary classification problems.

We are interested in situations where we receive a stream of data $\boldsymbol{X}_{(1)}, \boldsymbol{X}_{(2)}, \ldots$ over time t where each row of $\boldsymbol{X}_{(t)}$ is a sample in the $n \times p$ dimensional matrix with p denoting the number of predictor variables and $n = n_{(t)}$ may vary with time. Given these features, the data has a $n \times 1$ vector corresponding to the continuous response variable $\boldsymbol{y}_{(t)}$. For notational simplicity, we will use the $\tilde{}$ over features to indicate that there is an intercept term $\tilde{x} = [1\, x]$ and the subscript $(1 : \tau)$ to denote a vertical concatenation of samples up to time τ e.g. $\boldsymbol{X}_{(1:2)} = [\boldsymbol{X}_{(1)}, \boldsymbol{X}_{(2)}]$ is a $(n_{(1)} + n_{(2)}) \times p$ matrix. The rest of the paper is organized as follows: Section 2 gives a review of online linear and non-linear model algorithms. Section 3 discusses how to formulate a regression model in the minimum relative entropy framework and how this framework gives rise to a natural updating procedure. Section 4 validates the method by simulation and we discuss an application to a dataset of.

3.2 Review of Filtering

Methods that are capable of incorporating new data over time have numerous names in literature including: filtering, streaming, sequential modeling, and online learning, but almost all of these methods have the underlying assumption that the data is generated according to a Markov process. In this section we will review modeling under a stationary system (parameter estimation) for both linear and non-linearly generated observations and discuss modifications for modeling in a dynamic system.

3.2.1 Review of Linear Filters

Online learning for linear regression is a well studied problem in the literature. In linear regression, the response y_i and predictor variables $\boldsymbol{X}_i$ are assumed to have a linear relationship with additive Gaussian noise. The observed or measurement model is $y_i = \boldsymbol{X}_i\boldsymbol{\theta} + b + e_i$ where the noise terms e_i are independent, identically distributed Gaussian random variables with variance ϵ. Thus, minimizing the squared loss $||\boldsymbol{y} - (\boldsymbol{X}\boldsymbol{\theta} + b)||_2^2$ for weights $\boldsymbol{\theta}$ and intercept b is equivalent to maximizing the conditional Gaussian likelihood $y_i|\boldsymbol{X}_i \sim N(\boldsymbol{X}_i\boldsymbol{\theta} + b, \epsilon)$.

Solving for the maximum likelihood estimators (MLE) by applying stochastic gradient descent (ascent) on the log likelihood function $\ell(\cdot)$ leads to the least mean squares (LMS)

algorithm which has updates

$$\boldsymbol{\theta}' = \boldsymbol{\theta} + \frac{1}{\epsilon}\nabla_{\theta}\ell = \boldsymbol{\theta} + \frac{1}{\epsilon}\boldsymbol{X}_{(t)}^{T}(\boldsymbol{y}_{(t)} - \boldsymbol{X}_{(t)}\boldsymbol{\theta})$$
$$b' = b + \frac{1}{\epsilon}\nabla_{b}\ell = b + \frac{1}{\epsilon}\sum_{i=1}^{n}(y_{(t)i} - b)$$

However, this filter is only optimal when using an optimal step size $\frac{1}{\epsilon}$ and typically requires multiple passes through the mini-batches of data $\{\boldsymbol{X}_{(t)}, \boldsymbol{y}_{(t)}\}$. This is not ideal in a streaming setting where previous data is typically discarded or there are some constraints on the amount of previous data that can be stored.

If instead we adaptively estimated the step size, the recursive least squares (RLS) algorithm finds the optimum in a single pass through of the data. The RLS algorithm has update

$$\boldsymbol{\Theta}' = \boldsymbol{\Theta} + (\boldsymbol{P}'/\epsilon)\tilde{\boldsymbol{X}}_{(t)}^{T}(\boldsymbol{y}_{(t)} - \boldsymbol{\Theta})$$

where is $\boldsymbol{\Theta}$ is a $(p+1) \times 1$ vertically concatenated vector $[b, \boldsymbol{\theta}]$, $\tilde{\boldsymbol{X}}_{(t)} = [\mathbf{1}\, \boldsymbol{X}_{(t)}]$ is the feature vectors with an intercept term, and $\boldsymbol{P}$ is recursively estimated as

$$\boldsymbol{P}' = \left(\boldsymbol{P} - \boldsymbol{P}\tilde{\boldsymbol{X}}_{(t)}^{T}(\epsilon \mathrm{I} + \tilde{\boldsymbol{X}}_{(t)}\boldsymbol{P}\tilde{\boldsymbol{X}}_{(t)}^{T})^{-1}\tilde{\boldsymbol{X}}_{(t)}\boldsymbol{P}\right)$$

instead of set to be the identity in the LMS algorithm. Similar to the connections between minimizing the least squares problem and Gaussian maximum likelihood, the RLS algorithm can be viewed probabilistically as a Gaussian filter. Thus after sweeping through the τ mini-batches of data, the final estimator $\boldsymbol{\Theta}$ will be equivalent to solving for a MAP estimator of the full data problem where the prior is $\boldsymbol{\Theta} \sim N(\mathbf{0}, \epsilon \mathrm{I})$ or a MLE estimator as $\epsilon \to \infty$ and the prior tends to non-informative.

The RLS algorithm can also be viewed as a Newton method because it uses second order information. $\boldsymbol{P}'/\epsilon$ is an estimate of the inverse Hessian, which becomes clear when the recursive update is rewritten as $\boldsymbol{P}'^{-1} = \boldsymbol{P}^{-1} + (1/\epsilon)\tilde{\boldsymbol{X}}_{(t)}^{T}\tilde{\boldsymbol{X}}_{(t)}$. Thus the step size of the RLS algorithm $(\epsilon\boldsymbol{P}^{-1} + \tilde{\boldsymbol{X}}_{(t)}^{T}\tilde{\boldsymbol{X}}_{(t)})^{-1}$ is a weighted sum of the previous estimate and the amount of variance in the current data where ϵ can be interpreted as a forgetting factor.

3.2.2 Review of Non-Linear Filters

Now we consider the scenario where the measurement model, $y_i = f(\boldsymbol{X}_i)\boldsymbol{\theta} + b + e_i$, is no longer a direct linear relationship, but linear with respect to a function $f(\cdot)$ that maps the

original features to a new feature space with corresponding weights $\boldsymbol{\theta}$. Often this function can only be explicitly described through its inner products, so we define a parameter transformation $\omega(\boldsymbol{X}_i) = f(\boldsymbol{X}_i)\boldsymbol{\theta}$ and kernel function as $k(x, x') = \langle f(x), f(x') \rangle$. As e_i are still additive Gaussian noise, an ideal filter still aims to minimize the squared loss between y_i and $\omega(\boldsymbol{X}_i) + b$.

Because the least squares problem can be explicitly defined with inner products, the linearly optimal RLS algorithm can be extended to solve problems with a non-linear function $\omega(\cdot)$ using the kernel trick. This is first proposed in [56] where they derive a kernel based version of the RLS algorithm, dubbed KRLS, which is capable of recursively solving nonlinear least squares problems. However, because the KRLS algorithm is recursive, the computational complexity explodes as sample size increases. In the next section, we will explicitly show how this recursive kernel not only has computational issues, but is also not ideal in a streaming setting as it requires evaluations of the kernel function between any new point x' and all the previous data $\{\mathcal{D}_{(t)}\}_{t=1}^{\tau}$.

[56] and [57] deal with the problems of the recursive kernel by proposing algorithms that are sparse and thus require evaluations with fewer previous data points. They attain this sparsity by construction using their proposed approximate linear dependence (ALD) condition, which only cares about data points that are not approximately linear combinations of previous data points. Given a dictionary of m data points that are a subset of the total training set, a new point x' is only admitted into the dictionary if it violates the condition

$$\min_{\boldsymbol{\alpha}} \boldsymbol{\alpha}^T k(\boldsymbol{X}_{(t)}, \boldsymbol{X}_{(t)})\boldsymbol{\alpha} - 2\boldsymbol{\alpha}^T k(\boldsymbol{X}_{(t)}, x') + k(x', x') \leq tol$$

where the sparsity level (size of m) depends on the tolerance level. This condition is used to construct the kernel for an online SVR-like algorithm and an online KRLS algorithm in [57] and [56] respectively.

[58] approach the non-linear least squares problem using an online Gaussian Process (GP) model where they are also concerned with the linear dependence of the features. Similar to the ALD condition, they only include the new point x' into their "basis vector set" if it violates

$$k(x', x') - k(x', \boldsymbol{X}_{(t)})k(\boldsymbol{X}_{(t)}, \boldsymbol{X}_{(t)})^{-1}k(\boldsymbol{X}_{(t)}, x') < tol$$

and they also use this value to decide which inputs to keep when the set reaches the maximum size. [59] also have a sparse online GP model, but they construct sparsity (zeros) in the covariance matrix using compactified kernel functions.

3.2.3 Filtering in Dynamic Systems

Models for online parameter estimation assume a very specific Markov process where the observed response is generated according to the measurement model previously discussed, but the parameters of this measurement model are the latent variables in a stationary system model and thus constant. As a Gaussian filter, the RLS algorithm has obvious connections to this special stationary case of the Kalman filter; $(\boldsymbol{P}/\epsilon)\tilde{\boldsymbol{X}}_{(t)}^{T}$ is the Kalman gain matrix and the updates for $\boldsymbol{\Theta}'$, $\boldsymbol{P}'$ are the *a posteriori* state and covariance estimates respectively. From this Kalman filter perspective, it is clear why the RLS algorithm converges to the optimal parameter estimates in a single pass through of the data because it is well known that a Kalman filter is a Bayesian optimal filter for linear systems.

This stationary Markov process is often not applicable for real data though. The real world is not frozen in time, so data is often not generated from fixed constants, but instead parameters that slowly change over time. When the system model that dictates how the parameters evolve is linear and known, then the time evolution of the latent variable distribution is known resulting in the typical Kalman filter formulation. When the system model is not known, it is no longer obvious how to account for the changes in parameters over time. However, it is assumed that the current data observations are more reflective of the current state of the parameters than previous data. This is accounted for in models through a forgetting factor, which puts less weight on the *a prior* distribution as its parameters are estimated using the previous data. In Section 3, we will discuss this forgetting factor in much greater detail including the geometric intuition that naturally arises in our framework.

3.3 Sequential Maximum Entropy for Regression

In this section we will present algorithms that solve a dual problem to that of the filters discussed in the previous section. We call them dual algorithms because instead of solving the MLE or MAP, they solve a constrained relative entropy minimization, and the duality between maximum entropy and maximum likelihood has been well-studied. The advantage of this dual entropy problem is that it gives an intuitive framework on how to update a model. Solving the constrained relative entropy minimization gives the closest, in terms of Kullback-Leibler (KL) divergence, distribution to a prior subject to a set of data defined moment constraints. Thus we can update a model trained on previous data by projecting it onto a set that is constrained by the current data. In the following subsections, we will first show that this consecutive projecting can create optimal linear filters. Then we will extend

it to show that projecting can also create optimal non-linear filters. Finally we present a computationally efficient approximation to the optimal non-linear filter with bounded error.

3.3.1 Optimal Linear Filtering with MER

Similar to Bayesian conjugate priors, there exist relative entropy conjugate priors that with certain constraints produce optimal constrained relative entropy densities from the same parametric family as the prior. So using this conjugacy, we can update models in a similar fashion to Bayesian filtering. A general form of the constrained relative entropy problem is presented below.

Let $\mathrm{Q}(\psi)$ be a previously trained statistical model parameterized by ψ. Then the solution to

$$\begin{aligned} &\arg\min_{\mathrm{P}(\psi|\mathcal{D})} \mathrm{KL}(\mathrm{P}(\psi|\mathcal{D})||\mathrm{Q}(\psi)) \\ &\qquad \text{subject to} \\ &\mathrm{E}_{\mathrm{P}(\psi|\mathcal{D})}(L(\mathcal{D},\psi)) \in \mathcal{C} \\ &\int \mathrm{P}(\psi|\mathcal{D})\, d\psi = 1 \end{aligned}$$

is the closest statistical model to the previously trained model that has expected loss L with new data $\mathcal{D}$ that lies in some cost set $\mathcal{C}$. From [60], we know that the solution has the form $\mathrm{P}(\psi|\mathcal{D},\xi) = \frac{\mathrm{Q}(\psi)}{Z(\xi)} \exp\{\langle \xi, L(\mathcal{D},\psi)\rangle\}$ which is a exponential family distribution parametrized by ξ (natural parameters) with partition function $Z(\xi)$. As the ξ_i correspond to each constraint, they can also be thought of as Lagrange multipliers.

The authors of [61] show that, if the prior distribution is from the exponential family and the constraints are over only sufficient statistics, then the density that optimizes the constrained relative entropy problem is also a member of the exponential family. From information geometry, we know that the constraints induce an m-flat subspace and any exponential family manifold is m-flat in its expectation parameters. Thus when the constraints are linear in the sufficient statistics, they lie on the exponential family manifold in the dual expectation parameterized coordinate system. These are essentially guidelines on how to choose a model family and constraints so that sequentially solving a series of constrained relative entropy problems is tractable; each solution produces a model that can be projected in the next problem.

In [55], they present a constrained relative entropy minimization problem whose solution performs sequential binary classification in both supervised and semi-supervised sce-

narios. However, in this paper, we are concerned with regression and not classification. The following theorem describes a choice of relative entropy conjugate priors and constraints that produce a solution that is a regression model. This model we call sequential Maximum Entropy Regression (MER).

Theorem 3.3.1. *Let the prior at $t = 1$ factorize into $\mathrm{P}_0(\boldsymbol{\theta}) = N(\mathbf{0}, \mathrm{I}), \mathrm{P}_0(b) = N(\mathbf{0}, \sigma^2)$, $\mathrm{P}_0(\gamma_i) = N(0, \epsilon)$ and, $\mathrm{P}_0(\lambda) = Exp.(\infty)$ with constraints*

$$\mathrm{E}(y_{(t)i} - (\boldsymbol{X}_{(t)i}\boldsymbol{\theta} + b) - \gamma_i) = 0 \ \forall i$$
$$\mathrm{E}(\boldsymbol{\Theta}^T \tilde{\boldsymbol{X}}_{(t)}^T \tilde{\boldsymbol{X}}_{(t)} \boldsymbol{\Theta} - \lambda) \leq 0$$

Then at time point τ, the optimal posterior factorizes into distributions from the same families. The updated model for the regression weights is

$$\mathrm{P}\left(\boldsymbol{\Theta} | \boldsymbol{X}_{(1:\tau)}\right) = N\left(\tilde{\boldsymbol{H}}_{(\tau)}^{-1}(\tilde{\boldsymbol{H}}_{(\tau-1)}\boldsymbol{\mu}_{(\tau-1)} + \tilde{\boldsymbol{X}}_{(\tau)}^T \hat{\boldsymbol{\alpha}}_{(\tau)}), \tilde{\boldsymbol{H}}_{(\tau)}^{-1}\right)$$

where $\boldsymbol{\mu}_{\tau-1}$ is the mean of the previous model trained on data up to time $\tau - 1$ and $\tilde{\boldsymbol{H}}_{(\tau)} = \tilde{\boldsymbol{H}}_{(\tau-1)} + 2\beta_{(\tau)}\tilde{\boldsymbol{X}}_{(\tau)}^T \tilde{\boldsymbol{X}}_{(\tau)}$. The distributions of the other parameters do not depend on the data, thus the initial priors are used every for time point.

Because the initial prior belongs to the exponential family and the constraints are linear in the corresponding sufficient statistics $\boldsymbol{\Theta}, \boldsymbol{\Theta}\boldsymbol{\Theta}^T, \gamma_i, \lambda$, they satisfy the conjugacy requirements. The constraints are over a parametric family of regression functions $\boldsymbol{X}_{(t)}\boldsymbol{\theta} + b$ to ensure that the optimal solution to the constrained relative entropy minimization problem will be a regression model. The first set of constraints ensure the expected squared loss lies in an epsilon ball and the last constraint allows the covariance to be unknown, but bounds its to be finite.

The optimal Lagrange multipliers solve
$\underset{\boldsymbol{\alpha}_{(\tau)}, \beta_{(\tau)}}{\arg\min} - \log(Z(\boldsymbol{\alpha}_{(\tau)}, \beta_{(\tau)}))$, but when $\beta_{(\tau)}$ is given or set to a fixed value, then $\hat{\boldsymbol{\alpha}}_{(\tau)}$ can be solved in closed form. This sequential MER model has very similar form to the Kalman Filter perspective of RLS algorithm and for a certain choice β they have exactly the same solution allowing the sequential MER model to also perform optimal filtering.

Corollary 3.3.1.1. *When all previous $\beta_{(1)}, \ldots, \beta_{(\tau-1)}$ are set to $\frac{1}{2\epsilon}$ and $\beta_{(\tau)} = 0$, so that*

$$\hat{\boldsymbol{\alpha}}_{(\tau)} = \left(\epsilon \mathrm{I} + \tilde{\boldsymbol{X}}_{(\tau)} \tilde{\boldsymbol{H}}_{(\tau)}^{-1} \tilde{\boldsymbol{X}}_{(\tau)}^T\right)^{-1} \left(\boldsymbol{y}_{(\tau)} - \tilde{\boldsymbol{X}}_{(\tau)} \boldsymbol{\mu}_{(\tau-1)}\right)$$

then the model described in Theorem 3.3.1 has mean parameter updates that are optimal.

The $\beta_{(t)}$ Lagrange multipliers appear only in $\tilde{\boldsymbol{H}}_{(\tau)}$, which is the inverse covariance parameter of the model. Because the distribution $\mathrm{P}(y|x)$ is Gaussian, $\tilde{\boldsymbol{H}}_{(\tau)}$ is the negative Hessian of the log likelihood function of $n * \tau$ samples and gives an empirical estimate of the curvature of the distribution. Thus when decomposed as $\tilde{\boldsymbol{H}}_{(\tau)} = (\boldsymbol{\Sigma}_{(0)} + \sum_{t=1}^{\tau} 2\beta_{(t)}\tilde{\boldsymbol{X}}_{(t)}^T\tilde{\boldsymbol{X}}_{(t)})$, it is clear that the $\beta_{(t)}$ control how much weight each mini-batch is given for estimating the curvature. If all $\beta_{(t)} = 0$ and the prior $\mathrm{P}_0(\boldsymbol{\theta}) = N(\mathbf{0}, \eta\mathrm{I})$ is used at the first time point, the MER model will have equivalent performance to the LMS algorithm. In this case, the model no longer estimates the curvature using the data, but instead has constant curvature only along the orthogonal basis. When the system model is dynamic, the distribution $\mathrm{P}(y|x)$ is slowly changing over time and it is no longer guaranteed that its curvature does not change. Thus the relevance of each mini-batch of data is no longer equal; data from points closer in time are more representative of the current distribution than that data from the far past. Thus depending on how quickly we believe the system is changing, we can set the $\beta_{(t)}$ to decay or drop to 0 after a certain period.

3.3.2 Optimal Non-Linear Filtering with MER

Like the kernel version of the RLS algorithm, we can also express the MER algorithm using only inner products for the kernel trick. Using the parameter transformation $\omega(\tilde{\boldsymbol{X}}_i) = f(\boldsymbol{X}_i)\boldsymbol{\theta} + b$, we can rewrite the model described in Theorem 3.3.1 so that it is expressed only in kernel functions.

Proposition 1. *Let the prior at $t = 1$ factorize into $\mathrm{P}_0(\omega(\tilde{\boldsymbol{X}}_{(1)})) = N(\mathbf{0}, k(\tilde{\boldsymbol{X}}_{(1)}, \tilde{\boldsymbol{X}}_{(1)}))$, $\mathrm{P}_0(\gamma_i) = N(0, \epsilon)$ and, $\mathrm{P}_0(\lambda) = Exp.(\infty)$ with constraints*

$$\begin{aligned} &\mathrm{E}(y_{(t)i} - \omega(\tilde{\boldsymbol{X}}_{(t)i}) - \gamma_i) = 0 \ \forall i \\ &\mathrm{E}(\omega(\tilde{\boldsymbol{X}}_{(t)})^T\omega(\tilde{\boldsymbol{X}}_{(t)}) - \lambda) \leq 0 \end{aligned} \tag{3.1}$$

Then at time point τ, the updated model for the regression function is $\mathrm{P}(\omega(\tilde{\boldsymbol{X}}_{(\tau)})|\boldsymbol{X}_{(1:\tau)}) = N(\mu_{(\tau)}(\tilde{\boldsymbol{X}}_{(\tau)}), k_{(\tau)}(\tilde{\boldsymbol{X}}_{(\tau)}, \tilde{\boldsymbol{X}}_{(\tau)}))$ and the distributions for the other parameters are the same as in Theorem 3.3.1. When $\beta_{(\tau)} = 0$ and

$$\hat{\boldsymbol{\alpha}}_{(\tau)} = \left(\epsilon\mathrm{I} + k_{(\tau-1)}(\tilde{\boldsymbol{X}}_{(\tau)}, \tilde{\boldsymbol{X}}_{(\tau)})\right)^{-1}\left(\boldsymbol{y}_{(\tau)} - \mu_{(\tau-1)}(\tilde{\boldsymbol{X}}_{(\tau)})\right)$$

the mean and kernel functions are

$$\mu_{(\tau)}(\tilde{\boldsymbol{x}}) = \mu_{(\tau-1)}(\tilde{\boldsymbol{x}}) + k_{(\tau)}(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{X}}_{(\tau)})\hat{\boldsymbol{\alpha}}_{(\tau)}$$
$$k_{(\tau)}(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}') = k(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}') - k(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{X}}_{(1:\tau)})\big(\frac{1}{2B} + k(\tilde{\boldsymbol{X}}_{(1:\tau)}, \tilde{\boldsymbol{X}}_{(1:\tau)})\big)^{-1} k(\tilde{\boldsymbol{X}}_{(1:\tau)}, \tilde{\boldsymbol{x}}')$$

where $B = diag(\beta_{(1:\tau)})$ *and the kernel function can also be recursively defined as*

$$k_{(\tau)}(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}') = k_{(\tau-1)}(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}') - k_{(\tau-1)}(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{X}}_{(\tau)}) \left(\frac{1}{2\beta_{(\tau)}} \mathrm{I} + k_{(\tau-1)}(\tilde{\boldsymbol{X}}_{(\tau)}, \tilde{\boldsymbol{X}}_{(\tau)}) \right)^{-1} k_{(\tau-1)}(\tilde{\boldsymbol{X}}_{(\tau)}, \tilde{\boldsymbol{x}}')$$

Like in the linear case, $\beta_{(1)}, \ldots, \beta_{(\tau-1)} = \frac{1}{2\epsilon}$ and will provide the exact solution as solving the non-linear least squares problem in a batch setting. In general the impact of the $\beta_{(t)}$'s is similar to the linear case, which is revealed when the kernel function is written as $k_{(\tau)}(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}') = 1 + \langle f(\boldsymbol{x}), \boldsymbol{H}_{(\tau)}^{-1} f(\boldsymbol{x}') \rangle$. Recall $\boldsymbol{H}_{(\tau)}/(n * \tau)$ is the empirical covariance matrix (or operator) of all the data up to time τ. Thus it weights the kernel function so that dimensions with high variance affect the similarity measure less.

However, unlike in the linear case, the computational complexity and storage are no longer efficient. The mean is now a function that requires evaluating the new data against all previous data, which has $\mathcal{O}(\tau\, n_t)$ computational complexity. Additionally all previous data must be stored, which in a streaming setting is undesirable, especially when the storage costs grow with time. These costs do not account for the mean containing a kernel function that is a recursive function of previous kernel functions where all these functions must be evaluated on the new data making the computational complexity explode; 4^τ evaluations of $\mathcal{O}(pn_t^2 + n_t^3)$ complexity. Evaluating the kernel function in its non-recursive form, is also often not feasible as it requires storing a $(\tau\, n_t) \times (\tau\, n_t)$ matrix in memory and $\mathcal{O}(p(\tau\, n_t)^2 + (\tau\, n_t)^3)$ computational complexity. The computational burden of the recursive kernel problem can be removed by setting all $\beta_{(t)}$'s to 0, which produces a kernel version of the LMS algorithm. However the model will then contain all the drawbacks the LMS algorithm suffers from and the issue of storing all the data still remains.

3.3.3 Approximately Optimal Non-Linear Filtering with MER

Because of the computational complexity and storage issues of evaluating a function with respect to previous data, it is desirable to evaluate the function with the smallest set of data possible. This idea of reducing the training data to a sparse set or dictionary has been previously proposed in [56] and [58]. However, unlike the previously proposed methods, we will encourage sparsity not through explicit construction, but implicitly in our algorithm.

It is well known that penalizing predictions only when they lie outside a margin lead to models that are sparse in the number of training points with non-zero weights e.g. support vector machines (SVM), support vector regression (SVR). So by replacing the squared loss constraints in Proposition 1 with margin constraints, we propose a sparse version of the kernel MER model.

Proposition 2. *Let the prior at $t = 1$ factorize into $\mathrm{P}_0(\omega(\boldsymbol{X}_{(1)})) = N(\mathbf{0}, k(\boldsymbol{X}_{(1)}, \boldsymbol{X}_{(1)}))$, $\mathrm{P}_0(b) = N(0, \infty), \mathrm{P}_0(\gamma_i) = \mathrm{P}_0(\gamma_i') = C_{(1)} e^{C_{(1)}(\epsilon_{(1)} - \gamma_i)} \mathcal{I}(\gamma_i \in [\epsilon_{(1)}, \infty))$ and, $\mathrm{P}_0(\lambda) = Exp.(\infty)$ with constraints*

$$\begin{aligned} &\mathrm{E}(y_{(t)i} - (\omega(\boldsymbol{X}_{(t)i}) + b) - \gamma_i) \leq 0 \ \forall i \\ &\mathrm{E}(y_{(t)i} - (\omega(\boldsymbol{X}_{(t)i}) + b) + \gamma_i') \geq 0 \ \forall i \\ &\mathrm{E}(\omega(\boldsymbol{X}_{(t)})^T \omega(\boldsymbol{X}_{(t)}) - \lambda) \leq 0 \end{aligned} \tag{3.2}$$

Then at time point τ, the optimal posterior factorizes similarly to in Proposition 1 into distributions from the same families. The updated model for the regression function has the same mean and kernel functions where $\hat{\boldsymbol{\alpha}}_{(\tau)} = \boldsymbol{\alpha} - \boldsymbol{\alpha}'$ and $\boldsymbol{\alpha}, \boldsymbol{\alpha}'$ maximize

$$\begin{aligned} &-0.5(\boldsymbol{\alpha} - \boldsymbol{\alpha}')^T k_{(\tau-1)}(\boldsymbol{X}_{(\tau)}, \boldsymbol{X}_{(\tau)})(\boldsymbol{\alpha} - \boldsymbol{\alpha}') + (\boldsymbol{\alpha} - \boldsymbol{\alpha}')^T (\boldsymbol{y}_{(\tau)} - \mu_{(\tau-1)}(\boldsymbol{X}_{(\tau)})) \\ &- \epsilon_{(\tau)}(\boldsymbol{\alpha} + \boldsymbol{\alpha}') + \sum_{i=1}^{n_{(\tau)}} \log(1 - \frac{\alpha_i}{C_{(\tau)}}) + \log(1 - \frac{\alpha_i'}{C_{(\tau)}}) \\ &\textit{subject to } \sum_{i=1}^{n_{(\tau)}} \alpha_i - \alpha_i' = 0 \quad \alpha_i, \alpha_i' \geq 0 \ \forall i \end{aligned} \tag{3.3}$$

and to center the predictions, the optimal bias at every time point $b_{(\tau)} = \hat{b}$ are the solution to

$$\arg\min_b \sum_{i=1}^{n_{(\tau)}} \left| \left(y_{(\tau)i} - \mu_{(\tau)}(\boldsymbol{X}_{(\tau)i}) \right) - b \right|$$

These margin constraints and a similar prior are first proposed in [62]. While the updated model above is Gaussian with the same parameters functions as the one in Proposition 1, the crucial difference lies in $\hat{\boldsymbol{\alpha}}_{(\tau)}$, which solve a different objective function that encourages sparsity. At the first time point, when $\boldsymbol{\mu}_{(\tau-1)}(\boldsymbol{X}_{(\tau)}) = 0$, the objective is similar to the dual objective of SVR; there are log barrier terms to prevent α_i, α_i' from being greater than a cost $C_{(\tau)}$ instead of the inequality constraints. So similarly $C_{(\tau)}$ is the cost imposed on observations that lie outside of the $\epsilon_{(\tau)}$ sized margin. And the sparsity level of $\boldsymbol{\alpha} - \boldsymbol{\alpha}'$

will depend on these hyper-parameters. At earlier time points, when less data has been observed, it is important to establish a good model fit, so a complex model that is not as sparse is often favorable. When $\epsilon_{(t)}$ is small and $C_{(t)}$ is large, the model will be more complex as there is a high penalty for being even slightly off from the regression mean. However, once an accurate model is established, very little new information is contained in new points and it is desirable to relax $\epsilon_{(t)}$ and shrink $C_{(t)}$ making the model more robust to noise.

After the first time point, the objective in (3.3) differs from the SVR objective on account of updating a previously trained model. The additional term $\mu_{(\tau-1)}(\boldsymbol{X}_{(\tau)})$ is the mean of the previously trained model evaluated with the current data i.e. the *a priori* prediction. Instead of measuring the importance of a new point through the relationship between the features of the new point and features of the previous observed data like in [56] and [58], the model in Proposition 2 measures the importance of a new point by its difficulty in being predicted by the previous model. The part of the objective

$$(\alpha_i - \alpha_i')(y_{(\tau)i} - \mu_{(\tau-1)}(\boldsymbol{X}_{(\tau)i}))$$

puts higher weight on the α_i, α_i' pair where the residual between the current data and the previous model prediction is large. Thus unlike the previously proposed models that are agnostic to the response in their sparsity condition, the model in Proposition 2 is solely concerned about the relationship between current and previous samples through the response. This, along with the margin constraints, indicates that the sparse MER algorithm encourages sparsity in points where the model already has good prediction or whose response observations lie within a margin. Making it a far more natural criteria for sparsity because the goal of regression is to model the map between response and features.

Enforcing sparsity through the response is especially relevant in dynamic systems, where as previously discussed, the data are no longer identically distributed over time t. For example, consider the extreme case where the system has some of the features that have no relationship with the response (the feature weights are 0), and these useless features suddenly change to have high variance. So while the model fit is still good because the features have no impact on the response, an algorithm that only considers the dependence of samples through their features will start unnecessarily growing of their dictionary. Whereas the model in Proposition 2 would ignore any new samples that do not contain new information about the response.

While the constraints in Proposition 2 admit a sparse model, its covariance is still a recursive kernel function. The computational complexity and storage issues still remain because the mean function depends on the kernel function. Because the Lagrange multipli-

ers $(\alpha_i - \alpha_i')$ are only non-zero for observations that are "difficult", we can assume that the points that have corresponding non-zero weights are the ones that contain all the important information in the full dataset. Thus at every time point τ, the prior model only needs to trained on this subset of the data.

Theorem 3.3.2. *Let the priors at $t = 1$ be the same as in Proposition 2. Let at any point $t > 1$, the prior for the regression function update to* $\mathrm{P}_0(\omega(\boldsymbol{X}_{(t)})|\boldsymbol{X}_{\mathcal{S}}) = N(\mu_{\mathcal{S}}(\boldsymbol{X}_{(t)}), k_{\mathcal{S}}(\boldsymbol{X}_{(t)}, \boldsymbol{X}_{(t)}))$ *where* $k_{\mathcal{S}}(\boldsymbol{x}, \boldsymbol{x}') = \langle f(\boldsymbol{x}), \boldsymbol{H}_{\mathcal{S}}^{-1} f(\boldsymbol{x}')\rangle$. *Then the posterior that minimizes the constrained relative entropy problem with the constraints in* (3.2) *factorizes into distributions from the same families where when $\beta_{(t)} = 0$,* $\mathrm{P}(\omega(\boldsymbol{X}_{(t)})|\boldsymbol{X}_s, \boldsymbol{X}_{\mathcal{S}})$ *is Gaussian with parameters*

$$\mu_{s|\mathcal{S}}(\boldsymbol{x}) = \mu_{\mathcal{S}}(\boldsymbol{x}) + k_{\mathcal{S}}(\boldsymbol{x}, \boldsymbol{X}_s)\hat{\boldsymbol{\alpha}}_{s|\mathcal{S}}$$
$$k_{s|\mathcal{S}}(\boldsymbol{x}, \boldsymbol{x}') = \langle f(\boldsymbol{x}), (\boldsymbol{H}_{\mathcal{S}} + 2\beta_{(t)}\boldsymbol{X}_s^T\boldsymbol{X}_s)^{-1} f(\boldsymbol{x}')\rangle$$

where $\boldsymbol{X}_s$ are points with corresponding non-zero $\hat{\boldsymbol{\alpha}}_{s|\mathcal{S}}$ that maximize (3.3). *This model is an approximately optimal filter whose predictions can be bounded in ℓ_2 norm from the optimal batch solution*

$$||k(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{X}}_{(1:\tau)})\hat{\boldsymbol{\alpha}}_{(1:\tau)} - (\mu_{s|\mathcal{S}}(\boldsymbol{x}) + \hat{b})||_2^2 \lesssim \delta$$

when $\boldsymbol{X}_{(t)i}$ are independent and identically distributed for all t.

Note that for stationary systems with Gaussian noise, the batch model with natural parameters $\hat{\boldsymbol{\alpha}}_{(1:\tau)}$ achieves the Cramer-Rao lower bound. In this case, the mean squared error (MSE) of any prediction $||y - (\mu_{s|\mathcal{S}}(\boldsymbol{x}) + \hat{b})||_2^2$ can be bounded by $\delta + \sigma^2$ where σ^2 is the variance of the noise. But in general, Theorem 3.3.2 indicates that as long as the mean squared prediction error of the batch model is small, the error of the proposed sparse model will also be small. For dynamic systems, proving the theorem holds is much more difficult, but we empirically verify in simulations that the error is still small.

Let $n_{\mathcal{S}} = |\{i : \hat{\boldsymbol{\alpha}}_{(t)i} \neq 0\}|$ be the cardinality of the non-zero subset where $n_{\mathcal{S}} << \tau\, n_t$. The mean function now requires $O(p\, n_{\mathcal{S}} + n_{\mathcal{S}}^2)$ operations to predict a point and because the $\hat{\boldsymbol{\alpha}}_{(t)i}$ are sparse, the algorithm only needs to store the corresponding $\boldsymbol{X}_{(t)i}$. Solving (3.3) for the optimal $\boldsymbol{\alpha}_s$ has polynomial complexity in n_t if a conic interior point method is used, but if the log barrier terms are relaxed into hard inequality constraints, it can be solved in linear time with sequential minimal optimization (SMO). Updating the kernel function is more expensive, requiring $O(p\, n_{\mathcal{S}}^2 + n_{\mathcal{S}}^3)$ operations, but this function only needs to updated when either the previous $\beta_{(t)}$ change or there are non-zero $\hat{\boldsymbol{\alpha}}_{(s)}$. Additionally, the storage costs are

very manageable, requiring storing a $(n_{\mathcal{S}}) \times (n_{\mathcal{S}})$ matrix in memory. This computationally efficient model, which we call the sparse MER model is summarized below.

Algorithm 3.1 Sparse MER

Input: $\mathrm{P}_0(\omega(\boldsymbol{X}_{(1)}))\mathrm{P}_0(b), \mathrm{P}_0(\gamma_i)\mathrm{P}_0(\gamma_i')\mathrm{P}_0(\lambda)$
while $t < \infty$ **do**
  Set $\beta_{(t)} = 0$ and (optional) update previous β's
  Solve $\hat{\boldsymbol{\alpha}}_{(t)} = \hat{\boldsymbol{\alpha}} - \hat{\boldsymbol{\alpha}}'$ for the objective in (3.3)
  $\bar{b}_{(\tau)} = \bar{b}_{(\tau-1)} + (\hat{b} - \bar{b}_{(\tau-1)})/\tau$
  if $\hat{\alpha}_{(t)i} \neq 0$ for any i **then**
    Save new relevant points $\boldsymbol{X}_{\mathcal{S}} = [\boldsymbol{X}_{\mathcal{S}}, \boldsymbol{X}_s]$
    Update $\mathrm{P}(\omega(\boldsymbol{x}))$ through its parameters
  end if
end while
Return: $\mu_{s|\mathcal{S}}(\boldsymbol{x})$ for prediction

While setting $\hat{b}$ to be the median satisfies the constraints in (3.2), it may not be a very stable estimator when each mini-batch of data is small. This is not the only choice of $\hat{b}$ that satisfies the KKT conditions and balancing it with the average of all the previously estimated bias terms can make it more robust to noise.

3.4 Experiments

In this section, we compare the proposed sequential maximum margin classifiers to the batch model, which at every time point, is re-trained on all previous data. This is a lower bound on performance because from Theorem 3.3.2, we known that the MSE of the sparse kernel MER can be lower bounded by the MSE of the batch model.

3.4.1 Simulations

In the following simulations, the models receive roughly 100 samples ($n_{(t)} = [97, 103]$) at every time point and then are tested on an independent data set of 1000 test points. The mean square prediction error is averaged over the 100 trials. For the sparse kernel MER model, we show performance at three different sparsity levels with various sequences of $\epsilon_{(t)}$. In the low sparsity model we set $\epsilon_{(t)} = e^{15(0.001t)}$, in the medium sparsity model $\epsilon_{(t)} = \sqrt{0.001 + 0.2t}$, and in the high sparsity model $\epsilon_{(t)} = 1 + \lfloor t/10.1 \rfloor$. For all models, we set the cost penalty to be $C = 1000$.

In the first simulation, we generate data from a polynomial kernel of degree 2 because it has an explicit feature map. This allows us to compare against a RLS algorithm trained

as linear regression model with the expanded features. The performance of this model will be equivalent to the batch model as the RLS is an optimal linear filter. We set the ridge component or forgetting factor of the RLS to be $\epsilon_0 = 0.001$. Thus the generative model is $y_{(t)i} = f(\tilde{\boldsymbol{X}}_{(t)i})\Theta + e_i$ where $\boldsymbol{X}$ is generated from a zero mean Gaussian distribution with a random 20×20 covariance matrix and e_i are white noise. The bias b is generated uniformly from $[-5, 5]$ and each of the 230 feature weights θ_j are generated uniformly from $[-3, 3]$. Because the samples are identically distributed over time, we set all the previous $\beta_{(t)} = 1/\epsilon_0$, which in the fully dense scenario, will also be an optimal filter.

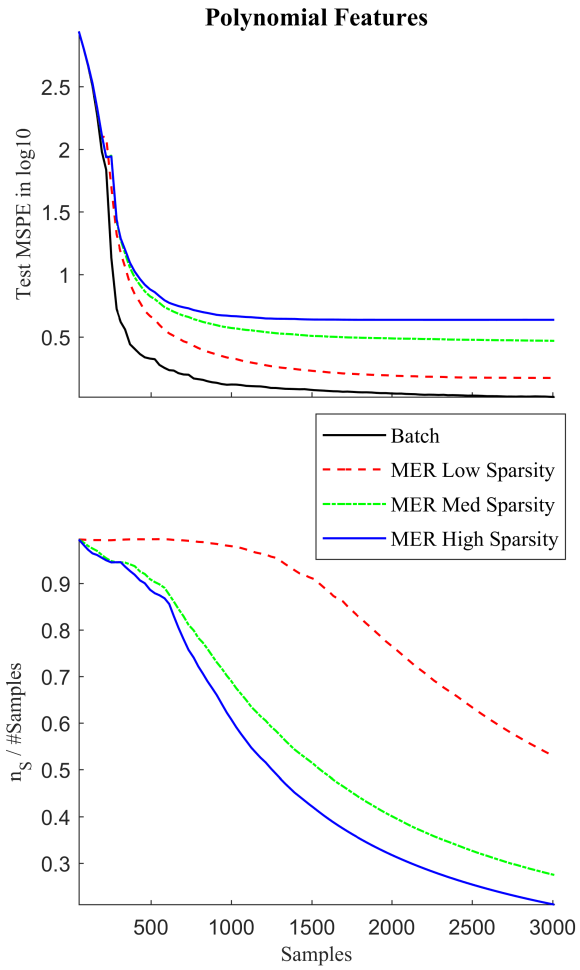


Figure 3.1: Mean squared prediction error of a batch model and the sparse kernel MER model at different sparsity levels.

Figure 3.1 shows the trade-off between lower mean squared prediction error and higher sparsity, which in turn gives a computationally faster model. At the final time point, the high sparsity model is only using 21% of the total samples while the low sparsity model is using 53% of the total samples. An even denser model than the low sparsity MER would

give a mean squared prediction error that is on par with the batch model, however it is not computationally feasible.

3.5 Conclusions

We have proposed a framework for generating regression models that can be sequentially updated with new data. Our framework can be used to generate models with equal performance to all the popular algorithms. It can also be use to generate approximate models that are computationally efficient by reducing the amount of data that must be seen at each iteration. In particular we show for stationary systems, we can bound the performance between our proposed sparse maximum entropy regression model and a model trained on the entire batch of data.

Appendix

Proof of Theorem 3.3.1. Let

$$\boldsymbol{\mu}_{(\tau-1)} = \boldsymbol{\Sigma}_{(\tau-1)}(\boldsymbol{\Sigma}^{-1}_{(\tau-2)}\boldsymbol{\mu}_{(\tau-2)} + \tilde{\boldsymbol{X}}^T_{(\tau-1)}\hat{\boldsymbol{\alpha}}_{(\tau-1)})$$
$$\boldsymbol{\Sigma}_{(\tau-1)} = \tilde{\boldsymbol{H}}^{-1}_{(\tau-1)} = \left(\boldsymbol{\Sigma}_{(0)} + \sum_{t=1}^{\tau-1} 2\beta_{(t)}\tilde{\boldsymbol{X}}^T_{(t)}\tilde{\boldsymbol{X}}_{(t)}\right)^{-1}$$

where $\boldsymbol{\mu}_{(0)} = \mathbf{0}$ and $\boldsymbol{\Sigma}_{(0)} = \begin{bmatrix} 1/\sigma^2 & 0 \\ 0 & \mathrm{I} \end{bmatrix}$. At time τ, let the priors be $\boldsymbol{\Theta} \sim N(\boldsymbol{\mu}_{(\tau-1)}, \boldsymbol{\Sigma}_{(\tau-1)})$, $\gamma_i \sim N(0, \epsilon)$, and $\lambda \sim Exp.(\nu)$ where $\nu \to \infty$. Then the minimizing distribution $\mathrm{P}(\boldsymbol{\Theta}, \boldsymbol{\gamma}, \lambda | \boldsymbol{X}_{(1:\tau)})$ factorizes into

$$= \frac{\mathrm{P}_0(\boldsymbol{\Theta})\prod_{i=1}^{n_{(\tau)}}\mathrm{P}_0(\gamma_i)\mathrm{P}_0(\lambda)}{Z(\boldsymbol{\alpha}_{(\tau)}, \beta_{(\tau)})} \exp\left\{\sum_{i=1}^{n_{(\tau)}} \alpha_{(\tau)i}(-y_{(\tau)i} + \tilde{\boldsymbol{X}}_{(\tau)i}\boldsymbol{\Theta} + \gamma_i) - \beta_{(\tau)}(\boldsymbol{\Theta}^T\tilde{\boldsymbol{X}}^T_{(\tau)}\tilde{\boldsymbol{X}}_{(\tau)}\boldsymbol{\Theta} - \lambda)\right\}$$
$$= \frac{\mathrm{P}_0(\boldsymbol{\Theta})}{Z_\Theta(\boldsymbol{\alpha}_{(\tau)}, \beta_{(\tau)})} \exp\left\{\sum_{i=1}^{n_{(\tau)}} \alpha_{(\tau)i}\tilde{\boldsymbol{X}}_{(\tau)i}\boldsymbol{\Theta} - \beta_{(\tau)}\boldsymbol{\Theta}^T\tilde{\boldsymbol{X}}^T_{(\tau)}\tilde{\boldsymbol{X}}_{(\tau)}\boldsymbol{\Theta}\right\}$$
$$\frac{\prod_{i=1}^{n_{(\tau)}}\mathrm{P}_0(\gamma_i)}{Z_\gamma(\boldsymbol{\alpha}_{(\tau)})} \exp\left\{\sum_{i=1}^{n_{(\tau)}} \alpha_{(\tau)i}\gamma_i\right\} \frac{\mathrm{P}_0(\lambda)e^{\beta_{(\tau)}\lambda}}{Z_\lambda(\beta_{(\tau)})} e^{-\boldsymbol{\alpha}^T_{(\tau)}\boldsymbol{y}_{(\tau)}} = \mathrm{P}(\boldsymbol{\Theta}|\boldsymbol{X}_{(1)}, \ldots, \boldsymbol{X}_{(\tau)})\mathrm{P}(\boldsymbol{\gamma})\mathrm{P}(\lambda)$$

where $\mathrm{P}(\boldsymbol{\Theta}|\boldsymbol{X}_{(1)},\ldots,\boldsymbol{X}_{(\tau)})$

$$
\begin{aligned}
&= \frac{\exp\left\{-0.5(\boldsymbol{\Theta}-\boldsymbol{\mu}_{(\tau-1)})^T\boldsymbol{\Sigma}_{(\tau-1)}^{-1}(\boldsymbol{\Theta}-\boldsymbol{\mu}_{(\tau-1)})\right\}}{\det(2\pi\boldsymbol{\Sigma}_{(\tau-1)})^{1/2}}\frac{\exp\left\{\boldsymbol{\alpha}_{(\tau)}^T\tilde{\boldsymbol{X}}_{(\tau)}\boldsymbol{\Theta}-\beta_{(\tau)}\boldsymbol{\Theta}^T\tilde{\boldsymbol{X}}_{(\tau)}^T\tilde{\boldsymbol{X}}_{(\tau)}\boldsymbol{\Theta}\right\}}{Z_\Theta(\boldsymbol{\alpha}_{(\tau)},\beta_{(\tau)})}\\
&= \exp\left\{-0.5\left(-2\boldsymbol{\alpha}_{(\tau)}^T\tilde{\boldsymbol{X}}_{(\tau)}\boldsymbol{\Theta}+2\beta_{(\tau)}\boldsymbol{\Theta}^T\tilde{\boldsymbol{X}}_{(\tau)}^T\tilde{\boldsymbol{X}}_{(\tau)}\boldsymbol{\Theta}+\boldsymbol{\Theta}^T\boldsymbol{H}_{(\tau-1)}\boldsymbol{\Theta}\right.\right.\\
&\left.\left.-2\boldsymbol{\Theta}^T\tilde{\boldsymbol{H}}_{(\tau-1)}\boldsymbol{\mu}_{(\tau-1)}+\boldsymbol{\mu}_{(\tau-1)}^T\tilde{\boldsymbol{H}}_{(\tau-1)}\boldsymbol{\mu}_{(\tau-1)}\right)\right\}/\left(\det(2\pi\boldsymbol{H}_{(\tau-1)}^{-1})^{1/2}Z_\Theta(\boldsymbol{\alpha}_{(\tau)},\beta_{(\tau)})\right)\\
&= \exp\left\{-0.5\left(\boldsymbol{\Theta}^T\tilde{\boldsymbol{H}}_{(\tau)}\boldsymbol{\Theta}-2\boldsymbol{\Theta}^T(\tilde{\boldsymbol{H}}_{(\tau-1)}\boldsymbol{\mu}_{(\tau-1)}+\tilde{\boldsymbol{X}}_{(\tau)}^T\boldsymbol{\alpha}_{(\tau)})\right.\right.\\
&\qquad\left.\left.+(\tilde{\boldsymbol{H}}_{(\tau-1)}\boldsymbol{\mu}_{(\tau-1)}+\tilde{\boldsymbol{X}}_{(\tau)}^T\boldsymbol{\alpha}_{(\tau)})^T\tilde{\boldsymbol{H}}_{(\tau)}^{-1}(\tilde{\boldsymbol{H}}_{(\tau-1)}\boldsymbol{\mu}_{(\tau-1)}+\tilde{\boldsymbol{X}}_{(\tau)}^T\boldsymbol{\alpha}_{(\tau)})\right)\right\}\\
&\quad\exp\left\{0.5\left(\boldsymbol{\alpha}_{(\tau)}^T\tilde{\boldsymbol{X}}_{(\tau)}\tilde{\boldsymbol{H}}_{(\tau)}^{-1}\tilde{\boldsymbol{X}}_{(\tau)}^T\boldsymbol{\alpha}_{(\tau)}-\boldsymbol{\mu}_{(\tau-1)}^T\tilde{\boldsymbol{H}}_{(\tau-1)}\boldsymbol{\mu}_{(\tau-1)}+\boldsymbol{\mu}_{(\tau-1)}^T\tilde{\boldsymbol{H}}_{(\tau-1)}\tilde{\boldsymbol{H}}_{(\tau)}^{-1}\tilde{\boldsymbol{H}}_{(\tau-1)}\boldsymbol{\mu}_{(\tau-1)}\right.\right.\\
&\qquad\left.\left.+2\boldsymbol{\mu}_{(\tau-1)}^T\tilde{\boldsymbol{H}}_{(\tau-1)}\tilde{\boldsymbol{H}}_{(\tau)}^{-1}\tilde{\boldsymbol{X}}_{(\tau)}^T\boldsymbol{\alpha}_{(\tau)}\right)\right\}/\left(\det(2\pi\boldsymbol{H}_{(\tau-1)}^{-1})^{1/2}Z_\Theta(\boldsymbol{\alpha}_{(\tau)},\beta_{(\tau)})\right)\\
&= \frac{e^{-0.5(\boldsymbol{\Theta}-\tilde{\boldsymbol{H}}_{(\tau)}^{-1}(\tilde{\boldsymbol{H}}_{(\tau-1)}\boldsymbol{\mu}_{(\tau-1)}+\tilde{\boldsymbol{X}}_{(\tau)}^T\boldsymbol{\alpha}_{(\tau)}))^T\tilde{\boldsymbol{H}}_{(\tau)}(\boldsymbol{\Theta}-\tilde{\boldsymbol{H}}_{(\tau)}^{-1}(\tilde{\boldsymbol{H}}_{(\tau-1)}\boldsymbol{\mu}_{(\tau-1)}+\tilde{\boldsymbol{X}}_{(\tau)}^T\boldsymbol{\alpha}_{(\tau)}))}}{\det(2\pi\boldsymbol{H}_{(\tau)}^{-1})^{1/2}}\\
&= N\left(\tilde{\boldsymbol{H}}_{(\tau)}^{-1}(\tilde{\boldsymbol{H}}_{(\tau-1)}\boldsymbol{\mu}_{(\tau-1)}+\tilde{\boldsymbol{X}}_{(\tau)}^T\hat{\boldsymbol{\alpha}}_{(\tau)}),\tilde{\boldsymbol{H}}_{(\tau)}^{-1}\right)
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{P}(\boldsymbol{\gamma}) &= \prod_{i=1}^{n_{(\tau)}}\mathrm{P}(\gamma_i) = \prod_{i=1}^{n_{(\tau)}}\frac{\exp\{-0.5\gamma_i^2/\epsilon\}}{\sqrt{2\pi\epsilon}}\frac{\exp\{\alpha_{(\tau)i}\gamma_i\}}{Z_{\gamma_i}(\alpha_{(\tau)i})}\\
&= \prod_{i=1}^{n_{(\tau)}}\frac{\exp\{-0.5(\gamma_i^2-2\epsilon\alpha_{(\tau)i}\gamma_i+\epsilon^2\alpha_{(\tau)i}^2)/\epsilon\}}{\sqrt{2\pi\epsilon}}\frac{\exp\{0.5\epsilon\alpha_{(\tau)i}^2\}}{Z_{\gamma_i}(\alpha_{(\tau)i})}\\
&= \prod_{i=1}^{n_{(\tau)}}\frac{\exp\{-0.5(\gamma_i-\epsilon\alpha_{(\tau)i})^2/\epsilon\}}{\sqrt{2\pi\epsilon}} = \prod_{i=1}^{n_{(\tau)}}N(\epsilon\alpha_{(\tau)i}\gamma_i,\epsilon)
\end{aligned}
$$

and

$$
\begin{aligned}
\mathrm{P}(\lambda) &= \frac{\nu\exp\{-\nu\lambda\}\exp\{\beta_{(\tau)}\lambda\}}{Z_\lambda(\beta_{(\tau)})} = \frac{\nu\exp\{-(\nu+\beta_{(\tau)})\lambda\}}{Z_\lambda(\beta_{(\tau)})}\\
&= Exp.(\nu+\beta_{(\tau)}) \to Exp.(\infty) \text{ as } \nu\to\infty.
\end{aligned}
$$

□

Proof of Corollary 3.3.1.1. The log partition functions $-\log Z_\Theta(\boldsymbol{\alpha}_{(\tau)},\beta_{(\tau)})$ and

$-\log Z_{\gamma_i}(\alpha_{(\tau)i})$ are defined respectively as

$$
\begin{aligned}
&= -\log \int_{-\infty}^{\infty} \frac{e^{-0.5(\boldsymbol{\Theta}-\boldsymbol{\mu}_{(\tau-1)})^T \boldsymbol{\Sigma}_{(\tau-1)}^{-1}(\boldsymbol{\Theta}-\boldsymbol{\mu}_{(\tau-1)})}}{\det(2\pi\boldsymbol{\Sigma}_{(\tau-1)})^{1/2}} e^{\boldsymbol{\alpha}_{(\tau)}^T \tilde{\boldsymbol{X}}_{(\tau)}\boldsymbol{\Theta} - \beta_{(\tau)}\boldsymbol{\Theta}^T \tilde{\boldsymbol{X}}_{(\tau)}^T \tilde{\boldsymbol{X}}_{(\tau)}\boldsymbol{\Theta}} d\boldsymbol{\Theta} \\
&= 0.5\boldsymbol{\mu}_{(\tau-1)}^T \tilde{\boldsymbol{H}}_{(\tau-1)}\boldsymbol{\mu}_{(\tau-1)} - 0.5\boldsymbol{\mu}_{(\tau-1)}^T \tilde{\boldsymbol{H}}_{(\tau-1)}\tilde{\boldsymbol{H}}_{(\tau)}^{-1}\tilde{\boldsymbol{H}}_{(\tau-1)}\boldsymbol{\mu}_{(\tau-1)} - \boldsymbol{\mu}_{(\tau-1)}^T \tilde{\boldsymbol{H}}_{(\tau-1)}\tilde{\boldsymbol{H}}_{(\tau)}^{-1}\tilde{\boldsymbol{X}}_{(\tau)}^T \boldsymbol{\alpha}_{(\tau)} \\
&- 0.5\boldsymbol{\alpha}_{(\tau)}^T \tilde{\boldsymbol{X}}_{(\tau)}\tilde{\boldsymbol{H}}_{(\tau)}^{-1}\tilde{\boldsymbol{X}}_{(\tau)}^T \boldsymbol{\alpha}_{(\tau)} - \log\det(2\pi\boldsymbol{H}_{(\tau)}^{-1})^{1/2} + \log\det(2\pi\boldsymbol{H}_{(\tau-1)}^{-1})^{1/2}
\end{aligned}
$$

$$
= -\log \int_{-\infty}^{\infty} \frac{e^{-0.5\gamma_i^2/\epsilon}}{\sqrt{2\pi\epsilon}} e^{\alpha_{(\tau)i}\gamma_i}\, d\gamma_i = -\log \int_{-\infty}^{\infty} \frac{e^{-0.5(\gamma_i^2 - 2\epsilon\alpha_{(\tau)i}\gamma_i)/\epsilon}}{\sqrt{2\pi\epsilon}}\, d\gamma_i = -0.5\epsilon\alpha_{(\tau)i}^2
$$

and

$$
\begin{aligned}
&-\log Z_\lambda(\beta_{(\tau)}) = -\log \int_0^\infty \nu e^{-\nu\lambda} e^{\beta_{(\tau)}\lambda}\, d\lambda = -\log\left(\frac{\nu}{\nu - \beta_{(\tau)}}\right) \\
&\Rightarrow \text{ if } \nu \to \infty, \text{ then } \log(1 - \beta_{(\tau)}/\nu) = 0 \text{ for finite } \beta_{(\tau)}.
\end{aligned}
$$

The optimal $\hat{\boldsymbol{\alpha}}_{(\tau)}$ are the solution to

$$
\arg\max_{\boldsymbol{\alpha}_{(\tau)}} -\log Z(\boldsymbol{\alpha}_{(\tau)}, \beta_{(\tau)}) \propto \arg\max_{\boldsymbol{\alpha}_{(\tau)}} -\boldsymbol{\alpha}_{(\tau)}^T \boldsymbol{y}_{(\tau)} - \sum_{i=1}^{n_{(\tau)}} \log Z_{\gamma_i}(\boldsymbol{\alpha}_{(\tau)}) - \log Z_\Theta(\boldsymbol{\alpha}_{(\tau)}, \beta_{(\tau)})
$$

so $\frac{\partial}{\partial \boldsymbol{\alpha}_{(\tau)}} -\log Z(\boldsymbol{\alpha}_{(\tau)}, \beta_{(\tau)})$

$$
\begin{aligned}
&= \frac{\partial}{\partial \boldsymbol{\alpha}_{(\tau)}} \boldsymbol{\alpha}_{(\tau)}^T \boldsymbol{y}_{(\tau)} - 0.5\epsilon\boldsymbol{\alpha}_{(\tau)}^T\boldsymbol{\alpha}_{(\tau)} - \boldsymbol{\alpha}_{(\tau)}^T \tilde{\boldsymbol{X}}_{(\tau)}\tilde{\boldsymbol{H}}_{(\tau)}^{-1}\tilde{\boldsymbol{H}}_{(\tau-1)}\boldsymbol{\mu}_{(\tau-1)} - 0.5\boldsymbol{\alpha}_{(\tau)}^T \tilde{\boldsymbol{X}}_{(\tau)}\tilde{\boldsymbol{H}}_{(\tau)}^{-1}\tilde{\boldsymbol{X}}_{(\tau)}^T\boldsymbol{\alpha}_{(\tau)} \\
&= \boldsymbol{y}_{(\tau)} - \tilde{\boldsymbol{X}}_{(\tau)}\tilde{\boldsymbol{H}}_{(\tau)}^{-1}\tilde{\boldsymbol{H}}_{(\tau-1)}\boldsymbol{\mu}_{(\tau-1)} + (\epsilon\mathrm{I} - \tilde{\boldsymbol{X}}_{(\tau)}\tilde{\boldsymbol{H}}_{(\tau)}^{-1}\tilde{\boldsymbol{X}}_{(\tau)}^T)\boldsymbol{\alpha}_{(\tau)} = 0 \\
&\Rightarrow \hat{\boldsymbol{\alpha}}_{(\tau)} = (\epsilon\mathrm{I} + \tilde{\boldsymbol{X}}_{(\tau)}\tilde{\boldsymbol{H}}_{(\tau)}^{-1}\tilde{\boldsymbol{X}}_{(\tau)}^T)^{-1}(\boldsymbol{y}_{(\tau)} - \tilde{\boldsymbol{X}}_{(\tau)}\tilde{\boldsymbol{H}}_{(\tau)}^{-1}\tilde{\boldsymbol{H}}_{(\tau-1)}\boldsymbol{\mu}_{(\tau-1)})
\end{aligned}
$$

At every time point $t = \tau$, set the previous $\beta_{(1)}, \ldots, \beta_{(\tau-1)} = \frac{1}{2\epsilon}$ and the current $\beta_{(\tau)} = 0$. Then $\tilde{\boldsymbol{H}}_{(\tau)} = \tilde{\boldsymbol{H}}_{(\tau-1)}$ and the mean of the updated model reduces to $\boldsymbol{\mu}_{(\tau)} = \boldsymbol{\mu}_{(\tau-1)} + \tilde{\boldsymbol{H}}_{(\tau-1)}^{-1}\tilde{\boldsymbol{X}}_{(\tau)}^T\hat{\boldsymbol{\alpha}}_{(\tau)}$ where the optimal $\hat{\boldsymbol{\alpha}}_{(\tau)} = \left(\epsilon\mathrm{I} + \tilde{\boldsymbol{X}}_{(\tau)}\tilde{\boldsymbol{H}}_{(\tau)}^{-1}\tilde{\boldsymbol{X}}_{(\tau)}^T\right)^{-1}\left(\boldsymbol{y}_{(\tau)} - \tilde{\boldsymbol{X}}_{(\tau)}\boldsymbol{\mu}_{(\tau-1)}\right)$. Plugging $\hat{\boldsymbol{\alpha}}_{(\tau)}$ into the mean update recovers the same update as the RLS algorithm.

□

Proof of Proposition 1. Define

$$\begin{aligned}
&\boldsymbol{\omega} = \omega(\boldsymbol{X}_{(\tau)}), \boldsymbol{\mu}_{(0)} = \mathbf{0}, \boldsymbol{\Sigma}_{(0)} = 1 + k(\boldsymbol{X}_{(1)}, \boldsymbol{X}_{(1)}) \\
&\boldsymbol{\mu}_{(\tau-1)} = \mu_{(\tau-1)}(\tilde{\boldsymbol{X}}_{(\tau)}) = \mu_{(\tau-2)}(\tilde{\boldsymbol{X}}_{(\tau)}) + k_{(\tau-1)}(\tilde{\boldsymbol{X}}_{(\tau)}, \tilde{\boldsymbol{X}}_{(\tau-1)})\hat{\boldsymbol{\alpha}}_{(\tau-1)} \\
&\boldsymbol{\Sigma}_{(\tau-1)} = k_{(\tau-1)}(\tilde{\boldsymbol{X}}_{(\tau)}, \tilde{\boldsymbol{X}}_{(\tau)})
\end{aligned}$$

At time τ, let the priors be $\boldsymbol{\omega} \sim N(\boldsymbol{\mu}_{(\tau-1)}, \boldsymbol{\Sigma}_{(\tau-1)})$, $\gamma_i \sim N(0, \epsilon)$, and $\lambda \sim Exp.(\nu)$ where $\nu \to \infty$. Then the minimizing distribution $\mathrm{P}(\omega(\boldsymbol{X}_{(\tau)}), \boldsymbol{\gamma}, \lambda | \boldsymbol{X}_{(1:\tau)})$ factorizes similarly as in Theorem 3.3.1 into

$$\begin{aligned}
&= \frac{\mathrm{P}_0(\omega(\tilde{\boldsymbol{X}}_{(\tau)}))}{Z_\omega(\boldsymbol{\alpha}_{(\tau)}), \beta_{(\tau)})} \exp\left\{\boldsymbol{\alpha}_{(\tau)}^T \omega(\boldsymbol{X}_{(\tau)}) - \beta_{(\tau)} \omega(\boldsymbol{X}_{(\tau)})^T \omega(\boldsymbol{X}_{(\tau)})\right\} \\
&\quad \frac{\prod_{i=1}^{n_{(\tau)}} \mathrm{P}_0(\gamma_i)}{Z_\gamma(\boldsymbol{\alpha}_{(\tau)})} \exp\left\{\sum_{i=1}^{n_{(\tau)}} \alpha_{(\tau)i}\gamma_i\right\} \frac{\mathrm{P}_0(\lambda) e^{\beta_{(\tau)}\lambda}}{Z_\lambda(\beta_{(\tau)})} e^{-\boldsymbol{\alpha}_{(\tau)}^T \boldsymbol{y}_{(\tau)}} \\
&= \mathrm{P}(\omega(\boldsymbol{X}_{(\tau)}) | \boldsymbol{X}_{(1)}, \ldots, \boldsymbol{X}_{(\tau)}) \mathrm{P}(\boldsymbol{\gamma}) \mathrm{P}(\lambda)
\end{aligned}$$

where $\mathrm{P}(\omega(\tilde{\boldsymbol{X}}_{(\tau)}) | \boldsymbol{X}_{(1)}, \ldots, \boldsymbol{X}_{(\tau)})$

$$\begin{aligned}
&= \frac{e^{-0.5(\boldsymbol{\omega}-\boldsymbol{\mu}_{(\tau-1)})^T \boldsymbol{\Sigma}_{(\tau-1)}^{-1} (\boldsymbol{\omega}-\boldsymbol{\mu}_{(\tau-1)})}}{\det(2\pi\boldsymbol{\Sigma}_{(\tau-1)})^{1/2}} \frac{e^{\boldsymbol{\alpha}_{(\tau)}^T \boldsymbol{\omega} - \beta_{(\tau)} \boldsymbol{\omega}^T \boldsymbol{\omega}}}{Z_\omega(\boldsymbol{\alpha}_{(\tau)}, \beta_{(\tau)})} \\
&= \frac{\exp\left\{-0.5\, \boldsymbol{\omega}^T (\boldsymbol{\Sigma}_{(\tau-1)}^{-1} + 2\beta_{(\tau)}\mathrm{I})\boldsymbol{\omega} + \boldsymbol{\omega}^T (\boldsymbol{\Sigma}_{(\tau-1)}^{-1} \boldsymbol{\mu}_{(\tau-1)} + \boldsymbol{\alpha}_{(\tau)}) - 0.5 \boldsymbol{\mu}_{(\tau-1)}^T \boldsymbol{\Sigma}_{(\tau-1)}^{-1} \boldsymbol{\mu}_{(\tau-1)}\right\}}{\left(\det\left(2\pi\boldsymbol{\Sigma}_{(\tau-1)}\right)^{-1/2} Z_\omega(\boldsymbol{\alpha}_{(\tau)}, \beta_{(\tau)})\right)} \\
&= \exp\Big\{ -0.5 \left(\boldsymbol{\omega} - k_{(\tau)}(\tilde{\boldsymbol{X}}_{(\tau)}, \tilde{\boldsymbol{X}}_{(\tau)})(\boldsymbol{\Sigma}_{(\tau-1)}^{-1} \boldsymbol{\mu}_{(\tau-1)} + \boldsymbol{\alpha}_{(\tau)})\right)^T k_{(\tau)}(\boldsymbol{X}_{(\tau)}, \boldsymbol{X}_{(\tau)})^{-1} \\
&\qquad \left(\boldsymbol{\omega} - k_{(\tau)}(\tilde{\boldsymbol{X}}_{(\tau)}, \tilde{\boldsymbol{X}}_{(\tau)})(\boldsymbol{\Sigma}_{(\tau-1)}^{-1} \boldsymbol{\mu}_{(\tau-1)} + \boldsymbol{\alpha}_{(\tau)})\right) \Big\} \Big/ \det\left(2\pi k_{(\tau)}(\tilde{\boldsymbol{X}}_{(\tau)}, \tilde{\boldsymbol{X}}_{(\tau)})\right)^{1/2} \\
&= N\left(k_{(\tau)}(\tilde{\boldsymbol{X}}_{(\tau)}, \tilde{\boldsymbol{X}}_{(\tau)})(\boldsymbol{\Sigma}_{(\tau-1)}^{-1} \boldsymbol{\mu}_{(\tau-1)} + \hat{\boldsymbol{\alpha}}_{(\tau)}), k_{(\tau)}(\tilde{\boldsymbol{X}}_{(\tau)}, \tilde{\boldsymbol{X}}_{(\tau)})\right)
\end{aligned}$$

for $k_{(\tau)}(\tilde{\boldsymbol{X}}_{(\tau)}, \tilde{\boldsymbol{X}}_{(\tau)}) = \boldsymbol{\Sigma}_{(\tau-1)} - \boldsymbol{\Sigma}_{(\tau-1)}(\frac{1}{2\beta_{(\tau)}}\mathrm{I} + \boldsymbol{\Sigma}_{(\tau-1)})^{-1} \boldsymbol{\Sigma}_{(\tau-1)} = \left(2\beta_{(\tau)}\mathrm{I} + \boldsymbol{\Sigma}_{(\tau-1)}^{-1}\right)^{-1}$ and the rest of distributions are the same as in Theorem 3.3.1.

The log partition $-\log\left(Z_\omega(\boldsymbol{\alpha}_{(\tau)}), \beta_{(\tau)}\right)$

$$
\begin{aligned}
&= -\log \int_{-\infty}^{\infty} \frac{\mathrm{P}_0(\omega(\tilde{\boldsymbol{X}}_{(\tau)}))}{\det(2\pi\boldsymbol{\Sigma}_{(\tau-1)})^{1/2}} e^{\boldsymbol{\alpha}_{(\tau)}^T\boldsymbol{\omega} - \beta_{(\tau)}\boldsymbol{\omega}^T\boldsymbol{\omega}} d\boldsymbol{\omega} \\
&= -0.5\Big(\log\det\left(\boldsymbol{\Sigma}_{(\tau)}\boldsymbol{\Sigma}_{(\tau-1)}^{-1}\right) - \boldsymbol{\mu}_{(\tau-1)}^T\boldsymbol{\Sigma}_{(\tau)}^{-1}\boldsymbol{\mu}_{(\tau-1)} + \boldsymbol{\mu}_{(\tau-1)}^T\boldsymbol{\Sigma}_{(\tau-1)}^{-1}\boldsymbol{\mu}_{(\tau-1)} \\
&\quad - \boldsymbol{\alpha}_{(\tau)}^T k_{(\tau)}(\tilde{\boldsymbol{X}}_{(\tau)}, \tilde{\boldsymbol{X}}_{(\tau)})\boldsymbol{\alpha}_{(\tau)} - 2\boldsymbol{\alpha}_{(\tau)}^T k_{(\tau)}(\tilde{\boldsymbol{X}}_{(\tau)}, \tilde{\boldsymbol{X}}_{(\tau)})\boldsymbol{\Sigma}_{(\tau-1)}^{-1}\boldsymbol{\mu}_{(\tau-1)}\Big)
\end{aligned}
$$

so when $\beta_{(\tau)} = 0$,

$$
\begin{aligned}
\hat{\boldsymbol{\alpha}}_{(\tau)} &= \arg\max_{\boldsymbol{\alpha}_{(\tau)}} -\boldsymbol{\alpha}_{(\tau)}^T\boldsymbol{y}_{(\tau)} - \sum_{i=1}^{n_{(\tau)}} \log Z_{\gamma_i}(\boldsymbol{\alpha}_{(\tau)}) - \log Z_\omega(\boldsymbol{\alpha}_{(\tau)}) \\
&= \left(\epsilon\mathrm{I} + k_{(\tau-1)}(\tilde{\boldsymbol{X}}_{(\tau)}, \tilde{\boldsymbol{X}}_{(\tau)})\right)^{-1}\left(\boldsymbol{y}_{(\tau)} - \mu_{(\tau-1)}(\tilde{\boldsymbol{X}}_{(\tau)})\right).
\end{aligned}
$$

□

Proof of Proposition 2. At time τ, let the priors be $\omega(\boldsymbol{X}_{(\tau)}) \sim N(\boldsymbol{\mu}_{(\tau-1)}, \boldsymbol{\Sigma}_{(\tau-1)})$ whose parameters are defined in Proposition 1, $b \sim N(0, \infty)$,
$\gamma_i, \gamma_i' \sim C_{(\tau)} e^{C_{(\tau)}(\epsilon_{(\tau)} - \gamma_i)}\mathcal{I}(\gamma_i \in [\epsilon_{(\tau)}, \infty))$ and, $\lambda \sim Exp.(\nu)$ where $\nu \to \infty$. Then the optimal distribution $\mathrm{P}(\omega(\boldsymbol{X}_{(\tau)}), b, \boldsymbol{\gamma}, \boldsymbol{\gamma}', \lambda | \{\mathcal{D}\}_{t=1}^{\tau})$ factorizes into

$$
\begin{aligned}
&= \frac{\mathrm{P}_0(\omega(\boldsymbol{X}_{(\tau)}))\mathrm{P}_0(b)\prod_{i=1}^{n_{(\tau)}}\mathrm{P}_0(\gamma_i)\mathrm{P}_0(\gamma_i')\mathrm{P}_0(\lambda)}{Z(\boldsymbol{\alpha}, \boldsymbol{\alpha}', \beta_{(\tau)})} \exp\Big\{ -\beta_{(\tau)}\omega(\boldsymbol{X}_{(\tau)})^T\omega(\boldsymbol{X}_{(\tau)}) - \lambda \\
&\quad \sum_{i=1}^{n_{(\tau)}} \alpha_i(-y_{(\tau)i} + (\omega(\boldsymbol{X}_{(\tau)i}) + b) + \gamma_i) - \alpha_i'(-y_{(\tau)i} + (\omega(\boldsymbol{X}_{(\tau)i}) + b) - \gamma_i')\Big\} \\
&= \exp\{-(\boldsymbol{\alpha} - \boldsymbol{\alpha}')^T\boldsymbol{y}_{(\tau)}\}\frac{\mathrm{P}_0(\omega(\boldsymbol{X}_{(\tau)}))}{Z_\omega(\boldsymbol{\alpha}, \boldsymbol{\alpha}', \beta_{(\tau)})} \exp\left\{(\boldsymbol{\alpha} - \boldsymbol{\alpha}')^T\omega(\boldsymbol{X}_{(\tau)}) - \beta_{(\tau)}\omega(\boldsymbol{X}_{(\tau)})^T\omega(\boldsymbol{X}_{(\tau)})\right\} \\
&\quad \frac{\mathrm{P}_0(b)}{Z_b(\boldsymbol{\alpha}, \boldsymbol{\alpha}')} \exp\left\{\mathbf{1}^T(\boldsymbol{\alpha} - \boldsymbol{\alpha}')\right\} \frac{\prod_{i=1}^{n_{(\tau)}}\mathrm{P}_0(\gamma_i)\mathrm{P}_0(\gamma_i')}{Z_\gamma(\boldsymbol{\alpha}, \boldsymbol{\alpha}')} \exp\left\{\boldsymbol{\alpha}^T\boldsymbol{\gamma} + \boldsymbol{\alpha}'^T\boldsymbol{\gamma}'\right\} \frac{\mathrm{P}_0(\lambda)\exp\{\beta_{(\tau)}\lambda\}}{Z_\lambda(\beta_{(\tau)})} \\
&= \mathrm{P}(\omega(\boldsymbol{X}_{(\tau)})|\boldsymbol{X}_{(1)}, \ldots, \boldsymbol{X}_{(\tau)})\mathrm{P}(b), \mathrm{P}(\boldsymbol{\gamma})\mathrm{P}(\boldsymbol{\gamma}')\mathrm{P}(\lambda)
\end{aligned}
$$

where $\mathrm{P}(\omega(\boldsymbol{X}_{(\tau)})|\boldsymbol{X}_{(1)}, \ldots, \boldsymbol{X}_{(\tau)})$ and $\mathrm{P}(\lambda)$ are the same as in Proposition 1 for

$\hat{\boldsymbol{\alpha}}_{(\tau)} = \hat{\boldsymbol{\alpha}} - \hat{\boldsymbol{\alpha}}'$. P($b$) and P($\boldsymbol{\gamma}$) respectively are

$$= \frac{\exp\{-0.5b^2/\sigma^2\}}{\sqrt{2\pi\sigma^2}} \frac{\exp\{b\mathbf{1}^T(\boldsymbol{\alpha}-\boldsymbol{\alpha}')\}}{Z_b(\boldsymbol{\alpha},\boldsymbol{\alpha}')} = \frac{\exp\{-0.5(b^2 - 2\sigma^2 b\mathbf{1}^T(\boldsymbol{\alpha}-\boldsymbol{\alpha}'))/\sigma^2\}}{\sqrt{2\pi\sigma^2}\, Z_b(\boldsymbol{\alpha},\boldsymbol{\alpha}')}$$
$$= \frac{\exp\{-0.5(b-\sigma^2\mathbf{1}^T(\boldsymbol{\alpha}-\boldsymbol{\alpha}'))^2/\sigma^2\}}{\sqrt{2\pi\sigma^2}} \frac{\exp\{0.5\sigma^2(\mathbf{1}^T(\boldsymbol{\alpha}-\boldsymbol{\alpha}'))^2\}}{Z_b(\boldsymbol{\alpha},\boldsymbol{\alpha}')}$$
$$= N(\sigma^2\mathbf{1}^T(\boldsymbol{\alpha}-\boldsymbol{\alpha}'),\sigma^2) \to N(0,\infty) \text{ as } \sigma\to\infty \text{ for } \mathbf{1}^T(\boldsymbol{\alpha}-\boldsymbol{\alpha}')=0$$

$$= \frac{\prod_{i=1}^{n_{(\tau)}} C_{(\tau)} \exp\{C_{(\tau)}(\epsilon_{(\tau)}-\gamma_i)\} e^{\alpha_i\gamma_i}}{Z_{\gamma_i}(\alpha_i)} = \prod_{i=1}^{n_{(\tau)}} \frac{C_{(\tau)}\exp\{\alpha_i\epsilon_{(\tau)}\}}{Z_{\gamma_i}(\alpha_i)} \exp\{(C_{(\tau)}-\alpha_i)(\epsilon_{(\tau)}-\gamma_i)\}$$
$$= \prod_{i=1}^{n_{(\tau)}} (C_{(\tau)}-\alpha_i)\exp\{(C_{(\tau)}-\alpha_i)(\epsilon_{(\tau)}-\gamma_i)\}$$

with support $\gamma_i \in [\epsilon_{(1)},\infty)$ and same distribution for P($\boldsymbol{\gamma}'$)

Let $-\log Z_\lambda(\beta_{(\tau)})$ be the same as in Theorem 3.3.1, use $\hat{\boldsymbol{\alpha}}-\hat{\boldsymbol{\alpha}}'$ in $-\log Z_\omega(\boldsymbol{\alpha},\boldsymbol{\alpha}')$. Define

$$\begin{aligned} -\log Z_b(\boldsymbol{\alpha},\boldsymbol{\alpha}') &= -\log \int_{-\infty}^{\infty} \frac{e^{-0.5(b^2-2\sigma^2 b\mathbf{1}^T(\boldsymbol{\alpha}-\boldsymbol{\alpha}'))/\sigma^2}}{\sqrt{2\pi\sigma^2}} db \\ &= -\log \int_{-\infty}^{\infty} \frac{e^{-0.5(b-\sigma^2\mathbf{1}^T(\boldsymbol{\alpha}-\boldsymbol{\alpha}'))^2/\sigma^2} e^{0.5\sigma^2(\mathbf{1}^T(\boldsymbol{\alpha}-\boldsymbol{\alpha}'))^2}}{\sqrt{2\pi\sigma^2}} db \\ &= -0.5\sigma^2(\mathbf{1}^T(\boldsymbol{\alpha}-\boldsymbol{\alpha}'))^2 = -0.5\sigma^2(\mathbf{1}^T(\boldsymbol{\alpha}-\boldsymbol{\alpha}'))^2 \end{aligned}$$

$\Rightarrow$ if $\sigma\to\infty$, then $\mathbf{1}^T(\boldsymbol{\alpha}-\boldsymbol{\alpha}')=0$ where $\boldsymbol{\alpha}\geq 0, \boldsymbol{\alpha}'\geq 0$ (inequality Lagrange multipliers), and

$$\begin{aligned} -\log Z_{\gamma_i}(\alpha_i) &= -\log \int_{\epsilon_{(\tau)}}^{\infty} C_{(\tau)} e^{\alpha_i\gamma_i} e^{C_{(\tau)}(\epsilon_{(\tau)}-\gamma_i)}\, d\gamma_i = -\log \frac{C_{(\tau)} e^{C_{(\tau)}\epsilon_{(\tau)}}}{\alpha_i - C_{(\tau)}} e^{\gamma_i(\alpha_i - C_{(\tau)})}\Big|_{\epsilon_{(\tau)}}^{\infty} \\ &= -\log \frac{-C_{(\tau)} e^{\epsilon_{(\tau)}\alpha_i}}{\alpha_i - C_{(\tau)}} = -\epsilon_{(\tau)}\alpha_i - \log(\frac{C_{(\tau)}}{C_{(\tau)}-\alpha_i}) = -\epsilon_{(\tau)}\alpha_i + \log(1-\frac{\alpha_i}{C_{(\tau)}}) \end{aligned}$$

where $-\log Z_{\gamma_i'}(\alpha_i')$ is defined similarly.

Then given $\beta_{(\tau)} = 0$, the optimal $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}'$ are the solution to

$$
\begin{aligned}
&= \arg\max_{\boldsymbol{\alpha},\boldsymbol{\alpha}'} - \log Z_\omega(\boldsymbol{\alpha},\boldsymbol{\alpha}') + (\boldsymbol{\alpha}-\boldsymbol{\alpha}')^T \boldsymbol{y}_{(\tau)} - \log Z_b(\boldsymbol{\alpha},\boldsymbol{\alpha}') - \sum_{i=1}^{n_{(\tau)}} \log Z_{\gamma_i,\gamma_i'}(\alpha_i,\alpha_i') \\
&= -0.5(\boldsymbol{\alpha}-\boldsymbol{\alpha}')^T k_{(\tau-1)}(\tilde{\boldsymbol{X}}_{(\tau)},\tilde{\boldsymbol{X}}_{(\tau)})(\boldsymbol{\alpha}-\boldsymbol{\alpha}') - (\boldsymbol{\alpha}-\boldsymbol{\alpha}')^T \mu_{(\tau-1)}(\boldsymbol{X}_{(\tau)}) + (\boldsymbol{\alpha}-\boldsymbol{\alpha}')^T \boldsymbol{y}_{(\tau)} \\
&+ \sum_{i=1}^{n_{(\tau)}} \log(1-\alpha_i/C_{(\tau)}) + \log(1-\alpha_i/C_{(\tau)}) - \epsilon_{(\tau)}(\alpha_i+\alpha_i') \\
&\text{subject to } \mathbf{1}^T(\boldsymbol{\alpha}-\boldsymbol{\alpha}') = 0, \boldsymbol{\alpha} \geq 0, \boldsymbol{\alpha}' \geq 0
\end{aligned}
$$

which are then used to solve for an optimal bias term $\hat{b}$. Since the distribution $\mathrm{P}(b) = N(0,\infty)$ is undefined, the bias term just needs to ensure that the expectation constraints in the original objective hold. Thus at every time point τ, $\hat{b}$ minimize

$$\sum_{i=1}^{n_{(\tau)}} \left| \left(y_{(\tau)i} - (\mu_{(\tau-1)}(\boldsymbol{X}_{(\tau)i}) + k_{(\tau)}(\boldsymbol{X}_{(\tau)i}, \boldsymbol{X}_{(\tau)})\hat{\boldsymbol{\alpha}}_{(\tau)}) \right) - b \right|$$

$\Rightarrow \hat{b}$ is the median of the residuals of a prediction without an intercept term.

□

Proof of Theorem 3.3.2. If there is no previously trained model, given all data $\{\boldsymbol{y}_{(1:\tau-1)}, \boldsymbol{X}_{(1:\tau-1)}\}$ up to time $\tau-1$ and $\beta_{(\tau-1)} = 0$, let $\hat{\boldsymbol{\alpha}}_{(1:\tau-1)}$ maximize the negative log partition function for the distribution that minimizes the constrained relative entropy problem using priors $\mathrm{P}_0(\omega(\tilde{\boldsymbol{X}}_{(1:\tau-1)})) = N(\mathbf{0}, k(\tilde{\boldsymbol{X}}_{(1:\tau-1)}, \tilde{\boldsymbol{X}}_{(1:\tau-1)}))$, $\mathrm{P}_0(\gamma_i) = N(0,\epsilon)$ and, $\mathrm{P}_0(\lambda) = Exp.(\infty)$ and the constraints in (3.1). Then $\hat{\boldsymbol{\alpha}}_{\mathcal{S}} = \boldsymbol{\alpha} - \boldsymbol{\alpha}'$ are the natural parameters for a sparse approximation to the optimal posterior distribution where $\boldsymbol{\alpha}, \boldsymbol{\alpha}'$ maximize the negative log partition function of the distribution that minimizes the constrained relative entropy problem using priors $\mathrm{P}_0(\omega(\boldsymbol{X}_{(1:\tau-1)})) = N(\mathbf{0}, k(\boldsymbol{X}_{(1:\tau-1)}, \boldsymbol{X}_{(1:\tau-1)}))$, $\mathrm{P}_0(b) = N(0,\sigma^2)$, $\mathrm{P}_0(\gamma_i) = \mathrm{P}_0(\gamma_i') = C_{\mathcal{S}} e^{C_{\mathcal{S}}(\epsilon_{\mathcal{S}}-\gamma_i)} \mathcal{I}(\gamma_i \in [\epsilon_{\mathcal{S}}, \infty))$ and, $\mathrm{P}_0(\lambda) = Exp.(\infty)$ and the constraints in (3.2), and $\hat{b}$ is the optimal bias term that centers the predictions.

At any point $\boldsymbol{x}$, for $\sigma^2 \to \infty$ such that $(\mathbf{1}^T\hat{\boldsymbol{\alpha}}_{(1:\tau-1)} - \hat{b}) \to 0$, there exists some δ_1 such that the difference in ℓ_2 norm between a prediction using $\hat{\boldsymbol{\alpha}}_{(1:\tau-1)}$ and one $\hat{\boldsymbol{\alpha}}_{\mathcal{S}}$ is small;

$$
\begin{aligned}
&||(\mathbf{1}^T + k(\boldsymbol{x}, \boldsymbol{X}_{(1:\tau-1)}))\hat{\boldsymbol{\alpha}}_{(1:\tau-1)} - (k(\boldsymbol{x}, \boldsymbol{X}_{\mathcal{S}})\hat{\boldsymbol{\alpha}}_{\mathcal{S}} + \hat{b})||_2^2 \\
&= ||k(\boldsymbol{x}, \boldsymbol{X}_{(1:\tau-1)})(\hat{\boldsymbol{\alpha}}_{(1:\tau-1)} - \boldsymbol{\alpha}_{\mathcal{S}}) + (\mathbf{1}^T\hat{\boldsymbol{\alpha}}_{(1:\tau-1)} - \hat{b})||_2^2 \\
&\leq ||k(\boldsymbol{x}, \boldsymbol{X}_{(1:\tau-1)})||_2^2 ||\hat{\boldsymbol{\alpha}}_{(1:\tau-1)} - \hat{\boldsymbol{\alpha}}_{\mathcal{S}}||_2^2 \leq \delta_1
\end{aligned}
$$

where δ_1 depends on the similarity between the new point $\boldsymbol{x}$ and the previous data, and the

hyper-parameters $C_{\mathcal{S}}, \epsilon_{\mathcal{S}}$ that control the sparsity level of $\hat{\boldsymbol{\alpha}}_{\mathcal{S}}$.

Now using the optimal posterior distribution $\mathrm{P}_0(\omega(\boldsymbol{X}_{(\tau)})) = N(k(\boldsymbol{X}_{(\tau)}, \boldsymbol{X}_{(1:\tau-1)}))\hat{\boldsymbol{\alpha}}_{(1:\tau-1)}, k_{(1:\tau-1)}(\boldsymbol{X}_{(\tau)}, \boldsymbol{X}_{(\tau)}))$ as the prior, given data $\boldsymbol{y}_{(\tau)}, \boldsymbol{X}_{(\tau)}$, the Lagrange multipliers $\hat{\boldsymbol{\alpha}}_{(\tau)}$ and $\hat{\boldsymbol{\alpha}}_{s|(1:\tau-1)}$ maximize the negative log partition function defined using constraints (3.1) and (3.2) with corresponding priors for $\beta_{(\tau)} = 0$. Again there exists some δ_2 such that the difference in ℓ_2 norm is small;

$$
\begin{aligned}
&||(\mathbf{1}^T\hat{\boldsymbol{\alpha}}_{(1:\tau-1)} + \mu_{(\tau-1)}(\boldsymbol{x}) + k_{(1:\tau-1)}(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{X}}_{(\tau)})\hat{\boldsymbol{\alpha}}_{(\tau)}) \\
&\quad - (\mu_{(\tau-1)}(\boldsymbol{x}) + k_{(1:\tau-1)}(\boldsymbol{x}, \boldsymbol{X}_{(\tau)})\hat{\boldsymbol{\alpha}}_{s|(1:\tau-1)} + \hat{b})||_2^2 \\
&\leq ||k_{(1:\tau-1)}(\boldsymbol{x}, \boldsymbol{X}_{(\tau)})||_2^2||\hat{\boldsymbol{\alpha}}_{(\tau)} - \hat{\boldsymbol{\alpha}}_{s|(1:\tau-1)}||_2^2 \leq \delta_2
\end{aligned}
$$

where δ_2 is a function of the prior's cost C_s and margin ϵ_s parameters and the similarity defined by the kernel function.

Finally let $\hat{\boldsymbol{\alpha}}_{s|(1:\tau-1)}$ and $\hat{\boldsymbol{\alpha}}_{s|\mathcal{S}}$ be the solutions to (3.3) when the kernel functions $k_{(1:\tau-1)}(\boldsymbol{x}, \boldsymbol{x}')$ and $k_{\mathcal{S}}(\boldsymbol{x}, \boldsymbol{x}')$ are used respectively. The kernel functions only differ in their estimators of the negative Hessian where $\boldsymbol{H}_{\mathcal{S}}$ is composed of a subset of the $\boldsymbol{X}_{(1:\tau-1)}$ data points in $\boldsymbol{H}_{(\tau-1)} = (\mathrm{I} + \sum_{t=1}^{\tau-1} 2\beta_{(t)} f(\boldsymbol{X}_{(t)})^T f(\boldsymbol{X}_{(t)}))$.

Assume the data points $f(\boldsymbol{X}_{(t)i}$ are independent and identically distributed from a "nice" distribution such that $f(\boldsymbol{X}_{(t)})$ converges to $f(\boldsymbol{x})$ in mean square. Then because $f(\boldsymbol{X}_{\mathcal{S}})$ is a subsequence of $\boldsymbol{X}_{(1:\tau-1)}$ and $f(\boldsymbol{x})$ lie in a Hilbert space, $\boldsymbol{H}_{(\tau-1)}$ and $\boldsymbol{H}_{\mathcal{S}}$ converge to the same thing as asymptotically. Thus as $\tau \to \infty$, $\hat{\boldsymbol{\alpha}}_{s|(1:\tau-1)}$ and $\hat{\boldsymbol{\alpha}}_{s|\mathcal{S}}$ solve the same objective function.

Using the above, the difference in ℓ_2 norm between the optimal prediction $\hat{y}$ and the sparse MER prediction $\hat{y}_s$ can be bounded as $||\hat{y} - \hat{y}_s||_2^2$

$$
\begin{aligned}
&= ||(\mathbf{1}^T + k(\boldsymbol{x}, \boldsymbol{X}_{(1:\tau-1)}))\hat{\boldsymbol{\alpha}}_{(1:\tau-1)} + k_{(1:\tau-1)}(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{X}}_{(\tau)})\hat{\boldsymbol{\alpha}}_{(\tau)}) \\
&\quad - (k(\boldsymbol{x}, \boldsymbol{X}_{\mathcal{S}})\hat{\boldsymbol{\alpha}}_{\mathcal{S}} + k_{\mathcal{S}}(\boldsymbol{x}, \boldsymbol{X}_{(\tau)})\hat{\boldsymbol{\alpha}}_{s|\mathcal{S}} + \hat{b})||_2^2 \\
&\leq ||k(\boldsymbol{x}, \boldsymbol{X}_{(1:\tau-1)})\hat{\boldsymbol{\alpha}}_{(1:\tau-1)} - k(\boldsymbol{x}, \boldsymbol{X}_{(1:\tau-1)})\hat{\boldsymbol{\alpha}}_{\mathcal{S}}||_2^2 \\
&\quad + ||k_{(1:\tau-1)}(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{X}}_{(\tau)})\hat{\boldsymbol{\alpha}}_{(\tau)} - k_{\mathcal{S}}(\boldsymbol{x}, \boldsymbol{X}_{(\tau)})\hat{\boldsymbol{\alpha}}_{s|\mathcal{S}}||_2^2 \\
&\leq \delta_1 + ||(k_{(1:\tau-1)}(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{X}}_{(\tau)})\hat{\boldsymbol{\alpha}}_{(\tau)} - k_{(1:\tau-1)}(\boldsymbol{x}, \boldsymbol{X}_{(\tau)})\hat{\boldsymbol{\alpha}}_{s|(1:\tau-1)}) \\
&\quad + (k_{(1:\tau-1)}(\boldsymbol{x}, \boldsymbol{X}_{(\tau)})\hat{\boldsymbol{\alpha}}_{s|(1:\tau-1)} - k_{\mathcal{S}}(\boldsymbol{x}, \boldsymbol{X}_{(\tau)})\hat{\boldsymbol{\alpha}}_{s|\mathcal{S}})||_2^2 \\
&\leq \delta_1 + \delta_2 + ||(k_{(1:\tau-1)}(\boldsymbol{x}, \boldsymbol{X}_{(\tau)})\hat{\boldsymbol{\alpha}}_s - k_{\mathcal{S}}(\boldsymbol{x}, \boldsymbol{X}_{(\tau)})\hat{\boldsymbol{\alpha}}_s)||_2^2 \lesssim \delta_1 + \delta_2 = \delta
\end{aligned}
$$

where $\hat{y}$ is equivalent to the prediction using a batch algorithm from Corollary 3.3.1.1.

□

CHAPTER 4

Sequential Maximum Entropy Discrimination with Partial Labels

In many real-world applications, data is not collected as one batch, but sequentially over time, and often it is not possible or desirable to wait until the data is completely gathered before analyzing it. Thus, we propose a framework to sequentially update a maximum margin classifier by taking advantage of the Maximum Entropy Discrimination principle. Our maximum margin classifier allows for a kernel representation to represent large numbers of features and can also be regularized with respect to a smooth sub-manifold, allowing it to incorporate unlabeled observations. We compare the performance of our classifier to its non-sequential equivalents in both simulated and real datasets.

4.1 Introduction

As the popularity of big data increases and more data is being gathered, the importance of sequential models that are able to continuously update with new data has increased. These models are particularly crucial in high throughput real-time applications such as speech or streaming text classification. To this end, we propose a sequential framework to update the probabilistic maximum margin classifier built from the Maximum Entropy Discrimination (MED) principle of [54].

The proposed sequential MED framework can be cast as recursive Bayesian estimation where the likelihood function is a log-linear model formed from a series of constraints and weighted by Lagrange multipliers. In the Gaussian case it shares similarities with the problem of Gaussian process classification, which has been previously studied [63, 64, 65, 66, 67, 68], but to the best of our knowledge, a method to recursively update the Gaussian process classifier has not been developed. In the single time point case, sequential MED can be specialized to the support vector machine [65] and Laplacian support vector machine [1] as previously discussed in [54] and [69].

We are interested in situations where we receive a stream of data $\boldsymbol{X}_{(1)}, \boldsymbol{X}_{(2)}, \ldots$ over time t where each $X_{(t)}$ is a matrix of dimension $p \times n$, with p denoting the number of feature variables and n denoting the number of i.i.d. samples, where $n = n_{(t)}$ may vary with time. In the fully labeled scenario, the data has corresponding labels $y_i = [1, -1] \, \forall i$ and t; however in the partially labeled scenario, at each time point t, only $l_{(t)} < n_{(t)}$ of the samples have labels. We define the observed data at any time point t as $\mathcal{D}_{(t)} = \{\boldsymbol{X}_{(t)}, \boldsymbol{y}_{(t)}\}$ and all observed data up to time τ as $\{\mathcal{D}_{(t)}\}_{t=1}^{\tau}$. Such scenarios would arise in a variety of domains such as a satellite that only transmits its data daily or a government agency that only releases its data quarterly with their corresponding reports. The rest of the paper is organized as follows: Section 2 and Section 3 will discuss how to sequentially update the corresponding MED models for supervised and semi-supervised classification. Section 4 validates the method by simulation and we present an application to a dataset of spoken letters of the English alphabet.

4.2 Sequential MED

Constrained relative entropy minimization is used to estimate the closest distribution to a given prior distribution subject to a set of moment constraints. The authors of [61] show that, if the prior distribution is from the exponential family, then the density that optimizes the constrained relative entropy problem is also a member of the exponential family. Similar to Bayesian conjugate priors, there exist relative entropy conjugate priors that facilitate evaluation of the closest distribution. These produce optimal constrained relative entropy densities, which can be thought of as posteriors, from the same parametric family as the prior. Maximum entropy discrimination (MED) [54] also admits conjugate priors as it a special case of constrained relative entropy minimization where one of the constraints is over a parametric family of discriminant functions $\mathcal{L}(\boldsymbol{X}|\boldsymbol{\Theta})$.

4.2.1 Review of MED for Maximum Margin Classification

In this paper, we are interested in maximum margin binary classifiers. In this case the discriminant function $\mathcal{L}(\boldsymbol{X}|\boldsymbol{\theta}, b) = f(\boldsymbol{X})\boldsymbol{\theta} + b$ is linear for some feature transformation $f(\cdot)$, feature weights vector $\boldsymbol{\theta}$, and bias term b. Slack variables γ_i are used to create a margin in the constraints $\mathrm{E}(y_i(f(\boldsymbol{X}_i)\boldsymbol{\theta} + b) - \gamma_i)$, the expected hinge loss with slack variables. The

MED objective function is

$$\min_{\mathrm{P}(\boldsymbol{\Theta},\boldsymbol{\gamma}|\mathcal{D})} \mathrm{KL}\left(\mathrm{P}(\boldsymbol{\Theta},\boldsymbol{\gamma}|\mathcal{D})||\mathrm{P}_0(\boldsymbol{\Theta},\boldsymbol{\gamma})\right) \tag{4.1}$$
$$\text{subject to } \iint \mathrm{P}(\boldsymbol{\Theta},\boldsymbol{\gamma}|\mathcal{D})\left(y_i(f(\boldsymbol{X}_i)\boldsymbol{\theta}+b)-\gamma_i\right)d\boldsymbol{\Theta}d\boldsymbol{\gamma} \geq 0 \;\forall i=1,\ldots,n$$

whose solution $\mathrm{P}(\boldsymbol{\Theta},\boldsymbol{\gamma}|\mathcal{D})$ is the constrained minimum relative entropy posterior. The associated MED decision rule $\hat{y}_{i'} = \mathrm{sgn}(\iint \mathrm{P}(\boldsymbol{\Theta}|\mathcal{D})(f(\boldsymbol{x}_{i'})\boldsymbol{\theta}+b)\,d\boldsymbol{\Theta})$ is a weighted combination of discriminant functions. The minimum relative entropy posterior has the form

$$\mathrm{P}(\boldsymbol{\Theta},\boldsymbol{\gamma}|\mathcal{D}) = \frac{\mathrm{P}_0(\boldsymbol{\Theta},\boldsymbol{\gamma})}{Z(\boldsymbol{\alpha})}\exp\left\{\sum_{i=1}^{n}\alpha_i\left(y_i(f(\boldsymbol{X})\boldsymbol{\theta}+b)-\gamma_i\right)\right\}$$

where $\boldsymbol{\alpha} = [\alpha_1,...,\alpha_n]^T \geq 0$ are Lagrange multipliers that minimize the partition function $Z(\boldsymbol{\alpha})$. It is common to set the initial prior distribution to the separable form:
$\mathrm{P}_0(\boldsymbol{\Theta},\boldsymbol{\gamma}) = \mathrm{P}_0(\boldsymbol{\theta})\mathrm{P}_0(b)\prod_{i=1}^{n}\mathrm{P}_0(\gamma_i)$. If in addition, we specify that
$\mathrm{P}_0(\gamma_i) = Ce^{-C(1-\gamma_i)}\mathcal{I}(\gamma_i \leq 1)$, $\mathrm{P}_0(\boldsymbol{\theta})$ is $N(\mathbf{0},\mathbf{I})$, and $\mathrm{P}_0(b)$ is a zero mean Bayesian non-informative (diffuse) prior, denoted $N(0,\infty)$, then the Lagrange multipliers can be obtained as the solution $\hat{\boldsymbol{\alpha}}$ to the constrained optimization

$$\max_{\boldsymbol{\alpha}} -\frac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{Y}f(\boldsymbol{X})f(\boldsymbol{X})^T\boldsymbol{Y}\boldsymbol{\alpha} + \sum_{i=1}^{n}\alpha_i + \log(1-\alpha_i/C)$$
$$\text{subject to } \sum_{i=1}^{n} y_i\alpha_i = 0 \text{ and } \alpha_1,\ldots,\alpha_n \geq 0$$

where $\boldsymbol{Y} = \mathrm{diag}(\boldsymbol{y})$. This objective function has a log barrier term $\log(1-\alpha_i/C)$ instead of the inequality constraints $\alpha_i \leq C$ commonly found in the dual form of the SVM. Except in some ill-defined cases where the maximum lies near the boundary of the feasible set, the $\hat{\alpha}_i$ will be identical to the optimal support vectors that maximize the SVM objective. The authors in [54, 69] show that the *maximum a posteriori* (MAP) estimator for $\boldsymbol{\theta}$ of the MED posterior is related to the Lagrange multipliers by $\hat{\boldsymbol{\theta}} = f(\boldsymbol{X})^T\hat{\boldsymbol{\alpha}}$, so the MED posterior mode is equivalent to a maximum margin classifier.

4.2.2 Updating MED

Under the separable prior assumptions above, the MED posterior $\mathrm{P}(\boldsymbol{\Theta},\boldsymbol{\gamma}|\mathcal{D})$ will take the factored form $\mathrm{P}(\boldsymbol{\theta}|\mathcal{D})\mathrm{P}(b|\mathcal{D})\mathrm{P}(\boldsymbol{\gamma})$. Due to the fact that the slack parameters γ_i do not depend on the data $\mathcal{D}$, the density $\mathrm{P}(\boldsymbol{\gamma})$ does not affect the MED decision rule given after

(4.1). Hence only $\mathrm{P}(\boldsymbol{\theta}|\mathcal{D})$ and $\mathrm{P}(b|\mathcal{D}$ are important. This remaining part of the MED posterior has the form: $\mathrm{P}(\boldsymbol{\theta}|\mathcal{D})\mathrm{P}(b|\mathcal{D}) = N(f(\boldsymbol{X})^T\boldsymbol{Y}\boldsymbol{\alpha}, \mathbf{I})N(0,\infty)$, which is a conjugate distribution. Due to this conjugacy the posterior distribution optimizing the objective in (4.1) can be propagated forward in time in a recursive manner. The updating procedure is given in the following theorem and corollaries.

Theorem 4.2.1. *Let the MED prior at $t = 1$ be $\boldsymbol{\theta} \sim N(\mathbf{0}, \mathbf{I}), b \sim N(0,\infty)$, and $\mathrm{P}_0(\gamma_i) = C_{(1)}e^{-C_{(1)}(1-\gamma_i)}\mathcal{I}(\gamma_i \leq 1)$. Then given data $\mathcal{D}_{(\tau)}$ at time point τ, the relative entropy conjugate priors are*

$$\begin{aligned}
\mathrm{P}_0\left(\boldsymbol{\theta}|\{\mathcal{D}_{(t)}\}_{t=1}^{\tau-1}\right) &= N\left(\sum_{t=1}^{\tau-1} f(\boldsymbol{X}_{(t)})^T\boldsymbol{Y}_{(t)}\hat{\boldsymbol{\alpha}}_{(t)}, \mathbf{I}\right) \\
\mathrm{P}_0\left(b|\{\mathcal{D}_{(t)}\}_{t=1}^{\tau-1}\right) &= N(0,\infty) \\
\mathrm{P}_0(\boldsymbol{\gamma}) &= \prod_{i=1}^{n_{(\tau)}} C_{(\tau)} \exp\left\{-C_{(\tau)}(1-\gamma_i)\right\}\mathcal{I}(\gamma_i \leq 1)
\end{aligned}$$

and the MED posterior $\mathrm{P}(\boldsymbol{\Theta}|\{\mathcal{D}\}_{t=1}^{\tau})$ can represented as

$$\mathrm{P}\left(\boldsymbol{\theta}|\{\mathcal{D}\}_{t=1}^{\tau}\right) = N\left(\boldsymbol{\mu}_0 + f(\boldsymbol{X}_{(\tau)})^T\boldsymbol{Y}_{(\tau)}\hat{\boldsymbol{\alpha}}_{(\tau)}, \mathbf{I}\right)$$

where $\boldsymbol{\mu}_0 = \sum_{t=1}^{\tau-1} f(\boldsymbol{X}_{(t)})^T\boldsymbol{Y}_{(t)}\hat{\boldsymbol{\alpha}}_{(t)}$ is the prior mean and $\mathrm{P}(b|\{\mathcal{D}\}_{t=1}^{\tau})$ is the same as the Bayes non-informative prior.

Introducing the kernel function $k(\boldsymbol{x}, \boldsymbol{x}') = \langle f(\boldsymbol{x}), f(\boldsymbol{x}')\rangle$ and the parameter transformation $\boldsymbol{\omega} = f(\boldsymbol{X})\boldsymbol{\theta}$, the posterior at time $\tau > 0$ can be represented in terms of this kernel.

Corollary 4.2.1.1. *The equivalent prior at $t = 1$ for the transformed parameter is $\boldsymbol{\omega} \sim N(\mathbf{0}, \boldsymbol{K}_{(1)})$ where $\boldsymbol{K}_{(1)} = f(\boldsymbol{X}_{(1)})f(\boldsymbol{X}_{(1)})^T$. Furthermore, the posterior at time τ is of Gaussian form $\mathrm{P}(\boldsymbol{\omega}|\{\mathcal{D}_{(t)}\}_{t=1}^{\tau}) = N(\boldsymbol{\mu}_{(\tau)}, \boldsymbol{K}_{(\tau)})$ where the mean parameter satisfies the recursions $\boldsymbol{\mu}_{(\tau)} = \boldsymbol{\mu}_{(\tau-1)} + \boldsymbol{K}_{(\tau)}\boldsymbol{Y}_{(\tau)}\hat{\boldsymbol{\alpha}}_{(\tau)}$.*

Since $\mathrm{P}(\boldsymbol{\theta}|\{\mathcal{D}_{(t)}\}_{t=1}^{\tau})$ is Gaussian, the MAP estimator is simply the mean parameter $\boldsymbol{\mu}_{(\tau)}$ given in the Corollary 4.2.1.1. Thus the decision rule reduces to $\hat{y}_{i'} = \mathrm{sgn}(f(\boldsymbol{x}_{i'})\hat{\boldsymbol{\theta}} + \hat{b})$ where the MAP estimator $\hat{\boldsymbol{\theta}}$ is a function of the previously estimated Lagrange multipliers $\hat{\boldsymbol{\alpha}}_{(1)}, \ldots, \hat{\boldsymbol{\alpha}}_{(\tau-1)}$ and the maximizing values $\hat{\boldsymbol{\alpha}}_{(\tau)}$ and $\hat{b}$ for the current time point τ.

Corollary 4.2.1.2. *Given all previous $\hat{\boldsymbol{\alpha}}_{(1)}, \ldots, \hat{\boldsymbol{\alpha}}_{(\tau-1)}$, the current optimal Lagrange multipliers $\hat{\boldsymbol{\alpha}}_{(\tau)}$ are the solution to*

$$\max_{\boldsymbol{\alpha}_{(\tau)}} -\frac{1}{2}\boldsymbol{\alpha}_{(\tau)}^T \boldsymbol{Y}_{(\tau)}\boldsymbol{K}_{(\tau)}\boldsymbol{Y}_{(\tau)}\boldsymbol{\alpha}_{(\tau)} + \sum_{i=1}^{n_{(\tau)}} \log(1-\frac{\alpha_{(\tau)i}}{C_{(\tau)}}) + \boldsymbol{\alpha}_{(\tau)}^T(\mathbf{1} - \boldsymbol{Y}_{(\tau)}\sum_{t=1}^{\tau-1} k(\boldsymbol{X}_{(\tau)}, \boldsymbol{X}_{(t)})\boldsymbol{Y}_{(t)}\hat{\boldsymbol{\alpha}}_{(t)})$$

subject to $\boldsymbol{y}_{(\tau)}^T\boldsymbol{\alpha}_{(\tau)} = 0$ *and* $\alpha_{(\tau)i} \geq 0$ *for all* $i = 1, \ldots, n_{(\tau)}$

and, holding the Lagrange multipliers fixed, the optimal bias $\hat{b} =$

$$\arg\min_b \sum_{s\in\{i|\hat{\alpha}_{(\tau)i}\neq 0\}} \left| \left(y_{(\tau)s} - \sum_{t=1}^{\tau} k(\boldsymbol{X}_{(\tau)s}, \boldsymbol{X}_{(t)})\boldsymbol{Y}_{(t)}\hat{\boldsymbol{\alpha}}_{(t)} \right) - b \right|$$

ensures that the expectation constraints in the objective hold.

The above dual formulation for the Lagrange multipliers $\boldsymbol{\alpha}_{(\tau)}$ has some interesting implications. Since the Lagrange multipliers from the previous time points are fixed at time step τ, the factor $\mathbf{1} - \boldsymbol{Y}_{(\tau)}\sum_{t=1}^{\tau-1} k(\boldsymbol{X}_{(\tau)}, \boldsymbol{X}_{(t)})\boldsymbol{Y}_{(t)}\hat{\boldsymbol{\alpha}}_{(t)}$ are constants and can be thought of as (unnormalized) weights for $\boldsymbol{\alpha}_{(\tau)}$, the Lagrange multipliers from the current time point. Thus the corresponding Lagrange multipliers for samples that are easily predicted using only the prior information will have lower weight than the Lagrange multipliers for samples that are difficult or incorrect.

4.3 Manifold Regularization

Next we consider the case wheres some of the labels are missing. Without loss of generality we will assume the first l points are labeled and the latter $n - l$ points are unlabeled.

We will adopt the semi-supervised MED classification framework of [69], called Laplacian MED (LapMED). LapMED introduces an additional "geometric" constraint

$$\iint \mathrm{P}(\boldsymbol{\theta}, \lambda) \left(\int_{x\in\mathcal{M}} \boldsymbol{\theta}^T f(\boldsymbol{x})\Delta_{\mathcal{M}} f(\boldsymbol{x})\boldsymbol{\theta}\, d\mathcal{P}_x - \lambda \right) d\boldsymbol{\theta} d\lambda \leq 0 \tag{4.2}$$

to (4.1) where $\mathcal{M} = \mathrm{supp}(\mathcal{P}_X) \subset \mathbb{R}^n$ is a compact submanifold, $\Delta_{\mathcal{M}}$ is the Laplace-Beltrami operator on $\mathcal{M}$, and λ controls the complexity of the decision boundary in the intrinsic geometry of $\mathcal{P}_X$. This constraint was motivated by the semi-supervised framework of [1] to encourage the function $f(x)$ to be smooth over the support set of the feature distribution $\mathcal{P}_X$, inducing a geometric interpolation of unlabeled points. Since the marginal distribution is unknown, from [70]

$$f(\boldsymbol{X})^T \boldsymbol{L} f(\boldsymbol{X}) \to \int_{x \in \mathcal{M}} f(\boldsymbol{x}) \Delta_{\mathcal{M}} f(\boldsymbol{x}) \, d\mathcal{P}_x, \text{ as } n \to \infty$$

where $\boldsymbol{L}$ is the normalized graph Laplacian formed with a heat kernel. The LapMED posterior can be approximated as $\mathrm{P}(\boldsymbol{\theta}, b, \boldsymbol{\gamma}, \lambda | \mathcal{D}) =$

$$\frac{\mathrm{P}_0(\boldsymbol{\theta}, b, \boldsymbol{\gamma}, \lambda)}{Z(\boldsymbol{\alpha}, \beta)} \exp \left\{ \sum_{i=1}^{l} \alpha_i \left(y_i (f(\boldsymbol{X})\boldsymbol{\theta} + b) - \gamma_i \right) + \beta \left(\lambda - \boldsymbol{\theta}^T f(\boldsymbol{X})^T \boldsymbol{L} f(\boldsymbol{X}) \boldsymbol{\theta} \right) \right\}$$

where $\beta \geq 0$ is a Lagrange multiplier for the smoothness constraint.

4.3.1 Sequential Laplacian MED

The distribution $\mathrm{P}(\boldsymbol{\Theta}, \boldsymbol{\gamma}, \lambda | \mathcal{D})$ that minimizes the objective with the additional constraint (4.2) can similarly be factorized and, like the distribution of slack parameters considered in Section 2, the distribution of the smoothness parameter λ is also independent of the data $\mathcal{D}$. Likewise, the distribution of the decision rule coefficients $\mathrm{P}(\boldsymbol{\Theta}|\mathcal{D})$ are conjugate distributions with their priors. Thus the updating procedure for the LapMED problem is similar to the updating procedure in Section 4.2.

Theorem 4.3.1. *At $t = 0$, the MED priors for $\boldsymbol{\theta}$ (or $\boldsymbol{\omega}$), b, and γ_i are the same as in Theorem 1, and the prior for λ is a Bayesian zero mean point prior, denoted $Exp.(\infty)$. Then given data $\mathcal{D}_{(\tau)}$ at time point τ, the MED conjugate prior and posterior are still $Exp.(\infty)$ for λ, the same as in Theorem 1 for b and γ_i, and Gaussian of form $N\left(\boldsymbol{\mu}_{(\tau)}, \boldsymbol{\Sigma}_{(\tau)}\right)$ for $\boldsymbol{\theta}$ (or $\boldsymbol{\omega}$). Define a $l \times n$ expansion matrix as $\boldsymbol{J} = [\mathbf{I}\ \mathbf{0}]$. Then the mean and covariance parameters for the distribution of $\boldsymbol{\theta}$ are*

$$\boldsymbol{\mu}_{(\tau)} = \boldsymbol{G}_{(\tau)}^{-1} \sum_{t=1}^{\tau} f(\boldsymbol{X}_{(t)})^T \boldsymbol{J}^T \boldsymbol{Y}_{(t)} \hat{\boldsymbol{\alpha}}_{(t)}, \quad \boldsymbol{\Sigma}_{(\tau)} = \boldsymbol{G}_{(\tau)}^{-1},$$

where $\boldsymbol{G}_{(\tau)} = \boldsymbol{G}_{(\tau-1)} + 2\beta_{(\tau)} f(\boldsymbol{X}_{(\tau)})^T \boldsymbol{L}_{(\tau)} f(\boldsymbol{X}_{(\tau)})$ is a recursive graph of vertex disjoint subgraphs, and for the distribution of $\boldsymbol{\omega}$ are

$$\boldsymbol{\mu}_{(\tau)} = \sum_{t=1}^{\tau} k_{(\tau)} \left(\boldsymbol{X}_{(\tau)}, \boldsymbol{X}_{(t)} \right) \boldsymbol{J}^T \boldsymbol{Y}_{(t)} \hat{\boldsymbol{\alpha}}_{(t)}, \; \boldsymbol{\Sigma}_{(\tau)} = k_{(\tau)} \left(\boldsymbol{X}_{(\tau)}, \boldsymbol{X}_{(\tau)} \right)$$

where $k_{(\tau)}(\boldsymbol{x}, \boldsymbol{x}') = \langle f(\boldsymbol{x}), \boldsymbol{G}_{(\tau)}^{-1} f(\boldsymbol{x}') \rangle$ *is a kernel function that is recursively defined as*

$$k_{(\tau)}(\boldsymbol{x}, \boldsymbol{x}') = k_{(\tau-1)}(\boldsymbol{x}, \boldsymbol{x}') - k_{(\tau-1)}(\boldsymbol{x}, \boldsymbol{X}_{(\tau)}) \Big(\left(2\beta_{(\tau)} \boldsymbol{L}_{(\tau)}\right)^{-1} + k_{(\tau-1)} \left(\boldsymbol{X}_{(\tau)}, \boldsymbol{X}_{(\tau)}\right) \Big)^{-1} k_{(\tau-1)}(\boldsymbol{X}_{(\tau)}, \boldsymbol{x}'). \tag{4.3}$$

Theorem 2 gives the posterior distribution for semi-supervised classification whose form is comparable to the form given in Corollary 4.2.1.1 for the supervised case. Indeed the forms are identical except for the presence of the precision matrix term $G_{(\tau)}$ in the semi-supervised case. As the sparsity of $G_{(\tau)}$ is associated with the graph Laplacian, the kernel function of the semi-supervised case is a regularized version of the kernel function that appears in Corallary 4.2.1.1. If we let $\beta_{(t)}$ be a fixed parameter, then $\hat{\boldsymbol{\alpha}}_{(t)}$ and $\hat{b}$ optimize an objective of the same form as in Corollary 4.2.1.2, but with kernel function $k_{(\tau)}(\boldsymbol{x}, \boldsymbol{x}')$. If $\beta_{(t)}$ is chosen to be 0, the sequential LapMED simply ignores the unlabeled data of time point t, and if all $\beta_{(i)}$'s are 0, then the unlabeled data is always ignored and the updating procedure is exactly the same as in the supervised scenario. These parameters are functions of the γ_A and γ_I, which are identical to the penalty parameters in the Laplacian SVM [1], associated with the reproducing kernel Hilbert space and data distribution respectively: $C_{(t)} = \frac{1}{2l_{(t)}\gamma_A}$ and $\beta_{(t)} = \frac{\gamma_I}{2\gamma_A n_{(t)}^2}$.

4.3.2 Approximating the Kernel Function

Because the kernel function in (4.3) is a function of the previous kernel functions, calculating a map to its associated Hilbert space $\mathcal{H}_{(\tau)}$ can be computationally expensive. Thus in this subsection, we derive an approximation to the map to $\langle \boldsymbol{x}, \boldsymbol{x}' \rangle_{\mathcal{H}_{(\tau)}}$, which is computationally easier than direct recursive calculation.

Recall that we approximate the constraint in (4.2), at any time point t, empirically with the graph Laplacian $\boldsymbol{L}_{(t)}$ formed using the data from that time point $\boldsymbol{X}_{(t)}$. However, the non-empirical constraint using the Laplace-Beltrami operator over the unknown marginal distribution $\mathcal{P}_x$, is actually the same at every time point. Thus as $n_{(\tau-1)} \to \infty$, the prior graph $\boldsymbol{G}_{(\tau-1)}$ converges to

$$B \int_{x \in \mathcal{M}} f(\boldsymbol{x}) \Delta_{\mathcal{M}} f(\boldsymbol{x}) \, d\mathcal{P}_x \approx B \sum_{i=1}^{\infty} \delta_i \xi_i^2 \upsilon_i(z) \upsilon_i(z) \tag{4.4}$$

where $B = 2 \sum_{t=1}^{\tau} \beta_{(t)}$, δ_i are the eigenvalues of the Laplace-Beltrami operator, and $\upsilon_i(z)$ and ξ_i are the infinite sequence of right singular functions and singular values of

$f(x) = \int k(x,z)f(z)\,dz$. The approximate decomposition arises since the left singular functions of f are the eigenfunctions of the Laplace-Beltrami operator [71] and [1]. Thus instead of empirically approximating the Laplacian as a sum of subgraphs $\boldsymbol{G}_{(\tau-1)} = \mathbf{I} + \sum_{t=1}^{\tau-1} 2\beta_{(t)} f(\boldsymbol{X}_{(t)})^T \boldsymbol{L}_{(t)} f(\boldsymbol{X}_{(t)})$, we can instead implement approximations to the eigen/singular values and singular functions in (4.4).

Assuming that the sample size n is large enough, the average eigenvalues of the $\tau - 1$ graph Laplacians would be a good estimator for the eigenvalues of the Laplace-Beltrami operator. Additionally the rows of the matix $\boldsymbol{V}^T$ from the singular value decomposition of $\boldsymbol{X}$ will contain the basis for its row space. Thus because the right singular functions form an orthonormal basis for the coimage of f, if the mapping approximately preserves the basis, the mapped average singular vectors $f(\bar{\boldsymbol{V}}_i)$ would be good estimators for the right singular functions $v_i(z)$ and correspondingly so for the singular values.

The posterior kernel function $k_{(\tau)}(\boldsymbol{x}, \boldsymbol{x}')$ using an approximation to the decomposition in (4.4) will no longer be a recursive function of prior kernel functions $k_{(\tau-1)}(\boldsymbol{x}, \boldsymbol{x}')$ that have the same form, like in (4.3). Instead for $\tau > 2$, it uses a prior kernel function $\tilde{k}_{(\tau-1)}(\boldsymbol{x}, \boldsymbol{x}') =$

$$k(\boldsymbol{x}, \boldsymbol{x}') - k(\boldsymbol{x}, \bar{\boldsymbol{V}}_{(\tau-1)}) \left(\frac{\text{diag}(\bar{\boldsymbol{s}}^2_{(\tau-1)} \bar{\boldsymbol{d}}_{(\tau-1)})^{-1}}{B} + k(\bar{\boldsymbol{V}}_{(\tau-1)}, \bar{\boldsymbol{V}}_{(\tau-1)}) \right)^{-1} k(\bar{\boldsymbol{V}}_{(\tau-1)}, \boldsymbol{x}').$$

where $k(\boldsymbol{x}, \boldsymbol{x}') = \langle f(\boldsymbol{x}), f(\boldsymbol{x}') \rangle$ is the non-regularized kernel function. So at time τ, the singular vectors of $\boldsymbol{X}_{(\tau-1)}$ are used to update the average singular vectors, in the above function, through

$$\bar{\boldsymbol{V}}_{(\tau-1)} = \bar{\boldsymbol{V}}_{(\tau-2)} + \frac{\boldsymbol{V}_{(\tau-1)} - \bar{\boldsymbol{V}}_{(\tau-2)}}{\tau - 1}$$

and similarly so for the average corresponding singular values $\bar{\boldsymbol{s}}_{(\tau-1)}$ and the average eigenvalues of the graph Laplacians $\bar{\boldsymbol{d}}_{(\tau-1)}$.

4.4 Experiments

In this section, we compare the proposed sequential maximum margin classifiers to popular supervised and semi-supervised maximum margin classifiers (SVM [65] and LapSVM [1]) where the model is trained using just the current time points data and where the model has been re-trained on all previous data. The former type of model is a lower bound on performance since it ignores all previous data and the latter type of model is an upper bound since it is re-trained on all previous data at every time point. Note the MED and SVM models only differ by a weak log-barrier term in the objective function making their

performance identical, and similarly so for LapMED and LapSVM. Thus their performance curves will referred to as Full SVM/MED and Full LapSVM/LapMED.

4.4.1 Simulations

In both of the following simulations, the models receive roughly 100 samples ($n_{(t)} = [97, 103]$) at every time point, the parameters are empirically chosen with a validation set, and then the models are tested on an independent data set of 1000 test points. The test accuracy $\frac{TP+TN}{1000}$ is the average accuracy over 100 trials of simulation.

In the first simulation, we generate data from 200 categorical distributions where 100 of the variables are sparse so they have high probability of being 0, another 50 of the variables have lower probability of being 0, and the final 50 variables are used to distinguish between the two classes. We use the term frequency - inverse document frequency (TF-IDF) kernel of [72], which is used in document processing and topic models. Figure 4.1 shows that the accuracy of the sequential model (SeqMED) improves as the model is updated with more training data and has much better results even after one model update versus the independent model (SVM) that ignores previous training data. Of course the sequential model does not improve as rapidly as the model that is re-trained on all the data (Full SVM/MED), but this is the price paid for lower computational complexity. For example, at $t = 30$, SeqMED updates and fits 100 coefficients for the new data whereas Full SVM/MED fits 3,000 coefficients for all the data.

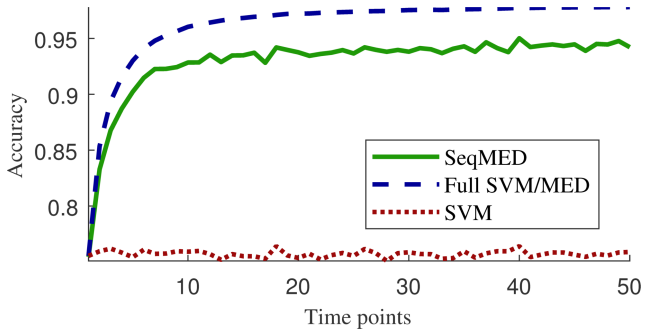


Figure 4.1: Accuracy of prediction for categorical fully labeled simulated data. The proposed sequential MED (SeqMED) classifier performs almost as well as the full batch implementation of the SVM/MED (Full SVM/MED).

In the second simulation, we generate data from the interior of a 3-dimensional sphere where one class is roughly at the center of the sphere and the other class is on the shell, but only 10% of the samples are labeled. We use a rbf kernel with width 1 for the kernel function and a heat kernel with width 0.01 and a 20 nearest neighbors graph for the graph

Laplacian. Figure 4.2 shows improvement in performance of the sequential model similar to in Figure 4.1. We use the approximate kernel function of Subsection 4.3.2 to perform each update, establishing that the approximation is adequate.

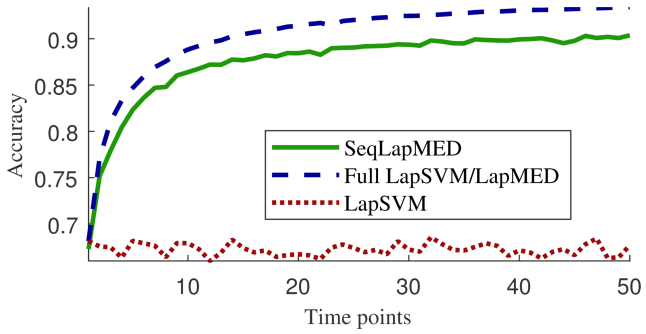


Figure 4.2: Accuracy of prediction for continuous simulated data with 10% labeled.

4.4.2 Data

We compare the proposed algorithms on the Isolet speech database from the UCI machine learning repository [73] following the experimental framework used in [1]. To train the models, we take the entire training set of 120 speakers (isolet1 - isolet4) and break them into 24 groups (time points) of 5 speakers where only the first speaker is labeled. At each time point, the models train on 260 samples ($t = 21$ and 23 only have 259) where 52 of the samples are labeled. The parameters are set in the same way as in [1] and the test set is similarly composed of the 1,559 samples from isolet5. Figure 4.3 shows that, after two time points, the sequential model always performs better than the model that ignores previous data, and comes close to performing as well as the fully re-trained model as time progresses.

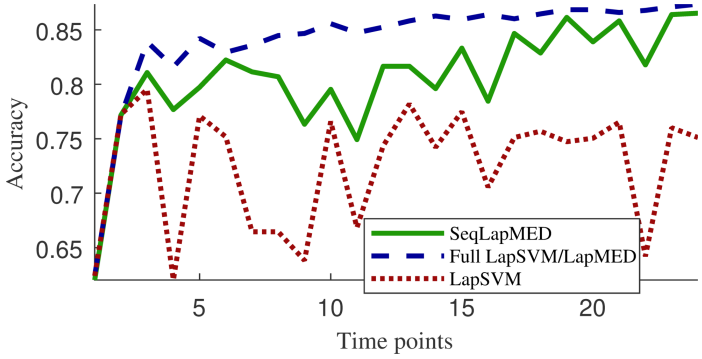


Figure 4.3: Accuracy of prediction on isolet5 for models trained on partially labeled speech isolets 1-4. The proposed semi-supervised sequential Laplacian MED classifier (SeqLapMED) comes close to the full Laplacian SVM [1] as time progresses.

4.5 Conclusions

We have proposed recursive versions of supervised and semi-supervised maximum margin classifiers in the minimum entropy discrimination (MED) classification framework. The proposed sequential maximum margin classifiers perform nearly as well as a much more computationally expensive fully re-trained maximum margin classifiers and significantly better than a classifier that ignores previous data.

Appendix

Proof of Theorem 4.2.1. Let $\boldsymbol{\mu}_{(\tau-1)} = \sum_{t=1}^{\tau-1} f(\boldsymbol{X}_{(t)})^T \boldsymbol{Y}_{(t)} \hat{\boldsymbol{\alpha}}_{(t)}$ where $\boldsymbol{\mu}_{(0)} = \mathbf{0}$. At time τ, let the priors be $\boldsymbol{\theta} \sim N(\boldsymbol{\mu}_{(\tau-1)}, \mathbf{I})$, $b \sim N(0, \sigma^2)$ where $\sigma^2 \to \infty$, and $\gamma_i \sim C_{(\tau)} e^{-C_{(\tau)}(1-\gamma_i)} \mathcal{I}(\gamma_i \leq 1)$. Then the posterior $\mathrm{P}(\boldsymbol{\theta}, b, \boldsymbol{\gamma} | \{\mathcal{D}\}_{t=1}^{\tau})$

$$
\begin{aligned}
&= \frac{\mathrm{P}_0(\boldsymbol{\theta})\mathrm{P}_0(b)\mathrm{P}_0(\boldsymbol{\gamma})}{Z(\hat{\boldsymbol{\alpha}}_{(\tau)})} \exp\left\{ \sum_{i=1}^{n_{(\tau)}} \hat{\alpha}_{(\tau)i} \left(y_{(\tau)i}(f(\boldsymbol{X}_{(\tau)})\boldsymbol{\theta} + b) - \gamma_i \right) \right\} \\
&= \frac{\mathrm{P}_0(\boldsymbol{\theta})}{Z_\theta(\hat{\boldsymbol{\alpha}}_{(\tau)})} \exp\left\{ \sum_{i=1}^{n_{(\tau)}} \hat{\alpha}_{(\tau)i} y_{(\tau)i} f(\boldsymbol{X}_{(\tau)})\boldsymbol{\theta} \right\} \frac{\mathrm{P}_0(\boldsymbol{b})}{Z_b(\hat{\boldsymbol{\alpha}}_{(\tau)})} \exp\left\{ b \sum_{i=1}^{n_{(\tau)}} y_{(\tau)i} \hat{\alpha}_{(\tau)i} \right\} \\
&\quad \frac{\prod_{i=1}^{n_{(\tau)}} \mathrm{P}_0(\gamma_i)}{Z_{\gamma_i}(\hat{\boldsymbol{\alpha}}_{(\tau)})} \exp\left\{ - \sum_{i=1}^{n_{(\tau)}} \hat{\alpha}_{(\tau)i} \gamma_i \right\} \\
&= \mathrm{P}(\boldsymbol{\theta} | \boldsymbol{X}_{(1)}, \boldsymbol{y}_{(1)}, \ldots, \boldsymbol{X}_{(\tau)}, \boldsymbol{y}_{(\tau)}) \mathrm{P}(\boldsymbol{b} | \boldsymbol{y}_{(1)}, \ldots, \boldsymbol{y}_{(\tau)}) \mathrm{P}(\boldsymbol{\gamma}).
\end{aligned}
$$

So the posterior of the weights $\mathrm{P}(\boldsymbol{\theta} | \boldsymbol{X}_{(1)}, \boldsymbol{y}_{(1)}, \ldots, \boldsymbol{X}_{(\tau)}, \boldsymbol{y}_{(\tau)})$

$$
\begin{aligned}
&= \frac{\exp\left\{ -0.5(\boldsymbol{\theta} - \boldsymbol{\mu}_{(\tau-1)})^T (\boldsymbol{\theta} - \boldsymbol{\mu}_{(\tau-1)}) + \hat{\boldsymbol{\alpha}}_{(\tau)}^T \boldsymbol{Y}_{(\tau)} f(\boldsymbol{X}_{(\tau)})\boldsymbol{\theta} \right\}}{(2\pi)^{p/2} Z(\hat{\boldsymbol{\alpha}}_{(\tau)})} \\
&= \frac{\frac{\exp\left\{ -0.5\left(\boldsymbol{\theta}^T\boldsymbol{\theta} - 2\boldsymbol{\mu}_{(\tau-1)}^T\boldsymbol{\theta} - 2\hat{\boldsymbol{\alpha}}_{(\tau)}^T \boldsymbol{Y}_{(\tau)} f(\boldsymbol{X}_{(\tau)})\boldsymbol{\theta} \right) \right\}}{(2\pi)^{p/2}}}{\int \frac{\exp\left\{ -0.5\left(\boldsymbol{\theta}^T\boldsymbol{\theta} - 2\boldsymbol{\mu}_{(\tau-1)}^T\boldsymbol{\theta} - 2\hat{\boldsymbol{\alpha}}_{(\tau)}^T \boldsymbol{Y}_{(\tau)} f(\boldsymbol{X}_{(\tau)})\boldsymbol{\theta} \right) \right\}}{(2\pi)^{p/2}} d\boldsymbol{\theta}} \\
&= \frac{\exp\left\{ -0.5 \left(\boldsymbol{\theta} - (\boldsymbol{\mu}_{(\tau-1)} + f(\boldsymbol{X}_{(\tau)})^T \boldsymbol{Y}_{(\tau)} \hat{\boldsymbol{\alpha}}_{(\tau)}) \right)^T \left(\boldsymbol{\theta} - (\boldsymbol{\mu}_{(\tau-1)} + f(\boldsymbol{X}_{(\tau)})^T \boldsymbol{Y}_{(\tau)} \hat{\boldsymbol{\alpha}}_{(\tau)}) \right) \right\}}{(2\pi)^{p/2}} \\
&\sim N(\boldsymbol{\mu}_{(\tau-1)} + f(\boldsymbol{X}_{(\tau)})^T \boldsymbol{Y}_{(\tau)} \hat{\boldsymbol{\alpha}}_{(\tau)}, \mathbf{I}),
\end{aligned}
$$

the posterior of the bias term $\mathrm{P}(\boldsymbol{b}|\boldsymbol{y}_{(1)},\ldots,\boldsymbol{y}_{(\tau)})$

$$= \frac{(2\pi\sigma^2)^{-1/2}\exp\left\{-0.5(b^2-2\sigma^2 b\boldsymbol{y}_{(\tau)}^T\hat{\boldsymbol{\alpha}}_{(\tau)})/\sigma^2\right\}}{\int(2\pi\sigma^2)^{-1/2}\exp\left\{-0.5(b^2-2\sigma^2 b\boldsymbol{y}_{(\tau)}^T\hat{\boldsymbol{\alpha}}_{(\tau)})/\sigma^2\right\}db} = \frac{\exp\{-0.5(b-\sigma^2\boldsymbol{y}_{(\tau)}^T\hat{\boldsymbol{\alpha}}_{(\tau)})^2/\sigma^2\}}{\sqrt{2\pi\sigma^2}}$$

$$\sim N(\sigma^2\boldsymbol{y}_{(\tau)}^T\hat{\boldsymbol{\alpha}}_{(\tau)},\sigma^2)$$

$$\Rightarrow \text{if } \sigma\to\infty, \text{then } N(\sigma^2\boldsymbol{y}_{(\tau)}^T\hat{\boldsymbol{\alpha}}_{(\tau)},\sigma^2)\to N(0,\infty)$$

as long as the optimal Lagrange multipliers satisfy $\boldsymbol{y}_{(\tau)}^T\hat{\boldsymbol{\alpha}}_{(\tau)}=0$, and the posterior of the margin parameters $\mathrm{P}(\boldsymbol{\gamma})$ do not depend on the data.

□

Proof of Corollary 4.2.1.1. At time τ, let $\boldsymbol{\omega}=f(\boldsymbol{X}_{(\tau)})\boldsymbol{\theta}$ have prior $N(\boldsymbol{\mu}_{(\tau-1)},\boldsymbol{K}_{(\tau)})$ where $\boldsymbol{\mu}_{(\tau-1)}=\sum_{t=1}^{\tau-1}k(\boldsymbol{X}_{(\tau)},\boldsymbol{X}_{(t)})\boldsymbol{Y}_{(t)}\hat{\boldsymbol{\alpha}}_{(t)}$. Then the posterior $\mathrm{P}(\boldsymbol{\omega}|\boldsymbol{X}_{(1)},\boldsymbol{y}_{(1)},\ldots,\boldsymbol{X}_{(\tau)},\boldsymbol{y}_{(\tau)})$

$$= \frac{\mathrm{P}_0(\boldsymbol{\omega})\exp\left\{\sum_{i=1}^{n_{(\tau)}}\hat{\alpha}_{(\tau)i}y_{(\tau)i}\boldsymbol{\omega}\right\}}{Z_\omega(\hat{\boldsymbol{\alpha}}_{(\tau)})} = \frac{\exp\left\{-0.5(\boldsymbol{\omega}-\boldsymbol{\mu}_{(\tau-1)})^T\boldsymbol{K}_{(\tau)}^{-1}(\boldsymbol{\omega}-\boldsymbol{\mu}_{(\tau-1)})+\hat{\boldsymbol{\alpha}}_{(\tau)}^T\boldsymbol{Y}_{(\tau)}\boldsymbol{\omega}\right\}}{|2\pi\boldsymbol{K}_{(\tau)}|^{1/2}Z_\omega(\hat{\boldsymbol{\alpha}}_{(\tau)})}$$

$$= \frac{\exp\{-0.5\left(\boldsymbol{\omega}-(\boldsymbol{\mu}_{(\tau-1)}+\boldsymbol{K}_{(\tau)}\boldsymbol{Y}_{(\tau)}\hat{\boldsymbol{\alpha}}_{(\tau)})\right)^T\boldsymbol{K}_{(\tau)}^{-1}\left(\boldsymbol{\omega}-(\boldsymbol{\mu}_{(\tau-1)}+\boldsymbol{K}_{(\tau)}\boldsymbol{Y}_{(\tau)}\hat{\boldsymbol{\alpha}}_{(\tau)})\right)\}}{|2\pi\boldsymbol{K}_{(\tau)}|^{1/2}}$$

$$\sim N(\boldsymbol{\mu}_{(\tau-1)}+\boldsymbol{K}_{(\tau)}\boldsymbol{Y}_{(\tau)}\hat{\boldsymbol{\alpha}}_{(\tau)},\boldsymbol{K}_{(\tau)}).$$

□

Proof of Corollary 4.2.1.2. The optimal Lagrange multipliers at $t=\tau$ are the solution to

$$\arg\max_{\boldsymbol{\alpha}_{(\tau)}} -\log\left(Z(\boldsymbol{\alpha}_{(\tau)})\right) = \arg\max_{\boldsymbol{\alpha}_{(\tau)}} -\log\left(Z_\theta(\boldsymbol{\alpha}_{(\tau)})\right)-\log\left(Z_b(\boldsymbol{\alpha}_{(\tau)})\right)-\log\left(Z_\gamma(\boldsymbol{\alpha}_{(\tau)})\right)$$

or

$$\arg\max_{\boldsymbol{\alpha}_{(\tau)}} -\log\left(Z(\boldsymbol{\alpha}_{(\tau)})\right) = \arg\max_{\boldsymbol{\alpha}_{(\tau)}} -\log\left(Z_\omega(\boldsymbol{\alpha}_{(\tau)})\right)-\log\left(Z_b(\boldsymbol{\alpha}_{(\tau)})\right)-\log\left(Z_\gamma(\boldsymbol{\alpha}_{(\tau)})\right)$$

where

$$
\begin{aligned}
-\log\left(Z_\theta(\boldsymbol{\alpha}_{(\tau)})\right) &= -\log\left(\int \frac{e^{\boldsymbol{\alpha}_{(\tau)}^T \boldsymbol{Y}_{(\tau)} f(\boldsymbol{X}_{(\tau)})\boldsymbol{\theta} - 0.5(\boldsymbol{\theta}-\boldsymbol{\mu}_{(\tau-1)})^T(\boldsymbol{\theta}-\boldsymbol{\mu}_{(\tau-1)})}}{(2\pi)^{p/2}} d\boldsymbol{\theta}\right) \\
&= -\boldsymbol{\alpha}_{(\tau)}^T \boldsymbol{Y}_{(\tau)} f(\boldsymbol{X}_{(\tau)})\boldsymbol{\mu}_{(\tau-1)} - 0.5\boldsymbol{\alpha}_{(\tau)}^T \boldsymbol{Y}_{(\tau)} f(\boldsymbol{X}_{(\tau)}) f(\boldsymbol{X}_{(\tau)})^T \boldsymbol{Y}_{(\tau)} \boldsymbol{\alpha}_{(\tau)} \\
-\log\left(Z_\omega(\boldsymbol{\alpha}_{(\tau)})\right) &= -\log\left(\int \frac{e^{\boldsymbol{\alpha}_{(\tau)}^T \boldsymbol{Y}_{(\tau)} \boldsymbol{\omega} - 0.5(\boldsymbol{\omega}-\boldsymbol{\mu}_{(\tau-1)})^T \boldsymbol{K}_{(\tau)}^{-1}(\boldsymbol{\omega}-\boldsymbol{\mu}_{(\tau-1)})}}{|2\pi \boldsymbol{K}_{(\tau)}|^{1/2}} d\boldsymbol{\omega}\right) \\
&= -\boldsymbol{\alpha}_{(\tau)}^T \boldsymbol{Y}_{(\tau)} \boldsymbol{\mu}_{(\tau-1)} - 0.5\boldsymbol{\alpha}_{(\tau)}^T \boldsymbol{Y}_{(\tau)} \boldsymbol{K}_{(\tau)} \boldsymbol{Y}_{(\tau)} \boldsymbol{\alpha}_{(\tau)} \\
-\log\left(Z_b(\boldsymbol{\alpha}_{(\tau)})\right) &= -\log\left(\int \frac{e^{-0.5(b-\sigma^2 \boldsymbol{y}_{(\tau)}^T \boldsymbol{\alpha}_{(\tau)})^2/\sigma^2}}{\sqrt{2\pi\sigma^2}} db\right) - \log\left(e^{0.5\sigma^2(\boldsymbol{y}_{(\tau)}^T \boldsymbol{\alpha}_{(\tau)})^2}\right) \\
&= -0.5\sigma^2(\boldsymbol{y}_{(\tau)}^T \boldsymbol{\alpha}_{(\tau)})^2 \quad \Rightarrow \text{if } \sigma \to \infty, \text{ then } \boldsymbol{y}_{(\tau)}^T \boldsymbol{\alpha}_{(\tau)} = 0 \\
-\log\left(Z_\gamma(\boldsymbol{\alpha}_{(\tau)})\right) &= -\sum_{i=1}^{n_{(\tau)}} \log\left(Z_{\gamma_i}(\boldsymbol{\alpha}_{(\tau)})\right) = -\sum_{i=1}^{n_{(\tau)}} \log\left(\int_{-\infty}^{1} C_{(\tau)} e^{-C_{(\tau)}(1-\gamma_i)} e^{-\alpha_{(\tau)i}\gamma_i}\, d\gamma_i\right) \\
&= -\sum_{i=1}^{n_{(\tau)}} \log\left(\frac{C_{(\tau)}}{C_{(\tau)} - \alpha_{(\tau)i}} e^{-C_{(\tau)} + \gamma_i(C_{(\tau)} - \alpha_{(\tau)i})}\Big|_{-\infty}^{1}\right) \\
&= -\sum_{i=1}^{n_{(\tau)}} \log\left(\frac{C_{(\tau)} e^{-\alpha_{(\tau)i}}}{C_{(\tau)} - \alpha_{(\tau)i}}\right) = \sum_{i=1}^{n_{(\tau)}} \alpha_{(\tau)i} + \log\left(1 - \frac{\alpha_{(\tau)i}}{C_{(\tau)}}\right).
\end{aligned}
$$

□

Proof of Theorem 4.3.1. At time τ, let the priors for b and γ_i be the same as in Theorem 4.2.1, $\lambda \sim Exp.(\nu)$ where $\nu \to \infty$, and $\boldsymbol{\theta}$ (or ω) $\sim N\left(\boldsymbol{\mu}_{(\tau-1)}, \boldsymbol{\Sigma}_{(\tau-1)}\right)$. Then the posterior $\mathrm{P}(\boldsymbol{\theta}, b, \boldsymbol{\gamma}, \lambda | \{\mathcal{D}\}_{t=1}^{\tau})$ and partition function $Z_\theta(\boldsymbol{\alpha}_{(\tau)}, \beta_{(\tau)})$ factorize similarly as

$$\mathrm{P}(\boldsymbol{\theta}|\boldsymbol{X}_{(1)}, \boldsymbol{y}_{(1)}, \ldots, \boldsymbol{X}_{(\tau)}, \boldsymbol{y}_{(\tau)})\mathrm{P}(\boldsymbol{b}|\boldsymbol{y}_{(1)}, \ldots, \boldsymbol{y}_{(\tau)})\mathrm{P}(\boldsymbol{\gamma})\mathrm{P}(\lambda)$$

and

$$Z_\theta(\boldsymbol{\alpha}_{(\tau)}, \beta_{(\tau)}) Z_b(\boldsymbol{\alpha}_{(\tau)}) Z_\lambda(\beta_{(\tau)}) \prod_{i=1}^{l_{(\tau)}} Z_{\gamma_i}(\boldsymbol{\alpha}_{(\tau)}).$$

The bias and margin terms are independent of $\beta_{(\tau)}$, so their posterior and partition functions are the same as in Theorem 4.2.1. The posterior of the smoothness parameter λ does not depend on the data and

$$
\begin{aligned}
&-\log\left(Z_\lambda(\beta_{(\tau)})\right) = -\log\left(\int_0^\infty \nu e^{-\nu\lambda} e^{\beta_{(\tau)}\lambda}\, d\lambda\right) = -\log\left(\frac{\nu}{\nu - \beta_{(\tau)}}\right) \\
&\Rightarrow \text{ if } \nu \to \infty, \text{ then } \log(1 - \beta_{(\tau)}/\nu) = 0.
\end{aligned}
$$

Let the parameters for the prior distribution of $\boldsymbol{\theta}$ be

$$\boldsymbol{\mu}_{(\tau-1)} = \boldsymbol{G}_{(\tau-1)}^{-1} \sum_{t=1}^{\tau-1} f(\boldsymbol{X}_{(t)})^T \boldsymbol{J}^T \boldsymbol{Y}_{(t)} \hat{\boldsymbol{\alpha}}_{(t)},\ \boldsymbol{\Sigma}_{(\tau-1)} = \boldsymbol{G}_{(\tau-1)}^{-1},$$

where $\boldsymbol{G}_{(\tau-1)} = \boldsymbol{G}_{(\tau-2)} + 2\beta_{(\tau-1)} f(\boldsymbol{X}_{(\tau-1)})^T \boldsymbol{L}_{(\tau-1)} f(\boldsymbol{X}_{(\tau-1)})$, and $\boldsymbol{G}_{(0)} = \mathbf{I}$, then the posterior $\mathrm{P}(\boldsymbol{\theta}|\boldsymbol{X}_{(1)}, \boldsymbol{y}_{(1)}, \ldots, \boldsymbol{X}_{(\tau)}, \boldsymbol{y}_{(\tau)})$

$$= \frac{\exp\left\{-0.5(\boldsymbol{\theta} - \boldsymbol{\mu}_{(\tau-1)})^T \boldsymbol{\Sigma}_{(\tau-1)}^{-1} (\boldsymbol{\theta} - \boldsymbol{\mu}_{(\tau-1)})\right\}}{\det(2\pi\boldsymbol{\Sigma}_{(\tau-1)})^{1/2}}$$

$$\frac{\exp\left\{\hat{\boldsymbol{\alpha}}_{(\tau)}^T \boldsymbol{Y}_{(\tau)} \boldsymbol{J} f(\boldsymbol{X}_{(\tau)})\boldsymbol{\theta} - \beta_{(\tau)} \boldsymbol{\theta}^T f(\boldsymbol{X}_{(\tau)})^T \boldsymbol{L}_{(\tau)} f(\boldsymbol{X}_{(\tau)})\boldsymbol{\theta}\right\}}{Z_\theta(\hat{\boldsymbol{\alpha}}_{(\tau)}, \beta_{(\tau)})}$$

$$= \exp\left\{ -0.5\left(\boldsymbol{\theta}^T \boldsymbol{G}_{(\tau-1)} \boldsymbol{\theta} + \left(\sum_{t=1}^{\tau-1} f(\boldsymbol{X}_{(t)})^T \boldsymbol{J}^T \boldsymbol{Y}_{(t)} \hat{\boldsymbol{\alpha}}_{(t)}\right)^T \boldsymbol{G}_{(\tau-1)}^{-1} \left(\sum_{t=1}^{\tau-1} f(\boldsymbol{X}_{(t)})^T \boldsymbol{J}^T \boldsymbol{Y}_{(t)} \hat{\boldsymbol{\alpha}}_{(t)}\right)\right.\right.$$

$$\left.\left. - 2\boldsymbol{\theta}^T \sum_{t=1}^{\tau-1} f(\boldsymbol{X}_{(t)})^T \boldsymbol{J}^T \boldsymbol{Y}_{(t)} \hat{\boldsymbol{\alpha}}_{(t)} - 2\hat{\boldsymbol{\alpha}}_{(\tau)}^T \boldsymbol{Y}_{(\tau)} \boldsymbol{J} f(\boldsymbol{X}_{(\tau)})\boldsymbol{\theta} + 2\beta_{(\tau)} \boldsymbol{\theta}^T f(\boldsymbol{X}_{(\tau)})^T \boldsymbol{L}_{(\tau)} f(\boldsymbol{X}_{(\tau)})\boldsymbol{\theta}\right)\right\}$$

$$\Big/ \left(\det(2\pi \boldsymbol{G}_{(\tau-1)}^{-1})^{1/2} Z_\theta(\hat{\boldsymbol{\alpha}}_{(\tau)}, \beta_{(\tau)})\right)$$

$$= \exp\left\{ -0.5\left(\boldsymbol{\theta}^T \boldsymbol{G}_{(\tau)} \boldsymbol{\theta} - 2\boldsymbol{\theta}^T \sum_{t=1}^{\tau} f(\boldsymbol{X}_{(t)})^T \boldsymbol{J}^T \boldsymbol{Y}_{(t)} \hat{\boldsymbol{\alpha}}_{(t)}\right.\right.$$

$$+ \left(\sum_{t=1}^{\tau} f(\boldsymbol{X}_{(t)})^T \boldsymbol{J}^T \boldsymbol{Y}_{(t)} \hat{\boldsymbol{\alpha}}_{(t)}\right)^T \boldsymbol{G}_{(\tau)}^{-1} \left(\sum_{t=1}^{\tau} f(\boldsymbol{X}_{(t)})^T \boldsymbol{J}^T \boldsymbol{Y}_{(t)} \hat{\boldsymbol{\alpha}}_{(t)}\right)\Bigg)$$

$$- \left(\sum_{t=1}^{\tau-1} f(\boldsymbol{X}_{(t)})^T \boldsymbol{J}^T \boldsymbol{Y}_{(t)} \hat{\boldsymbol{\alpha}}_{(t)}\right)^T \frac{\boldsymbol{G}_{(\tau-1)}^{-1}}{2} \left(\sum_{t=1}^{\tau-1} f(\boldsymbol{X}_{(t)})^T \boldsymbol{J}^T \boldsymbol{Y}_{(t)} \hat{\boldsymbol{\alpha}}_{(t)}\right)$$

$$\left. + \left(\sum_{t=1}^{\tau} f(\boldsymbol{X}_{(t)})^T \boldsymbol{J}^T \boldsymbol{Y}_{(t)} \hat{\boldsymbol{\alpha}}_{(t)}\right)^T \frac{\boldsymbol{G}_{(\tau)}^{-1}}{2} \left(\sum_{t=1}^{\tau} f(\boldsymbol{X}_{(t)})^T \boldsymbol{J}^T \boldsymbol{Y}_{(t)} \hat{\boldsymbol{\alpha}}_{(t)}\right)\right)\right\}$$

$$\Big/ \left(\det(2\pi \boldsymbol{G}_{(\tau-1)}^{-1})^{1/2} Z_\theta(\hat{\boldsymbol{\alpha}}_{(\tau)}, \beta_{(\tau)})\right)$$

$$= \frac{e^{-0.5\left(\boldsymbol{\theta} - \boldsymbol{G}_{(\tau)}^{-1} \sum_{t=1}^{\tau} f(\boldsymbol{X}_{(t)})^T \boldsymbol{J}^T \boldsymbol{Y}_{(t)} \hat{\boldsymbol{\alpha}}_{(t)}\right)^T \boldsymbol{G}_{(\tau)} \left(\boldsymbol{\theta} - \boldsymbol{G}_{(\tau)}^{-1} \sum_{t=1}^{\tau} f(\boldsymbol{X}_{(t)})^T \boldsymbol{J}^T \boldsymbol{Y}_{(t)} \hat{\boldsymbol{\alpha}}_{(t)}\right)}}{\det(2\pi \boldsymbol{G}_{(\tau)}^{-1})^{1/2}}$$

$$\sim N\left(\boldsymbol{G}_{(\tau)}^{-1} \sum_{t=1}^{\tau} f(\boldsymbol{X}_{(t)})^T \boldsymbol{J}^T \boldsymbol{Y}_{(t)} \hat{\boldsymbol{\alpha}}_{(t)}, (\boldsymbol{G}_{(\tau)})^{-1}\right)$$

and $-\log\left(Z_\theta(\boldsymbol{\alpha}_{(\tau)}), \beta_{(\tau)}\right)$

$$
\begin{aligned}
&= -\log\frac{\det(2\pi\boldsymbol{G}_{(\tau)}^{-1})^{1/2}}{\det(2\pi\boldsymbol{G}_{(\tau-1)}^{-1})^{1/2}} - 0.5(\sum_{t=1}^{\tau} f(\boldsymbol{X}_{(t)})^T\boldsymbol{J}^T\boldsymbol{Y}_{(t)}\boldsymbol{\alpha}_{(t)})^T\boldsymbol{G}_{(\tau)}^{-1}(\sum_{t=1}^{\tau} f(\boldsymbol{X}_{(t)})^T\boldsymbol{J}^T\boldsymbol{Y}_{(t)}\boldsymbol{\alpha}_{(t)}) \\
&\qquad\qquad - 0.5(\sum_{t=1}^{\tau-1} f(\boldsymbol{X}_{(t)})^T\boldsymbol{J}^T\boldsymbol{Y}_{(t)}\boldsymbol{\alpha}_{(t)})^T\boldsymbol{G}_{(\tau-1)}^{-1}(\sum_{t=1}^{\tau-1} f(\boldsymbol{X}_{(t)})^T\boldsymbol{J}^T\boldsymbol{Y}_{(t)}\boldsymbol{\alpha}_{(t)}) \\
&= \log\left(\det(2\pi\boldsymbol{G}_{(\tau-1)}^{-1})\right) - \log\left(\det(2\pi\boldsymbol{G}_{(\tau)}^{-1})\right) - 0.5\Bigg((\sum_{t=1}^{\tau-1} f(\boldsymbol{X}_{(t)})^T\boldsymbol{J}^T\boldsymbol{Y}_{(t)}\boldsymbol{\alpha}_{(t)})^T \\
&\qquad \left(\boldsymbol{G}_{(\tau-1)}\left(2\beta_{(\tau)}f(\boldsymbol{X}_{(\tau)})^T\boldsymbol{L}_{(\tau)}f(\boldsymbol{X}_{(\tau)})\right)^{-1}\boldsymbol{G}_{(\tau-1)} + \boldsymbol{G}_{(\tau-1)}\right)^{-1}(\sum_{t=1}^{\tau-1} f(\boldsymbol{X}_{(t)})^T\boldsymbol{J}^T\boldsymbol{Y}_{(t)}\boldsymbol{\alpha}_{(t)})\Bigg) \\
&\qquad - 0.5\,\boldsymbol{\alpha}_{(\tau)}^T\boldsymbol{Y}_{(\tau)}\boldsymbol{J}f(\boldsymbol{X}_{(\tau)})\boldsymbol{G}_{(\tau)}^{-1}f(\boldsymbol{X}_{(\tau)})^T\boldsymbol{J}^T\boldsymbol{Y}_{(\tau)}\boldsymbol{\alpha}_{(\tau)} \\
&\qquad - \boldsymbol{\alpha}_{(\tau)}^T\boldsymbol{Y}_{(\tau)}\boldsymbol{J}f(\boldsymbol{X}_{(\tau)})\boldsymbol{G}_{(\tau)}^{-1}\sum_{t=1}^{\tau-1} f(\boldsymbol{X}_{(t)})^T\boldsymbol{J}^T\boldsymbol{Y}_{(t)}\boldsymbol{\alpha}_{(t)} \\
&= 0.5\left(Const._{\beta_{(\tau)}} - \boldsymbol{\alpha}_{(\tau)}^T\boldsymbol{Y}_{(\tau)}\boldsymbol{J}f(\boldsymbol{X}_{(\tau)})\boldsymbol{G}_{(\tau)}^{-1}f(\boldsymbol{X}_{(\tau)})^T\boldsymbol{J}^T\boldsymbol{Y}_{(\tau)}\boldsymbol{\alpha}_{(\tau)}\right) \\
&\qquad - \boldsymbol{\alpha}_{(\tau)}^T\boldsymbol{Y}_{(\tau)}\boldsymbol{J}f(\boldsymbol{X}_{(\tau)})\boldsymbol{G}_{(\tau)}^{-1}\sum_{t=1}^{\tau-1} f(\boldsymbol{X}_{(t)})^T\boldsymbol{J}^T\boldsymbol{Y}_{(t)}\boldsymbol{\alpha}_{(t)}
\end{aligned}
$$

where $Const._{\beta_{(\tau)}}$ can be dropped from the objective when $\beta_{(t)}$ are fixed parameters.
Or let the parameters for the prior distribution of $\boldsymbol{\omega} = f(\boldsymbol{X}_{(\tau)})\boldsymbol{\theta}$ be

$$
\begin{aligned}
&\boldsymbol{\mu}_{(\tau-1)} = \sum_{t=1}^{\tau-1} k_{(\tau-1)}(\boldsymbol{X}_{(\tau)}, \boldsymbol{X}_{(t)})\boldsymbol{J}^T\boldsymbol{Y}_{(t)}\hat{\boldsymbol{\alpha}}_{(t)} \\
&\boldsymbol{\Sigma}_{(\tau-1)} = k_{(\tau-1)}(\boldsymbol{X}_{(\tau)}, \boldsymbol{X}_{(\tau)}) \text{ where } k_{(0)}(\boldsymbol{x}, \boldsymbol{x}') = \langle f(\boldsymbol{x}), f(\boldsymbol{x}')\rangle \\
&k_{(\tau-1)}(\boldsymbol{x}, \boldsymbol{x}') = k_{(\tau-2)}(\boldsymbol{x}, \boldsymbol{x}') \\
&\quad - k_{(\tau-2)}(\boldsymbol{x}, \boldsymbol{X}_{(\tau-1)})\left(\left(2\beta_{(\tau-1)}\boldsymbol{L}_{(\tau-1)}\right)^{-1} + k_{(\tau-2)}\left(\boldsymbol{X}_{(\tau-1)}, \boldsymbol{X}_{(\tau-1)}\right)\right)^{-1} k_{(\tau-2)}(\boldsymbol{X}_{(\tau-1)}, \boldsymbol{x}').
\end{aligned}
$$

The posterior $\mathrm{P}(\boldsymbol{\omega}|\boldsymbol{X}_{(1)}, \boldsymbol{y}_{(1)}, \ldots, \boldsymbol{X}_{(\tau)}, \boldsymbol{y}_{(\tau)})$

$$
\begin{aligned}
&= \frac{\exp\left\{-0.5(\boldsymbol{\omega}-\boldsymbol{\mu}_{(\tau-1)})^T\boldsymbol{\Sigma}_{(\tau-1)}^{-1}(\boldsymbol{\omega}-\boldsymbol{\mu}_{(\tau-1)})\right\}}{\det(2\pi\boldsymbol{\Sigma}_{(\tau-1)})^{1/2}} \frac{\exp\left\{\hat{\boldsymbol{\alpha}}_{(\tau)}^T\boldsymbol{Y}_{(\tau)}\boldsymbol{J}\boldsymbol{\omega} - \beta_{(\tau)}\boldsymbol{\omega}^T\boldsymbol{L}_{(\tau)}\boldsymbol{\omega}\right\}}{Z_\omega(\hat{\boldsymbol{\alpha}}_{(\tau)}, \beta_{(\tau)})} \\
&= \exp\Bigg\{ -0.5\Big(\boldsymbol{\omega}^T\left(k_{(\tau-1)}(\boldsymbol{X}_{(\tau)}, \boldsymbol{X}_{(\tau)})^{-1} + 2\beta_{(\tau)}\boldsymbol{L}_{(\tau)}\right)\boldsymbol{\omega} \\
&+ \big(\sum_{t=1}^{\tau-1} k_{(\tau-1)}(\boldsymbol{X}_{(\tau)}, \boldsymbol{X}_{(t)})\boldsymbol{J}^T\boldsymbol{Y}_{(t)}\hat{\boldsymbol{\alpha}}_{(t)}\big)^T k_{(\tau-1)}(\boldsymbol{X}_{(\tau)}, \boldsymbol{X}_{(\tau)})^{-1}\big(\sum_{t=1}^{\tau-1} k_{(\tau-1)}(\boldsymbol{X}_{(\tau)}, \boldsymbol{X}_{(t)})\boldsymbol{J}^T\boldsymbol{Y}_{(t)}\hat{\boldsymbol{\alpha}}_{(t)}\big) \\
&- 2\boldsymbol{\omega}^T k_{(\tau-1)}(\boldsymbol{X}_{(\tau)}, \boldsymbol{X}_{(\tau)})^{-1}\big(\sum_{t=1}^{\tau-1} k_{(\tau-1)}(\boldsymbol{X}_{(\tau)}, \boldsymbol{X}_{(t)})\boldsymbol{J}^T\boldsymbol{Y}_{(t)}\hat{\boldsymbol{\alpha}}_{(t)} + k_{(\tau-1)}(\boldsymbol{X}_{(\tau)}, \boldsymbol{X}_{(\tau)})\boldsymbol{J}^T\boldsymbol{Y}_{(\tau)}\hat{\boldsymbol{\alpha}}_{(\tau)}\big)\Big)\Bigg\} \\
&\Big/ \left(\det\left(2\pi k_{(\tau-1)}(\boldsymbol{X}_{(\tau)}, \boldsymbol{X}_{(\tau)})\right)^{1/2} Z_\omega(\hat{\boldsymbol{\alpha}}_{(\tau)}, \beta_{(\tau)})\right) \\
&= \frac{e^{-0.5\left(\boldsymbol{\omega}-\sum_{t=1}^{\tau} k_{(\tau)}\left(\boldsymbol{X}_{(\tau)}, \boldsymbol{X}_{(t)}\right)\boldsymbol{J}^T\boldsymbol{Y}_{(t)}\hat{\boldsymbol{\alpha}}_{(t)}\right)^T k_{(\tau)}(\boldsymbol{X}_{(\tau)}, \boldsymbol{X}_{(\tau)})^{-1}\left(\boldsymbol{\omega}-\sum_{t=1}^{\tau} k_{(\tau)}\left(\boldsymbol{X}_{(\tau)}, \boldsymbol{X}_{(t)}\right)\boldsymbol{J}^T\boldsymbol{Y}_{(t)}\hat{\boldsymbol{\alpha}}_{(t)}\right)}}{\det\left(2\pi k_{(\tau)}(\boldsymbol{X}_{(\tau)}, \boldsymbol{X}_{(\tau)})\right)^{1/2}} \\
&\sim N\left(\sum_{t=1}^{\tau} k_{(\tau)}(\boldsymbol{X}_{(\tau)}, \boldsymbol{X}_{(t)})\boldsymbol{J}^T\boldsymbol{Y}_{(t)}\hat{\boldsymbol{\alpha}}_{(t)}, k_{(\tau)}(\boldsymbol{X}_{(\tau)}, \boldsymbol{X}_{(\tau)})\right)
\end{aligned}
$$

and $-\log\left(Z_\omega(\boldsymbol{\alpha}_{(\tau)}), \beta_{(\tau)}\right)$

$$
\begin{aligned}
&= -0.5\Bigg(\log\left(\det\left(k_{(\tau)}(\boldsymbol{X}_{(\tau)}, \boldsymbol{X}_{(\tau)})\, k_{(\tau-1)}(\boldsymbol{X}_{(\tau)}, \boldsymbol{X}_{(\tau)})^{-1}\right)\right) \\
&+ \Big(\sum_{t=1}^{\tau-1} k_{(\tau-1)}(\boldsymbol{X}_{(\tau)}, \boldsymbol{X}_{(t)})\boldsymbol{J}^T\boldsymbol{Y}_{(t)}\boldsymbol{\alpha}_{(t)}\Big)^T k_{(\tau-1)}(\boldsymbol{X}_{(\tau)}, \boldsymbol{X}_{(\tau)})^{-1}\Big(\sum_{t=1}^{\tau-1} k_{(\tau-1)}(\boldsymbol{X}_{(\tau)}, \boldsymbol{X}_{(t)})\boldsymbol{J}^T\boldsymbol{Y}_{(t)}\boldsymbol{\alpha}_{(t)}\Big) \\
&- \Big(\sum_{t=1}^{\tau} k_{(\tau)}(\boldsymbol{X}_{(\tau)}, \boldsymbol{X}_{(t)})\boldsymbol{J}^T\boldsymbol{Y}_{(t)}\boldsymbol{\alpha}_{(t)}\Big)^T k_{(\tau)}(\boldsymbol{X}_{(\tau)}, \boldsymbol{X}_{(\tau)})^{-1}\Big(\sum_{t=1}^{\tau} k_{(\tau)}(\boldsymbol{X}_{(\tau)}, \boldsymbol{X}_{(t)})\boldsymbol{J}^T\boldsymbol{Y}_{(t)}\boldsymbol{\alpha}_{(t)}\Big)\Bigg) \\
&= 0.5\left(Const._{\beta_{(\tau)}} - \boldsymbol{\alpha}_{(\tau)}^T\boldsymbol{Y}_{(\tau)}\boldsymbol{J}k_{(\tau)}(\boldsymbol{X}_{(\tau)}, \boldsymbol{X}_{(\tau)})\boldsymbol{J}^T\boldsymbol{Y}_{(\tau)}\boldsymbol{\alpha}_{(\tau)}\right) \\
&- \boldsymbol{\alpha}_{(\tau)}^T\boldsymbol{Y}_{(\tau)}\boldsymbol{J}\sum_{t=1}^{\tau-1} k_{(\tau)}(\boldsymbol{X}_{(\tau)}, \boldsymbol{X}_{(t)})\boldsymbol{J}^T\boldsymbol{Y}_{(t)}\boldsymbol{\alpha}_{(t)}.
\end{aligned}
$$

□

CHAPTER 5

Maximum Entropy Discrimination with Partial Labels for Anomaly Detection

Data-driven anomaly detection methods suffer from the drawback of detecting all instances that are statistically rare, irrespective of whether the detected instances have real-world significance or not. In this paper, we are interested in the problem of specifically detecting anomalous instances that are known to have high real-world utility, while ignoring the low-utility statistically anomalous instances. To this end, we propose a novel method called Latent Laplacian Maximum Entropy Discrimination (LatLapMED) as a potential solution. This method uses the EM algorithm to simultaneously incorporate the Geometric Entropy Minimization principle for identifying statistical anomalies, and the Maximum Entropy Discrimination principle to incorporate utility labels, in order to detect high-utility anomalies. We apply our method in both simulated and real datasets to demonstrate that it has superior performance over existing alternatives that independently pre-process with unsupervised anomaly detection algorithms before classifying.

5.1 Introduction

Anomaly detection is a very pervasive problem applicable to a variety of domains including network intrusion, fraud detection, and system failures. It is a crucial task in many applications because failure to detect anomalous activity could result in highly undesirable outcomes. For example, (i) detection of anomalous medical claims is important to identify fraud; (ii) detection of fraudulent credit card transactions is necessary to help prevent identity theft; and (iii) detection of abnormal network traffic is necessary to identify hacking.

Many techniques have been developed for anomaly detection. These methods can be broadly classified into two categories: (i) rule-based systems, and (ii) statistical data-driven approaches. The rule-based systems are based on domain expertise and look for specific

types of anomalies while the data-driven approaches look to identify anomalies by identifying statistically rare patterns. Examples of data-driven methods include parametric methods that assume a known family for the nominal (non-anomalous) distribution and non-parametric methods such as those using unsupervised or semi-supervised support vector machines (SVMs) [74, 75] or based on minimum volume set estimation [76, 77, 78].

The advantage of data-driven approaches over rule-based methods is that they can identify novel types of anomalies that are unknown to the domain expert. In the network traffic example, they can be used to identify previously unknown types of network attacks that would not have been detected by rule-based systems. The disadvantage is that sometimes the anomalies, while statistically rare, are not interesting to the domain expert. For instance, the data-driven methods would detect routine monthly backup events due to the high volume of network traffic.

5.1.1 Related work

To identify the domain expert's interests, one could simply have the user label instances as high or low utility through active learning frameworks like the algorithms in [79, 80], and subsequently use popular supervised or semi-supervised classification methods [81, 1, 82, 83] to discriminate between the high-utility and low-utility instances. The drawback with this approach in contrast to our proposed approach is that these methods do not exploit the following key idea: only statistically rare points can be of high- utility, or equivalently, all nominal points are low-utility. As a result, the existing methods are less successful in detecting high utility instances given the limited number of labeled instances.

To incorporate this idea that nominal points are low-utility, one could pre-identify anomalous/nominal points using a statistical anomaly detection method [74, 75, 76, 77, 78], and subsequently use the instances labeled as nominal by the anomaly detection method as additional nominal labels for the classifier [84]. However, as we demonstrate in our experimental results, this strategy is not optimal because the detected anomalies are independent of the utility labels that are available. In contrast, our algorithm holistically utilizes the labeled information to accurately detect anomalies, and the detected anomalies to improve utility classification.

A similar approach to ours was taken in [85] where the authors also distinguish between high utility anomalies and low utility statistical outliers by incorporating human expert utility labels (which they acquire using an active learning loop). Their algorithm is set up to ensure that the anomaly scores of all labeled anomalies (high-utility) is higher than a threshold, and the scores of all labeled nominals (low-utility) is lower than that threshold.

Another related approach is the Bayesian posterior probability model of [86]. Their algorithm makes similar assumptions about anomalous points being far away, in distance, from the nominal points.

We construct our model using the Maximum Entropy Discrimination (MED) [54] framework, a variant of the classical minimum relative entropy principle, but with a discriminant function in some of the constraints. By choosing different priors, discriminant functions, or constraints, the MED framework can be used for corrupt measurements [87], infinite mixture classifiers [88], and Markov networks [89] among other applications. In our case, we choose to add constraints that are hinge loss style discriminant functions with latent variables and a regularizer on the smoothness of the discriminant function.

5.1.2 Proposed Work

In this paper, we develop a novel method called Latent Laplacian Maximum Entropy Discrimination (LatLapMED) which detects high-utility anomalies that are of interest to the domain expert by exploiting the idea that all high-utility points are statistically rare. We are interested in situations where we have data $\boldsymbol{X}$ of sample size n, but their labels y_i, which denote high utility $(y_i = 1)$ or not $(y_i = -1)$ are only partially observed. Some of the samples $\boldsymbol{X}_i$ are also anomalous with *latent* variables indicating whether they are $(\eta_i = 1)$ or are not $(\eta_i = 0)$. Without loss of generality, we assume the labels are observed for the first $l << n$ points and that the first a points are anomalous (all labeled points are anomalous so $l \leq a << n$).

By adding constraints to the MED framework to incorporate partially labeled observations, the subsequent decision boundary will be able to separate the high-utility anomalous points from the other points despite this incomplete information. However the nominal distribution is unknown, so one way to identify anomalies is by using the Geometric Entropy Minimization (GEM) principle [76, 78]. This idea of integrating the GEM principle into the MED framework has been previously studied by [87], who look at classifying nominal points in a fully supervised setting. In our algorithm, we use exploit the probabilistic nature of the MED framework and solve it with the EM algorithm so that the E-step estimates the latent variables with GEM and the M-Step maximizes over only the anomalous points.

5.1.2.1 Notation

The dataset is of size n where a sample $\boldsymbol{X}_i \in \mathbb{R}^p$. For notational simplicity, we assume the first l samples are labeled as high utility $(y_i = 1)$ or not $(y_i = -1)$ and the first a points are anomalous with indicator variables $(\eta_i = 1)$ and the rest are not $(\eta_i = 0)$. We

denote KL$(\cdot||\cdot)$ to be the Kullback-Leibler divergence, P$(\cdot)$ and P$_0(\cdot)$ to be a probability density and prior respectively, E$(\cdot)$ as the expectation of random variables with respect to their distribution, $\mathcal{I}(\cdot)$ to be an indicator function, $M(\cdot|\cdot)$ to be a discrimination function, $Z(\cdot)$ to be the partition function or normalizing constant, and $||\cdot||_2$ and $||\cdot||_F$ to be the ℓ_2 and Frobenius norm respectively. The following are parameters for: the decision boundary $\boldsymbol{\Theta} = \{\boldsymbol{\theta}, b\}$, the margin of each labeled sample γ_i, and the smoothness of the discrimination function λ. Their corresponding Lagrange multipliers are α_i and β. We define the following matrices: $\boldsymbol{I}$ as the identity, $\mathbf{0}$ as a zero vector, $\mathcal{L}$ as the normalized graph Laplacian matrix, $\boldsymbol{K}$ as the Gram matrix of a kernel function $k(\cdot, \cdot)$, $\boldsymbol{Y}$ as a diagonal matrix of the labels, $\boldsymbol{J}$ as a 0-1 expansion matrix, and $\boldsymbol{H}$ as a diagonal matrix of the anomaly indicators with $\boldsymbol{h}$ as only its non-zero rows. Anything with a "hat" $\hat{}$ is an estimator of its true value which has the same symbol, but no "hat".

The rest of this paper is organized as follows: Section 2 will briefly review the MED framework and discuss constructing maximum margin classifiers with it. Section 3 will propose an additional constraint to incorporate unlabeled points and derive a probabilistic interpretation of the Laplacian SVM. Section 4 will describe the proposed Latent Laplacian MED method, which uses the EM algorithm to simultaneously estimate unobserved anomalous labels and form a utility decision boundary. Section 5 contains simulation results of the performance of our proposed method, an application to a dataset of Reddit subforums, and two applications to datasets of botnet traffic (CTU-13).

5.2 Maximum Entropy Discrimination

Maximum entropy is a classical method of estimating an unknown distribution subject to the expected values of a set of constraints where the expectation is with respect to the unknown distribution. When the prior distribution is not uniform, this can be generalized as minimizing the relative entropy (or Kullback-Leibler divergence). The MED framework [54] extends the minimum relative entropy principle to have discriminant power by requiring one of the constraints to be over a parametric family of decision boundaries $M(\boldsymbol{X}|\boldsymbol{\Theta})$. Thus, it creates models that have both the classification robustness of discriminative approaches and the ability to deal with uncertain or incomplete observations of generative approaches.

The basic MED objective function is

$$\min_{\mathrm{P}(\boldsymbol{\Theta},\boldsymbol{\gamma}|\boldsymbol{X},\boldsymbol{y})} \mathrm{KL}\left(\mathrm{P}(\boldsymbol{\Theta},\boldsymbol{\gamma}|\boldsymbol{X},\boldsymbol{y})||\mathrm{P}_0(\boldsymbol{\Theta},\boldsymbol{\gamma})\right)$$

$$\text{subject to}$$

$$\iint \mathrm{P}(\boldsymbol{\Theta},\boldsymbol{\gamma})\left(y_1 M(\boldsymbol{X}_1|\boldsymbol{\Theta}) - \gamma_1\right) d\boldsymbol{\Theta} d\boldsymbol{\gamma} \geq 0$$

$$\vdots$$

$$\iint \mathrm{P}(\boldsymbol{\Theta},\boldsymbol{\gamma})\left(y_n M(\boldsymbol{X}_n|\boldsymbol{\Theta}) - \gamma_n\right) d\boldsymbol{\Theta} d\boldsymbol{\gamma} \geq 0$$

which has solution,

$$\mathrm{P}(\boldsymbol{\Theta},\boldsymbol{\gamma}|\boldsymbol{X},\boldsymbol{y}) = \frac{\mathrm{P}_0(\boldsymbol{\Theta},\boldsymbol{\gamma})}{Z(\boldsymbol{\alpha})} \exp\left\{\sum_{i=1}^{n} \alpha_i \left(y_i M(\boldsymbol{X}_i|\boldsymbol{\Theta}) - \gamma_i\right)\right\}$$

where the rows $\boldsymbol{X}_i \in \mathbb{R}^p$ are samples, $y_i \in \{-1, 1\}$ are labels, $\mathrm{P}_0(\boldsymbol{\Theta},\boldsymbol{\gamma})$ is the joint prior, and $\boldsymbol{\alpha} = [\alpha_1, ..., \alpha_n]^T \geq 0$ are Lagrange multipliers, which can be found by maximizing the negative log partition function $-\log\left(Z(\boldsymbol{\alpha})\right)$. Because the posterior distribution $\mathrm{P}(\boldsymbol{\Theta},\boldsymbol{\gamma}|\boldsymbol{X},\boldsymbol{y})$ is over the decision and margin parameters $\boldsymbol{\Theta}$ and $\boldsymbol{\gamma}$, the MED framework gives a distribution of solutions. This gives additional flexibility because the decision rule $\hat{y}_{i'} = \text{sign}(\iint \mathrm{P}(\boldsymbol{\Theta},\boldsymbol{\gamma}|\boldsymbol{X},\boldsymbol{y}) M(\boldsymbol{X}_{i'}|\boldsymbol{\Theta}) d\boldsymbol{\gamma} d\boldsymbol{\Theta})$ is a weighted combination of discriminant functions, and different priors on $\boldsymbol{\gamma}$ can permit different degrees of separability in the classification. If the support of this prior includes negative values, the decision boundary can be found on non-separable data.

5.2.1 Interpretation as a Maximum Margin Classifier

Specifically in the case when the discriminant function $M(\boldsymbol{X}|\boldsymbol{\theta}, b) = \boldsymbol{X}\boldsymbol{\theta} + b$ is linear, and the prior distribution is $\mathrm{P}_0(\boldsymbol{\Theta},\boldsymbol{\gamma}) = \mathrm{P}_0(\boldsymbol{\theta})\mathrm{P}_0(b)\prod_{i=1}^{n}\mathrm{P}_0(\gamma_i)$ where
$\mathrm{P}_0(\gamma_i) = Ce^{-C(1-\gamma_i)}\mathcal{I}(\gamma_i \leq 1)$, $\mathrm{P}_0(\boldsymbol{\theta})$ is $N(\boldsymbol{0}, \boldsymbol{I})$, and $\mathrm{P}_0(b)$ is a Gaussian non-informative prior, [54] shows that the MED solution is very similar to a support vector machine (SVM). The *maximum a posteriori* (MAP) estimator for $\boldsymbol{\theta}$ is $\sum_{i=1}^{n} \alpha_i y_i \boldsymbol{X}_i^T$ where α_i maximize

$$-\log\left(Z(\boldsymbol{\alpha})\right) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{i'=1}^{n} \alpha_i \alpha_{i'} y_i y_{i'} \boldsymbol{X}_i \boldsymbol{X}_{i'}^T + \sum_{i=1}^{n}\left(\alpha_i + \log(1 - \frac{\alpha_i}{C})\right)$$

$$\text{subject to } \sum_{i=1}^{n} y_i \alpha_i = 0 \text{ and } \alpha_i \geq 0 \text{ for all } i$$

which has a log barrier term $\log(1 - \alpha_i/C)$ instead of the inequality constraints $\alpha_i \leq C$ found in the dual form of an SVM. Otherwise the two objective functions are equivalent, so the $\hat{\alpha}_i$ are roughly the optimal support vectors and would only differ from actual support vectors when the posterior mode lies near the boundary of its support.

The connection between SVMs and Gaussian process classification has been previously studied in many works including [63, 64, 65, 90, 67]. The model in [67] is also a probabilistic interpretation of an SVM and also uses a MAP estimator with a Gaussian process prior. However, the MED framework is more generalizable and intuitive because we can easily tailor the posterior to have specific properties by narrowing the feasible set of posteriors through additional goodness-of-fit constraints expressed as statistical expectations of fitting errors. In the following sections we will show how the probabilistic interpretation of an SVM can incorporate partially labeled points and latent variables through additional constraints.

5.3 MED with Partially Labeled Observations

In order to incorporate unlabeled points, we use the semi-supervised framework of [1], which requires the decision boundary to be smooth with respect to the marginal distribution of all the data, $\mathcal{P}_X$. This is because we assume that unlabeled points have the same label as their labeled neighbors and prefer decision boundaries in low density regions. So we can restrict the choice of posteriors to be one that induces a decision boundary with at least a certain level of expected smoothness by the additional constraint

$$\iint \mathrm{P}(\boldsymbol{\theta}, \lambda) \left(\int_{x \in \mathcal{M}} ||\nabla_{\mathcal{M}} M(\boldsymbol{X}|\boldsymbol{\theta})||_2^2 \, d\mathcal{P}_X - \lambda \right) d\boldsymbol{\theta} d\lambda \leq 0$$

where $\mathcal{M} = \text{supp}(\mathcal{P}_X) \subset \mathbb{R}^n$ is a compact submanifold, $\nabla_{\mathcal{M}}$ is the gradient along it, and λ controls the complexity of the decision boundary in the intrinsic geometry of $\mathcal{P}_X$. Note the bias/intercept term b does not appear in the constraint.

Since the marginal distribution of the data is unknown, we must approximate the constraint. From [70],

$$M(\boldsymbol{X}|\boldsymbol{\theta})^T \mathcal{L} M(\boldsymbol{X}|\boldsymbol{\theta}) \to \int_{x \in \mathcal{M}} ||\nabla_{\mathcal{M}} M(\boldsymbol{X}|\boldsymbol{\theta})||_2^2 \, d\mathcal{P}_X$$

where $\mathcal{L}$ is the normalized graph Laplacian formed with a heat kernel using all the data.

Thus, we define the empirical objective function for this semi-supervised problem as

$$\min_{\mathrm{P}(\boldsymbol{\theta},b,\boldsymbol{\gamma},\lambda|\boldsymbol{X},\boldsymbol{y})} \mathrm{KL}\left(\mathrm{P}(\boldsymbol{\theta},b,\boldsymbol{\gamma},\lambda|\boldsymbol{X},\boldsymbol{y})||\mathrm{P}_0(\boldsymbol{\theta},b,\boldsymbol{\gamma},\lambda)\right)$$

$$\text{subject to}$$

$$\iiint \mathrm{P}(\boldsymbol{\theta},b,\boldsymbol{\gamma})\left(y_i M(\boldsymbol{X}_1|\boldsymbol{\theta},b)-\gamma_1\right)d\boldsymbol{\theta}dbd\boldsymbol{\gamma}\geq 0$$

$$\vdots$$

$$\iiint \mathrm{P}(\boldsymbol{\theta},b,\boldsymbol{\gamma})\left(y_l M(\boldsymbol{X}_l|\boldsymbol{\theta},b)-\gamma_l\right)d\boldsymbol{\theta}dbd\boldsymbol{\gamma}\geq 0$$

$$\iint \mathrm{P}(\boldsymbol{\theta},\lambda)(M(\boldsymbol{X}|\boldsymbol{\theta})^T\mathcal{L}M(\boldsymbol{X}|\boldsymbol{\theta})-\lambda)\,d\boldsymbol{\theta}d\lambda\leq 0$$

which has solution, $\mathrm{P}(\boldsymbol{\theta},b,\boldsymbol{\gamma},\lambda|\boldsymbol{X},\boldsymbol{y})=$

$$\frac{\mathrm{P}_0(\boldsymbol{\theta},b,\boldsymbol{\gamma},\lambda)}{Z(\boldsymbol{\alpha},\beta)}\exp\left\{\sum_{i=1}^{l}\alpha_i\left(y_i M(\boldsymbol{X}_i|\boldsymbol{\theta},b)-\gamma_i\right)+\beta\left(\lambda-M(\boldsymbol{X}|\boldsymbol{\theta})^T\mathcal{L}M(\boldsymbol{X}|\boldsymbol{\theta})\right)\right\}$$

where the α_i's are Lagrange multipliers for the mean goodness-of-fit constraint on $\mathcal{M}(\boldsymbol{X}|\boldsymbol{\theta})$ and $\beta\geq 0$ is the Lagrange multiplier for the smoothness constraint on $M(\boldsymbol{X}|\boldsymbol{\theta})$.

5.3.1 Laplacian MED as a Maximum Margin Classifier

When one uses a linear discriminant function, the same independent priors as in Section 5.2.1, but with additional exponential non-informative prior $\mathrm{P}_0(\lambda)$, the MAP estimator is thus a maximum margin classifier. This estimator is defined as $\hat{\boldsymbol{\theta}}=\sum_{i=1}^{l}(\boldsymbol{I}+2\beta\boldsymbol{X}^T\boldsymbol{L}\boldsymbol{X})^{-1}\boldsymbol{X}_i^T y_i\alpha_i$ where the Lagrange multipliers $\boldsymbol{\alpha},\beta$ maximize the negative log partition function $-\log\left(Z(\boldsymbol{\alpha},\beta)\right)=$

$$\begin{aligned}&-\frac{1}{2}\sum_{i=1}^{l}\sum_{i'=1}^{l}\alpha_i\alpha_{i'}y_i y_{i'}\boldsymbol{X}_i(\boldsymbol{I}+2\beta\boldsymbol{X}^T\boldsymbol{L}\boldsymbol{X})^{-1}\boldsymbol{X}_{i'}^T \\ &+\sum_{i=1}^{l}\left(\alpha_i+\log(1-\alpha_i/C)\right)+\log\left(\det(\boldsymbol{I}+2\beta\boldsymbol{X}^T\mathcal{L}\boldsymbol{X})\right)\end{aligned} \tag{5.1}$$

subject to $\sum_{i=1}^{l}y_i\alpha_i=0,\alpha_1,\ldots,\alpha_l\geq 0$, and $\beta\geq 0$.

Since the smoothness constraint is formulated using the semi-supervised framework of [1], the above objective function is very similar to their proposed Laplacian SVM (LapSVM). This is more obviously seen by extending (5.1) to nonlinear discriminant func-

tions though a kernel function $k(\cdot,\cdot)$ and treating β as a fixed parameter to be chosen separately.

Proposition 3. *Let* $M(\boldsymbol{X}|\boldsymbol{\theta},b)=\boldsymbol{X}\boldsymbol{\theta}+b$ *and* $P_0(\boldsymbol{\theta},b,\boldsymbol{\gamma},\lambda)=P_0(\boldsymbol{\theta})P_0(b)\prod_{i=1}^{l}P_0(\gamma_i)P_0(\lambda)$ *where* $P_0(\gamma_i)=Ce^{-C(1-\gamma_i)}\mathcal{I}(\gamma_i\leq 1)$*,* $P_0(\boldsymbol{\theta})$ *is* $N(\mathbf{0},\boldsymbol{I})$*, and* $P_0(\lambda)$ *and* $P_0(b)$ *approach exponential and Gaussian non-informative priors. Then for a given parameter* $\beta\geq 0$*, the dual problem to maximizing the posterior* $P(\boldsymbol{\theta},b,\boldsymbol{\gamma},\lambda|\boldsymbol{X},\boldsymbol{y})$ *for* $\boldsymbol{\theta}$ *is*

$$\arg\max_{\boldsymbol{\alpha}}\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{Y}\boldsymbol{J}\boldsymbol{K}(\boldsymbol{I}+2\beta\mathcal{L}\boldsymbol{K})^{-1}\boldsymbol{J}^T\boldsymbol{Y}\boldsymbol{\alpha}+\sum_{i=1}^{l}\log(1-\alpha_i/C)$$

$$\textit{subject to}\ \sum_{i=1}^{l}y_i\alpha_i=0,\ \ \alpha_1,\ldots,\alpha_l\geq 0$$

where $\boldsymbol{K}$ *is the Gram matrix of the kernel function,* $\boldsymbol{Y}=\text{diag}(y_1,...,y_l)$ *and* $\boldsymbol{J}=[\boldsymbol{I}\ \mathbf{0}]$ *is a* $l\times n$ *expansion matrix. The decision rule in this dual form is*
$\hat{y}_{i'}=\text{sign}\big(k(\boldsymbol{X}_{i'},\boldsymbol{X})(\boldsymbol{I}+2\beta\mathcal{L}\boldsymbol{K})^{-1}\boldsymbol{J}^T\boldsymbol{Y}\hat{\boldsymbol{\alpha}}+\hat{b}\big)$ *where*
$\hat{b}=\arg\min_{b}\sum_{s\in\{i|\hat{\alpha}_i\neq 0\}}|(y_s-\hat{y}_s)-b|$ *is equivalent to an SVM bias term [91].*

Again the log barrier term produces a relaxation of the inequality constraints $\alpha_i\leq C$ and decreases the objective function if the optimum is near the boundary of the support. The parameters can be written as $C=\frac{1}{2l\gamma_A}$ and $\beta=\frac{\gamma_I}{2\gamma_A n^2}$ so that they are functions of γ_A and γ_I, the penalty parameters in the LapSVM for the norms associated with the reproducing kernel Hilbert space (RKHS) and data distribution $\mathcal{P}_X$ respectively. Due to these similarities, we will call the classifier of Proposition 3 the Laplacian MED (LapMED).

5.4 MED with Latent Variables

Now that we have established a method to incorporate unlabeled points in MED, we will present a method to also incorporate latent variables. This joint method of simultaneously incorporating unlabeled points and latent variables is our proposed Latent Laplacian MED (LatLapMED) method. We will first consider the case where we can observe the latent variables, so that we have a complete posterior distribution. Then we will derive a lower bound for the observed posterior distribution and discuss how to deal with estimating the latent variables when they are not observed. This will allow us to apply the EM algorithm, which alternates between estimating the latent variables and maximizing the lower bound.

5.4.1 The Complete Posterior

If we observe the anomaly indicator variables η_i, then we can construct a posterior that depends on these variables by modifying the constraints on mean goodness-of-fit. The discriminant function $M(\boldsymbol{X}_i, \eta_i|\boldsymbol{\theta}, b) = \eta_i(\boldsymbol{X}_i\boldsymbol{\theta} + b)$ can be used to create a maximum margin classifier that gives positive or negative values for anomalous points and zeros for nominal points. This is reasonable because all labeled points are anomalous, so if they are mistakenly classified as nominal, the loss function embedded in the constraints

$$\begin{aligned} \iiint \mathrm{P}(\boldsymbol{\theta}, b, \boldsymbol{\gamma})\,(y_i\eta_1(\boldsymbol{X}_i\boldsymbol{\theta} + b) - \gamma_1)\,d\boldsymbol{\theta}dbd\boldsymbol{\gamma} \geq 0 \\ \vdots \\ \iiint \mathrm{P}(\boldsymbol{\theta}, b, \boldsymbol{\gamma})\,(y_l\eta_l(\boldsymbol{X}_l\boldsymbol{\theta} + b) - \gamma_l)\,d\boldsymbol{\theta}dbd\boldsymbol{\gamma} \geq 0 \end{aligned} \tag{5.2}$$

will penalize the labeled points as if they were inside the margin.

Additionally if some of the anomalous points are not labeled, then we will use the same semi-supervised framework as before and add a smoothness constraint. Since the discriminant function $M(\boldsymbol{X}_i, \eta_i|\boldsymbol{\theta}, b)$ will always give zeros for nominal points, it really only needs to be smooth with respect to the marginal distribution of the anomalies $\mathcal{P}_{X_\eta}$. Thus because $\int_{x\in\mathcal{M}} ||\nabla_{\mathcal{M}} M(\boldsymbol{X}|\boldsymbol{\theta})||_2^2\, d\mathcal{P}_X = \int_{x\in\mathcal{M}_{\boldsymbol{\eta}}} ||\nabla_{\mathcal{M}_\eta} M(\boldsymbol{X}|\boldsymbol{\theta})||_2^2\, d\mathcal{P}_{X_\eta}$, there are two choices for the empirical smoothness constraint that converge to the same limit,

$$\iint \mathrm{P}(\boldsymbol{\theta}, \lambda)(\boldsymbol{\theta}^T\boldsymbol{X}^T\boldsymbol{H}^T\mathcal{L}\boldsymbol{H}\boldsymbol{X}\boldsymbol{\theta} - \lambda)\,d\boldsymbol{\theta}d\lambda \leq 0 \tag{5.3}$$

$$\iint \mathrm{P}(\boldsymbol{\theta}, \lambda)(\boldsymbol{\theta}^T\boldsymbol{X}^T\boldsymbol{h}^T\mathcal{L}_\eta\boldsymbol{h}\boldsymbol{X}\boldsymbol{\theta} - \lambda)\,d\boldsymbol{\theta}d\lambda \leq 0 \tag{5.4}$$

where $\mathcal{L}_\eta$ is the normalized graph Laplacian of the anomalous points, $\boldsymbol{H} = \mathrm{diag}(\boldsymbol{\eta})$, and $\boldsymbol{h}$ is a $a \times n$ submatrix of only the non-zero rows of $\boldsymbol{H}$.

The solution to the MED problem, using constraints (5.2) and either (5.3) or (5.4), is a posterior distribution $\mathrm{P}(\boldsymbol{\theta}, b, \boldsymbol{\gamma}, \lambda|\boldsymbol{X}, \boldsymbol{\eta}, \boldsymbol{y})$ and its MAP estimator can also be a maximum margin classifier when the priors are the ones in Proposition 3. If possible, it is more ideal to use constraint (5.4) because the maximum margin classifier forms a decision boundary with just the a anomalous points; so it takes considerable less computation time than the equivalent classifier using constraint (5.3).

Lemma 5.4.1. *Using the same priors as in Proposition 3, but with discriminant function* $M(\boldsymbol{X}_i, \eta_i|\boldsymbol{\theta}, b) = \eta_i(\boldsymbol{X}_i\boldsymbol{\theta} + b)$*, the dual problem to maximizing the posterior of the MED problem with constraints* (5.2) *and* (5.4) *is maximizing*

$$\arg\max_{\boldsymbol{\alpha}} \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\boldsymbol{\alpha}^T \boldsymbol{Y}\boldsymbol{J}\boldsymbol{K}_\eta(\boldsymbol{I} + 2\beta\mathcal{L}_\eta\boldsymbol{K}_\eta)^{-1}\boldsymbol{J}^T\boldsymbol{Y}\boldsymbol{\alpha} + \sum_{i=1}^{l}\log(1-\alpha_i/C)$$

$$\textit{subject to } \sum_{i=1}^{l}\alpha_i y_i = 0,\ \alpha_1,\ldots,\alpha_l \geq 0$$

where $\boldsymbol{K}_\eta$ is an $a \times a$ submatrix of the Kernel matrix and $\boldsymbol{J} = [\boldsymbol{I}\ \boldsymbol{0}]$ is now a $l \times a$ expansion matrix.

Now, we consider the more realistic scenario where the anomaly indicator variables are latent. Because $\mathcal{L}_\eta$ depends on both $\boldsymbol{X}$ and $\boldsymbol{\eta}$, it is simpler to use constraint (5.3) to derive a posterior. Additionally, the posterior distribution is no longer concave and thus difficult to maximize so we will derive a lower bound to maximize instead.

5.4.2 A Lower Bound

Since the anomaly indicator variables η_i are not actually observable, the posterior distribution we can observe is of the form

$$\mathrm{P}(\boldsymbol{\theta}, b, \boldsymbol{\gamma}, \lambda | \boldsymbol{X}, \boldsymbol{y}) = \frac{\mathrm{P}_0(\boldsymbol{\theta}, b, \boldsymbol{\gamma}, \lambda)\sum \mathrm{P}(\boldsymbol{X}, \boldsymbol{\eta}, \boldsymbol{y} | \boldsymbol{\theta}, b, \boldsymbol{\gamma}, \lambda)}{\sum \mathrm{P}(\boldsymbol{X}, \boldsymbol{\eta}, \boldsymbol{y} | \boldsymbol{\alpha})} \tag{5.5}$$

where the summation $\sum$ is over all $\eta_i \in \{0, 1\}$. So, we need a lower bound for the negative log expected partition function $-\log\left(\sum \mathrm{P}(\boldsymbol{X}, \boldsymbol{\eta}, \boldsymbol{y}|\boldsymbol{\alpha})\right)$ that is practical to maximize.

Lemma 5.4.2. *Let* (5.5) *be the posterior of the MED problem with constraints* (5.2) *and* (5.3)*, then using the same assumptions as Lemma 5.4.1, the dual problem to MAP estimation is maximizing $-\log\left(\sum P(\boldsymbol{X}, \boldsymbol{\eta}, \boldsymbol{y}|\boldsymbol{\alpha})\right)$ for $\boldsymbol{\alpha}$. This objective has a lower bound proportional to*

$$\sum_{i=1}^{l}\alpha_i - \frac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{Y}\boldsymbol{J}\boldsymbol{K}\left(\boldsymbol{I} + 2\beta E_\eta(\boldsymbol{H}^T\mathcal{L}\boldsymbol{H})\boldsymbol{K}\right)^{-1}\boldsymbol{J}^T\boldsymbol{Y}\boldsymbol{\alpha} + \sum_{i=1}^{l}\log(1-\alpha_i/C)$$

$$\textit{subject to } \sum_{i=1}^{l}\alpha_i y_i = 0,\ \alpha_1,\ldots,\alpha_l \geq 0$$

where E_η is the expectation with respect to $P(\boldsymbol{\eta}|\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\alpha}^{t-1})$ and $\boldsymbol{\alpha}^{t-1}$ are the optimal Lagrange multipliers of the previous iteration.

With this lower bound, we have an objective to maximize in the M-step of the EM algorithm. In the following subsection, we give a way to estimate $\mathrm{E}(\boldsymbol{H}^T\mathcal{L}\boldsymbol{H}|\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\alpha}^{t-1}) = \mathcal{L} \odot \mathrm{E}(\boldsymbol{\eta}\boldsymbol{\eta}^T|\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\alpha}^{t-1})$ for the E-step.

5.4.3 Estimating the Latent Variables

Since $\eta_i = 1$ when the data point $\boldsymbol{X}_i$ does not come from the nominal distribution, we can define it as an indicator variable $\eta_i = \mathcal{I}(\boldsymbol{X}_i \notin \Omega_\phi)$ where Ω_ϕ is a minimum entropy set of level ϕ. So η_i can be viewed as the test function for a statistical test of whether the density of $\boldsymbol{X}_i$ is equal to the density of the nominal points or not, and Ω_ϕ is the optimal acceptance region of the test. However because the nominal distribution of $\boldsymbol{X}$ is unknown, the GEM principle [76, 78] estimates the optimal acceptance region using the property that if $\lim_{K,N\to\infty} \frac{K}{N} = \phi$, a greedy K point k nearest neighbors graph (K-kNNG) converges almost surely to the minimum υ-entropy set containing at least $(1-\phi)\%$ of the mass. Thus, for any ij element of the matrix $\mathrm{E}(\boldsymbol{\eta}\boldsymbol{\eta}^T|\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\alpha}^{t-1})$, we have

$$\mathrm{E}(\eta_i\eta_j|\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\alpha}^{t-1}) = \mathrm{E}\left(\mathcal{I}(\boldsymbol{X}_i, \boldsymbol{X}_j \notin \Omega_\phi)|\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\alpha}^{t-1}\right) \approx \mathcal{I}(\boldsymbol{X}_i, \boldsymbol{X}_j \notin \hat{\Omega}_\phi) = \hat{\eta}_i\hat{\eta}_j$$

where $\hat{\Omega}_\phi$ is the estimated acceptance region.

However, if $\hat{\Omega}_\phi$ uses the standard K-kNNG with edge lengths equal to Euclidean distances, the graph does not incorporate label information or how the points lie relative to the decision boundary. Since the neighbors of an anomalous point are also most likely anomalous, we instead use a similarity metric that penalizes a point for having anomalous neighbors. So the edge length between a point i and its neighbor j is

$$|e_{i(j)}| = \begin{cases} ||\boldsymbol{X}_i - \boldsymbol{X}_j||_2 + \hat{d}_j^{t-1} & \text{if } \hat{d}_j^{t-1} > \rho \text{ or } y_j = 1 \\ ||\boldsymbol{X}_i - \boldsymbol{X}_j||_2 & \text{otherwise} \end{cases} \tag{5.6}$$

where $\hat{d}_j^{t-1}$ is the signed perpendicular distance between $\boldsymbol{X}_j$ and the decision boundary, $\rho \geq 0$ is some threshold, and y_j is the label of $\boldsymbol{X}_j$. Using a graph with the above edges in the GEM principle, we can estimate the optimal acceptance region, given a decision boundary and labels, by

$$\hat{\Omega}_\phi = \underset{\mathcal{X}_{N,K}\subset\mathcal{X}_N}{\arg\min} \sum_{i=1}^{K}\sum_{j=1}^{k} |e_{i(j)}| \tag{5.7}$$

where $\mathcal{X}_{N,K}$ is a size K subset of the set of all points $\mathcal{X}_N$ and $\{e_{i(1)}, ..., e_{i(k)}\}$ are the edges between point i and its k neighbors.

So using the GEM principle described above, $\mathcal{L} \odot \hat{\boldsymbol{\eta}}\hat{\boldsymbol{\eta}}^T = \hat{\boldsymbol{H}}^T\mathcal{L}\hat{\boldsymbol{H}}$ is an estimator for $\mathrm{E}(\boldsymbol{H}^T\mathcal{L}\boldsymbol{H}|\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\alpha}^{t-1})$. However, if the MED problem uses constraint (5.4), the E-step would need an estimator for $\mathrm{E}(\boldsymbol{h}^T\mathcal{L}_\eta\boldsymbol{h}|\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\alpha}^{t-1})$ instead.

Lemma 5.4.3. *Assume that $\mathcal{L} \odot \hat{\boldsymbol{\eta}}\hat{\boldsymbol{\eta}}^T = \hat{\boldsymbol{H}}^T \mathcal{L} \hat{\boldsymbol{H}}$ is a good estimator for $E(\boldsymbol{H}^T \mathcal{L} \boldsymbol{H} | \boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\alpha}^{t-1})$ and that the first m neighbors of any anomalous points are also anomalous. Then $\hat{\boldsymbol{h}}^T \hat{\mathcal{L}}_\eta \hat{\boldsymbol{h}}$ is a good estimator for $E(\boldsymbol{h}^T \mathcal{L}_\eta \boldsymbol{h} | \boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\alpha}^{t-1})$ where $\hat{\boldsymbol{h}}$ is the $a \times n$ submatrix of the nonzero rows of $\hat{\boldsymbol{H}}$ and $\hat{\mathcal{L}}_\eta$ is the Laplacian matrix on only the set of data points $\{\boldsymbol{X}_i : \hat{\eta}_i = 1\}$.*

5.4.4 Maximum Margin Classification with the EM Algorithm

From the previous three subsections, it is obvious that the EM algorithm for MAP estimation of the unobserved posterior distribution $\mathrm{P}(\boldsymbol{\theta}, b, \boldsymbol{\gamma}, \lambda | \boldsymbol{X}, \boldsymbol{y})$ is also a maximum margin classifier, which we call Latent Laplacian MED (LatLapMED).

Theorem 5.4.4. *Under Lemmas 5.4.1 and 5.4.3, the E-step of the EM algorithm is just getting estimators $\hat{\eta}_i = \mathcal{I}(\boldsymbol{X}_i \notin \hat{\Omega}_\phi)$ for the function of unknown parameters $E(\eta_i | \boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\alpha}^{t-1})$. And, the M-step for maximizing $P(\boldsymbol{\theta}, b, \boldsymbol{\gamma}, \lambda | \boldsymbol{X}, \boldsymbol{y})$ is a maximum margin classifier of the form,*

$$\arg\max_{\boldsymbol{\alpha}} \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \boldsymbol{\alpha}^T \boldsymbol{Y} \boldsymbol{J} \hat{\boldsymbol{K}}_\eta (\boldsymbol{I} + 2\beta \hat{\mathcal{L}}_\eta \hat{\boldsymbol{K}}_\eta)^{-1} \boldsymbol{J}^T \boldsymbol{Y} \boldsymbol{\alpha} + \sum_{i=1}^{l} \log(1 - \frac{\alpha_i}{C})$$

$$\textit{subject to } \sum_{i=1}^{l} \alpha_i y_i = 0, \ \alpha_1, \ldots, \alpha_l \geq 0$$

$$\arg\min_{b} \sum_{s \in \{i | \hat{\alpha}_i \neq 0\}} |(y_s - k(\boldsymbol{X}_s, \boldsymbol{X}_{\hat{\eta}})(\boldsymbol{I} + 2\beta \hat{\mathcal{L}}_\eta \hat{\boldsymbol{K}}_\eta)^{-1} \boldsymbol{J}^T \boldsymbol{Y} \hat{\boldsymbol{\alpha}}) - b|$$

where $\hat{\Omega}_\phi$ is approximated with the GEM principle described in subsection 5.4.3 and $\hat{\boldsymbol{K}}_\eta$ is $a \times a$ submatrix of only $\{\boldsymbol{X}_i : \hat{\eta}_i = 1\}$.

LatLapMED exploits the idea that all high utility points are anomalous because the similarity metric in $\hat{\Omega}_\phi$ is dependent on the decision boundary and label information; thus it will be skewed away from points with high utility neighbors. This is crucial because any high utility points incorrectly estimated as nominal will not be considered in the M-step and thus cannot be predicted as high utility. In contrast, it is not that vital to correctly estimate low utility anomalous points because it is not of interest to distinguish between them and the nominal. As the decision boundary moves every EM iteration, it changes the penalties that neighboring nodes can incur in the similarity metric. Since the normalized margin is 1, setting $\rho = 1$ is typical; however, if the data is difficult to classify, it may be appropriate to set $\rho > 1$ because there is less confidence in the classification. Thus the threshold ρ can be set empirically using prior domain knowledge of the structure of the data or by cross-validation.

Corollary 5.4.4.1. *Once the EM algorithm converges, the decision rule is*

$$\hat{y}_{i'} = \begin{cases} -1 & \text{if } \hat{\eta}_{i'} = 0 \\ \text{sign}\left(k(\boldsymbol{X}_{i'}, \boldsymbol{X}_{\hat{\eta}})(\boldsymbol{I} + 2\beta\hat{\mathcal{L}}_\eta\hat{\boldsymbol{K}}_\eta)^{-1}\boldsymbol{J}^T\boldsymbol{Y}\hat{\boldsymbol{\alpha}} + \hat{b}\right) & \text{otherwise} \end{cases}$$

where $\boldsymbol{X}_{\hat{\eta}}$ are the data points estimated to be anomalous and $\hat{\boldsymbol{\alpha}}, \hat{b}$ are optimal parameters.

In this work, we approximate $\mathrm{E}_{\boldsymbol{\eta}}(\boldsymbol{\eta}\boldsymbol{\eta}^T|\boldsymbol{X}, \boldsymbol{\alpha}^{t-1})$ using the GEM principle with similarity metric (5.6) because the expectation distribution is unknown. Because the GEM principle is nonparametric, it does not impose, potentially incorrect, distributional assumptions on the unknown distribution of anomalies, which may be extremely difficult to parametrically characterize. Other estimators, derived using either a different similarity metric in the GEM principle or another nonparametric method altogether, could be used instead in the E-step. We believe our estimator is a good choice because it is asymptotically consistent and empirically we find it is sufficient enough such that the objective in the M-step increases every iteration. The LatLapMED algorithm, summarized below, produces a joint estimate of both anomaly and utility labels. This simultaneous estimation allows the method to incorporate additional information that would be lost when estimating the anomaly and utility labels independently.

Algorithm 5.1 LatLapMED

Input: $\phi, \rho, k, C, \beta, \boldsymbol{X}, \boldsymbol{y}$
repeat
  E-Step:
    1) Given $\hat{\boldsymbol{d}}^{t-1} = \hat{\boldsymbol{K}}_\eta(\boldsymbol{I} + 2\beta\hat{\mathcal{L}}_\eta\hat{\boldsymbol{K}}_\eta)^{-1}\boldsymbol{J}^T\boldsymbol{Y}\hat{\boldsymbol{\alpha}}^{t-1} + \hat{b}^{t-1}$
    2) $\hat{\eta}_i = \mathcal{I}(\boldsymbol{X}_i \notin \hat{\Omega}_\phi)$ where $\hat{\Omega}_\phi$ is the solution of (5.7)
  M-Step:
    1) Given $\hat{\boldsymbol{\eta}}$, form new submatrices $\hat{\boldsymbol{K}}_\eta$ and $\hat{\mathcal{L}}_\eta$
    2) Solve the objectives in Theorem 5.4.4 to get $\hat{\boldsymbol{\alpha}}^t, \hat{b}^t$
until convergence
Return: $\hat{\boldsymbol{\eta}}, \hat{\boldsymbol{\alpha}}, \hat{b}$

5.4.4.1 Computational Complexity

The E-step uses a K-kNNG for the estimators $\hat{\eta}_i$. This K-kNNG is defined by the Euclidean distance between points, which is constant over all EM iterations, and a penalty, which changes between EM iterations. The Euclidean distances are calculated and the k neighbors are sorted only once at initialization, which have computational complexity $O(n^2p+n^2\log(n))$. At every EM iteration, the E-step just needs to add the n penalties given

from the previous M-step and sort the n total edge lengths, adding computational complexity $O(n\log(n))$ per iteration. The M-step maximizes a quadratic objective (formed from $O(a^3)$ matrix operations) over the Lagrange multipliers α_i, which can be solved with conic interior point methods in polynomial a time, where $a << n$. Alternatively, the M-step objective can be approximated as a quadratic program and solved using sequential minimal optimization in linear a time. Thus, the LatLapMED algorithm has overall computation time $O(n^2p + n^2\log(n) + \#\text{iter}(n\log(n) + a^3 + a^q))$ where $1 \leq q << \infty$ depends on how the objective is solved. If we assume that the computational time of the E-step dominates significantly over the computational time of the M-step because $a << n$, then this reduces to roughly $O(n^2p + n^2\log(n) + \#\text{iter}(n\log(n)))$. We have had no problem implementing the algorithm even for n as high as 100,000 points. Parallelization of the initial sorted distances for the K-kNNG can also improve its computational speed to $O(\frac{n^2p+n^2\log(n)}{\#nodes})$.

The final LatLapMED posterior $\text{P}(\boldsymbol{\theta}, b, \boldsymbol{\gamma}, \lambda | \boldsymbol{X}, \hat{\boldsymbol{\eta}}, \boldsymbol{y})$, where $\hat{\boldsymbol{\eta}}$ is the estimated latent variables at EM convergence, is a probabilistic model with a mode that performs maximum margin classification. Thus LatLapMED has the classification robustness of discriminant methods, but the natural flexibility of generative methods to incorporate latent variables. Additionally the generative nature also provides for sequential classification by using the posterior distribution as a new prior for new data in the MED framework. This allows LatLapMED to be very applicable to real world problems where data is often continuously collected in a stream. Alternatively, it can also be used to process a very large dataset, $n >> 10^5$, in smaller batches allowing for the algorithm to be feasible for very large n.

5.4.4.2 Limitations and Future Work

While the computational complexity of the LatLapMED algorithm is feasible for moderately large datasets, it is still more computationally expensive than many competing methods. However the performance improvement may make it worthwhile to implement the proposed algorithm in challenging anomaly detection problems. Strategies for reducing computational complexity through parallelization, specialized hardware approaches, or implementation of second order acceleration methods are also possible. Additionally, the problem of online sequential anomaly detection and classification is open. One possible approach would be to make the E-step be only weakly dependent on of the prior information to make it adaptive to changes in the prior over time. An alternative solution would be to modify the K-kNNG in the GEM algorithm to incorporate a time varying prior through weighted edges or a suitable choice of level set boundary that varies with the prior. Finally the number of tuning parameters in the LatLapMED algorithm might be reduced by using hyperpriors or empirical risk minimization methods.

5.5 Experiments

In this section, we apply the LatLapMED algorithm to both simulated and real data sets and demonstrate that the proposed method outperforms alternative two-stage methods that first estimate the anomaly labels and then predict the utility labels of only the estimated anomalous points. For combination in the two-stage methods, we consider three algorithms for non-parametric anomaly detection and both popular supervised and semi-supervised algorithms for classification; these are shown in Table 5.1.

Table 5.1: Algorithms Used to Form Two-Stage Methods

Anomaly Detector		Supervised Classifier		Semi-Supervised
GEM		SVM		LapMED
1SVM	+	RF	or	LapSVM
SSAD		NN		LDS

The one class SVM (1SVM) of [74] and the standard GEM with euclidean distance K-kNNG of [76] are unsupervised, but the semi-supervised anomaly detection (SSAD) algorithm of [75] incorporates the labeled points as known anomalous points. In the three supervised methods: SVM, random forests (RF) of [81], and neural networks (NN), we train the algorithms with labeled points and predict the labels of only the anomalous unlabeled points, and in the three semi-supervised methods: the LapMED from Section 5.3.1, the LapSVM of [1], and the low density separation (LDS) algorithm of [82], we train the algorithms on all anomalous points to classify their unlabeled ones. Because these two-stage methods naively perform anomaly detection independently of classification, there is no synergy between the two stages unlike in the LatLapMED method, which binds the two actions through the EM algorithm.

For all of the following experiments, we choose the parameters of classifiers based on the methods described in their original papers. We verify that our parameter choices are acceptable because under "oracle" conditions where the anomaly labels are known, all classifiers can classify relatively equally as well, which is to be expected. All methods are implemented in MATLAB, but most of the optimization is done with an optimization package written in another language. Specifically, we use LIBSVM [92] for the SVM classifiers, CVX [93, 94] for optimization of the LapMED objective, CVX or LIBQP [95] for SSAD, the code provided in [82] for LDS, and the corresponding MATLAB toolboxes for random forest and neural networks. Thus the GEM routine in the E-step of LatLapMED is solved purely in MATLAB, but the LapMED objective in the M-step is solved with CVX. Because the high utility class is much smaller than the low utility class, we choose to use

precision and recall to measure performance due to the benefits argued in [96].

5.5.1 Simulation Results

We simulate datasets of sample size 7,000 where the variables come from a multivariate folded t-distribution with location $\boldsymbol{\mu} = \mathbf{0}$, a random positive definite scale matrix $\boldsymbol{\Sigma}$, and 30 degrees of freedom. We calculate the utility scores for each point by $score_i = \max_h \frac{1}{|\mathcal{C}_h|} \sum_{j \in \mathcal{C}_h} \boldsymbol{X}_{ij} - \frac{1}{p-|\mathcal{C}_h|} \sum_{j \notin \mathcal{C}_h} \boldsymbol{X}_{ij}$ where $\mathcal{C}_h$ is a random set of column indicies for random utility component h. Thus 5% of the data is anomalous and the top 25% of anomalies with the highest utility scores are defined as having high utility. We observed 30% of the high utility anomalies and an equal number of low utility anomalies.

In the exact simulations below, we use the parameters listed in Table 5.2. For SSAD, we allow the regularization parameter for margin importance κ to vary. For LatLapMED, we set $\rho = 1$ because we believe, in the space of only the anomalies, the data is pretty separable and easily classified. Figure 5.1a shows the "oracle" scenario, where anomaly labels are known; so the nominal points, $\eta_i = 0$, are automatically given a label $\hat{y}_i = -1$, and the semi-supervised methods (LapMED, LapSVM, LDS) train and classify on only points with $\eta_i = 1$ while the supervised methods (SVM, RF, NN) predict on the unlabeled $\eta_i = 1$ points. Under this scenario, all classifiers have relatively equal precision and recall, which indicates that our parameter choices are acceptable since each classifier has its advantages and disadvantages. Note that LapMED and LapSVM are essentially the same model as discussed theoretically in Section 5.3.1. Additionally note that when the anomaly labels are known, we have the complete posterior for LatLapMED described in Section 5.4.1, which has the same mode as the LapMED posterior given only anomalous data.

In a realistic scenario, as opposed to the "oracle" one, the anomaly labels are unknown and must be estimated. So we compare our LatLapMED method, which estimates the anomaly and utility labels simultaneously, with two-stage methods that first perform either GEM or 1-class SVM for anomaly detection and then uses one of the above classifiers to label the utility of the anomalous points. In Figure 5.1b, we show similar boxplot plots to the ones in Figure 5.1a, but in this scenario, the anomaly labels are latent. While the LatLapMED method has similar precision as the alternative two-stage methods, it has much better recall. This indicates that LatLapMED is able leverage more information from the labeled anomalous points than a naive two-stage method that treats the utility label information and anomaly status of points as independent.

Table 5.2: Parameters Used in the Algorithms

Anomaly Detector	**Parameters**
GEM	$k = 10$ neighbors in kNN graph, the K points = ϕn
1SVM	$\sigma = 1$ in rbf kernel, $\nu = \phi$
SSAD	$\sigma = 1$ in rbf kernel, κ, label $C = 1$, unlabel $C = 1/\phi$
Classifier	**Parameters**
SVM	$\sigma = 1$ in rbf kernel, cost $C = 50$
LapSVM	$\Uparrow$, $\beta = \frac{10Cl}{a^2}$, Laplacian: $k = 50$, $\tau = 100$ in heat kernel
LapMED	$\Uparrow$ (same as above in LapSVM)
LDS	$k = 50$ neigh., $\sigma = 1$ in rbf, $C = 50$, softening = 1.5
RF	50 weak learners, default params. in MATLAB toolbox
NN	50 neurons, default params. in MATLAB toolbox
Joint Method	**Parameters**
LatLapMED	$\Uparrow$ (same as above in LapMED), threshold $\rho = 1, \phi$

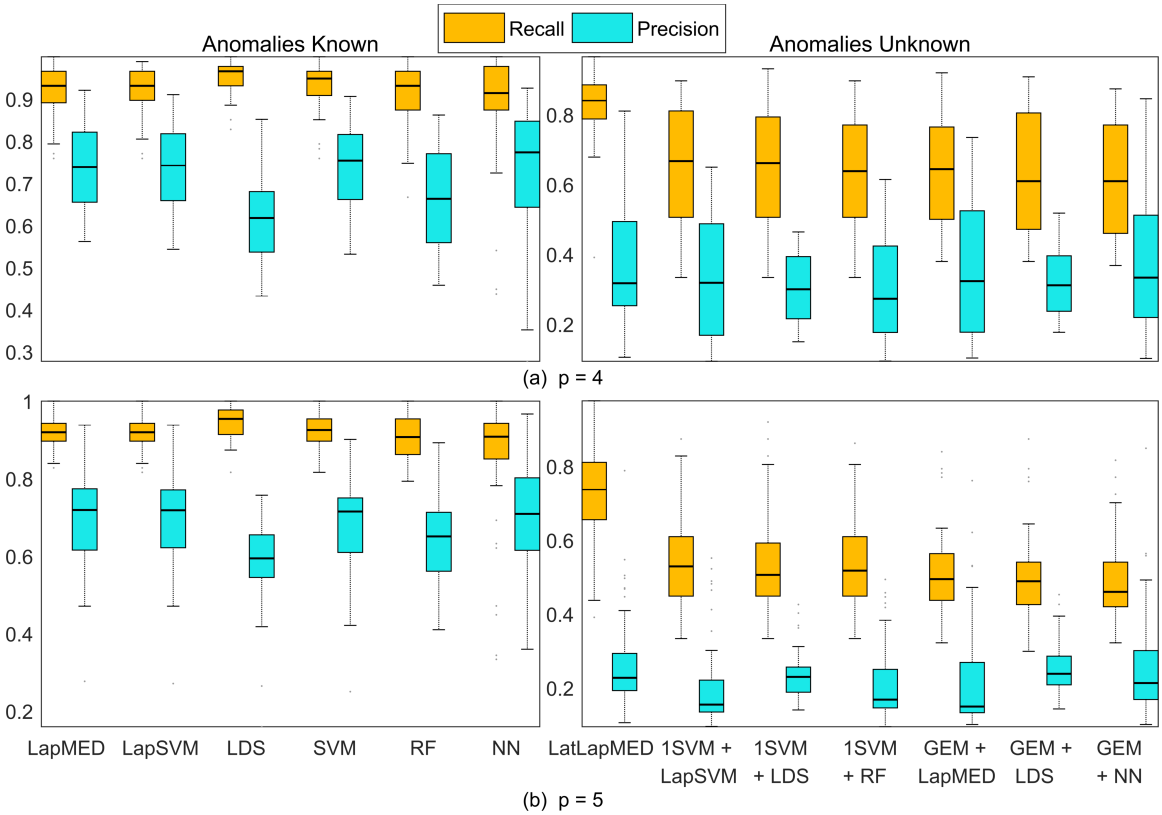
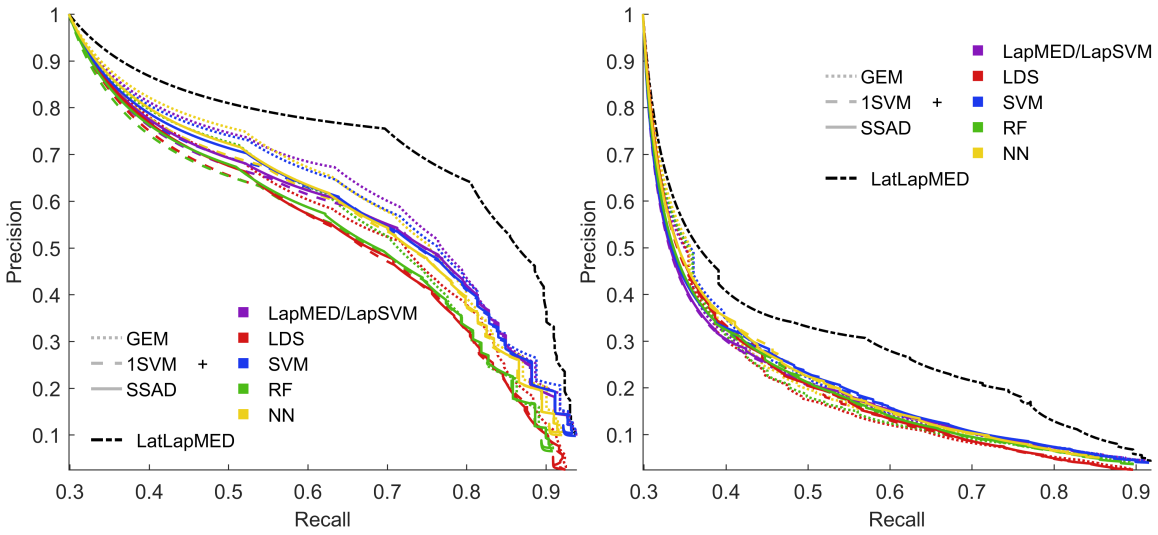


Figure 5.1: Boxplots showing 50 trials of precision and recall of different methods. a) Under the "oracle" scenario, where anomaly labels are known, all classifiers have relatively equal performance. b) The anomaly labels are unknown, but the percentage of the data that is anomalous is known to be $\phi = 0.05$. All methods have relatively equal precision, but the LatLapMED method has much better recall because it does not treats the utility and anomaly labels as independent.

Figure 5.2 compares LatLapMED against all combinations of alternative two-stage methods in $p = 3$ and $p = 6$ dimensions respectively. The Precision-Recall (PR) curves (averaged over 50 trials) show that for all levels of ϕ LatLapMED always dominates all of the naive two-stage methods. It is well known that as dimensionality increases, non-parametric estimation becomes more difficult, so the performance of all methods degrade because anomaly detection becomes more difficult. However, Table 5.3 shows that LatLapMED always has superior performance over the other methods irrespective of the dimension.



(a) $p = 3$ (b) $p = 6$

Figure 5.2: PR curves for various anomaly levels ϕ in 3 and 6 dimensions. The area under the PR curves are listed in Table 5.3. The LatLapMED method significantly outperforms all the naive two-stage methods.

Figure 5.3 gives an in-depth view of LatLapMED compared to some alternative two-stage methods. The anomaly level ϕ of the methods is set to be between 0.05 and 0.06 to control the number of false positives. The number of false negatives in LatLapMED is much lower than that of the other methods. This is because unlike the two-stage methods, if LatLapMED misses some high-utility points when estimating anomalies, it can correct for them in the next EM iteration. Figure 5.4 shows how both the number of false positives and false negatives decrease as the EM algorithm in LatLapMED iterates. In comparison to the naive two-stage GEM+LapMED, which would be equivalent to LatLapMED with only one EM iteration, LatLapMED is able to recover over 50% of the high utility points initially missed in the first EM iteration.

Table 5.3: Area Under the PR Curve (AUC-PR)

	$p = 3$	$p = 4$	$p = 5$	$p = 6$
GEM+LapMED	0.69246	0.55129	0.43539	0.42204
1SVM+LapSVM	0.66449	0.54049	0.439	0.41696
SSAD+LapSVM	0.66899	0.54406	0.44189	0.41842
GEM+LDS	0.65386	0.53299	0.44842	0.41061
1SVM+LDS	0.63228	0.52483	0.44607	0.41522
SSAD+LDS	0.63614	0.52994	0.45126	0.41557
GEM+SVM	0.68738	0.56141	0.43449	0.42702
1SVM+SVM	0.6675	0.55206	0.43766	0.42286
SSAD+SVM	0.67246	0.55739	0.44122	0.42413
GEM+RF	0.65483	0.52194	0.42837	0.4151
1SVM+RF	0.63192	0.51556	0.43516	0.41667
SSAD+RF	0.63823	0.52169	0.43913	0.41737
GEM+NN	0.68344	0.54716	0.44525	0.41907
1SVM+NN	0.66064	0.54109	0.44639	0.42216
SSAD+NN	0.66673	0.54613	0.45027	0.42216
LatLapMED	0.76253	0.66417	0.51792	0.47854

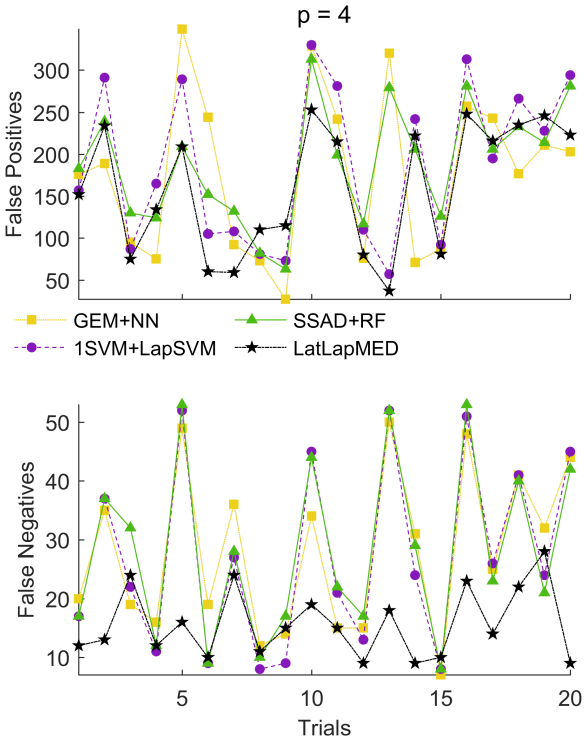


Figure 5.3: The number of false positives and false negatives in 20 different trials with $\phi \in [0.05, 0.06]$ to control the number of false positives. LatLapMED has far fewer false negatives for the same number of false positives compared to the other methods.

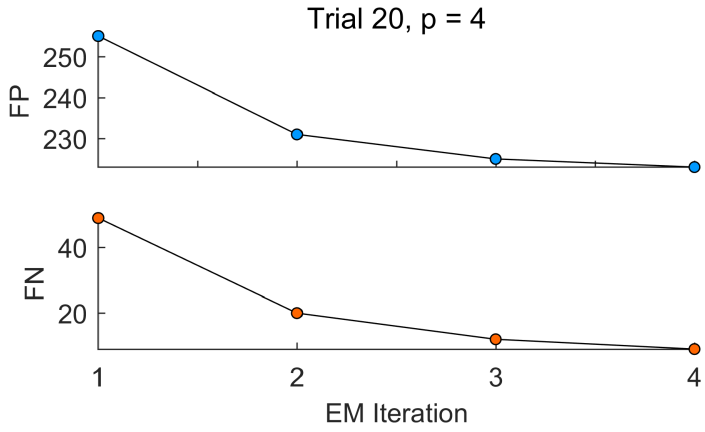


Figure 5.4: The number of false positives (FP) and false negatives (FN) predicted by LatLapMED decrease as the EM iterations in the algorithm increase. This is due to the synergy between the anomaly detection in the E-step and the classification in the M-step.

Table 5.4: Mean and standard deviation of CPU times over 50 trials.

	Average CPU time	Standard Deviation
SSAD+LapSVM	3.9784	0.11991
SSAD+LDS	4.2316	0.18557
SSAD+SVM	3.9491	0.29182
SSAD+RF	4.2800	0.23246
SSAD+NN	4.2534	0.26558
GEM+LapMED	1.4428	0.13992
GEM+LDS	1.7278	0.19305
GEM+SVM	1.0253	0.05716
GEM+RF	1.3775	0.09409
GEM+NN	1.2916	0.11141
1SVM+LapSVM	0.3750	0.09295
1SVM+LDS	0.7897	0.19116
1SVM+SVM	0.2106	0.06824
1SVM+RF	0.5984	0.17712
1SVM+NN	0.4781	0.11465
LatLapMED	2.9944	0.64291

In Table 5.4, we show the mean and standard deviation of the CPU time in seconds for each algorithm over 50 trials. The algorithms were run on a quad-core Intel i7-6700HQ CPU at 3.20GHz using Matlab. While we have not numerically optimized each algorithm, we used as many built-in functions and optimizers, which are written in compiled languages

(C++, Fortran), to show the best performance. LatLapMED is slower than many of the two-stage methods, but it is not exorbitantly slower, and it is still faster than two-stage methods that use SSAD.

5.5.2 Experiment on Reddit data

We apply LatLapMED to the May 2015 comments of the Reddit comment dataset [97]. We form a sample of subreddits with variables: *Avg. Number of Users*, *Avg. Gilded*, and *Avg. Score*, where only subreddits with at least 100 comments are included and additionally only the top 7,000 most controversial subreddits are chosen (from approximately 10,000). The anomalous data points are defined as those that lie in the tail 3% of any variable's marginal distribution and we are interested in only the controversial subreddits among these anomalous points. Thus, we treat the average controversy of each subreddit as a utility score, with again 30% visible and the top 25% as high utility. This mimics the situation where a domain expert is given roughly 1.5% of the dataset that is considered to be anomalous, and asked to label it.

Here the cost regularization parameter $C = 2$ is chosen to be smaller than in the simulations because we expect the margin to be noisier, and similarly the softening parameter in LDS is increased to 100. The other parameters, which basically describe the structure of the classifiers, are the same as in the simulations. We choose ϕ to control the false positive rate (FPR) to be around 0.05, which corresponds to a commonly chosen Type 1 error level. Table 5.5 shows the rates of LatLapMED, all the competing naive two-stage methods, and an "oracle" LapMED, which we use as a lower/upper bound on the best LatLapMED could do. For a Type 1 error level of 0.05, LatLapMED does considerably better than the competing methods. It has the lowest false negative rates (FNR) and the highest recall. While 1SVM+RF and GEM+NN have slightly higher precision than LatLapMED, they also have much lower recall.

Additionally compared to "oracle" LapMED, LatLapMED does not do considerably worse. Its precision is not nearly as high as the "oracle" method's; however, 191 out of the 338 subreddits incorrectly predicted to be controversial (false positives), actually have controversy scores in the top 25%, but since they are not anomalous, they are not labeled as high utility by our criteria. This is very promising because it implies that our method is able to additionally find high utility points that may not lie far enough in the tails of the empirical distribution. The recall of LatLapMED is almost as high as that of the "oracle" method's with LatLapMED only failing to label as controversial (false negatives) the subreddits ['vegetarian', 'DesignPorn'] compared to the "oracle". Otherwise, both meth-

Table 5.5: False Positive Rate, False Negative Rate, Recall, Precision for Reddit Data

	FPR	FNR	Recall	Precision
"oracle" LapMED	0.02224	0.13008	0.86992	0.41797
GEM+LapMED	0.050022	0.17886	0.82114	0.22697
1SVM+LapSVM	0.050167	0.1626	0.8374	0.22991
SSAD+LapSVM	0.050167	0.17886	0.82114	0.22646
GEM+LDS	0.050894	0.19512	0.80488	0.22049
1SVM+LDS	0.048422	0.17073	0.82927	0.23448
SSAD+LDS	0.049004	0.1626	0.8374	0.23409
GEM+SVM	0.056275	0.15447	0.84553	0.21181
1SVM+SVM	0.049295	0.1626	0.8374	0.23303
SSAD+SVM	0.048131	0.17886	0.82114	0.2338
GEM+RF	0.050749	0.17073	0.82927	0.22616
1SVM+RF	0.047114	0.17886	0.82114	0.23765
SSAD+RF	0.049731	0.1626	0.8374	0.23146
GEM+NN	0.047259	0.17886	0.82114	0.23709
1SVM+NN	0.047404	0.18699	0.81301	0.23474
SSAD+NN	0.047695	0.19512	0.80488	0.23185
LatLapMED	0.049149	0.14634	0.85366	0.23702

ods failed to find the other 16 subreddits: ['pathofexile', 'Cleveland', 'Liberal', 'mississauga', 'Eesti', 'Images', 'uofmn', 'trackertalk', 'Kuwait', 'asianbros', 'saskatchewan', 'rule34_comics', 'boop', 'macedonia', 'wanttobelieve', 'DebateACatholic'] . While some of these topics are definitely controversial, others such as 'mississauga' and 'saskatchewan' (providences of Canada) or 'uofmn' (University of Minnesota) seem to have unreasonably high controversy scores. It is not particularly worrisome that LatLapMED failed to predict these topics as controversial because the "oracle" also incorrectly classified them, so many of them could be considered mislabeled by the domain expert.

5.5.3 Experiment on CTU-13 data

Finally, we apply LatLapMED to the CTU-13 dataset, which is of botnet traffic on a university network that was captured by CTU University, Czech Republic, in 2011 [98]. The dataset contains real botnet traffic mixed with normal traffic and background traffic. The authors of [98] processed the captured traffic into bidirectional NetFlows and manually labeled them. The dataset contains 13 different scenarios and for our experiments below we

considered two scenarios, 1 and 8. Scenario 1 contains the malware Neris.exe, which is a bot that sent spam, connected to an HTTP CC, and used HTTP to do ClickFraud. Scenario 8 has malware QvodSetuPuls23.exe, which contacted many different Chinese C&C hosts, received large amounts of encrypted data, and scanned and cracked the passwords of machines. We are interested in identifying the botnet traffic (high-utility points) from the rare, but uninteresting normal traffic (low-utility, anomalous points) and the background traffic (nominal points) in a situation where instead of manually labeling all points, only a small subset is labeled.

For each scenario, we randomly sample 38,000 NetFlows of background traffic and 1000 NetFlows each of normal and botnet traffic, making 5% of the samples anomalous. We allow 300 of the normal and 300 of the botnet traffic to have visible labels so a domain expert would only be manually labeling 1.5% of all the samples in the dataset. We used 9 of the features provided by the NetFlows dataset: duration of the flow, direction of the flow, total packets, total bytes, source bytes, source and destination port numbers and IP addresses (in integer format). Thus each of the two datasets have dimensions $p = 9$ features and $n = 40{,}000$ total samples, of which $a = 2{,}000$ are anomalous and $l = 600$ are labeled. In order to have multiple trials, we perform this sampling 10 times so that we have 10 almost independent experiments for each scenario. The following results are the average of these 10 trials.

Because many of the features are discrete and not continuous, we use cosine distances and cosine kernels instead of euclidean distances and the radial basis kernel; otherwise, the parameters are the same as in Table 5.2. We choose ϕ so that the Type 1 error level (or FPR) is 0.01. Like in the Reddit experiments, we compare LatLapMED against all the competing naive two-stage methods and an "oracle" LapMED and summarize the performance in Tables 5.6 and 5.7 for scenarios 1 and 8 respectively. The only two-stage method we do not compare against are those using SSAD due to its unmanageably high computational complexity.

The average error rates of the "oracle" method shown in Tables 5.6 and 5.7 indicates that identifying the botnet traffic is not extremely difficult when the anomalies are known. However, when the anomaly indicator variables are latent or unknown, the tables show that the problem is more difficult. Nonetheless, in both scenarios, LatLapMED has the lowest false negative rates (FNR) and the highest precision and recall. The most competitive two-stage methods do not come close to the performance of LatLapMED, and particularly in scenario 8, LatLapMED has significantly higher precision and recall. This is a direct result of the fact that all malware are statistical outliers, so incorporating label information into anomaly detection helps to identify botnet traffic.

Table 5.6: Mean False Positive Rate, False Negative Rate, Recall, Precision for Scenario 1

	FPR	FNR	Recall	Precision
"oracle' LapMED	0.0038308	0.3457	0.6543	0.82145
GEM+LapMED	0.012279	0.6916	0.3084	0.39202
GEM+LDS	0.010464	0.6929	0.3071	0.4317
GEM+SVM	0.010049	0.6936	0.3064	0.44034
GEM+RF	0.0062231	0.6924	0.3076	0.56775
GEM+NN	0.010844	0.691	0.309	0.43359
1SVM+LapSVM	0.012713	0.6836	0.3164	0.38969
1SVM+LDS	0.0098	0.6816	0.3184	0.45581
1SVM+SVM	0.0093462	0.6853	0.3147	0.4641
1SVM+RF	0.0078769	0.6742	0.3258	0.51662
1SVM+NN	0.0089462	0.6762	0.3238	0.48675
LatLapMED	0.0094692	0.402	0.598	0.61899

Table 5.7: Mean False Positive Rate, False Negative Rate, Recall, Precision for Scenario 8

	FPR	FNR	Recall	Precision
"oracle' LapMED	0.0031538	0.15	0.85	0.87359
GEM+LapMED	0.011538	0.6979	0.3021	0.41185
GEM+LDS	0.011118	0.6997	0.3003	0.42204
GEM+SVM	0.011267	0.6969	0.3031	0.42978
GEM+RF	0.007859	0.6967	0.3033	0.49887
GEM+NN	0.0095487	0.696	0.304	0.4506
1SVM+LapSVM	0.0092205	0.6916	0.3084	0.4729
1SVM+LDS	0.010682	0.6919	0.3081	0.43073
1SVM+SVM	0.011746	0.6898	0.3102	0.43154
1SVM+RF	0.010272	0.6807	0.3193	0.44506
1SVM+NN	0.0096103	0.6818	0.3182	0.46041
LatLapMED	0.011064	0.1779	0.8221	0.65125

We also measure the CPU times of the two scenarios using the same Intel CPU and code as described in the simulations of subsection 5.5.1. Tables 5.8 and 5.9 show that while LatLapMED is significantly slower than all the competing methods, it on average takes less than 1 minute to process on dataset of 40,000 NetFlows, which is still very reasonable.

Table 5.8: Mean and standard deviation of CPU times (in seconds) for Scenario 1

	Average CPU time	Standard Deviation
GEM+LapMED	31.2625	0.58902
GEM+LDS	27.7656	0.68016
GEM+SVM	26.55	0.68274
GEM+RF	26.9656	0.82544
GEM+NN	27.0109	1.2891
1SVM+LapSVM	4.1141	0.10951
1SVM+LDS	3.3328	0.22134
1SVM+SVM	1.9734	0.08137
1SVM+RF	2.6484	0.18001
1SVM+NN	2.5859	0.57919
LatLapMED	56.2547	15.4514

Table 5.9: Mean and standard deviation of CPU times (in seconds) for Scenario 8

	Average CPU time	Standard Deviation
GEM+LapMED	31.7359	0.66264
GEM+LDS	29.4672	0.28389
GEM+SVM	26.4953	0.24859
GEM+RF	26.9734	0.31724
GEM+NN	26.9922	0.6696
1SVM+LapSVM	4.3141	0.0814
1SVM+LDS	6.8063	0.22136
1SVM+SVM	1.9953	0.10154
1SVM+RF	2.6875	0.10725
1SVM+NN	2.6219	0.55448
LatLapMED	50.3656	0.84805

5.6 Conclusion

We have proposed a novel data-driven method called latent Laplacian minimum entropy discrimination (LatLapMED) for detecting anomalous points that are of high utility. LatLapMED extends the MED framework to simultaneously handle semi-supervised utility labels and incorporate anomaly information. Through this extended framework, LatLapMED exploits the key idea that high-utility points are also anomalous, which allows it to work

successfully when provided with a very small number of utility labels. Our simulation results show its advantages over combinations of standard anomaly detection and classification algorithms. In particular, theses two-stage approaches perform worse because they treat statistical rarity and label information as independent components, which LatLapMED overcomes by explicitly combining them through a latent variable model and the EM algorithm. This performance increase is shown in the EM iterations of LatLapMED where using previous label information helps identify anomalies and vice versa. Finally, we applied our method to the Reddit and CTU-13 botnet datasets to show its applicability in real life situations where only certain high-utility anomalies are of interest to the end user.

Appendix

5..1 Proofs for Section III

Proof of Proposition 3. The posterior $\mathrm{P}(\boldsymbol{\theta}, b, \boldsymbol{\gamma}, \lambda | \boldsymbol{X}, \boldsymbol{y})$ is a log concave distribution where the log posterior can be treated as a Lagrangian function. So the MAP estimator $\hat{\boldsymbol{\theta}}$ is the solution to $\frac{\partial}{\partial \boldsymbol{\theta}} \log\left(\mathrm{P}(\boldsymbol{\theta}, b, \boldsymbol{\gamma}, \lambda | \boldsymbol{X}, \boldsymbol{y})\right) = \sum_{i=1}^{l} \alpha_i y_i \boldsymbol{X}_i^T - (\boldsymbol{I} + 2\beta \boldsymbol{X}^T \mathcal{L} \boldsymbol{X})\boldsymbol{\theta} = \mathbf{0}$ and the Lagrange multipliers $\boldsymbol{\alpha}$ are the solution to

$$\frac{\partial}{\partial \boldsymbol{\alpha}} \log\left(\mathrm{P}(\boldsymbol{\theta}, b, \boldsymbol{\gamma}, \lambda | \boldsymbol{X}, \boldsymbol{y})\right) |_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} = \frac{\partial}{\partial \boldsymbol{\alpha}} - \log\left(Z(\boldsymbol{\alpha})\right) = \mathbf{0}$$

where $-\log\left(Z(\boldsymbol{\alpha})\right) =$ Bias + Smoothness + $\sum_{i=1}^{l}$ Margin$_i$ + Weight, defined as

$$\text{Bias: } -\log\left(\int_{-\infty}^{\infty}\frac{e^{-b^2/2\sigma^2}}{2\pi\sigma^2}\exp\left\{\sum_{i=1}^{l}\alpha_i y_i b\right\}db\right) = -\frac{\sigma^2}{2}\left(\sum_{i=1}^{l}\alpha_i y_i\right)^2$$

$$\Rightarrow \text{ if } \sigma\to\infty, \text{ then } \sum_{i=1}^{l}\alpha_i y_i = 0$$

$$\text{Smoothness: } -\log\left(\int_{0}^{\infty}Be^{-B\lambda}e^{\beta\lambda}\,d\lambda\right) = \log\left(1-\frac{\beta}{B}\right)\to 0 \text{ as } B\to\infty$$

$$\text{Margin: } -\log\left(\int_{-\infty}^{1}Ce^{-C(1-\gamma_i)}e^{-\alpha_i\gamma_i}\,d\gamma_i\right) = \alpha_i + \log(1-\alpha_i/C)$$

$$\text{Weight: } -\log\left(\int_{-\infty}^{\infty}\frac{e^{-\boldsymbol{\theta}^T\boldsymbol{\theta}/2}}{(2\pi)^{p/2}}\exp\left\{\sum_{i=1}^{l}\alpha_i y_i\boldsymbol{X}_i\boldsymbol{\theta} - \beta\boldsymbol{\theta}^T\boldsymbol{X}^T\mathcal{L}\boldsymbol{X}\boldsymbol{\theta}\right\}d\boldsymbol{\theta}\right)$$

$$= -\frac{1}{2}\left(\sum_{i=1}^{l}\alpha_i y_i\boldsymbol{X}_i\right)(\boldsymbol{I}+2\beta\boldsymbol{X}^T\mathcal{L}\boldsymbol{X})^{-1}\left(\sum_{i=1}^{l}\boldsymbol{X}_i^T\alpha_i y_i\right) + \log\left(\det(\boldsymbol{I}+2\beta\boldsymbol{X}^T\mathcal{L}\boldsymbol{X})\right)$$

$$= -0.5\,\boldsymbol{\alpha}^T\boldsymbol{Y}\boldsymbol{J}(\boldsymbol{K}^{-1}+2\beta\mathcal{L})^{-1}\boldsymbol{J}^T\boldsymbol{Y}\boldsymbol{\alpha} + \text{tr}\left(\log(\boldsymbol{I}+2\beta\mathcal{L}\boldsymbol{K})\right)$$

$$\propto -0.5\,\boldsymbol{\alpha}^T\boldsymbol{Y}\boldsymbol{J}\boldsymbol{K}(\boldsymbol{I}+2\beta\mathcal{L}\boldsymbol{K})^{-1}\boldsymbol{J}^T\boldsymbol{Y}\boldsymbol{\alpha}.$$

Thus the relationship between the probabilistic primal estimator and the kernel dual estimator is $\boldsymbol{X}\hat{\boldsymbol{\theta}} = \boldsymbol{X}(\boldsymbol{I}+2\beta\boldsymbol{X}^T\boldsymbol{L}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{J}^T\boldsymbol{Y}\hat{\boldsymbol{\alpha}} = \boldsymbol{K}(\boldsymbol{I}+2\beta\mathcal{L}\boldsymbol{K})^{-1}\boldsymbol{J}^T\boldsymbol{Y}\hat{\boldsymbol{\alpha}}$.

□

5..2 Proofs for Section IV

Proof of Lemma 5.4.1. Note that all labeled points are anomalous so $y_i\eta_i = y_i$ for all $i \in [1, l]$ or $\boldsymbol{J}\boldsymbol{H} = \boldsymbol{J}$. Thus following the same procedure as Proposition 3, the MAP estimator for the posterior P$(\boldsymbol{\theta}, b, \boldsymbol{\gamma}, \lambda|\boldsymbol{X}, \boldsymbol{\eta}, \boldsymbol{y})$ is $\hat{\boldsymbol{\theta}} = (\boldsymbol{I}+2\beta\boldsymbol{X}^T\boldsymbol{h}^T\mathcal{L}_\eta\boldsymbol{h}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{J}^T\boldsymbol{Y}\boldsymbol{\alpha}$ where

the Lagrange multipliers $\boldsymbol{\alpha}$ are the solution to $\arg\max_{\boldsymbol{\alpha}} -\log\left(Z(\boldsymbol{\alpha})\right)$, which has terms

$$\text{Bias: } -\log\left(\int_{-\infty}^{\infty}\frac{e^{-b^2/2\sigma^2}}{2\pi\sigma^2}\exp\left\{\sum_{i=1}^{l}\alpha_i y_i\eta_i b\right\}db\right) = -\frac{\sigma^2}{2}\left(\sum_{i=1}^{l}\alpha_i y_i\eta_i\right)^2$$

$$\Rightarrow \text{ if } \sigma\to\infty, \text{ then } \sum_{i=1}^{l}\alpha_i y_i\eta_i = \sum_{i=1}^{l}\alpha_i y_i = 0$$

Smooth: Same as Proposition 3

Margin: Same as Proposition 3

Weight:

$$-\log\left(\int_{-\infty}^{\infty}\frac{e^{-\boldsymbol{\theta}^T\boldsymbol{\theta}/2}}{(2\pi)^{p/2}}\exp\left\{\sum_{i=1}^{l}\alpha_i y_i\eta_i\boldsymbol{X}_i\boldsymbol{\theta} - \beta\boldsymbol{\theta}^T\boldsymbol{X}^T\boldsymbol{h}^T\mathcal{L}_\eta\boldsymbol{h}\boldsymbol{X}\boldsymbol{\theta}\right\}d\boldsymbol{\theta}\right)$$

$$\propto -\frac{1}{2}\left(\sum_{i=1}^{l}\alpha_i y_i\eta_i\boldsymbol{X}_i\right)(\boldsymbol{I}+2\beta\boldsymbol{X}^T\boldsymbol{h}^T\mathcal{L}_\eta\boldsymbol{h}\boldsymbol{X})^{-1}\left(\sum_{i=1}^{l}\boldsymbol{X}_i^T\alpha_i y_i\eta_i\right)$$

$$= -\frac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{Y}\boldsymbol{J}\boldsymbol{H}(2\beta\boldsymbol{h}^T\mathcal{L}_\eta\boldsymbol{h}+\boldsymbol{K}^{-1})^{-1}\boldsymbol{H}\boldsymbol{J}^T\boldsymbol{Y}\boldsymbol{\alpha}$$

$$= -\frac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{Y}\boldsymbol{J}\left((2\beta\boldsymbol{h}^T\mathcal{L}_\eta\boldsymbol{h}+\boldsymbol{K}^{-1})^{-1}\odot\boldsymbol{\eta}\boldsymbol{\eta}^T\right)\boldsymbol{J}^T\boldsymbol{Y}\boldsymbol{\alpha}$$

Instead of $n\times n$ matrix operations, the Weight term can be compressed to $a\times a$ matrix operations by permuting the rows and columns of the matrices so that the first a rows/cols correspond to $\eta_i = 1$. Then $\left((2\beta\boldsymbol{h}^T\mathcal{L}_\eta\boldsymbol{h}+\boldsymbol{K}^{-1})^{-1}\odot\boldsymbol{\eta}\boldsymbol{\eta}^T\right)$ is 0 everywhere except for the top left $a\times a$ block, which (using block matrix inversion) can be expressed as $[(2\beta\boldsymbol{h}^T\mathcal{L}_\eta\boldsymbol{h}+\boldsymbol{K}^{-1})^{-1}]_{11} = (\mathcal{L}_\eta+\boldsymbol{K}_\eta^{-1})^{-1} = \boldsymbol{K}_\eta(\boldsymbol{I}+\mathcal{L}_\eta\boldsymbol{K}_\eta)^{-1}$ where $\boldsymbol{K}_\eta$ is the top left $a\times a$ block of the original Gram matrix.

Thus the primal dual relationship is $\boldsymbol{X}\hat{\boldsymbol{\theta}} = \boldsymbol{K}_\eta(\boldsymbol{I}+2\beta\mathcal{L}_\eta\boldsymbol{K}_\eta)^{-1}\boldsymbol{J}^T\boldsymbol{Y}\hat{\boldsymbol{\alpha}}$.

□

Proof of Lemma 5.4.2. Because $\mathrm{P}(\boldsymbol{X},\boldsymbol{\eta},\boldsymbol{y}|\boldsymbol{\alpha})$ is log convex, the function $-\log(\cdot)$ is concave on its domain. Thus by Jensen's Inequality, $-\log\left(\mathrm{P}(\boldsymbol{X},\boldsymbol{y}|\boldsymbol{\alpha})\right)$

$$= -\log\left(\sum_{\eta_1=0}^{1}\cdots\sum_{\eta_n=0}^{1}q(\boldsymbol{\eta})\frac{\mathrm{P}(\boldsymbol{X},\boldsymbol{\eta},\boldsymbol{y}|\boldsymbol{\alpha})}{q(\boldsymbol{\eta})}\right) \geq \mathrm{E}_\eta\left(-\log\mathrm{P}(\boldsymbol{X},\boldsymbol{y},\boldsymbol{\eta}|\boldsymbol{\alpha})\right) - \mathrm{E}_\eta\left(-\log q(\boldsymbol{\eta})\right)$$

where $q(\boldsymbol{\eta})$ is an arbitrary distribution. A natural choice for the distribution is $q(\boldsymbol{\eta}) = \mathrm{P}(\boldsymbol{\eta}|\boldsymbol{X},\boldsymbol{y},\boldsymbol{\alpha}^{t-1})$ where $\boldsymbol{\alpha}^{t-1}$ are the optimal Lagrange multipliers of the previous iteration. Since the second term $\mathrm{E}_\eta\left(\log\left(\mathrm{P}(\boldsymbol{\eta}|\boldsymbol{X},\boldsymbol{y},\boldsymbol{\alpha}^{t-1})\right)\right)$ does not depend on $\boldsymbol{\alpha}$, it can be dropped

so the lower bound is proportional to just the first term $\mathrm{E}_\eta\left(-\log\left(Z(\boldsymbol{\alpha})\right)\right) =$

$$\sum_{i=1}^{l}\alpha_i + \mathrm{E}_\eta\left(-\frac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{Y}\boldsymbol{J}\boldsymbol{H}(2\beta\boldsymbol{H}^T\boldsymbol{\mathcal{L}}\boldsymbol{H}+\boldsymbol{K}^{-1})^{-1}\boldsymbol{H}\boldsymbol{J}^T\boldsymbol{Y}\boldsymbol{\alpha}\right) + \sum_{i=1}^{l}\log(1-\alpha_i/C)$$
$$\text{subject to } \sum_{i=1}^{l}\alpha_i y_i = 0, \alpha_1, \ldots, \alpha_l \geq 0.$$

By Jensen's inequality again and ($\boldsymbol{J}\boldsymbol{H} = \boldsymbol{J}$), the quadratic term above has lower bound

$$\begin{aligned}
&\geq -\frac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{Y}\boldsymbol{J}\left(\boldsymbol{K}^{-1} + 2\mathrm{E}_\eta(\boldsymbol{H}^T\boldsymbol{\mathcal{L}}\boldsymbol{H})\right)^{-1}\boldsymbol{J}^T\boldsymbol{Y}\boldsymbol{\alpha} \\
&= -\frac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{Y}\boldsymbol{J}\boldsymbol{K}\left(\boldsymbol{I} + 2\mathrm{E}_\eta(\boldsymbol{H}^T\boldsymbol{\mathcal{L}}\boldsymbol{H})\boldsymbol{K}\right)^{-1}\boldsymbol{J}^T\boldsymbol{Y}\boldsymbol{\alpha}.
\end{aligned}$$

□

Proof of Lemma 5.4.3. Define k_L and k_{L_η} as the number of neighbors in the kNNG of the graph Laplacians $\mathcal{L}$ and $\mathcal{L}_\eta$. There are kNNG with at least k_L neighbors in $\boldsymbol{H}^T\boldsymbol{\mathcal{L}}\boldsymbol{H}$, which is formed on all the data and then pruned to just contain just the anomalous nodes, that will contain the subgraph in $\boldsymbol{h}^T\mathcal{L}_\eta\boldsymbol{h}$, which is a kNNG of only anomalous nodes with k_{L_η} neighbors. This is true for any $\boldsymbol{\eta}$ or its estimators $\hat{\boldsymbol{\eta}}$. So, there exists some m (defined as the first m points of any anomalous point are also anomalous) and $k_L \geq k_{L_\eta}$ such that

$$\begin{aligned}
&||\mathrm{E}(\boldsymbol{H}^T\boldsymbol{\mathcal{L}}\boldsymbol{H}|\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\alpha}^{t-1}) - \mathrm{E}(\boldsymbol{h}^T\mathcal{L}_\eta\boldsymbol{h}|\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\alpha}^{t-1})||_F \leq \delta(m) \\
&||\hat{\boldsymbol{h}}^T\hat{\mathcal{L}}_\eta\hat{\boldsymbol{h}} - \hat{\boldsymbol{H}}^T\boldsymbol{\mathcal{L}}\hat{\boldsymbol{H}}||_F \leq \delta'(m)
\end{aligned}$$

with equality and $\delta(m) = \delta'(m) = 0$ when $k_L = k_{L_\eta} = m$ because then the pruned graph is exactly the graph in $\mathcal{L}_\eta$.

And since the GEM principle described in Section 5.4.3 gives a good estimator, then

$$||\hat{\boldsymbol{H}}^T\boldsymbol{\mathcal{L}}\hat{\boldsymbol{H}} - \mathrm{E}(\boldsymbol{H}^T\boldsymbol{\mathcal{L}}\boldsymbol{H}|\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\alpha}^{t-1})||_F \leq \zeta$$

has small ζ. So if m is sufficiently large relative to k_{L_η} so that $\delta'(m)$ and $\delta(m)$ are small,

then $\hat{\boldsymbol{h}}^T\hat{\mathcal{L}}_\eta\hat{\boldsymbol{h}}$ is a good estimator

$$\begin{aligned}
&||\hat{\boldsymbol{h}}^T\hat{\mathcal{L}}_\eta\hat{\boldsymbol{h}} - \mathrm{E}(\boldsymbol{h}^T\mathcal{L}\boldsymbol{h}|\boldsymbol{X},\boldsymbol{y},\boldsymbol{\alpha}^{t-1})||_F \\
&\leq ||\hat{\boldsymbol{h}}^T\hat{\mathcal{L}}_\eta\hat{\boldsymbol{h}} - \hat{\boldsymbol{H}}^T\mathcal{L}\hat{\boldsymbol{H}}||_F + ||\hat{\boldsymbol{H}}^T\mathcal{L}\hat{\boldsymbol{H}} - \mathrm{E}(\boldsymbol{H}^T\mathcal{L}\boldsymbol{H}|\boldsymbol{X},\boldsymbol{y},\boldsymbol{\alpha}^{t-1})||_F \\
&\quad + ||\mathrm{E}(\boldsymbol{H}^T\mathcal{L}\boldsymbol{H}|\boldsymbol{X},\boldsymbol{y},\boldsymbol{\alpha}^{t-1}) - \mathrm{E}(\boldsymbol{h}^T\mathcal{L}_\eta\boldsymbol{h}|\boldsymbol{X},\boldsymbol{y},\boldsymbol{\alpha}^{t-1})||_F \\
&\leq \delta'(m) + \zeta + \delta(m)
\end{aligned}$$

(by triangle inequality) because $\delta'(m) + \zeta + \delta(m)$ is also small.

□

Proof of Theorem 5.4.4. By Jensen's inequality, the log observed posterior has tight lower bound, $\log\left(\mathrm{P}(\boldsymbol{\theta}, b, \boldsymbol{\gamma}, \lambda|\boldsymbol{X}, \boldsymbol{y})\right)$

$$\geq \log\left(\mathrm{P}_0(\boldsymbol{\theta}, b, \boldsymbol{\gamma}, \lambda)\right) + \mathrm{E}_\eta\left(\log\left(\mathrm{P}(\boldsymbol{X}, \boldsymbol{\eta}, \boldsymbol{y}|\boldsymbol{\theta}, b, \boldsymbol{\gamma}, \lambda)\right)\right) - \mathrm{E}_\eta\left(\log\left(\mathrm{P}(\boldsymbol{X}, \boldsymbol{\eta}, \boldsymbol{y}|\boldsymbol{\alpha})\right)\right)$$

where the expectation is with respect to $\mathrm{P}(\boldsymbol{\eta}|\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\alpha}^{t-1})$. When the posterior is the MED solution using constraints (5.2) and (5.4), maximizing the lower bound for $\boldsymbol{\theta}$ gives the primal form for the M-step as the solution to derivative of the lower bound

$$\sum_{i=1}^{l} \alpha_i y_i \mathrm{E}_\eta(\eta)\boldsymbol{X}_i^T - \left(\boldsymbol{I} + 2\beta\boldsymbol{X}^T\mathrm{E}_\eta(\boldsymbol{h}^T\mathcal{L}_\eta\boldsymbol{h})\boldsymbol{X}\right)\boldsymbol{\theta} = \boldsymbol{0}.$$

Following the same procedure as Lemma 5.4.2, the dual form for the M-step has a lower bound with quadratic term

$$-\frac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{Y}\boldsymbol{J}\boldsymbol{K}\left(\boldsymbol{I} + 2\mathrm{E}_\eta(\boldsymbol{h}^T\mathcal{L}_\eta\boldsymbol{h})\boldsymbol{K}\right)^{-1}\boldsymbol{J}^T\boldsymbol{Y}\boldsymbol{\alpha}.$$

So using the same block matrix inversion procedure as Lemma 5.4.1, the dual objective for the M-step is

$$\begin{aligned}
&\sum_{i=1}^{l}\alpha_i + \log\left(1 - \frac{\alpha_i}{C}\right) - \frac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{Y}\boldsymbol{J}\hat{\boldsymbol{K}}_\eta(\boldsymbol{I} + 2\beta\hat{\mathcal{L}}_\eta\hat{\boldsymbol{K}}_\eta)^{-1}\boldsymbol{J}^T\boldsymbol{Y}\boldsymbol{\alpha} \\
&\qquad \text{subject to } \sum_{i=1}^{l}\alpha_i y_i = 0,\ \alpha_1, \ldots, \alpha_l \geq 0
\end{aligned}$$

where $\hat{\mathcal{L}}_\eta$ is the Laplacian matrix on only the set of data points $\{\boldsymbol{X}_i : \hat{\eta}_i = 1\}$ and $\hat{\boldsymbol{K}}_\eta$ is the $a \times a$ submatrix of these same data points.

□

Proof of Corollary 5.4.4.1. The primal dual relationship is

$$\boldsymbol{X}\hat{\boldsymbol{\theta}} = k(\boldsymbol{X}, \boldsymbol{X}_{\hat{\eta}})(\boldsymbol{I} + 2\beta\hat{\mathcal{L}}_{\eta}\hat{\boldsymbol{K}}_{\eta})^{-1}\boldsymbol{J}^{T}\boldsymbol{Y}\hat{\boldsymbol{\alpha}}.$$

So for any point $\boldsymbol{X}_{i'}$, the prediction is

$$\hat{\eta}_{i'}\left(\boldsymbol{X}_{i'}\hat{\boldsymbol{\theta}} + \hat{b}\right) = \hat{\eta}_{i'}\left(k(\boldsymbol{X}_{i'}, \boldsymbol{X}_{\hat{\eta}})(\boldsymbol{I} + 2\beta\hat{\mathcal{L}}_{\eta}\hat{\boldsymbol{K}}_{\eta})^{-1}\boldsymbol{J}^{T}\boldsymbol{Y}\hat{\boldsymbol{\alpha}} + \hat{b}\right).$$

Because all nominal points are low utility, for simplicity they will be given the predicted label -1.

□

CHAPTER 6

Anomaly Detection in Partially Observed Traffic Networks

In this chapter, we address the problem of detecting anomalous activity in traffic networks where the network is not directly observed. Given knowledge of what the node-to-node traffic in a network should be, any activity that differs significantly from this baseline would be considered anomalous. We propose a Bayesian hierarchical model for estimating the traffic rates and detecting anomalous changes in the network. The probabilistic nature of the model allows us to perform statistical goodness-of-fit tests to detect significant deviations from a baseline network. We show that due to the more defined structure of the hierarchical Bayesian model, such tests perform well even when the empirical models estimated by the EM algorithm are misspecified. We apply our model to both simulated and real datasets to demonstrate its superior performance over existing alternatives.

6.1 Introduction

In today's connected world, communication is increasingly voluminous, diverse, and essential. Phone calls, delivery services, and the Internet are all modern amenities that send massive amounts of traffic over immense networks. Thus network security, such as the ability to detect network intrusions or illegal network activity, plays a vital role in defending these network infrastructures. For example, (i) computer networks can protect themselves from malware such as botnets by identifying unusual network flow patterns; (ii) supply chains can prevent cargo theft by monitoring the schedule of shipments or out-of-route journeys between warehouses; (iii) law enforcement agencies can uncover smuggling operations by detecting alternative modes of transporting goods.

Identifying unusual network activity requires a good estimator of the true network traffic, including the anomalous activity, in order to distinguish it from a baseline of what the

network should look like. However, often it is not possible to for an external observer to observe the network directly due to constraints such as cost, protocols, or legal restrictions. This makes the problem of estimating the rate of traffic between nodes in a network difficult because the edges between nodes are latent unobserved variables. Network tomography approaches have been previously proposed for estimating network topology or reconstructing link traffic from incomplete measurements and limited knowledge about network connectivity. However for network anomography, the detection of anomalous deviations of traffic in the network, highly accurate estimation of all network traffic may not be necessary. It often suffices to detect perturbations within the network at an aggregate or global scale. This paper addresses the problem of network anomography rather than that of network tomography or traffic estimation.

6.1.1 Related Work

Broadly defined, the network tomography problem is to reconstruct complete network properties, e.g., source-destination (SD) traffic or network topology, based on incomplete data. The term "network tomography" was introduced in [99] where the objective is to estimate unknown source destination traffic intensities given observations of link traffic and known network topology. Since the publication of [99], the scope of the term network tomography has been used in a much broader sense (see the review papers [100, 101, 102], and [103]). For example, a variety of passive or active packet probing strategies have been used for topology reconstruction of the Internet, including unicast, multicast, or multi-multicast [104, 105], and [106]; or using different statistical measures including packet loss, packet delay, or correlation [107, 108, 109], and [110].

In the formulation of [99], the network tomography objective is to determine the total amount of traffic between SD pairs given knowledge of the physical network topology and the total amount of traffic flowing over links, called the link data. This leads to the linear model for the observations $\boldsymbol{y}^t = \boldsymbol{A}\boldsymbol{x}^t$ where $\boldsymbol{A}$ is the known routing matrix defining the routing paths, and at each time point t, $\boldsymbol{y}^t$ is a vector of the observed total traffic on the links and $\boldsymbol{x}^t$ is a vector of the unobserved message traffic between SD pairs. Using the model that the elements of $\boldsymbol{x}^t$ are independent and Poisson distributed, an expectation-maximization (EM) maximum likelihood estimator (MLE) and a method of moments estimator are proposed in [99] for the Poisson rate parameters $\boldsymbol{\lambda}$. The authors of [111] propose a Bayesian conditionally Poisson model, which uses a Markov chain Monte Carlo (MCMC) method to iteratively draw samples from the joint posterior of $\boldsymbol{\lambda}$ and $\boldsymbol{x}$. The authors of [112] and [113] assume the message traffic is instead from a Normal distribution, obtaining a computation-

ally simpler estimator of the SD traffic rates. The authors of [114] relax the assumption that the traffic is an independent and identically Poisson distributed sequence and instead consider the network as a directly observable Markov chain. Under this weaker assumption, they derive a threshold estimator for the Hoeffding test in order to detect if the network contains anomalous activity.

In [115] the authors propose an EM approach for Poisson maximum likelihood estimation when the network topology is unknown; however, their solution is only computationally feasible for very small networks and it does not account for observations of traffic through interior nodes. This has led to simpler and more scalable solutions in the form of gravity models where the rate of traffic between each SD pair is modeled by $x_{sd} = (N_s N_d)/N$ where N_s and N_d are the total traffic out of the source node and into the destination node respectively and N is the total traffic in the network. Standard gravity models do not account for the interior nodes, thus in [116] and [117] tomogravity and entropy regularized tomogravity models were proposed, which incorporate the interior node information in the second stage of their algorithm. The authors of [118] generalize the tomogravity model from a rank one (time periods are independent) to a low rank approximation (time periods are correlated) and allow additional observations on individual SD pairs. Similarly, the authors of [119] and [120] use a low rank model with network traffic maps to incorporate a sparse anomaly matrix, and they solve their multiple convex objectives with the alternating direction method of multipliers (ADMM) algorithm.

Dimensionality reduction has also been used directly for anomaly detection in the SD traffic flows in networks. Under the assumption that traffic links have low rank structure, the authors in [121] and [122] use Principle Component Analysis (PCA) to separate the anomalous traffic from the nominal traffic. This low rank framework is generalized to applying PCA in networks that are temporally low rank or have dynamic routing matrices, in [123]. The authors of [123] also coin the term "network anomography" to reflect the influence of network topology reconstruction, which is a necessary component to detecting anomalies in a network with unknown structure. However, later work in [124] discusses the limitations of PCA for detecting anomalous network traffic, e.g., it is sensitive to (i) the choice of subspace size; (ii) the way traffic measurements are aggregated; (iii) large anomalies. The low rank plus sparse framework is extended to online setting with a subspace tracking algorithm in [125].

Specifically for Internet Protocol (IP) networks, some works prefer to perform anomaly detection on the flows from the IP packets instead of the SD flows. The authors of [126] use PCA to separate the anomalous and nominal flows from sketches (random aggregations of IP flows) while the authors of [127] model the sketches as time series and detect change

points with forecasting. The works of [128] and [129] also perform change point detection using windowed hypothesis testing with generalized likelihood ratio or relative entropy respectively.

Because our approach in this paper is based on traffic networks or SD models, these types of approaches were the focus of our related works subsection. However, networks can also be represented as graph models or as features of the network characteristics. This subsection would be incomplete if it did not mention anomaly detection approaches to other types of network models. So, we refer to some survey papers that cover many of the recent techniques in graph based approaches: [130] and [131]. In particular, similar to the low rank approaches for SD networks, there are low rank approaches to graph models such as [132] who assume the inverse covariance matrix of their wireless sensor network data has a graph structure and solve a low rank penalized Gaussian graphical model problem and [133] who impose graph smoothness by a low rank assumption on graph Laplacian of the features of the network. [134] also uses a low rank approach on their KDD intrusion data set, but they directly apply the low rank assumption to the network characteristics of their data.

6.1.2 Our Contribution

In this paper, we consider networks where an exterior node (a node in an SD pair) only transmits and receives messages from a few other nodes, but because, as an external observer (one that is not located on a node), we cannot observe network directly, we do not know which SD pairs have traffic and which do not. Thus, we develop a novel framework to detect anomalous traffic in sparse networks with unknown sparsity pattern. Our contributions are the following. 1) In order to estimate the network traffic, we propose a parametric hierarchical model that alternates between estimating the unobserved network traffic and optimizing for the best fit rates of traffic using the EM algorithm. 2) We warm-start the algorithm with the solution to non-parametric minimum relative entropy model that directly projects the rates of traffic onto the nearest attainable sparse network. 3) Since we do not make assumptions of fixed edge structure in our model, it allows us to accommodate the possibility of anomalous edges in the actual network structure because anomalies will never be known in advance. 4) Using our probabilistic model's estimator of actual traffic rates, we test for anomalous network activity by comparing it to a baseline to determine which deviations are anomalies and which are estimation noise. We develop specific statistical tests, based on the generalized likelihood ratio framework, to control for the false positive rate of our probabilistic model, and show that even when our models are misspecified, our

tests can accurately detect anomalous activity in the network.

The rest of the paper is organized in the following way. Section II proposes a problem formulation of the network we are interested in and our assumptions about it. Section III describes our proposed hierarchical Bayesian model, which is solved with a generalized EM algorithm and warm-starting the EM with a solution that satisfies the minimum relative entropy principle. Section IV describes our anomaly detection scheme through statistical goodness of fit tests and Section V describes the computational complexity of our method. Section VI contains simulation results of the performance of our proposed estimators and applications to the CTU-13 dataset of botnet traffic and a dataset of NYC taxicab traffic. Finally, Section VII concludes the paper.

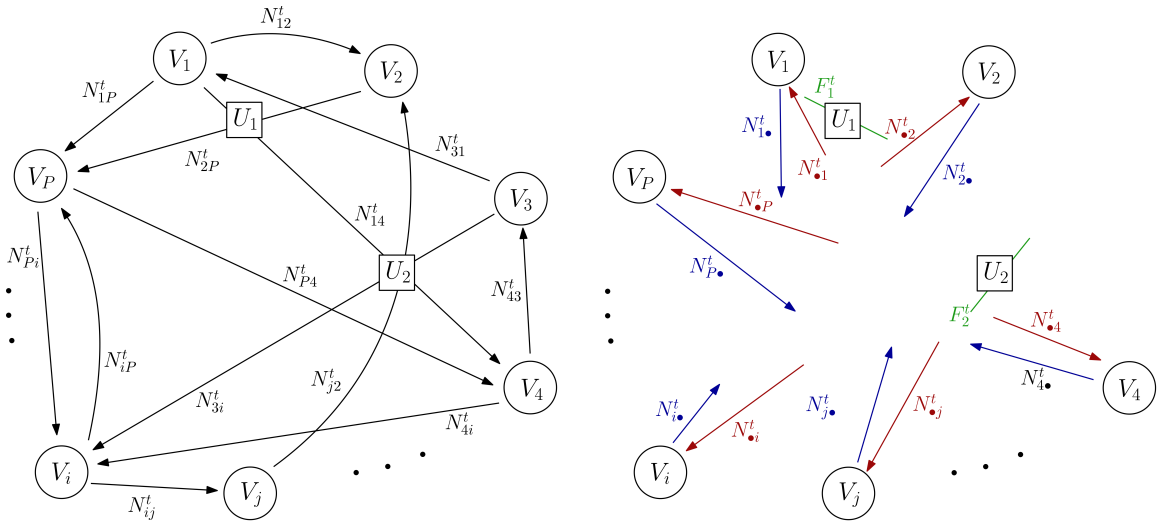
6.2 Proposed Formulation

We give a simple diagram of a notional network in Fig. 6.1a. An exterior node, V_i, sends messages, N_{ij}^t, at a rate, Λ_{ij}, to another exterior node, V_j, at each time point, t. Messages can flow through interior nodes, such as U_1, but the interior nodes do not absorb or create messages. Because the magnitude of flow is just the total number of messages that have been sent from one node to another, network traffic between nodes is a counting process. For tractability, it is common to assume the messages are independent and identically distributed (i.i.d.) and the total number of messages in a time period is from some parametric distribution. The Poisson distribution is the most natural choice because it models events occurring independently with a constant rate, and it is used by [99], [115], [111], [112], and [113] although the latter two works use a Normal approximation to the Poisson for additional tractability.

When the network is observed directly, the edge structure and rates can be easily estimated using a sample of observations at different time points. Under these Poisson process assumptions, the uniformly minimum variance unbiased estimator is simply the maximum likelihood estimator (MLE) of the Poisson distribution. However, this is a very strong and unrealistic assumption because it implies that we, as an external observer, are able to track every single message being passed in the network. Thus, we are interested in the much weaker assumption that we can only monitor the nodes themselves. Fig. 6.1b shows what we can actually observe from the network under this weaker assumption. While we also observe the total amount of traffic, unlike in [99], we do not know the network topology.

Since we can only monitor the nodes, we can only observe the total ingress and egress of the exterior nodes. Thus we know an exterior node, V_i, transmits $N_{i\cdot}^t$ messages and receives $N_{\cdot i}^t$ messages, but we do not know which of the other nodes it is interacting with.

We can also observe the flow through interior nodes, but we cannot distinguish where the messages come from or are going to. For instance, in Fig. 6.1a, an interior node, such as U_1, will observe all messages, $F_1^t = N_{14}^t + N_{2P}^t$, that flow through it, but it will not be able to distinguish the number of messages from each SD pair or whether all the SD pairs actually send messages.



(a) Proposed Network: V_i - exterior nodes, U_i - interior nodes, N_{ij}^t - messages from node i to node j at time point t

(b) Actual Observed Network: $N_{i\cdot}^t$ - total egress of exterior nodes, $N_{\cdot i}^t$ - total ingress of exterior nodes, F_i^t - total flow through interior nodes

Figure 6.1: Diagram of a network with P exterior nodes and 2 interior nodes.

A network with P exterior nodes can naturally be mathematically formulated as a $P \times P$ matrix, which is observed T times. Let $\boldsymbol{N}^t$ be the unobserved traffic matrix at time instance t and let the elements of the matrix, N_{ij}^t, be the amount of traffic between nodes i and j. The row and column sums of the traffic are denoted by $\boldsymbol{R} = [N_{1\cdot} \dots N_{P\cdot}]'$ and $\boldsymbol{C} = [N_{\cdot 1} \dots N_{\cdot P}]'$ respectively, and $\boldsymbol{F} = [F_h]$ are the observed flows through interior nodes, which are indexed by h. The traffic at each time instance t is generated from a distribution with mean $\boldsymbol{\Lambda}$, the true intensity/rate parameter of the matrix, and $\boldsymbol{\Lambda}_0$ is the baseline parameter of a network without any anomalies. This mathematical formulation is shown below.

$$\boldsymbol{N}^t = \begin{bmatrix} 0 & N_{12}^t & N_{13}^t & \cdots & N_{1P}^t \\ N_{21}^t & 0 & N_{23}^t & \cdots & N_{2P}^t \\ N_{31}^t & N_{32}^t & 0 & \cdots & N_{3P}^t \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ N_{P1}^t & N_{P2}^t & N_{P3}^t & \cdots & 0 \end{bmatrix}$$

<u>Observations</u>

$N_{\cdot j}^t = \sum_{i=1}^P N_{ij}^t$

$N_{i\cdot}^t = \sum_{j=1}^P N_{ij}^t$

$F_h^t = \sum N_{ij}^t$

N_{ij}^t for some ij

We assume *a priori* that the distribution of the rate matrix is centered around some baseline rate matrix $\mathbf{\Lambda_0}$, which are the assumed rates when there is no anomalous activity. We then update this prior distribution using the observations $\mathcal{D} = \{\boldsymbol{R}^t, \boldsymbol{C}^t, \boldsymbol{F}^t\}_{t=1}^T$ in order to get a distribution of the rates $\mathrm{P}(\mathbf{\Lambda}|\mathcal{D})$, which does account for potential anomalous activity.

6.3 Hierarchical Poisson Model with EM

We propose a generative model that assumes a series of statistical distributions govern the generation of the network. We assume that the messages N_{ij}^t passed through the network are Poisson distributed with rates Λ_{ij}. However, because we cannot observe the traffic network directly, we do not have the complete Poisson likelihood and use the EM algorithm. In the following subsections, we will show a series of generative models with increasing complexity that attain successively higher accuracy. Then we will discuss warm-starting the EM algorithm at a robust initial solution to compensate for its sensitivity to initialization.

6.3.1 Proposed Hierarchical Bayesian Model

6.3.1.1 Maximum Likelihood by EM

The simplest hierarchical model assumes all priors are uniform, thus the only distributional assumption is that likelihood $\mathrm{P}(\boldsymbol{N}^1, \ldots, \boldsymbol{N}^T|\mathbf{\Lambda})$ is $\prod_{t=1}^T \prod_{ij} Poisson(\Lambda_{ij})$. The maximum likelihood estimator for the Poisson rates $\mathbf{\Lambda}$ can be approximated by lower bounds of the observed likelihood $\mathrm{P}(\mathcal{D}|\mathbf{\Lambda})$ using the maximum likelihood expectation maximization (MLEM) algorithm. The MLEM alternates between computing a lower bound on the likelihood function $\mathrm{P}(\mathcal{D}|\mathbf{\Lambda})$, the E-step, and maximizing the lower bound, the M-step. A general expression for the E-step bound can be expressed as:

$$\log \mathrm{P}(\mathcal{D}|\mathbf{\Lambda}) \geq \sum_{t=1}^{T} \mathrm{E}_{\mathrm{q}^t} \left(\log \mathrm{P}(\boldsymbol{R}^t, \boldsymbol{C}^t, \boldsymbol{F}^t, \boldsymbol{N}^t|\mathbf{\Lambda}) \right) + \mathrm{H}(\mathrm{q}^t) \tag{6.1}$$

where q^t is an arbitrarily chosen distribution of $\boldsymbol{N}^t$, $\mathrm{E}_{\mathrm{q}^t}$ denotes statistical expectation with respect to the reference distribution q^t, and $\mathrm{H}(\mathrm{q}^t)$ is the Shannon entropy of q^t. The choice of q^t that makes the bound (6.1) the tightest, and results in the fastest convergence of the MLEM algorithm, is $\mathrm{q}^t = \mathrm{P}(\boldsymbol{N}^t|\boldsymbol{R}^t, \boldsymbol{C}^t, \boldsymbol{F}^t, \mathbf{\Lambda})$, (see Section 11.4.7 of [135]); however,

this is not a tractable distribution. When the observations consist of the row and column sums of the matrix $\boldsymbol{N}^t$, this distribution is the multivariate Fisher's noncentral hypergeometric distribution, and when the flows are also observed the distribution is unknown. Unfortunately, use of this optimal distribution leads to an intractable E-step in the MLEM algorithm due to the coupling (dependence) between the row and column sums of $\boldsymbol{N}^t$. As an alternative we can weaken the bound on the likelihood function (6.1) by using a different distribution q that leads to an easier E-step. To this aim, we propose to use a distribution q that decouples the row sum from the column sum; equivalent to assuming that each sum is independent, e.g., as if each were computed with different realizations of $\boldsymbol{N}^t$.

Proposition 4. *Assume t_1, t_2 and t_3 are different time points so that observations at these time points are independent*

$$\mathrm{P}(\mathcal{D}|\boldsymbol{\Lambda}) = \prod_{t_1=1}^{T} \mathrm{P}(\boldsymbol{R}^{t_1}|\boldsymbol{\Lambda}) \prod_{t_2=1}^{T} \mathrm{P}(\boldsymbol{C}^{t_2}|\boldsymbol{\Lambda}) \prod_{t_3=1}^{T} \mathrm{P}(\boldsymbol{F}^{t_3}|\boldsymbol{\Lambda}).$$

Then the tightest lower bound of the observed data log likelihood is

$$\log \mathrm{P}(\mathcal{D}|\boldsymbol{\Lambda}) \geq \sum_{\tau=1}^{3} \sum_{t_\tau=1}^{T} \mathrm{H}(q^{t_\tau}) + \mathrm{E}_{q^{t_\tau}} \left(\log \mathrm{P}(\boldsymbol{N}^{t_\tau}|\boldsymbol{\Lambda}) \right)$$

where $q^{t_1} = \mathrm{P}(\boldsymbol{N}^{t_1}|\boldsymbol{R}^{t_1}, \boldsymbol{\Lambda})$, $q^{t_2} = \mathrm{P}(\boldsymbol{N}^{t_2}|\boldsymbol{C}^{t_2}, \boldsymbol{\Lambda})$, and $q^{t_3}(\boldsymbol{N}^{t_3}) = \mathrm{P}(\boldsymbol{N}^{t_3}|\boldsymbol{F}^{t_3}, \boldsymbol{\Lambda})$ are multinomial distributions.

In the EM algorithm, the expectation in the E-step is taken with respect to the distribution estimated using the previous iteration's estimate of the parameter $\hat{\boldsymbol{\Lambda}}^k$, and the M-step does not depend on the entropy terms in the lower bound in Proposition 4, which are constant with respect to $\boldsymbol{\Lambda}$. Since the likelihoods are all Poisson, the E-step reduces to computing the means of multinomial distributions and the M-step for any ij pair is given by the Poisson MLE with the unknown N_{ij}^t terms replaced by their mean values. Explicitly the M-step objective is

$$\hat{\Lambda}_{ij}^{k+1} = \arg\max_{\Lambda_{ij}} \; -\Lambda_{ij} + \log(\Lambda_{ij}) N_{ij}^{total} \tag{6.2}$$

where $N_{ij}^{total} = \sum_{t_1=1}^{T} \mathrm{E}(N_{ij}^{t_1}|\boldsymbol{R}^{t_1}, \hat{\boldsymbol{\Lambda}}^k) + \sum_{t_2=1}^{T} \mathrm{E}(N_{ij}^{t_2}|\boldsymbol{C}^{t_2}, \hat{\boldsymbol{\Lambda}}^k) + \sum_{t_3=1}^{T} \mathrm{E}(N_{ij}^{t_3}|\boldsymbol{F}^{t_3}, \hat{\boldsymbol{\Lambda}}^k)$ and the expectations are with respect to the multinomial distributions of Proposition 4. Thus the Poisson MLE equals $\hat{\Lambda}_{ij}^{k+1} = N_{ij}^{total}/3T$.

6.3.1.2 Maximum a Posteriori by EM

Because there are P^2 unobserved variables and only $\mathcal{O}(P)$ observed variables, the expected log likelihoods have a lot of local maxima. In order to make the EM objective better defined and incorporate the baseline Poisson rate information $\mathbf{\Lambda}_0$, a prior can be added to the likelihood model of the previous subsection. The EM objective of this new model is now the expected log posterior and the estimator in the M-step is the maximum a posteriori (MAP) estimator. It is natural to choose a conjugate prior of the form $\mathrm{P}(\mathbf{\Lambda}) = \prod_{ij} \mathrm{P}(\Lambda_{ij})$ where each $\Lambda_{ij} \sim Gamma(\epsilon_{ij}\Lambda_{0\,ij} + 1, \epsilon_{ij})$ (shape, rate) as this choice yields a closed form expression for the posterior distribution. These priors have modes at the baseline rates $\Lambda_{0\,ij}$. The hyperparameters ϵ_{ij} can be thought of as the belief we have in the correctness of the baseline so as $\epsilon \to 0$, the prior variance goes to infinity, and the prior becomes non-informative because we have no confidence in the baseline, while as $\epsilon \to \infty$, the prior variance goes to zero, and the prior degenerates into the point $\Lambda_{0\,ij}$ because we are certain the baseline is correct.

Given a matrix of hyperparameters $\boldsymbol{\epsilon}$, the complete data posterior distribution is $\mathrm{P}(\mathbf{\Lambda}|\boldsymbol{\epsilon}, \boldsymbol{N}^1, \ldots, \boldsymbol{N}^T) = \prod_{ij} \mathrm{P}(\Lambda_{ij}|\epsilon_{ij}, N_{ij}^1, \ldots, N_{ij}^T)$ where each posterior is of the form of $Gamma(\epsilon_{ij}\Lambda_{0\,ij} + 1 + \sum_{t=1}^{T} N_{ij}, \epsilon_{ij} + T)$. Because we can only observe the network indirectly $\mathcal{D} = \{\boldsymbol{R}^t, \boldsymbol{C}^t, \boldsymbol{F}^t\}_{t=1}^T$, we again must estimate the mode of this posterior using the EM algorithm, which is very similar to the algorithm for the likelihood model. The only difference is the M-step in which an additional term of the form $\sum_{ij}(\epsilon_{ij}\Lambda_{0\,ij}) \log(\Lambda_{ij}) - \epsilon_{ij}\Lambda_{ij}$ is added to (6.2). Thus at every EM iteration, the entries of the MAP estimator matrix $\hat{\mathbf{\Lambda}}^{k+1}$ are

$$\hat{\Lambda}_{ij}^{k+1} = \frac{\epsilon_{ij}\Lambda_{0\,ij} + N_{ij}^{total}}{\epsilon_{ij} + 3T} \tag{6.3}$$

where N_{ij}^{total} is the same as in (6.2).

6.3.1.3 Bayesian Hierarchical Model

Choosing the hyperparameters ϵ_{ij} can be difficult because it is not always possible to quantify our belief in the correctness of the baseline rates. We can rectify this by allowing the ϵ_{ij} to be random with hyperpriors $\epsilon_{ij} \sim Uniform(0, \infty)$. We choose uninformative hyperpriors for $\epsilon_{ij} > 0$. A notional diagram for the proposed hierarchical model is shown in Fig. 6.2.

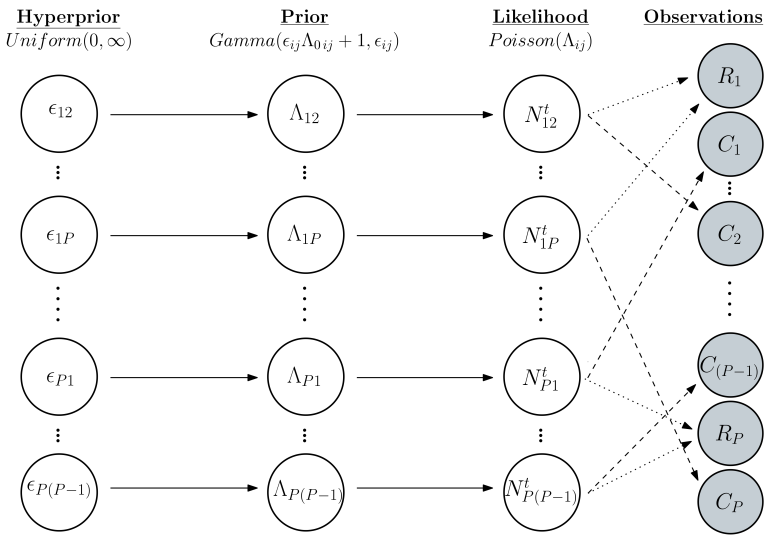


Figure 6.2: The statistical process believed to underlie our network.

With these uninformative priors the posterior takes the form

$$\begin{aligned}
\mathrm{P}(\boldsymbol{\Lambda}|\boldsymbol{N}^1,\ldots,\boldsymbol{N}^T) &= \int \frac{\mathrm{P}(\boldsymbol{N}^1,\ldots,\boldsymbol{N}^T|\boldsymbol{\Lambda})\mathrm{P}(\boldsymbol{\Lambda}|\boldsymbol{\epsilon})\mathrm{P}(\boldsymbol{\epsilon})}{\mathrm{P}(\boldsymbol{N}^1,\ldots,\boldsymbol{N}^T)}\,d\boldsymbol{\epsilon} \\
&= \int \frac{\mathrm{P}(\boldsymbol{N}^1,\ldots,\boldsymbol{N}^T|\boldsymbol{\Lambda})\mathrm{P}(\boldsymbol{\Lambda}|\boldsymbol{\epsilon})}{\mathrm{P}(\boldsymbol{N}^1,\ldots,\boldsymbol{N}^T|\epsilon)}\frac{\mathrm{P}(\boldsymbol{N}^1,\ldots,\boldsymbol{N}^T|\boldsymbol{\epsilon})\mathrm{P}(\boldsymbol{\epsilon})}{\mathrm{P}(\boldsymbol{N}^1,\ldots,\boldsymbol{N}^T)}\,d\boldsymbol{\epsilon} \\
&= \int \mathrm{P}(\boldsymbol{\Lambda}|\boldsymbol{\epsilon},\boldsymbol{N}^1,\ldots,\boldsymbol{N}^T)\mathrm{P}(\boldsymbol{\epsilon}|\boldsymbol{N}^1,\ldots,\boldsymbol{N}^T)\,d\boldsymbol{\epsilon} \qquad (6.4)
\end{aligned}$$

where $\mathrm{P}(\boldsymbol{\epsilon}|\boldsymbol{N}^1,\ldots,\boldsymbol{N}^T) = \int \mathrm{P}(\boldsymbol{\Lambda},\boldsymbol{\epsilon}|\boldsymbol{N}^1,\ldots,\boldsymbol{N}^T)\,d\boldsymbol{\Lambda}$. The observed (incomplete data) log posterior $\log \mathrm{P}(\boldsymbol{\Lambda}|\mathcal{D})$ has lower bound proportional to

$$\log\left(\int \exp\left\{\mathrm{E_q}\left(\log \mathrm{P}(\boldsymbol{\Lambda}|\boldsymbol{\epsilon},\boldsymbol{N}^1,\ldots,\boldsymbol{N}^T)\right)\right\} \exp\left\{\mathrm{E_q}\left(\log\int \mathrm{P}(\boldsymbol{\Lambda},\boldsymbol{\epsilon}|\boldsymbol{N}^1,\ldots,\boldsymbol{N}^T)\,d\boldsymbol{\Lambda}\right)\right\} d\boldsymbol{\epsilon}\right)$$

which is tight when $\mathrm{q} = \mathrm{P}(\boldsymbol{N}^1,\ldots,\boldsymbol{N}^T|\mathcal{D},\boldsymbol{\Lambda})$, as shown in (6.9) in the Appendix.

However, marginalizing the joint posterior $\int \mathrm{P}(\boldsymbol{\Lambda},\boldsymbol{\epsilon}|\boldsymbol{N}^1,\ldots,\boldsymbol{N}^T)\,d\boldsymbol{\Lambda}$ is often not feasible, so instead it is popular to use empirical Bayes to approximate it with a point-estimate.

We propose an empirical Bayes approach to maximizing the log posterior as an alternative to maximization of (6.4), $\hat{\boldsymbol{\epsilon}} = \arg\max_{\boldsymbol{\epsilon}} \mathrm{P}(\boldsymbol{\epsilon}|\boldsymbol{N}^1,\ldots,\boldsymbol{N}^T)$. This empirical Bayes approximation can be embedded in the EM algorithm so that once we have an estimate for $\boldsymbol{\epsilon}$, an estimator for $\boldsymbol{\Lambda}$ is obtained by maximizing the expected log conditional posterior $\mathrm{E_q}\left(\log \mathrm{P}(\boldsymbol{\Lambda}|\hat{\boldsymbol{\epsilon}},\boldsymbol{N}^1,\ldots,\boldsymbol{N}^T)\right)$.

Theorem 6.3.1. *Using the time independence in Proposition 4 and the empirical Bayes approximation, the E-step of the EM algorithm for the hierarchal model is*

$$\hat{N}_{ij}^{t_1} = \mathrm{E}(N_{ij}^{t_1} | \boldsymbol{R}^{t_1}, \hat{\boldsymbol{\Lambda}}^k) = \frac{\hat{\Lambda}_{ij}^k}{\sum_{j=1}^P \hat{\Lambda}_{ij}^k} R_i^{t_1}$$

$$\hat{N}_{ij}^{t_2} = \mathrm{E}(N_{ij}^{t_2} | \boldsymbol{C}^{t_2}, \hat{\boldsymbol{\Lambda}}^k) = \frac{\hat{\Lambda}_{ij}^k}{\sum_{i=1}^P \hat{\Lambda}_{ij}^k} C_j^{t_2},$$

$$\hat{N}_{ij}^{t_3} = \mathrm{E}(N_{ij}^{t_3} | \boldsymbol{F}^{t_3}, \hat{\boldsymbol{\Lambda}}^k) = \frac{\hat{\Lambda}_{ij}^k}{\sum_{ij} \hat{\Lambda}_{ij}^k} F_h^{t_3} \textit{ for any pair } ij,$$

and the M-step is

$$\begin{aligned}
\hat{\epsilon}_{ij}^{k+1} &= \arg\max_{\epsilon_{ij}} \sum_{\tau=1}^{3} \sum_{t_\tau=1}^{T} \log \frac{\Gamma(\hat{N}_{ij}^{t_\tau} + \epsilon_{ij}\Lambda_{0\,ij} + 1)}{\Gamma(\epsilon_{ij}\Lambda_{0\,ij} + 1)} \\
&\quad + \sum_{\tau=1}^{3} \sum_{t_\tau=1}^{T} (\epsilon_{ij}\Lambda_{0\,ij} + 1) \log \frac{\epsilon_{ij}}{1+\epsilon_{ij}} - \hat{N}_{ij}^{t_\tau} \log(1+\epsilon_{ij}) \\
&\quad \textit{and} \\
\hat{\Lambda}_{ij}^{k+1} &= \arg\max_{\Lambda_{ij}} (\hat{\epsilon}_{ij}\Lambda_{0\,ij}) \log(\Lambda_{ij}) - \hat{\epsilon}_{ij}\Lambda_{ij} - 3T\Lambda_{ij} \\
&\quad + \log(\Lambda_{ij}) \left(\sum_{t_1=1}^{T} \hat{N}_{ij}^{t_1} + \sum_{t_2=1}^{T} \hat{N}_{ij}^{t_2} + \sum_{t_3=1}^{T} \hat{N}_{ij}^{t_3} \right).
\end{aligned}$$

Since the function that lower bounds the observed log likelihood changes after every iteration of the EM algorithm, the prior should also change after every iteration. Intuitively, the earlier iterations of the EM algorithm will have expected log likelihoods that are more misspecified than the later iterations. This suggests spreading the prior distribution in the earlier iterations. The empirical Bayes approximation of Theorem 6.3.1 effectively does this by allowing the variance of the prior to be chosen using the data instead of fixing it as a constant. In this manner, the empirical Bayes approximation can be thought of as a Bayesian analog to the regularized EM algorithm of [136].

6.3.2 Warm Starting with Minimum Relative Entropy

The EM algorithm is well known to be sensitive to initialization, especially if the objective has a lot of local maxima. Thus if instead of a random initialization, the EM algorithm is warm-started, it is more likely to converge to a good maximum and also potentially converge faster. A good choice for an initialization point is a more robust estimator of the

rate matrix such as the solution to a model with fewer distributional assumptions. Thus instead of modeling an explicit generative model, we can instead adopt the minimum relative entropy (MRE) principle [137, 138, 60], and [61]. Geometrically, this reduces to an information projection of the prior distribution, as shown in Fig. 6.3.

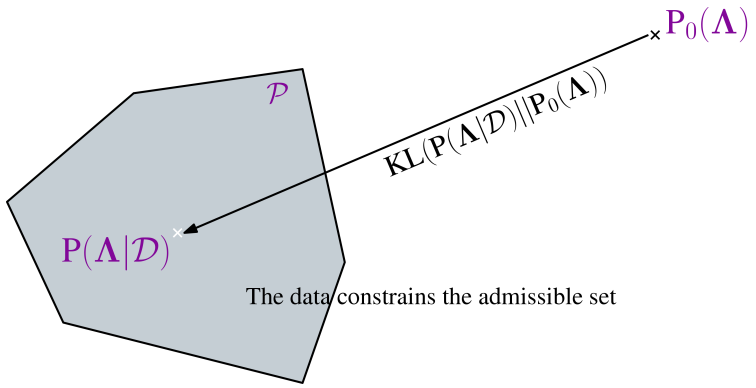


Figure 6.3: A projection of the prior, $\mathrm{P}_0(\mathbf{\Lambda})$, onto a feasible set $\mathcal{P}$ of distributions that satisfy the observed data, $\mathcal{D}$.

The constrained minimum relative entropy distribution is the density that is closest to a given prior distribution and lies in a feasible set, $\mathcal{P}$. This feasible set is formed from constraints that require their expected values, with respect to the minimum relative entropy distribution, to match properties of the observations, $\mathcal{D}$ (the total ingress, egress, and flows). And because relative entropy is the Kullback-Leibler (KL) divergence between probability distributions, this is used as the metric for closeness. This closeness criterion is well suited to the anomaly detection problem of interest to us because anomalous activity is rare, so the distribution of the actual rates, $\mathbf{\Lambda}$, should be similar to the prior distribution $\mathrm{P}_0(\mathbf{\Lambda}) = \mathrm{P}(\mathbf{\Lambda}|\mathbf{\Lambda}_0)$, which is parameterized by the baselines rates $\mathbf{\Lambda_0}$.

The MRE objective is

$$\min_{\mathrm{P}(\mathbf{\Lambda}|\boldsymbol{R},\boldsymbol{C},\boldsymbol{F})} \mathrm{KL}\left(\mathrm{P}(\mathbf{\Lambda}|\boldsymbol{R},\boldsymbol{C},\boldsymbol{F})||\mathrm{P}_0(\mathbf{\Lambda})\right)$$

$$\text{subject to}$$

$$\int \mathrm{P}(\mathbf{\Lambda}|\boldsymbol{R},\boldsymbol{C},\boldsymbol{F})(\mathbf{\Lambda}\mathbf{1} - \bar{\boldsymbol{R}})\, d\mathbf{\Lambda} = \mathbf{0}$$

$$\int \mathrm{P}(\mathbf{\Lambda}|\boldsymbol{R},\boldsymbol{C},\boldsymbol{F})(\mathbf{1}'\mathbf{\Lambda} - \bar{\boldsymbol{C}})\, d\mathbf{\Lambda} = \mathbf{0}$$

$$\int \mathrm{P}(\mathbf{\Lambda}|\boldsymbol{R},\boldsymbol{C},\boldsymbol{F})(\boldsymbol{A}\mathbf{\Lambda}\boldsymbol{B} - \bar{\boldsymbol{F}})\, d\mathbf{\Lambda} = \mathbf{0}$$

where $\mathbf{0}$ and $\mathbf{1}$ are vectors of zeros and ones respectively, $\bar{\boldsymbol{C}} = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{C}^t$ and $\bar{\boldsymbol{R}} = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{R}^t$ are the average rates of observed total traffic into and out of each node, and $\boldsymbol{A}$

and $\boldsymbol{B}$ are 0-1 matrices summing the rates that flow through each of the interior nodes with average observations $\bar{\boldsymbol{F}} = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{F}^t$. Using the Legendre transform of the Lagrangian to get the Hamiltonian, the optimal density has the form

$$\mathrm{P}(\boldsymbol{\Lambda}|\boldsymbol{R}, \boldsymbol{C}, \boldsymbol{F}) = \frac{\mathrm{P}_0(\boldsymbol{\Lambda})}{Z(\boldsymbol{\rho}, \boldsymbol{\gamma}, \boldsymbol{\phi})} \exp\left\{\boldsymbol{\rho}'(\boldsymbol{\Lambda}\mathbf{1} - \bar{\boldsymbol{R}}) + \boldsymbol{\gamma}'(\mathbf{1}'\boldsymbol{\Lambda} - \bar{\boldsymbol{C}}) + \boldsymbol{\phi}'(\boldsymbol{A}\boldsymbol{\Lambda}\boldsymbol{B} - \bar{\boldsymbol{F}})\right\} \quad (6.5)$$

where $\boldsymbol{\rho}, \boldsymbol{\gamma}, \boldsymbol{\phi}$ are Lagrange multipliers that maximize the negative log partition function $-\log(Z(\boldsymbol{\rho}, \boldsymbol{\gamma}, \boldsymbol{\phi}))$.

Proposition 5. *Let* $\mathrm{P}_0(\boldsymbol{\Lambda}) = \prod_{ij} \mathrm{P}_0(\Lambda_{ij})$ *be independent Laplace distributions with mean parameter* $\Lambda_{0\,ij}$ *and scale parameter* 1*, then the constrained mode of the MRE distribution is the solution to*

$$\underset{\boldsymbol{\Lambda}\in\mathbb{R}^+}{\arg\max} \; -||\boldsymbol{\Lambda} - \boldsymbol{\Lambda}_0||_1 + \hat{\boldsymbol{\rho}}'(\boldsymbol{\Lambda}\mathbf{1} - \bar{\boldsymbol{R}}) + \hat{\boldsymbol{\gamma}}'(\mathbf{1}'\boldsymbol{\Lambda} - \bar{\boldsymbol{C}})' + \hat{\boldsymbol{\phi}}'(\boldsymbol{A}\boldsymbol{\Lambda}\boldsymbol{B} - \bar{\boldsymbol{F}})$$

where $\hat{\boldsymbol{\rho}}, \hat{\boldsymbol{\gamma}}, \hat{\boldsymbol{\phi}} = \underset{\boldsymbol{\rho},\boldsymbol{\gamma},\boldsymbol{\phi}}{\arg\max} -\log(Z(\boldsymbol{\rho}, \boldsymbol{\gamma}, \boldsymbol{\phi}))$.

Maximizing the above expression over $\boldsymbol{\Lambda}$ (constrained to only positive real numbers) can be seen as a slight relaxation of the more direct objective of minimizing the loss function

$$\underset{\boldsymbol{\Lambda}\in\mathbb{R}^+}{\arg\min} ||\boldsymbol{\Lambda} - \boldsymbol{\Lambda}_0||_1 \quad \text{subject to} \quad \boldsymbol{\Lambda}\mathbf{1} = \bar{\boldsymbol{R}},\ \mathbf{1}'\boldsymbol{\Lambda} = \bar{\boldsymbol{C}},\ \boldsymbol{A}\boldsymbol{\Lambda}\boldsymbol{B} = \bar{\boldsymbol{F}} \quad (6.6)$$

where $||\cdot||_1$ is the element wise ℓ_1 norm. The loss function in (6.6) has the advantage that it can be easily implemented in any constrained convex solver such as CVX [94].

The objective in (6.6) is an easily interpretable formulation for estimating the rate matrix, which does not depend on the unobserved traffic N_{ij}^t. And, because it does not put distributional assumptions on the "likelihood", it is more robust to model mismatch, at the cost of accuracy. The generality of the solution to (6.6), while not precise enough on its own, makes it a good candidate to be further refined by the EM algorithm in the Hierarchical Poisson model.

6.4 Testing For Anomalies

Since the estimators in the previous section are maximizers of probabilistic models, a natural way to test for anomalies in the rate matrix $\boldsymbol{\Lambda}$ is to compare goodness of fit of the fitted model using hypothesis testing. By testing the null hypothesis $\mathrm{vec}(\boldsymbol{\Lambda}) = \mathrm{vec}(\boldsymbol{\Lambda}_0)$

against the alternative hypothesis $\text{vec}(\mathbf{\Lambda}) \neq \text{vec}(\mathbf{\Lambda}_0)$, we can control the false positive rate (FPR) (Type 1 error), of incorrectly declaring anomalous activity in the rate matrix, using a level-α test. In this section we will represent a statistical model with the notation $\mathcal{M}(\cdot)$, as the results apply for both log likelihood and log posterior models.

Depending on if the statistical models are likelihoods or posteriors, the statistic

$$\psi = -2 \sum_{t=1}^{T} \Big(\log(\mathcal{M}_t(\mathbf{\Lambda}_0)) - \log(\mathcal{M}_t(\hat{\mathbf{\Lambda}})) \Big) \tag{6.7}$$

would be either a log likelihood ratio (LR) statistic or a log posterior density ratio (PDR) statistic [139] respectively, where $\hat{\mathbf{\Lambda}} = \underset{\mathbf{\Lambda} \in \mathbb{R}^+}{\arg\max}\, \mathcal{M}(\mathbf{\Lambda})$. Thus testing ψ against a threshold can be seen as a generalized log likelihood ratio test or generalized log posterior ratio test with a composite alternative hypothesis.

Proposition 6. *Under the standard regularity conditions for the log LR statistic or under the sufficient conditions of the Bernstein-von Mises theorem for the log PDR statistic, ψ will be asymptotically $\chi^2_{P^2-P}$ distributed under the null hypothesis.*

Next we show that the statistic ψ in (6.7) is a good estimator of the KL divergence between the true model at its maximum and the true model at the baseline. And even if the models are misspecified, the statistic

$$\hat{\psi} = -2 \sum_{t=1}^{T} \Big(\log(\hat{\mathcal{M}}_t^k(\mathbf{\Lambda}_0)) - \log(\hat{\mathcal{M}}_t^k(\hat{\mathbf{\Lambda}})) \Big)$$

can still be a good estimator for goodness-of-fit, where the k in $\hat{\mathcal{M}}^k(\mathbf{\Lambda}_0)$ and $\hat{\mathcal{M}}^k(\hat{\mathbf{\Lambda}})$ indicates the iteration of the EM algorithm.

Proposition 7. *The statistic ψ/T is a consistent estimator for*

$$\Psi = 2\,\text{KL}\left(\mathcal{M}(\mathbf{\Lambda}^*) || \mathcal{M}(\mathbf{\Lambda}_0)\right),$$

the KL divergence between the true model and the true model under the null hypothesis. The statistic $\hat{\psi}/T$ is a consistent estimator for

$$2\,\text{KL}\left(\mathcal{M}(\mathbf{\Lambda}^*) || \mathcal{M}(\mathbf{\Lambda}_0)\right) - 2\left(\text{KL}(\mathcal{M}(\mathbf{\Lambda}^*) || \hat{\mathcal{M}}^k(\hat{\mathbf{\Lambda}}^*)) - \text{KL}(\mathcal{M}(\mathbf{\Lambda}_0) || \hat{\mathcal{M}}^k(\mathbf{\Lambda}_0))\right) \tag{6.8}$$

where $\hat{\mathcal{M}}^k(\hat{\mathbf{\Lambda}}^*)$ is the closest population local maximum at iteration k.

The second term in (6.8) can be seen as the difference between the true model misspecification error and the model misspecification error of the null hypothesis. So if conditions are satisfied so that the EM algorithm converges to the global maximum as the number of iterations $k \to \infty$ or if the model is equally as misspecified under the truth as under the null hypothesis such that the differences in the second term in (6.8) cancel to 0, then the statistic $\hat{\psi}/T$ is also a consistent estimator of Ψ. The justification for using misspecified models can also be geometrically interpreted as follows. Because the models estimated from the EM algorithm are from the correct parametric family of distributions, the misspecified models still lie on the same Riemannian manifold as the correct models. Below, we provide an algorithm for performing hypothesis testing on the statistic $\hat{\psi}$.

Algorithm 6.1 Anomaly Test

Input: models $\hat{\mathcal{M}}_1^k, \ldots, \hat{\mathcal{M}}_T^k$, critical value $c = F^{-1}(\alpha)$
where F is $\chi^2_{p^2-p}$ CDF, α is test level
Solve $\hat{\mathbf{\Lambda}} = \underset{\mathbf{\Lambda} \in \mathbb{R}^+}{\arg\max} \sum_{t=1}^T \log(\hat{\mathcal{M}}_t^k(\mathbf{\Lambda}))$
$\hat{\psi} = -2 \sum_{t=1}^T \Big(\log(\hat{\mathcal{M}}_t^k(\mathbf{\Lambda}_0)) - \log(\hat{\mathcal{M}}_t^k(\hat{\mathbf{\Lambda}})) \Big)$
if $\hat{\psi} > c$ **then**
Reject $\text{vec}(\mathbf{\Lambda}) = \text{vec}(\mathbf{\Lambda}_0)$
else
Do not reject $\text{vec}(\mathbf{\Lambda}) = \text{vec}(\mathbf{\Lambda}_0)$
end if
Return: Reject or Not

Algorithm 1 calculates the statistic $\hat{\psi}$ as a log ratio of the modes of the model under the null and alternative hypothesis. It then tests $\hat{\psi}$ against a critical value c, which is related to the false positive level.

Under the null hypothesis, the statistic $\hat{\psi}$ can be decomposed as sampling error $-2 \sum_{t=1}^T \log \hat{\mathcal{M}}_t^k(\hat{\mathbf{\Lambda}}^*) - \max_{\mathbf{\Lambda} \in \mathbb{R}^+} \log \hat{\mathcal{M}}_t^k(\mathbf{\Lambda})$ plus model error $-2 \sum_{t=1}^T \log \hat{\mathcal{M}}_t^k(\mathbf{\Lambda}_0) - \log \hat{\mathcal{M}}_t^k(\hat{\mathbf{\Lambda}}^*)$. Thus for the level-$\alpha$ test $\text{P}(\hat{\psi} > c | \mathcal{H}_0) = \alpha$, a Type-I error can occur due to either sampling error or model error or a combination of both. Since typically the finite sample distribution of the statistic ψ is unknown, the asymptotic distribution described in Proposition 6 can be used to choose the critical value c of $\text{P}(\psi > c | \mathcal{H}_0) = \alpha$. Assuming the model error is small, or small relative to the sampling error, we can also use Proposition 6 to choose the critical value of a test with a misspecified statistic $\text{P}(\hat{\psi} > c | \mathcal{H}_0) = \alpha$. In the following section, we will show in simulations that the asymptotic distribution of the correct statistic ψ is adequate for choosing the critical value of a test using the misspecified statistic $\hat{\psi}$.

6.5 Computational Complexity

In Algorithm 2, we present our hierarchical Poisson EM model warm started at the MRE estimator and analyze its computational complexity.

Algorithm 6.2 HP-MRE

Input: observations $\mathcal{D} = \{\boldsymbol{R}^t, \boldsymbol{C}^t, \boldsymbol{F}^t\}_{t=1}^T$, test level α
Initialize: $\hat{\boldsymbol{\Lambda}}$ as the solution to (6.6)
repeat
  E-Step: Calculate $\hat{N}_{ij}^{t_1}, \hat{N}_{ij}^{t_2}, \hat{N}_{ij}^{t_3}$ for all i, j in Theorem 6.3.1
  M-Step: Solve for $\hat{\epsilon}_{ij}^{k+1}$ and $\hat{\Lambda}_{ij}^{k+1}$ for all i, j in Theorem 6.3.1
until convergence
Test: Calculate $\hat{\psi}$ and reject if it is greater than critical value c
Return: Reject or Not

Warm starting the EM algorithm at the MRE solution (6.6) requires using interior-point methods, which have polynomial complexity in the number of variables. Since the MRE objective has P^2 linear variables and $2P^2$ second order cone problem variables, the computational cost is of order $\mathcal{O}(\#IPiter(3P^2)^r)$ where r is the polynomial degree (often 3) and $\#IPiter$ is the number of iterations of the interior point algorithm.

The E-Step consists of calculating the multinomial means using the observed data. Assume that the number of flows in the interior nodes are roughly P, so that each of the row sums, column sums, and interior node flows are the summation of P values. Then for each independent time instance t_τ, there are P summations of P values in denominator and a multiplication and division operation on each of the P^2 entries in the numerator. The total computational cost of the E-step is of order $\mathcal{O}(\tau TP^2)$ where τ is the number of different time points in Proposition 4 (2 + number of interior nodes).

In the M-step, the estimator $\hat{\epsilon}_{ij}^{k+1}$ can only be solved numerically because the score function of the negative binomial distribution is a non-linear equation. Because we can derive the gradient of the score function, we can use a trust-region method with a Newton conjugate gradient subproblem (each subproblem has linear complexity in time points). Given $\hat{\epsilon}_{ij}^{k+1}$, the estimator $\hat{\Lambda}_{ij}^{k+1}$ can be solved in closed form (6.3) with scalar operations, making its complexity linear in time points. Thus the total computational cost of the M-step is of order $\mathcal{O}((1 + \#CGiter)TP^2)$ where $\#CGiter$ is the number of conjugate gradient iterations.

Given the final iterations EM estimators, evaluating the models at each i, j entry only involves scalar operations, and getting the log ratio statistic $\hat{\psi}$ requires summing over all i, j entries and the T time points; so the total complexity of the anomaly test statistics

is of order $\mathcal{O}(TP^2)$. Thus, overall Algorithm 2 has computational complexity of order $\mathcal{O}(\#IPiter(3P^2)^r + \#EMiter((\tau+1)TP^2 + \#CGiterTP^2))$. Note that our choice in algorithms for the numerical optimizations were based more on convenience (using popular standard packages e.g. CVX, Matlab's fsolve) than optimal performance, so the computational complexities listed in this section are certainly not the best case scenarios. Nonetheless, even using non-optimal numerical algorithms, we show, in the following section, that our method can run in a reasonable amount of time in both simulations and large real world problems.

6.6 Simulation and Data Examples

In this section, we model network traffic in both simulated and real datasets as hierarchical Poisson posteriors to get estimators of the true network traffic rates. These estimators, from the hierarchical Poisson posteriors where the EM algorithm is initialized randomly or at the MRE estimator (Rand-HP or MRE-HP), are tested against baseline rates to detect anomalous activity in the network, as shown in Algorithm 1. We compare the performance of our proposed models to the maximum likelihood EM (MLEM) model of [115] (with the same time independence assumptions of Proposition 4 for feasibility), the Traffic and Anomaly Map (TA-Map) method of [120], and an "Oracle" that unrealistically observes the network directly. The "Oracle" estimator is the uniformly minimum variance unbiased estimator and achieves the Cramer-Rao lower bound [140].

The Traffic and Anomaly Map method is the state-of-the-art for estimating the rates in networks with traffic anomalies. Specifically for the TA-Map method we use the objective of (P1) in [120], but with the low rank decomposition of (P4) in [120] where $\boldsymbol{X} = \boldsymbol{L}\boldsymbol{Q}'$ and $\boldsymbol{Q} = \mathbf{1}$ is a vector of ones because the rates do not change over time. Since the anomalies also do not change over time, they can be expanded as $\boldsymbol{A}\boldsymbol{Q}'$ where $\boldsymbol{A}$ is a $P^2 \times 1$ vector of rates of anomalous activity. We use $\boldsymbol{\Lambda}_0$ to form the routing matrix for the vector of nominal rates $\boldsymbol{L}$ and a full routing matrix for the vector of anomalous rates $\boldsymbol{A}$ since we do not know any structural knowledge about them. Additionally, converting the notation of [120] to the notation of this paper, $\boldsymbol{Y} = [\boldsymbol{C}, \boldsymbol{R}, \boldsymbol{F}]$, $\boldsymbol{Z}_\Pi$ are defined as the edges that are observed, and $\boldsymbol{L} + \boldsymbol{A} = \text{vec}(\boldsymbol{\Lambda})$, where $\boldsymbol{L}$ and $\boldsymbol{A}$ are solved using CVX on (P1) in [120]. We empirically choose the penalty parameters $\lambda_\star = 0.5$ and $\lambda_1 = 0.1$.

6.6.1 Simulation Results

We simulate networks where the baseline rate matrix has 10 exterior nodes and 2 interior nodes. The probability of an edge between any two nodes in the baseline network is 0.65, the baseline rates $\Lambda_{0\,ij}$ are drawn from $Gamma(1.75, 1)$ distributions, and each interior node observes the total flow of a random 7 edges. We consider scenarios where anomalous activity can take place in either the edges or the nodes. In the first scenario, the anomalous activity can cause increases in the rates of some of the edges, new edges to appear or disappear, or both. So, the rates of anomalous activity $\Lambda_{ij} - \Lambda_{0\,ij}$ are drawn from $Gamma(0.75, 1)$ distributions where the probability of anomalous activity between any two nodes is 0.2. In the second scenario, there is a hidden node that is interacting with the other nodes, thus affecting the observed total flows of the known nodes. So the entries of the true rate matrix are drawn from $Gamma(1.75, 1)$ distributions, but the true rate matrix has 11 exterior nodes and the baseline rate matrix is the 10×10 submatrix of known nodes. Like in the first scenario, the probability of an edge between the hidden node and another node is 0.2. All simulations contain 200 trials, with anomalous activity in approximately half of them.

In Fig. 6.4 we explore the accuracy of correctly identifying anomalous activity as a function of the percentage of observed edges, where we observe $T = 100$ time points (samples). We measure accuracy as $\frac{\#TP+\#TN}{\#Trials}$ where the number of true positives (TP) and true negatives (TN) are the number of times a method correctly detects that there is anomalous activity or no anomalous activity respectively. For the probabilistic models (MLEM, Rand-HP, MRE-HP) , we use the likelihood or posterior density ratio tests described in Section 6.4 where the critical value is calculated using the inverse cumulative distribution function of the $\chi^2_{P^2-P}$ distribution at 0.05. The Traffic and Anomaly Map method uses a threshold on the maximum (absolute) value of the anomaly matrix $\boldsymbol{A}$ where the threshold is chosen so that it has 0.05 Type-I error. While the accuracy of all the probabilistic models increases as the percentage of observed edges increases, the MLEM has low accuracy unless over 80% of the network is observed whereas the two Hierarchical Poisson models have high accuracy even when no part of the network is directly observed. The TA-Map method also has poor performance at all percentages of the network observed. This may due to issues the TA-Map method has at separating $\boldsymbol{L}$ and $\boldsymbol{A}$ into the correct separate matrices even when the total estimator $\boldsymbol{L} + \boldsymbol{A}$ is accurate.

While the Rand-HP and MRE-HP models have approximately the same accuracy at detecting anomalies (MRE-HP does slightly better when only a few of the edges are observed), initializing the EM algorithm of the Hierarchical Poisson model at the MRE solution has additional benefits. Fig. 6.5 shows that the EM algorithm in the Hierarchical

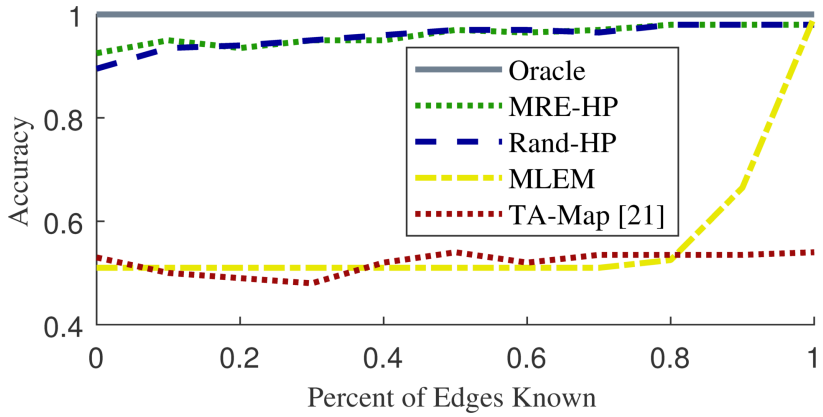


Figure 6.4: The network has 10 exterior nodes, 2 interior nodes, 35% sparsity, and a 0.5 probability of having anomalous activity, where $T = 100$ samples are observed. The accuracy of correctly detecting if the network has anomalous activity increases as the number of edges observed increases. The proposed Rand-HP, and MRE-HP models outperform the state-of-the-art TA-Map anomaly detector.

Poisson model with random initialization takes longer to converge than if it is initialized at the MRE solution. This is because, if the EM algorithm is initialized in a place where likelihood is very noisy, it may have difficultly deciding on the best of the nearby local maxima, but the MRE solution is often already close to a good local maximum.

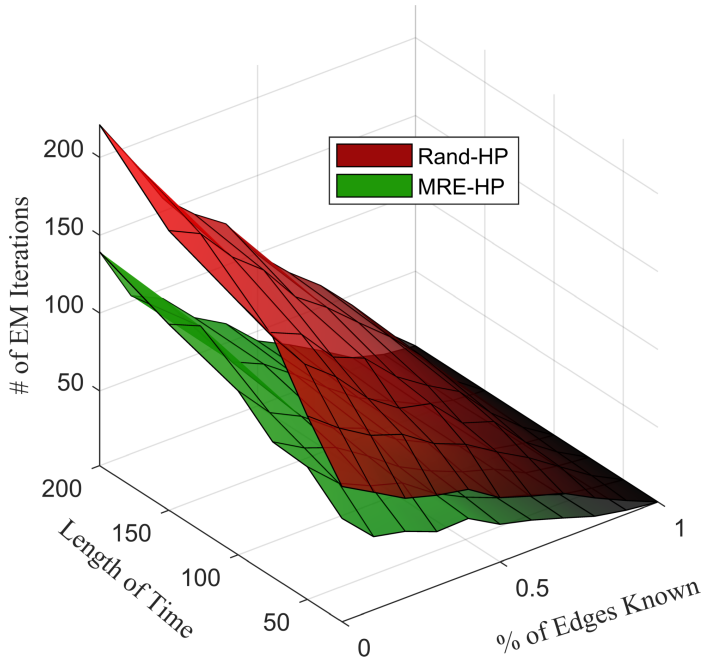


Figure 6.5: The number of iterations required for the EM algorithm to converge as the observation time and number of edges observed vary. By warm-starting the EM algorithm at the MRE estimator, the number of iteration is much fewer everywhere because it is already close to a good local maximum.

Fig. 6.6 shows the mean squared error (MSE) of the estimated rate matrices $||\hat{\mathbf{\Lambda}} - \mathbf{\Lambda}||_F^2$. The MRE-HP model gains some of the advantages of the MRE estimator making its MSE much lower than that of the Rand-HP model. As the percentage of observed edges in the network increases, all estimators' errors decrease to the Oracle estimator's error, which is the lowest possible MSE among all unbiased estimators. However, both the TA-Map method and the MLEM model do not have good performance except when almost all of the network is observed, at which point every estimator performs well. Note that estimating the traffic is not the end goal in the considered anomaly detection problem. We demonstrate this by comparing Fig. 6.6 to Fig. 6.4, where we can see that estimating the traffic well (having low MSE) does not guarantee the method high accuracy. Low MSE implies that a method's estimates do not have a large difference with the true rates, however depending on where the differences occur, it can be enough to cause the method to incorrectly detect anomalous activity.

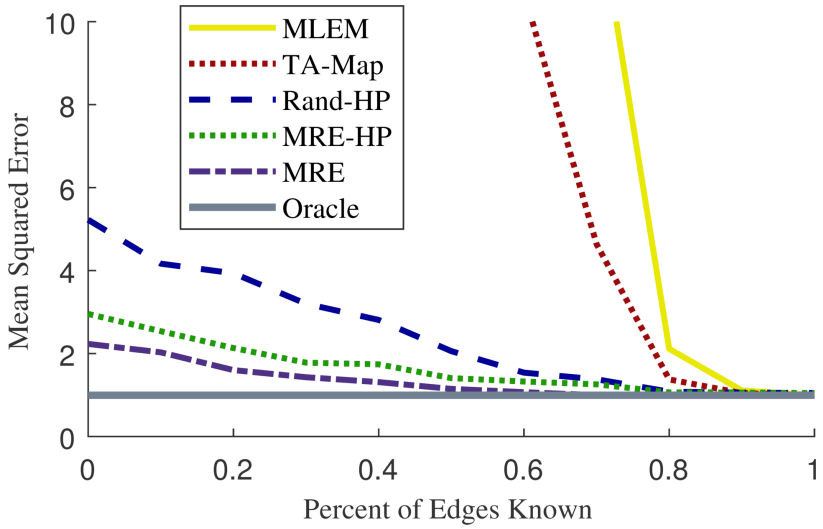


Figure 6.6: The MSE decreases as the number of edges observed increases. The proposed MRE, Rand-HP, and MRE-HP models outperform the state-of-the-art TA-Map method.

Fig. 6.7 shows the ROC curves of the anomaly detection performance of the MRE-HP, MLEM, and TA-Map methods for both the anomalous rates and the hidden node scenarios, where only 20% of the edges are observed. The accuracy of the MRE-HP model increases with the total observation time T, and it can detect anomalous activity almost perfectly with only 100 time points, as evidenced by its area under the curve (AUC) being very close to 1. The stars over the lines are the FPR vs TPR when using the critical values found by calculating the inverse cumulative distribution function of the $\chi^2_{P^2-P}$ distribution at 0.05. The ROC curve for testing a misspecified LR test statistic using the MLEM is just the point at $(1, 1)$ because the Poisson MLE model is so misspecified, it always rejects the null

hypothesis. The TA-Map method, while it does not always rejects the null hypothesis like the MLEM model, performs about as bad as random guessing (a diagonal line from $(0, 0)$ to $(1, 1)$). These results are consistent with the accuracy results shown in Fig. 6.4.

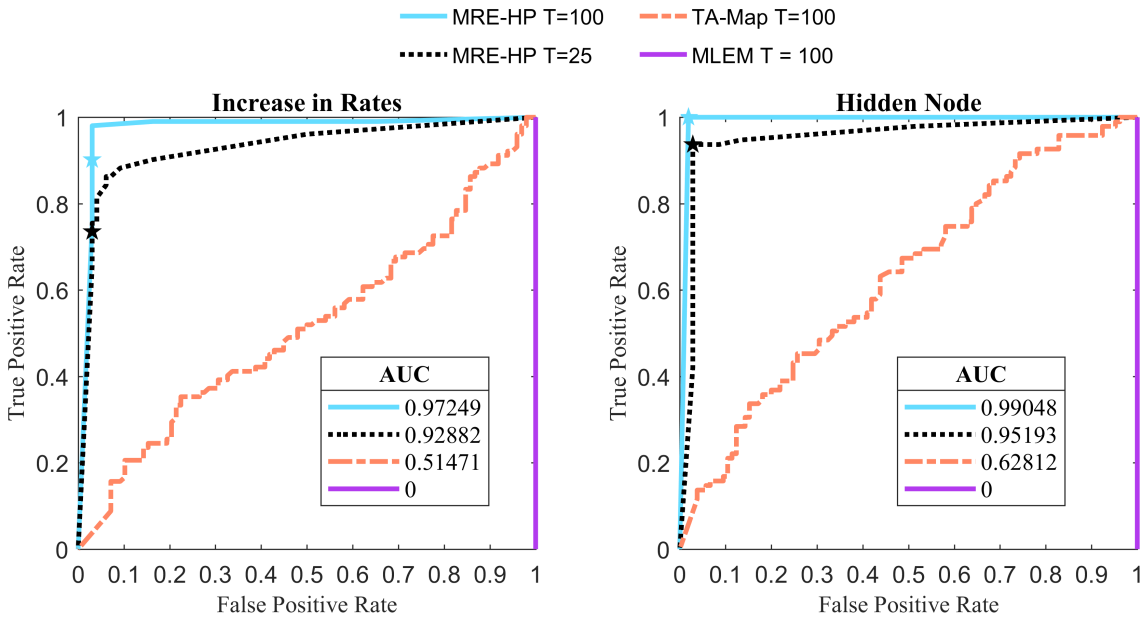


Figure 6.7: ROC curves where 20% of edges in the network are observed and roughly half of the networks have anomalous activity. The proposed MRE-HP model can detect anomalous activity almost perfectly while the TA-Map and MLEM methods have poor performance.

In Table 6.1, we show the corresponding CPU timings of each method in the two scenarios used in Fig. 6.7. The algorithms were run on an Intel Xeon E5-2630 processor at 2.30GHz without any explicit parallelization; however some of the built-in Matlab functions are by default multi-threaded (such as ones that call BLAS or LAPACK libraries). While the MRE-HP is slower than the competing methods, its computation time is still very fast and on average less than half a minute. Also, note the significant performance improvement provided by MRE-HP in the considered anomaly detection problem (see Fig. 6.4 and Fig. 6.7).

Table 6.1: Fig. 6.7 CPU Times (in seconds) over 200 Trials

	Increase in Rates		Hidden Node	
	Average	Standard Dev.	Average	Standard Dev.
MRE-HP	18.594	28.611	18.901	30.785
MLEM	0.0398	0.0112	0.0380	0.0119
TA-Map	3.1860	0.1347	3.1861	0.1912

6.6.2 CTU-13 Dataset

The proposed model was applied to botnet traffic networks from the CTU-13 dataset, which come from 13 different scenarios of botnets executing malware attacks captured by CTU University, Czech Republic, in 2011 [98]. The dataset contains real botnet traffic mixed with normal traffic and background traffic and the authors of [98] processed the captured traffic into bidirectional NetFlows and manually labeled them. Because the objective is to detect if there is botnet traffic among the regular users, we will only use the sub-network of nodes that are being used for normal traffic, but the traffic on this sub-network can be of any type: normal, background, or botnet. Thus, baseline traffic on the network is either normal or background traffic and the anomalous traffic is from botnets. And because the botnet traffic originates and also potentially ceases from nodes that are not the regular users, the anomalous activity is due to unobserved hidden nodes.

The observations consist of the total ingress and egress of each node along with the total flows of 10 interior nodes, where each interior node receives flow from $0.7P$ other nodes, in addition to observing 20% of the edges in the network. An observation or sample is all the traffic that occurs in a one-hour time period. For each of the scenarios, we test the probabilistic models at an alpha level of 0.05 under both regimes where the null hypothesis is true (no botnet traffic) and not true (botnet traffic). For the TA-Map method of [120], we use the ROC curves from the simulations to choose the threshold that yields a Type-I error equal to 0.05. Table 6.2 summarizes the characteristics of each of the 13 difference scenarios.

Table 6.3 shows that the Hierarchical Poisson model initialized at the MRE solution always correctly rejects the null hypothesis when it is not true. However, the model incorrectly rejects the null hypothesis in Scenario 3. This scenario has far more nodes than any of the other scenarios, and as the number of nodes increase, the number of entries that must be estimated, $\mathcal{O}(P^2)$, vastly outweigh the number of observations, $\mathcal{O}(P)$. This gives rise to a large model misspecification error in this scenario, which would negatively impact the accuracy of Algorithm I. Like in the simulations, the Poisson MLE model always rejects the null hypothesis due to its massive model misspecification error and the TA-Map method also has poor performance in the scenarios that are computationally feasible for the method (the ones marked NA are too computationally expensive). Overall MRE-HP has good performance detecting anomalous activity, especially compared to the other methods.

Table 6.2: CTU Network Characteristics

Scenario	Time T (Hours)	# of Nodes P	# of Edges Normal Traffic	# of Hidden Nodes	# of Edges Botnet Traffic
1	7	510	1566	2280	4428
2	6	114	249	283	337
3	68	333	977	2463	2466
4	5	414	1737	9	27
5	2	246	652	59	67
6	3	200	380	2	5
7	2	93	161	11	14
8	20	3031	8799	57	106
9	6	485	1799	706	3372
10	6	260	1088	25	131
11	1	53	162	7	19
12	2	290	697	861	1829
13	17	272	814	267	345

Table 6.3: CTU Network Test

Scenario	When $\mathcal{H}_0$ is True			When $\mathcal{H}_A$ is True		
	MRE-HP	MLE	TA-Map	MRE-HP	MLE	TA-Map
1	✓	×	NA	✓	✓	NA
2	✓	×	×	✓	✓	✓
3	×	×	NA	✓	✓	NA
4	✓	×	NA	✓	✓	NA
5	✓	×	NA	✓	✓	NA
6	✓	×	NA	✓	✓	NA
7	✓	×	×	✓	✓	✓
8	✓	×	NA	✓	✓	NA
9	✓	×	NA	✓	✓	NA
10	✓	×	NA	✓	✓	NA
11	✓	×	×	✓	✓	✓
12	✓	×	NA	✓	✓	NA
13	✓	×	NA	✓	✓	NA

In Table 6.4, we show the CPU timings of the algorithms for the 13 scenarios in the CTU-13 dataset under both hypothesis, where the algorithms are run on the same processor

described in the simulations. Even for scenario 8, the computational times of MRE-HP are feasible despite running on a rather out-of-date processor with a low clock speed. Again we mark NA for the scenarios that are computationally infeasible for the TA-Map method (the memory requirements are above 32GB even for scenario 6). The MRE-HP method despite being slower than the TA-Map on smaller networks (see Table 6.1), scales much more efficiently to larger networks.

Table 6.4: CTU Network CPU Times (in seconds)

Scenario	When $\mathcal{H}_0$ is True			When $\mathcal{H}_A$ is True		
	MRE-HP	MLE	TA-Map	MRE-HP	MLE	TA-Map
1	513.72	25.381	NA	1303.2	62.148	NA
2	19.258	5.2586	426.69	206.71	1.0702	683.26
3	790.60	70.706	NA	468.39	55.564	NA
4	2038.0	10.834	NA	3607.5	30.863	NA
5	539.85	2.0568	NA	263.56	3.2602	NA
6	69.452	1.6095	NA	58.427	12.556	NA
7	10.164	0.8487	360.09	17.043	0.6761	366.83
8	62602	8071.2	NA	55591	2087.8	NA
9	5439.3	97.082	NA	903.46	51.762	NA
10	648.88	7.8440	NA	174.66	2.9925	NA
11	4.2550	0.7101	55.126	17.645	0.4735	56.367
12	1864.8	3.6154	NA	514.97	5.5562	NA
13	355.20	17.620	NA	792.81	76.237	NA

6.6.3 Taxi Dataset

The proposed model was applied to a dataset consisting of yellow and green taxicabs rides from the New York City Taxi and Limousine Commission (NYC TLC) [141] and [142]. For every NYC taxicab ride, the dataset contains the pickup and drop-off locations as geographic coordinates (latitude and longitude). Green taxicabs are not allowed to pickup passengers below West 110th Street and East 96th Street in Manhattan, but occasionally they risk the chance of getting punished and ignore the regulations. In an article on June 10th 2014, the New York Post explains how the city began hiring more TLC inspectors to catch illegal pickups and enforce the location rules [143]. Thus we are interested in identifying if there are green taxicabs operating in lower Manhattan when we only know the yellow taxicab network. We treat the 18 Neighborhood Tabulation Areas (NTA) in lower

Manhattan as nodes and associate any pickups or drop-offs within an NTA's boundaries as traffic entering or leaving the node. We form edges from only frequently occurring routes of traffic, which we define as having activity at least an average of every 20 minutes for yellow and twice a month for green taxicabs. For samples, we use the yellow and green taxicab rides from between January and May of 2014 and aggregate them into daily totals.

Like in the previous example, we indirectly observe samples of the total ingress and egress of each node, and the total flows of 10 interior nodes that each observe the flows of $0.7P$ nodes. This creates a total traffic network with $P = 18$ nodes and 187 non-zero edges (39% sparsity) where the baseline network (yellow taxicab rides) has 163 of the edges. There is anomalous activity (green taxicab rides) on 56 of the edges, where 32 of these edges are also in the baseline network and 24 are not. We observe the network for a total of $T = 150$ days. Fig. 6.8 shows the baseline network formed from yellow taxicab rides and the unknown anomalous activity due to illegal pickups from green taxicabs.

Table 6.5 shows, for different percentages of edges observed, whether the correct decision (reject or not) is made when the null hypothesis is true (no green taxi traffic) and when it is not true (green taxi traffic). The Hierarchical Poisson model initialized at the MRE solution always makes the correct decision while the Poisson MLE model, except for when the network can be directly observed, always rejects the null hypothesis. These two models are tested at an alpha level of 0.05. The Traffic and Anomaly Map method, which has a 0.05 Type-I error threshold chosen from the ROC curves of the simulations, also has poor performance.

Table 6.5: Taxi Network Test

%	When $\mathcal{H}_0$ is True			When $\mathcal{H}_A$ is True		
Edges	MRE-HP	MLE	TA-Map	MRE-HP	MLE	TA-Map
0	✓	×	✓	✓	✓	×
10%	✓	×	✓	✓	✓	×
20%	✓	×	✓	✓	✓	×
30%	✓	×	✓	✓	✓	×
40%	✓	×	✓	✓	✓	×
50%	✓	×	✓	✓	✓	×
60%	✓	×	×	✓	✓	×
70%	✓	×	×	✓	✓	✓
80%	✓	×	×	✓	✓	✓
90%	✓	×	×	✓	✓	✓
100%	✓	✓	×	✓	✓	✓

From the results of Table 6.5, we know the Hierarchical Poisson model initialized at the MRE solution is always able to detect changes in the network at a global scale, but we are also interested in the recovery of the individual green taxicab routes. When 70% of the network is observed, the model is able to detect 52 of the 56 edges that contain anomalous activity with only a 2% false positive rate. The 4 missed edges and 5 false alarms are shown in Fig. 6.9.

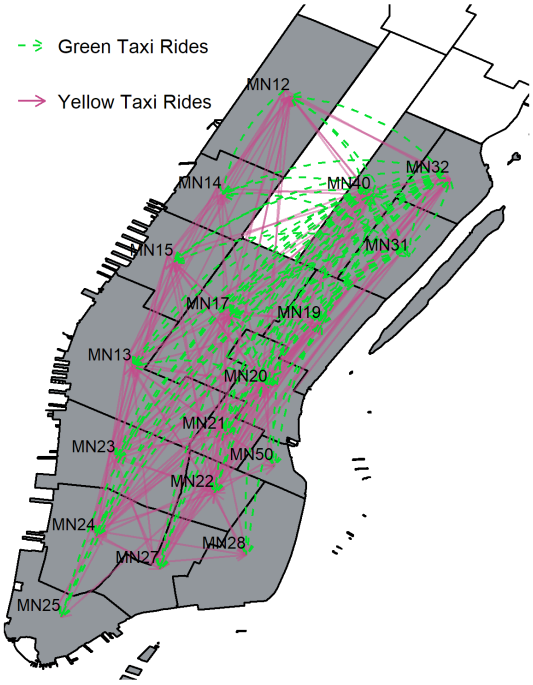


Figure 6.8: A network of taxicab rides in lower Manhattan where the nodes are the 18 NTAs. The traffic from yellow taxicab rides (solid purple lines) form the baseline network and the traffic from green taxicab rides (dashed green lines) are anomalous activity in the network.

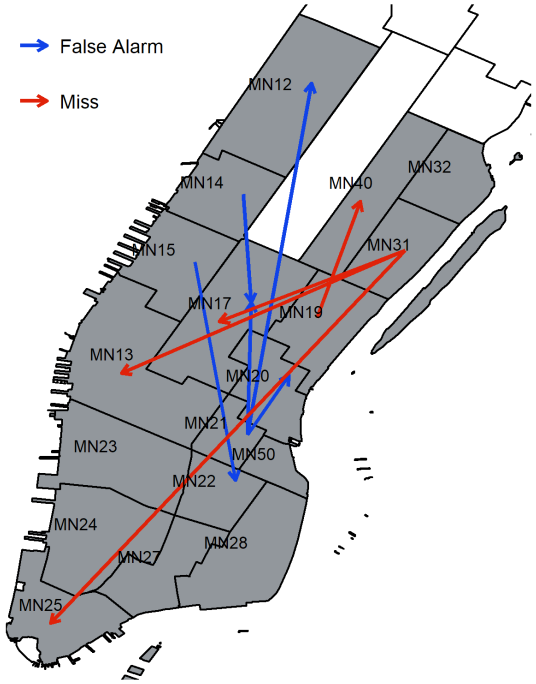


Figure 6.9: A miss (red line) is an edge that the MRE-HP model fails to identify as containing anomalous activity and a false alarm (blue line) is an edges that is incorrectly identified as containing anomalous activity. The majority of the misses depart from MN31 (Lenox Hill and Roosevelt Island), which may contain legal activity because green taxis are allowed to pick up passengers from Roosevelt Island.

Out of the 4 misses, 3 of them are from green taxicab pickups from MN31, which contains the Lenox Hill and Roosevelt Island areas. Green taxicabs are allowed to pick up passengers from Roosevelt Island, but not from Lenox Hill, so some of the traffic on these 3 routes could be legal and not anomalous activity. The other miss, from MN19 to MN40, only had 11 rides in 150 days, making it harder to distinguish from just perturbation noise in the samples.

6.7 Conclusion

We have developed a framework and a probabilistic model for detecting anomalous activity in the traffic rates of sparse networks. Our framework is realistic and robust in that, at minimum, it only requires observing the total egress and ingress of the nodes. Because it imposes no fixed assumptions of edge structure, our framework allows the estimator to handle noisy observations and anomalous activity. Our simulation results show the advantages of our model over competing methods in detecting anomalous activity. Through application of our models to the CTU-13 botnet datasets, we show that the model is scalable and robust to various scenarios, and with the NYC taxi dataset, we show an application of our model and framework to an already identified real-world problem.

Appendix

Proof of Proposition 4. By Jensen's inequality, $\log\left(\mathrm{P}(\mathcal{D}|\boldsymbol{\Lambda})\right)$

$$
\begin{aligned}
&= \log\left(\prod_{t_1=1}^{T}\mathrm{P}(\boldsymbol{R}^{t_1}|\boldsymbol{\Lambda})\prod_{t_2=1}^{T}\mathrm{P}(\boldsymbol{C}^{t_2}|\boldsymbol{\Lambda})\prod_{t_3=1}^{T}\mathrm{P}(\boldsymbol{F}^{t_3}|\boldsymbol{\Lambda})\right)\\
&\geq \sum_{t_1=1}^{T}\mathrm{E}_{\mathrm{q}^{t_1}}\left(\log\mathrm{P}(\boldsymbol{R}^{t_1},\boldsymbol{N}^{t_1}|\boldsymbol{\Lambda})\right)-\mathrm{E}_{\mathrm{q}^{t_1}}\left(\log\mathrm{q}(\boldsymbol{N}^{t_1})\right)+\sum_{t_2=1}^{T}\mathrm{E}_{\mathrm{q}^{t_2}}\left(\log\mathrm{P}(\boldsymbol{C}^{t_2},\boldsymbol{N}^{t_2}|\boldsymbol{\Lambda})\right)\\
&-\mathrm{E}_{\mathrm{q}^{t_2}}\left(\log\mathrm{q}(\boldsymbol{N}^{t_2})\right)+\sum_{t_3=1}^{T}\mathrm{E}_{\mathrm{q}^{t_3}}\left(\log\mathrm{P}(\boldsymbol{F}^{t_3},\boldsymbol{N}^{t_3}|\boldsymbol{\Lambda})\right)-\mathrm{E}_{\mathrm{q}^{t_3}}\left(\log\mathrm{q}(\boldsymbol{N}^{t_3})\right)\\
&=\sum_{t_1=1}^{T}\mathrm{E}_{\mathrm{q}^{t_1}}\left(\log\mathrm{P}(\boldsymbol{R}^{t_1}|\boldsymbol{N}^{t_1},\boldsymbol{\Lambda})\right)+\sum_{t_2=1}^{T}\mathrm{E}_{\mathrm{q}^{t_2}}\left(\log\mathrm{P}(\boldsymbol{C}^{t_2}|\boldsymbol{N}^{t_2},\boldsymbol{\Lambda})\right)\\
&+\sum_{t_3=1}^{T}\mathrm{E}_{\mathrm{q}^{t_3}}\left(\log\mathrm{P}(\boldsymbol{F}^{t_3}|\boldsymbol{N}^{t_3},\boldsymbol{\Lambda})\right)+\sum_{\tau=1}^{3}\sum_{t_\tau=1}^{T}\mathrm{E}_{\mathrm{q}^{t_\tau}}\left(\log\mathrm{P}(\boldsymbol{N}^{t_\tau}|\boldsymbol{\Lambda})\right)+\mathrm{H}\left(\mathrm{q}(\boldsymbol{N}^{t_\tau})\right)
\end{aligned}
$$

and $\mathrm{P}(\boldsymbol{R}^{t_1}|\boldsymbol{N}^{t_1},\boldsymbol{\Lambda})=\mathrm{P}(\boldsymbol{C}^{t_2}|\boldsymbol{N}^{t_2},\boldsymbol{\Lambda})=\mathrm{P}(\boldsymbol{F}^{t_3}|\boldsymbol{N}^{t_3},\boldsymbol{\Lambda})=1$. The inequality is tight (by KL divergence) when $\mathrm{q}(\boldsymbol{N}^{t_1})=\mathrm{P}(\boldsymbol{N}^{t_1}|\boldsymbol{R}^{t_1},\boldsymbol{\Lambda})$, $\mathrm{q}(\boldsymbol{N}^{t_2})=\mathrm{P}(\boldsymbol{N}^{t_2}|\boldsymbol{C}^{t_2},\boldsymbol{\Lambda})$, and $\mathrm{q}(\boldsymbol{N}^{t_3})=\mathrm{P}(\boldsymbol{N}^{t_3}|\boldsymbol{F}^{t_3},\boldsymbol{\Lambda})$ are multinomial distributions. □

Proof of Theorem 6.3.1. Define $\mathcal{N}=\{\boldsymbol{N}^{t_\tau}\colon \forall t_\tau=1,\ldots,T \text{ and } \tau=1,\ldots,3\}$ as the set of all network traffic at different time points t_τ for the entire sample window $1,\ldots,T$. So $\cap\mathcal{N}$ is the intersection of the set and $\mathrm{P}(\cap\mathcal{N})$ is its joint probability. By Jensen's inequality,

$$
\begin{aligned}
&\log \mathrm{P}(\boldsymbol{\Lambda}|\mathcal{D}) = \log \int \mathrm{P}(\boldsymbol{\Lambda}, \boldsymbol{\epsilon}|\mathcal{D})\, d\boldsymbol{\epsilon} = \log \int \frac{\mathrm{P}(\mathcal{D}|\boldsymbol{\Lambda}, \boldsymbol{\epsilon})\mathrm{P}(\boldsymbol{\Lambda}|\boldsymbol{\epsilon})\mathrm{P}(\boldsymbol{\epsilon})}{\mathrm{P}(\mathcal{D})}\, d\boldsymbol{\epsilon} \\
&= \log \int \left(\int \cdots \int \mathrm{P}\left(\mathcal{D}, \cap\mathcal{N}|\boldsymbol{\Lambda}, \boldsymbol{\epsilon}\right) \frac{\mathrm{P}(\boldsymbol{\Lambda}|\boldsymbol{\epsilon})\mathrm{P}(\boldsymbol{\epsilon})}{\mathrm{P}(\mathcal{D})} d\mathcal{N} \right) d\boldsymbol{\epsilon} \\
&= \log \int \mathrm{E}_{\mathrm{q}} \left(\frac{\mathrm{P}(\mathcal{D}, \cap\mathcal{N}|\boldsymbol{\Lambda}, \boldsymbol{\epsilon})}{\prod_{\tau=1}^{3}\prod_{t_\tau=1}^{T} \mathrm{q}(\boldsymbol{N}^{t_\tau})} \frac{\mathrm{P}(\boldsymbol{\Lambda}|\boldsymbol{\epsilon})\mathrm{P}(\boldsymbol{\epsilon})}{\mathrm{P}(\mathcal{D})} \right) d\boldsymbol{\epsilon} \\
&\geq \log \int \exp \left\{ \mathrm{E}_{\mathrm{q}} \left(\log \mathrm{P}(\mathcal{D}, \cap\mathcal{N}|\boldsymbol{\Lambda}, \boldsymbol{\epsilon}) + \log \frac{\mathrm{P}(\boldsymbol{\Lambda}|\boldsymbol{\epsilon})\mathrm{P}(\boldsymbol{\epsilon})}{\mathrm{P}(\mathcal{D})} - \sum_{\tau=1}^{3}\sum_{t_\tau=1}^{T} \log \mathrm{q}(\boldsymbol{N}^{t_\tau}) \right) \right\} d\boldsymbol{\epsilon} \\
&= \log \int \exp \left\{ \mathrm{E}_{\mathrm{q}} \left(\log \frac{\mathrm{P}(\mathcal{D}, \cap\mathcal{N}, \boldsymbol{\Lambda}|\boldsymbol{\epsilon})}{\mathrm{P}(\mathcal{D}, \cap\mathcal{N}|\boldsymbol{\epsilon})} \frac{\mathrm{P}(\mathcal{D}, \cap\mathcal{N}|\boldsymbol{\epsilon})\mathrm{P}(\boldsymbol{\epsilon})}{\mathrm{P}(\mathcal{D})} \right) \right\} d\boldsymbol{\epsilon} - \mathrm{E}_{\mathrm{q}} \left(\sum_{\tau=1}^{3}\sum_{t_\tau=1}^{T} \log \mathrm{q}(\boldsymbol{N}^{t_\tau}) \right) \\
&= \log \int \exp \left\{ \mathrm{E}_{\mathrm{q}} \left(\log \mathrm{P}(\boldsymbol{\Lambda}| \cap \mathcal{N}, \boldsymbol{\epsilon}) \right) + \mathrm{E}_{\mathrm{q}} \left(\log \mathrm{P}(\cap\mathcal{N}, \boldsymbol{\epsilon}|\mathcal{D}) \right) \right\} d\boldsymbol{\epsilon} + \sum_{\tau=1}^{3}\sum_{t_\tau=1}^{T} \mathrm{H}\left(\mathrm{q}(\boldsymbol{N}^{t_\tau}) \right) \\
&= \log \int \exp \left\{ \mathrm{E}_{\mathrm{q}} \left(\log \mathrm{P}(\boldsymbol{\Lambda}| \cap \mathcal{N}, \boldsymbol{\epsilon}) \right) + \mathrm{E}_{\mathrm{q}} \left(\log \mathrm{P}(\boldsymbol{\epsilon}| \cap \mathcal{N}) \right) \right\} d\boldsymbol{\epsilon} \qquad (6.9)
\end{aligned}
$$

where this bound is tight (by KL divergence) when $\mathrm{q} = \mathrm{P}(\cap\mathcal{N}|\mathcal{D}, \boldsymbol{\Lambda}, \boldsymbol{\epsilon}) = \prod_{\tau=1}^{3}\prod_{t_\tau=1}^{T} \mathrm{P}(\boldsymbol{N}^{t_\tau}|\boldsymbol{R}^{t_\tau}, \boldsymbol{\Lambda})\mathrm{P}(\boldsymbol{N}^{t_\tau}|\boldsymbol{C}^{t_\tau}, \boldsymbol{\Lambda})\mathrm{P}(\boldsymbol{N}^{t_\tau}|\boldsymbol{F}^{t_\tau}, \boldsymbol{\Lambda})$ are multinomial distributions. And, maximizing $\mathrm{E}_{\mathrm{q}} \left(\log \mathrm{P}(\cap\mathcal{N}, \boldsymbol{\epsilon}|\mathcal{D}) \right)$

$$
\begin{aligned}
&= \mathrm{E}_{\mathrm{q}} \left(\log \mathrm{P}(\mathcal{D}| \cap \mathcal{N}, \boldsymbol{\epsilon}) + \log \mathrm{P}(\cap\mathcal{N}, \boldsymbol{\epsilon}) - \log \mathrm{P}(\mathcal{D}) \right) \\
&= \mathrm{E}_{\mathrm{q}} \left(\log(1) + \log \mathrm{P}(\cap\mathcal{N}|\boldsymbol{\epsilon}) \right) + \log \mathrm{P}(\boldsymbol{\epsilon}) - \log \mathrm{P}(\mathcal{D}) \\
&= \log \mathrm{P}(\boldsymbol{\epsilon}) - \log \mathrm{P}(\mathcal{D}) + \sum_{\tau=1}^{3}\sum_{t_\tau=1}^{T} \mathrm{E}_{\mathrm{q}^{t_\tau}} \log \mathrm{P}(\boldsymbol{N}^{t_\tau}|\boldsymbol{\epsilon})
\end{aligned}
$$

is equivalent to maximizing a lower bound of $\log \mathrm{P}(\boldsymbol{\epsilon}|\mathcal{D})$

$$
\begin{aligned}
&= \log \frac{\prod_{t_1=1}^{T}\prod_{t_2=1}^{T}\prod_{t_3=1}^{T} \mathrm{P}(\boldsymbol{R}^{t_1}, \boldsymbol{C}^{t_2}, \boldsymbol{F}^{t_3}|\boldsymbol{\epsilon})\mathrm{P}(\boldsymbol{\epsilon})}{\mathrm{P}(\mathcal{D})} \\
&= \log \mathrm{P}(\boldsymbol{\epsilon}) - \log \mathrm{P}(\mathcal{D}) + \log \prod_{t=1}^{T} \mathrm{E}_{\mathrm{q}^{t_1}} \left(\frac{\mathrm{P}(\boldsymbol{R}^{t_1}, \boldsymbol{N}^{t_1}|\boldsymbol{\epsilon})}{\mathrm{q}^{t_1}(\boldsymbol{N}^{t_1})} \right) \\
&+ \log \prod_{t_2=1}^{T} \mathrm{E}_{\mathrm{q}^{t_2}} \left(\frac{\mathrm{P}(\boldsymbol{C}^{t_2}, \boldsymbol{N}^{t_2}|\boldsymbol{\epsilon})}{\mathrm{q}^{t_2}(\boldsymbol{N}^{t_2})} \right) + \log \prod_{t_3=1}^{T} \mathrm{E}_{\mathrm{q}^{t_3}} \left(\frac{\mathrm{P}(\boldsymbol{F}^{t_3}, \boldsymbol{N}^{t_3}|\boldsymbol{\epsilon})}{\mathrm{q}^{t_3}(\boldsymbol{N}^{t_3})} \right)
\end{aligned}
$$

$$
\begin{aligned}
&\geq \log \mathrm{P}(\boldsymbol{\epsilon}) - \log \mathrm{P}(\mathcal{D}) + \sum_{t_1=1}^{T} \mathrm{E}_{\mathrm{q}^{t_1}} (\log \mathrm{P}(\boldsymbol{R}^{t_1} | \boldsymbol{N}^{t_1}), \boldsymbol{\epsilon}) + \sum_{t_2=1}^{T} \mathrm{E}_{\mathrm{q}^{t_2}} \left(\log \mathrm{P}(\boldsymbol{C}^{t_2} | \boldsymbol{N}^{t_2}, \boldsymbol{\epsilon}) \right) \\
&+ \sum_{t_3=1}^{T} \mathrm{E}_{\mathrm{q}^{t_3}} \left(\log \mathrm{P}(\boldsymbol{F}^{t_3} | \boldsymbol{N}^{t_3}, \boldsymbol{\epsilon}) \right) + \sum_{\tau=1}^{3} \sum_{t^{(\tau)}=1}^{T} \mathrm{E}_{\mathrm{q}^{t_\tau}} \left(\log \mathrm{P}(\boldsymbol{N}^{t_\tau} | \boldsymbol{\epsilon}) \right) - \mathrm{E}_{\mathrm{q}^{t_\tau}} \left(\log \mathrm{q}(\boldsymbol{N}^{t_\tau}) \right) \\
&\propto \log \mathrm{P}(\boldsymbol{\epsilon}) - \log \mathrm{P}(\mathcal{D}) + \sum_{\tau=1}^{3} \sum_{t_\tau=1}^{T} \mathrm{E}_{\mathrm{q}^{t_\tau}} \left(\log \mathrm{P}(\boldsymbol{N}^{t_\tau} | \boldsymbol{\epsilon}) \right)
\end{aligned}
$$

for any distributions of $\mathrm{q}(\boldsymbol{N}^{t_1}), \mathrm{q}(\boldsymbol{N}^{t_2}), \mathrm{q}(\boldsymbol{N}^{t_3})$.

Since $\boldsymbol{N}_{ij}^{t_\tau} | \epsilon_{ij} \sim NegBin(\epsilon_{ij}\Lambda_{0\,ij} + 1, \frac{1}{1+\epsilon_{ij}})$ is the negative binomial distribution and $\epsilon_{ij} \sim Unif(0, \infty)$, the M-step is $\hat{\epsilon_{ij}}$

$$
\begin{aligned}
&= \arg\max_{\epsilon_{ij}} \log \mathrm{P}(\epsilon_{ij}) + \sum_{\tau=1}^{3} \sum_{t_\tau=1}^{T} \mathrm{E}_{\mathrm{q}^{t_\tau}} \left(\log \mathrm{P}(\boldsymbol{N}_{ij}^{t_\tau} | \epsilon_{ij}) \right) \\
&\propto \arg\max_{\epsilon_{ij}} \sum_{\tau=1}^{3} \sum_{t_\tau=1}^{T} \mathrm{E}_{\mathrm{q}^{t_\tau}} \left(\log \Gamma(N_{ij}^{t_\tau} + \epsilon_{ij}\Lambda_{0\,ij} + 1) \right) + \log(\epsilon_{ij}) 3T(\epsilon_{ij}\Lambda_{0\,ij} + 1) \\
&- \log(1 + \epsilon_{ij}) 3T(\epsilon_{ij}\Lambda_{0\,ij} + 1) - 3T \log \Gamma(\epsilon_{ij}\Lambda_{0\,ij} + 1) - \log(1 + \epsilon_{ij}) \sum_{\tau=1}^{3} \sum_{t_\tau=1}^{T} \mathrm{E}_{\mathrm{q}^{t_\tau}} (N_{ij}^{t_\tau}) \\
&\geq \arg\max_{\epsilon_{ij}} 3T \left((\epsilon_{ij}\Lambda_{0\,ij} + 1) \log \frac{\epsilon_{ij}}{1 + \epsilon_{ij}} - \log \Gamma(\epsilon_{ij}\Lambda_{0\,ij} + 1) \right) \\
&+ \sum_{\tau=1}^{3} \sum_{t_\tau=1}^{T} \log \Gamma(\mathrm{E}_{\mathrm{q}^{t_\tau}} (N_{ij}^{t_\tau}) + \epsilon_{ij}\Lambda_{0\,ij} + 1) - \log(1 + \epsilon_{ij}) \sum_{\tau=1}^{3} \sum_{t^\tau=1}^{T} \mathrm{E}_{\mathrm{q}^{t_\tau}} (N_{ij}^{t_\tau})
\end{aligned}
$$

and given estimates of the hyperparameters $\hat{\epsilon}_{ij}$, estimators for the rates $\hat{\Lambda}_{ij}$

$$
\begin{aligned}
&= \arg\max_{\Lambda_{ij}} \mathrm{E}_{\mathrm{q}} \left(\log \mathrm{P}(\cap \mathcal{N} | \boldsymbol{\Lambda}, \hat{\boldsymbol{\epsilon}}) + \log \mathrm{P}(\boldsymbol{\Lambda} | \hat{\boldsymbol{\epsilon}}) - \log \mathrm{P}(\cap \mathcal{N}) \right) \\
&\propto \arg\max_{\Lambda_{ij}} \log \mathrm{P}(\boldsymbol{\Lambda} | \hat{\boldsymbol{\epsilon}}) + \sum_{\tau=1}^{3} \sum_{t_\tau=1}^{T} \mathrm{E}_{\mathrm{q}^{t_\tau}} \left(\log \mathrm{P}(\boldsymbol{N}^{t_\tau} | \boldsymbol{\Lambda}) \right) \\
&\propto \arg\max_{\Lambda_{ij}} (\hat{\epsilon}_{ij}\Lambda_{0\,ij}) \log(\Lambda_{ij}) - \hat{\epsilon}_{ij}\Lambda_{ij} - 3T\Lambda_{ij} + \sum_{\tau=1}^{3} \sum_{t_\tau=1}^{T} \mathrm{E}_{\mathrm{q}^{t_\tau}} (N_{ij}^{t_\tau}).
\end{aligned}
$$

Thus when $\mathrm{E}_{\mathrm{q}^{t_1}} (N_{ij}^{t}) = \mathrm{E}(N_{ij}^{t_1} | \boldsymbol{R}^{t_1}, \hat{\boldsymbol{\Lambda}}^k)$ where $\hat{\boldsymbol{\Lambda}}^k$ are the previous iterations' estimators for the rate matrix, the lower bound will push up against the observed log posterior $\log \mathrm{P}(\boldsymbol{\Lambda} | \mathcal{D})$. This makes the E-step just the means of the independent Multinomial dis-

tributions $\prod_{i=1}^{P} Multi(R_i^{t_1}, \frac{\hat{\Lambda}_{i1}^k}{\sum_{j=1}^{P}\hat{\Lambda}_{ij}^k}, \ldots, \frac{\hat{\Lambda}_{iP}^k}{\sum_{j=1}^{P}\hat{\Lambda}_{ij}^k})$ like in the previous models. The same holds when given the column sums $\boldsymbol{C}^{t_2}$ or flows $\boldsymbol{F}^{t_3}$. □

Proof of Proposition 5. The positive estimator $\hat{\boldsymbol{\Lambda}}$ that maximizes the MRE distribution is the solution to $\underset{\boldsymbol{\Lambda}\in\mathbb{R}^+}{\arg\max}\ \log\left(\mathrm{P}(\boldsymbol{\Lambda}|\boldsymbol{R},\boldsymbol{C},\boldsymbol{F})\right)$

$$
\begin{aligned}
&= \underset{\boldsymbol{\Lambda}\in\mathbb{R}^+}{\arg\max}\ \log(\prod_{ij}\exp\left\{-|\Lambda_{ij}-\Lambda_{0ij}|\right\}) - \log(Z\left(\boldsymbol{\rho},\boldsymbol{\gamma},\boldsymbol{\phi}\right))\\
&+ \log(\exp\{\hat{\boldsymbol{\rho}}'(\boldsymbol{\Lambda}\mathbf{1}-\bar{\boldsymbol{R}}) + \hat{\boldsymbol{\gamma}}'(\mathbf{1}'\boldsymbol{\Lambda}-\bar{\boldsymbol{C}}) + \hat{\boldsymbol{\phi}}'(\boldsymbol{A}\boldsymbol{\Lambda}\boldsymbol{B}-\bar{\boldsymbol{F}})\})\\
&= \underset{\boldsymbol{\Lambda}\in\mathbb{R}^+}{\arg\max}\ -\sum_{ij}|\Lambda_{ij}-\Lambda_{0ij}| + \hat{\boldsymbol{\rho}}'(\boldsymbol{\Lambda}\mathbf{1}-\bar{\boldsymbol{R}}) + \hat{\boldsymbol{\gamma}}'(\boldsymbol{\Lambda}'\mathbf{1}-\bar{\boldsymbol{C}}) + \hat{\boldsymbol{\phi}}'(\boldsymbol{A}\boldsymbol{\Lambda}\boldsymbol{B}-\bar{\boldsymbol{F}})\\
&= \underset{\boldsymbol{\Lambda}\in\mathbb{R}^+}{\arg\min}\ ||\boldsymbol{\Lambda}-\boldsymbol{\Lambda}_0||_1 - \hat{\boldsymbol{\rho}}'(\boldsymbol{\Lambda}\mathbf{1}-\bar{\boldsymbol{R}}) - \hat{\boldsymbol{\gamma}}'(\boldsymbol{\Lambda}'\mathbf{1}-\bar{\boldsymbol{C}}) - \hat{\boldsymbol{\phi}}'(\boldsymbol{A}\boldsymbol{\Lambda}\boldsymbol{B}-\bar{\boldsymbol{F}})
\end{aligned}
$$

where $||\cdot||_1$ is the element wise ℓ_1 norm and the optimal Lagrange multipliers $\hat{\boldsymbol{\rho}},\hat{\boldsymbol{\gamma}},\hat{\boldsymbol{\phi}}$ are the solution to $\underset{\boldsymbol{\rho},\boldsymbol{\gamma},\boldsymbol{\phi}}{\arg\max}\ -\log\left(Z(\boldsymbol{\rho},\boldsymbol{\gamma},\boldsymbol{\phi})\right)$

$$
\begin{aligned}
&= \underset{\boldsymbol{\rho},\boldsymbol{\gamma},\boldsymbol{\phi}}{\arg\max}\ \sum_{i=1}^{P}\rho_i\bar{R}_i + \sum_{j=1}^{P}\gamma_j\bar{C}_j + \sum_h \phi_h\bar{F}_h - \log 2\\
&- \sum_{ij}\Lambda_{0ij}(\rho_i+\gamma_j+\sum_h\phi_h A_{hi}B_j) + \log(1+LM_{ij}) + \log(1-LM_{ij})\\
&= \underset{\boldsymbol{\rho},\boldsymbol{\gamma},\boldsymbol{\phi}}{\arg\max}\ \sum_{i=1}^{P}\rho_i(\bar{R}_i - \sum_{j=1}^{P}\Lambda_{0ij}) + \sum_{j=1}^{P}\gamma_j(\bar{C}_j - \sum_{i=1}^{P}\Lambda_{0ij})\\
&+ \sum_h \phi_h(\bar{F}_h - \sum_{ij}A_{hi}\Lambda_{0ij}B_j) + \sum_{ij}\log(1-LM_{ij}^2)
\end{aligned}
\tag{6.10}
$$

where $LM_{ij} = \rho_i + \gamma_j + \sum_h \phi_h A_{hi}B_j$.

The Lagrangian of the loss function in (6.6) is $||\boldsymbol{\Lambda}-\boldsymbol{\Lambda}_0||_1$
$+ \boldsymbol{\rho}'(\boldsymbol{\Lambda}\mathbf{1}-\bar{\boldsymbol{R}}) + \boldsymbol{\gamma}'(\mathbf{1}'\boldsymbol{\Lambda}-\bar{\boldsymbol{C}}) + \boldsymbol{\phi}'(\boldsymbol{A}\boldsymbol{\Lambda}\boldsymbol{B}-\bar{\boldsymbol{F}})$ with optimal Lagrange multipliers that are the solution to dual problem

$$
\begin{aligned}
&= \underset{\boldsymbol{\rho},\boldsymbol{\gamma},\boldsymbol{\phi}}{\arg\max}\ -\sum_{ij}f^*(-\rho_i-\gamma_j-\sum_h\phi_h A_{hi}B_j) - \sum_{i=1}^{P}\rho_i\bar{R}_i - \sum_{j=1}^{P}\gamma_j\bar{C}_j - \sum_h\phi_h\bar{F}_h\\
&= \underset{\boldsymbol{\rho},\boldsymbol{\gamma},\boldsymbol{\phi}}{\arg\max}\ \sum_{ij}\Lambda_{0ij}(LM_{ij}) - \sum_{i=1}^{P}\rho_i\bar{R}_i - \sum_{j=1}^{P}\gamma_j\bar{C}_j - \sum_h\phi_h\bar{F}_h \quad \text{s.t.} \quad |LM_{ij}|<1\ \forall i,j
\end{aligned}
$$

because $f^*(-\rho_i - \gamma_j - \sum_h \phi_h A_{hi} B_j)$ are the convex conjugates defined as

$$
\begin{aligned}
&= \max_{\Lambda_{ij}} -\Lambda_{ij}(\rho_i + \gamma_j + \sum_h \phi_h A_{hi} B_j) - |\Lambda_{ij} - \Lambda_{0ij}| \\
&= \max_{\Lambda_{ij}} \begin{cases} \Lambda_{0ij} - \Lambda_{ij}(1 + LM_{ij}) & \text{if} \quad \Lambda_{ij} \geq \Lambda_{0ij} \\ \Lambda_{ij}(1 - LM_{ij}) - \Lambda_{0ij} & \text{if} \quad \Lambda_{ij} < \Lambda_{0ij} \end{cases} \\
&= \begin{cases} \infty \quad \text{if} \quad |\rho_i + \gamma_j + \sum_h \phi_h A_{hi} B_j| > 1 \\ -\Lambda_{0ij}(\rho_i + \gamma_j + \sum_h \phi_h A_{hi} B_j) \quad \text{otherwise.} \end{cases}
\end{aligned}
$$

The dual can be relaxed with log barrier terms to an unconstrained problem that is equivalent to (6.7) making minimizing the Lagrangian of (6.6) for $\mathbf{\Lambda}$ equivalent to maximizing the MRE distribution.

□

Proof of Proposition 6. Using Remark 1.7 of [144], then for regular models, the MAP estimator will have the same asymptotic properties as the MLE. Thus, the standard proof for the asymptotic distribution for the log likelihood ratio [145] applies to the log posterior density ratio. □

Proof of Proposition 7. Let $\mathcal{M}(\mathbf{\Lambda}^*)$ be the true model, then the test statistic ψ

$$
\begin{aligned}
&= -2 \sum_{t=1}^{T} \log(\mathcal{M}_t(\mathbf{\Lambda}_0)) - \log(\mathcal{M}_t(\hat{\mathbf{\Lambda}})) = -2 \left(\sum_{t=1}^{T} \log(\mathcal{M}_t(\mathbf{\Lambda}_0)) - \max_{\mathbf{\Lambda} \in \mathbb{R}^+} \sum_{t=1}^{T} \log(\mathcal{M}_t(\mathbf{\Lambda})) \right) \\
&= 2 \sum_{t=1}^{T} \log(\mathcal{M}_t(\mathbf{\Lambda}^*)) - \log(\mathcal{M}_t(\mathbf{\Lambda}_0)) - 2 \min_{\mathbf{\Lambda} \in \mathbb{R}^+} \sum_{t=1}^{T} \log(\mathcal{M}_t(\mathbf{\Lambda}^*)) - \log(\mathcal{M}_t(\mathbf{\Lambda}))
\end{aligned}
$$

and as $T \to \infty$,

$$
\frac{\psi}{T} \to 2\,\mathrm{KL}\left(\mathcal{M}(\mathbf{\Lambda}^* || \mathcal{M}(\mathbf{\Lambda}_0)\right) - 2 \min_{\mathbf{\Lambda} \in \mathbb{R}^+} \mathrm{KL}\left(\mathcal{M}(\mathbf{\Lambda}^*) || \mathcal{M}(\mathbf{\Lambda})\right) = 2\,\mathrm{KL}\left(\mathcal{M}(\mathbf{\Lambda}^* || \mathcal{M}(\mathbf{\Lambda}_0))\right) = \Psi.
$$

The misspecified test statistic $\hat{\psi}$

$$
\begin{aligned}
&= -2 \sum_{t=1}^{T} \log(\hat{\mathcal{M}}_t^k(\mathbf{\Lambda}_0)) - \log(\hat{\mathcal{M}}_t^k(\hat{\mathbf{\Lambda}})) = -2 \sum_{t=1}^{T} \log(\hat{\mathcal{M}}_t^k(\mathbf{\Lambda}_0)) - \max_{\mathbf{\Lambda} \in \mathbb{R}^+} \sum_{t=1}^{T} \log(\hat{\mathcal{M}}_t^k(\mathbf{\Lambda})) \\
&= 2 \sum_{t=1}^{T} \log(\mathcal{M}_t(\mathbf{\Lambda}^*)) - \log(\mathcal{M}_t(\mathbf{\Lambda}_0)) + 2 \sum_{t=1}^{T} \log(\mathcal{M}_t(\mathbf{\Lambda}_0)) - \log(\hat{\mathcal{M}}_t^k(\mathbf{\Lambda}_0)) \qquad (6.11) \\
&\quad - 2 \sum_{t=1}^{T} \log(\mathcal{M}_t(\mathbf{\Lambda}^*)) - \log(\hat{\mathcal{M}}_t^k(\hat{\mathbf{\Lambda}}^*)) - 2 \min_{\mathbf{\Lambda} \in \mathbb{R}^+} \sum_{t=1}^{T} \log(\hat{\mathcal{M}}_t^k(\hat{\mathbf{\Lambda}}^*)) - \log(\hat{\mathcal{M}}_t^k(\mathbf{\Lambda}))
\end{aligned}
$$

and as $T \to \infty$,

$$\begin{aligned}\hat{\psi}/T \to\ & 2\,\mathrm{KL}\left(\mathcal{M}(\mathbf{\Lambda}^*)||\mathcal{M}(\mathbf{\Lambda}_0)\right) + 2\,\mathrm{KL}\left(\mathcal{M}(\mathbf{\Lambda}_0)||\hat{\mathcal{M}}^k(\mathbf{\Lambda}_0)\right) \\ & - 2\,\mathrm{KL}\left(\mathcal{M}(\mathbf{\Lambda}^*)||\hat{\mathcal{M}}^k(\hat{\mathbf{\Lambda}}^*)\right) - 2\min_{\mathbf{\Lambda}\in\mathbb{R}^+}\mathrm{KL}\left(\hat{\mathcal{M}}^k(\hat{\mathbf{\Lambda}}^*)||\hat{\mathcal{M}}^k(\mathbf{\Lambda})\right) \\ =\ & \Psi - 2\left(\mathrm{KL}\left(\mathcal{M}(\mathbf{\Lambda}^*)||\hat{\mathcal{M}}^k(\hat{\mathbf{\Lambda}}^*)\right) - \mathrm{KL}\left(\mathcal{M}(\mathbf{\Lambda}_0)||\hat{\mathcal{M}}^k(\mathbf{\Lambda}_0)\right)\right)\end{aligned}$$

where $\Psi = 2\,\mathrm{KL}\left(\mathcal{M}(\mathbf{\Lambda}^*||\mathcal{M}(\mathbf{\Lambda}_0))\right)$ and $\hat{\mathcal{M}}^k(\hat{\mathbf{\Lambda}}^*)$ is the closest population local maximum at iteration k. If as $k \to \infty$, the EM model $\hat{\mathcal{M}}^k$ converges to the true model $\mathcal{M}$, then $\hat{\psi}/T \to \Psi$

□

CHAPTER 7

Conclusion and Future Work

In conclusion, real data is complex and contains many intricacies such as partial labels, latent variables, and anomalies. This thesis developed methods to deal with the complications that arise in real data. This thesis also address the very contemporary problem of updating models when data is being continuously collected.

In chapter 2, we proposed a penalized ensemble Kalman filter that is designed for superior performance in non-linear high dimensional systems. We give theoretical results that prove that the Kalman gain matrix used in our algorithm will converge to the population Kalman gain matrix under the non-simplistic asymptotic case of high-dimensional scaling, where the sample size and the dimensionality increase to infinity. We show the performance of our filter in data generated from fluid dynamics simulators, which are know to be strongly non-linear and chaotic.

In chapter 3, we proposed a framework, using minimum constrained relative entropy, to generate and update regression models. This framework can be used to build an optimal non-linear filter and also an approximation to it that is computationally efficient. For stationary systems, we can bound the performance between our proposed sparse approximate model and a model trained on the entire batch of data.

In chapter 4, we proposed recursive versions of supervised and semi-supervised maximum margin classifiers in the minimum entropy discrimination classification framework. Our proposed models perform nearly as well as a much more computationally expensive batch model and significantly better than a model that cannot incorporate previous data.

In chapter 5, we proposed a method for detecting anomalous points that are of high utility by exploiting the key idea that high-utility points are also anomalous. The method simultaneously uses semi-supervised utility labels and incorporates anomaly information through the EM algorithm. We show in simulations, that the performance increases with EM iterations because using previous label information helps identify anomalies and vice versa. We applied our method to the Reddit and CTU-13 botnet datasets to show its applicability in real life situations.

In chapter 6, we developed a framework and probabilistic model for detecting anomalous activity in the traffic rates of sparse networks where, in the most restrictive case, only the total egress and ingress of the nodes are observed. We show real-world applicability of our models to datasets containing botnet and taxi traffic.

7.1 Future Work

One area I would like to further explore is the connection between the exponential family and information projections with Kullback-Leibler divergence. While the geometry of exponential family likelihoods and their duality with maximum entropy has been well studied [146]; the intuition is less clear for posteriors. Much of the work in this thesis uses the principle of minimum relative entropy to building models and while it presents some geometric intuition, I am interested in further studying the relationship between Bayes rule and projecting with relative entropy. Understanding this could have significant impact on how to design models. Many of the most popular algorithms explicitly or implicitly use exponential family distributions and every exponential family induces a Bregman divergence [147]. However, some loss functions, do not correspond to a Bregman divergence associated with a distribution. I believe that the framework discussed in chapters 3 through 5 can be used to build a class of posterior models that have all the benefits of the exponential family, but can induce a much larger class of loss functions. In contrast to directly optimizing the loss function, these models would also have the advantages of generative models such as filtering and being solvable with the EM algorithm.

Another area I am interested in is misspecified models; specifically the kind described in the famous quote by George Box, All models are wrong but some are useful. Parametric models, particularly ones from the exponential family, are useful because they are generally easier to solve, but data from the real world is usually not generated from such simple distributions. Even when we model the generating process as multi-layered hierarchal distributions or with complicated non-linear functions, these models are often not feasible to solve, especially not quickly. This need for fast computational complexity has lead to the development of algorithms that approximate not the true generating process, but other more complex models. This is inspired by the work of chapter 3, where we present a sparse approximation to an optimal model, which is computationally much faster. I am interested in studying essentially the model equivalent to the trade-off between bias and variance, the trade-off between wrong-ness and usefulness. My goal is to develop a framework for designing less computationally intensive models that are close to their complex counterparts, but useful in that they preserve the all the necessary aspects such as the mean, mode, and

standard deviation.

BIBLIOGRAPHY

[1] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, pp. 2399–2434, Dec. 2006.

[2] G. Evensen, "Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics," *Journal of Geophysical Research: Oceans*, vol. 99, no. C5, pp. 10143–10162, 1994.

[3] G. Burgers, P. Jan van Leeuwen, and G. Evensen, "Analysis scheme in the ensemble kalman filter," *Monthly weather review*, vol. 126, no. 6, pp. 1719–1724, 1998.

[4] I. M. Johnstone, "On the distribution of the largest eigenvalue in principal components analysis," *The Annals of Statistics*, vol. 29, no. 2, pp. 295–327, 2001.

[5] I. M. Johnstone and A. Yu Lu, "Sparse principle component analysis," *Unpublished Manuscript*, 2004.

[6] T. M. Hamill, J. S. Whitaker, and C. Snyder, "Distance-dependent filtering of background error covariance estimates in an ensemble kalman filter," *Monthly Weather Review*, vol. 129, no. 11, pp. 2776–2790, 2001.

[7] P. L. Houtekamer and H. L. Mitchell, "A sequential ensemble kalman filter for atmospheric data assimilation," *Monthly Weather Review*, vol. 129, no. 1, pp. 123–137, 2001.

[8] E. Ott, B. R. Hunt, I. Szunyogh, A. V. Zimin, E. J. Kostelich, M. Corazza, E. Kalnay, D. Patil, and J. A. Yorke, "A local ensemble kalman filter for atmospheric data assimilation," *Tellus A*, vol. 56, no. 5, pp. 415–428, 2004.

[9] P. L. Houtekamer, H. L. Mitchell, G. Pellerin, M. Buehner, M. Charron, L. Spacek, and B. Hansen, "Atmospheric data assimilation with an ensemble kalman filter: Results with real observations," *Monthly Weather Review*, vol. 133, no. 3, pp. 604–620, 2005.

[10] X. Wang, T. M. Hamill, J. S. Whitaker, and C. H. Bishop, "A comparison of hybrid ensemble transform kalman filteroptimum interpolation and ensemble square root filter analysis schemes," *Monthly Weather Review*, vol. 135, no. 3, pp. 1055–1076, 2007.

[11] J. L. Anderson, "An adaptive covariance inflation error correction algorithm for ensemble filters," *Tellus A*, vol. 59, no. 2, pp. 210–224, 2007.

[12] J. L. Anderson, "Spatially and temporally varying adaptive covariance inflation for ensemble filters," *Tellus A*, vol. 61, no. 1, pp. 72–83, 2009.

[13] H. Li, E. Kalnay, and T. Miyoshi, "Simultaneous estimation of covariance inflation and observation errors within an ensemble kalman filter," *Quarterly Journal of the Royal Meteorological Society*, vol. 135, no. 639, pp. 523–533, 2009.

[14] C. Bishop and D. Hodyss, "Ensemble covariances adaptively localized with eco-rap. part 1: tests on simple error models," *Tellus A*, vol. 61, no. 1, 2009.

[15] C. H. Bishop and D. Hodyss, "Ensemble covariances adaptively localized with eco-rap. part 2: a strategy for the atmosphere," *Tellus A*, vol. 61, no. 1, pp. 97–111, 2009.

[16] P. L. Houtekamer, H. L. Mitchell, and X. Deng, "Model error representation in an operational ensemble kalman filter," *Monthly Weather Review*, vol. 137, no. 7, pp. 2126–2143, 2009.

[17] W. F. Campbell, C. H. Bishop, and D. Hodyss, "Vertical covariance localization for satellite radiances in ensemble kalman filters," *Monthly Weather Review*, vol. 138, no. 1, pp. 282–290, 2010.

[18] S. J. Greybush, E. Kalnay, T. Miyoshi, K. Ide, and B. R. Hunt, "Balance and ensemble kalman filter localization techniques," *Monthly Weather Review*, vol. 139, no. 2, pp. 511–522, 2011.

[19] T. Miyoshi, "The gaussian approach to adaptive covariance inflation and its implementation with the local ensemble transform kalman filter," *Monthly Weather Review*, vol. 139, no. 5, pp. 1519–1535, 2011.

[20] G. Evensen, "The ensemble kalman filter: Theoretical formulation and practical implementation," *Ocean dynamics*, vol. 53, no. 4, pp. 343–367, 2003.

[21] G. Evensen, "Sampling strategies and square root analysis schemes for the enkf," *Ocean dynamics*, vol. 54, no. 6, pp. 539–560, 2004.

[22] C. H. Bishop, B. J. Etherton, and S. J. Majumdar, "Adaptive sampling with the ensemble transform kalman filter. part i: Theoretical aspects," *Monthly Weather Review*, vol. 129, no. 3, pp. 420–436, 2001.

[23] J. S. Whitaker and T. M. Hamill, "Ensemble data assimilation without perturbed observations," *Monthly Weather Review*, vol. 130, no. 7, pp. 1913–1924, 2002.

[24] M. K. Tippett, J. L. Anderson, C. H. Bishop, T. M. Hamill, and J. S. Whitaker, "Ensemble square root filters," *Monthly Weather Review*, vol. 131, no. 7, pp. 1485–1490, 2003.

[25] B. R. Hunt, E. J. Kostelich, and I. Szunyogh, "Efficient data assimilation for spatiotemporal chaos: a local ensemble transform kalman filter," *Physica D: Nonlinear Phenomena*, vol. 230, no. 1-2, pp. 112–126, 2007.

[26] H. C. Godinez and J. D. Moulton, "An efficient matrix-free algorithm for the ensemble kalman filter," *Computational Geosciences*, vol. 16, no. 3, pp. 565–575, 2012.

[27] L. Nerger, T. Janji, J. Schrter, and W. Hiller, "A unification of ensemble square root kalman filters," *Monthly Weather Review*, vol. 140, no. 7, pp. 2335–2345, 2012.

[28] J. Tödter and B. Ahrens, "A second-order exact ensemble square root filter for nonlinear data assimilation," *Monthly Weather Review*, vol. 143, no. 4, pp. 1347–1367, 2015.

[29] N. Papadakis, E. Mémin, A. Cuzol, and N. Gengembre, "Data assimilation with the weighted ensemble kalman filter," *Tellus A*, vol. 62, no. 5, pp. 673–697, 2010.

[30] C. Snyder, T. Bengtsson, P. Bickel, and J. Anderson, "Obstacles to high-dimensional particle filtering," *Monthly Weather Review*, vol. 136, no. 12, pp. 4629–4640, 2008.

[31] P. J. van Leeuwen, "Nonlinear data assimilation in geosciences: an extremely efficient particle filter," *Quarterly Journal of the Royal Meteorological Society*, vol. 136, no. 653, pp. 1991–1999, 2010.

[32] M. Ades and P. J. van Leeuwen, "An exploration of the equivalent weights particle filter," *Quarterly Journal of the Royal Meteorological Society*, vol. 139, no. 672, pp. 820–840, 2013.

[33] J. Lei and P. Bickel, "A moment matching ensemble filter for nonlinear non-gaussian data assimilation," *Monthly Weather Review*, vol. 139, no. 12, pp. 3964–3973, 2011.

[34] M. Frei and H. R. Künsch, "Bridging the ensemble kalman and particle filters," *Biometrika*, vol. 100, no. 4, pp. 781–800, 2013.

[35] S. Nakano, "Hybrid algorithm of ensemble transform and importance sampling for assimilation of non-gaussian observations," *Tellus A*, vol. 66, no. 0, 2014.

[36] S. Robert and H. R. Künsch, "Local Ensemble Kalman Particle Filters for efficient data assimilation," *ArXiv e-prints*, May 2016.

[37] S. J. Julier and J. K. Uhlmann, "New extension of the kalman filter to nonlinear systems," in *AeroSense'97*, pp. 182–193, International Society for Optics and Photonics, 1997.

[38] E. A. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pp. 153–158, Ieee, 2000.

[39] G. Ueno and T. Tsuchiya, "Covariance regularization in inverse space," *Quarterly Journal of the Royal Meteorological Society*, vol. 135, no. 642, pp. 1133–1156, 2009.

[40] E. D. Nino-Ruiz, A. Sandu, and X. Deng, "A parallel ensemble kalman filter implementation based on modified cholesky decomposition," in *Proceedings of the 6th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems*, ScalA '15, pp. 4:1–4:8, 2015.

[41] P. Ravikumar, M. J. Wainwright, G. Raskutti, B. Yu, *et al.*, "High-dimensional covariance estimation by minimizing 1-penalized log-determinant divergence," *Electronic Journal of Statistics*, vol. 5, pp. 935–980, 2011.

[42] J. Janková, S. Van De Geer, *et al.*, "Confidence intervals for high-dimensional inverse covariance estimation," *Electronic Journal of Statistics*, vol. 9, no. 1, pp. 1205–1229, 2015.

[43] R. Foygel and M. Drton, "Extended bayesian information criteria for gaussian graphical models," in *Advances in neural information processing systems*, pp. 604–612, 2010.

[44] J. Lv and J. S. Liu, "Model selection principles in misspecified models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 76, no. 1, pp. 141–167, 2014.

[45] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.

[46] C.-J. Hsieh, M. A. Sustik, I. S. Dhillon, P. K. Ravikumar, and R. Poldrack, "Big & quic: Sparse inverse covariance estimation for a million variables," in *Advances in Neural Information Processing Systems*, pp. 3165–3173, 2013.

[47] G. Gaspari and S. E. Cohn, "Construction of correlation functions in two and three dimensions," *Quarterly Journal of the Royal Meteorological Society*, vol. 125, no. 554, pp. 723–757, 1999.

[48] G. Schwarz, "Estimating the dimension of a model," *Ann. Statist.*, vol. 6, pp. 461–464, 03 1978.

[49] T. Bengtsson, C. Snyder, and D. Nychka, "Toward a nonlinear ensemble filter for high-dimensional systems," *Journal of Geophysical Research: Atmospheres*, vol. 108, no. D24, 2003.

[50] M. Frei and H. R. Künsch, "Mixture ensemble kalman filters," *Computational Statistics & Data Analysis*, vol. 58, pp. 127–138, 2013.

[51] M. Würsch and G. C. Craig, "A simple dynamical model of cumulus convection for data assimilation research," *Meteorologische Zeitschrift*, vol. 23, pp. 483–490, 12 2014.

[52] S. Robert, *modifiedSWEQ: Simplified cumulus convection with the modified SWEQ*, 2014. R package version 0.1, https://github.com/robertsy/modifiedSWEQ.

[53] R. Vershynin, "How close is the sample covariance matrix to the actual covariance matrix?," *Journal of Theoretical Probability*, vol. 25, no. 3, pp. 655–686, 2012.

[54] T. Jaakkola, M. Meila, and T. Jebara, "Maximum entropy discrimination," in *Advances in Neural Information Processing Systems 12* (S. Solla, T. Leen, and K. Müller, eds.), pp. 470–476, MIT Press, 2000.

[55] E. Hou and A. O. Hero, "Sequential maximum margin classifiers for partially labeled data," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2826–2830, IEEE, 2018.

[56] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least-squares algorithm," *IEEE Transactions on Signal Processing*, vol. 52, pp. 2275–2285, Aug 2004.

[57] Y. Engel, S. Mannor, and R. Meir, "Sparse online greedy support vector regression," in *Machine Learning: ECML 2002* (T. Elomaa, H. Mannila, and H. Toivonen, eds.), (Berlin, Heidelberg), pp. 84–96, Springer Berlin Heidelberg, 2002.

[58] L. Csató and M. Opper, "Sparse on-line gaussian processes," *Neural computation*, vol. 14, no. 3, pp. 641–668, 2002.

[59] A. Ranganathan, M.-H. Yang, and J. Ho, "Online sparse gaussian process regression and its applications," *IEEE Transactions on Image Processing*, vol. 20, no. 2, pp. 391–404, 2010.

[60] Y. Altun and A. Smola, "Unifying divergence minimization and statistical inference via convex duality," in *International Conference on Computational Learning Theory*, pp. 139–153, Springer, 2006.

[61] O. Koyejo and J. Ghosh, "A representation approach for relative entropy minimization with expectation constraints," in *ICML WDDL workshop*, 2013.

[62] T. Jebara, *Machine learning: discriminative and generative*, vol. 755. Springer Science & Business Media, 2012.

[63] G. Wahba, "Support vector machines, reproducing kernel hilbert spaces and the randomized gacv," *Advances in Kernel Methods-Support Vector Learning*, vol. 6, pp. 69–87, 1999.

[64] T. S. Jaakkola and D. Haussler, "Probabilistic kernel regression models.," in *AISTATS*, 1999.

[65] A. J. Smola, B. Schölkopf, and K.-R. Müller, "The connection between regularization operators and support vector kernels," *Neural networks*, vol. 11, no. 4, pp. 637–649, 1998.

[66] M. Opper and O. Winther, "Gaussian process classification and svm: Mean field results and leave-one-out estimator," *Advances in Large Margin Classifiers*, 1999.

[67] P. Sollich, "Bayesian methods for support vector machines: Evidence and predictive class probabilities," *Machine Learning*, vol. 46, no. 1, pp. 21–52, 2002.

[68] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[69] E. Hou, K. Sricharan, and A. O. Hero, "Latent laplacian maximum entropy discrimination for detection of high-utility anomalies," *IEEE Transactions on Information Forensics and Security*, vol. 13, pp. 1446–1459, June 2018.

[70] A. Grigoryan, "Heat kernels on weighted manifolds and applications," *Cont. Math*, vol. 398, pp. 93–191, 2006.

[71] R. R. Lederman and V. Rokhlin, "On the analytical and numerical properties of the truncated laplace transform i.," *SIAM Journal on Numerical Analysis*, vol. 53, no. 3, pp. 1214–1235, 2015.

[72] C. Elkan, "Deriving tf-idf as a fisher kernel," in *SPIRE*, vol. 3772, pp. 295–300, Springer, 2005.

[73] M. Lichman, "UCI machine learning repository," 2013.

[74] P. B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. C. Platt, "Support vector method for novelty detection," in *Advances in Neural Information Processing Systems 12*, pp. 582–588, 2000.

[75] N. Görnitz, M. Kloft, K. Rieck, and U. Brefeld, "Toward supervised anomaly detection.," *Journal of Artificial Intelligence Research (JAIR)*, vol. 46, pp. 235–262, 2013.

[76] A. O. Hero, "Geometric entropy minimization (gem) for anomaly detection and localization," in *Advances in Neural Information Processing Systems 19*, pp. 585–592, 2007.

[77] C. Scott and R. D. Nowak, "Learning minimum volume sets," *Journal of Machine Learning Research (JMLR)*, vol. 7, pp. 665–704, 2006.

[78] K. Sricharan and A. O. Hero, "Efficient anomaly detection using bipartite k-nn graphs," in *Advances in Neural Information Processing Systems 24*, pp. 478–486, 2011.

[79] D. Pelleg and A. W. Moore, "Active learning for anomaly and rare-category detection," in *Advances in Neural Information Processing Systems 17*, pp. 1073–1080, 2005.

[80] J. W. Stokes, J. C. Platt, J. Kravis, and M. Shilman, "Aladin: Active learning of anomalies to detect intrusions," *Technique Report. Microsoft Network Security Redmond, WA*, vol. 98052, 2008.

[81] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[82] O. Chapelle and A. Zien, "Semi-supervised classification by low density separation," in *Tenth International Workshop on Artificial Intelligence and Statistics (AISTATS 2005)*, pp. 57–64, 2005.

[83] T. Jebara, *Machine Learning: Discriminative and Generative*. Springer Science & Business Media, 2004.

[84] K. Veeramachaneni, I. Arnaldo, V. Korrapati, C. Bassias, and K. Li, "Aiˆ2: Training a big data machine to defend," in *2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS)*, pp. 49–54, 2016.

[85] S. Das, W. K. Wong, T. Dietterich, A. Fern, and A. Emmott, "Incorporating expert feedback into active anomaly discovery," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 853–858, 2016.

[86] H. Li and Z. Han, "Catch me if you can: An abnormality detection approach for collaborative spectrum sensing in cognitive radio networks," *IEEE Transactions on Wireless Communications*, vol. 9, no. 11, pp. 3554–3565, 2010.

[87] T. Xie, N. M. Nasrabadi, and A. O. Hero, "Learning to classify with possible sensor failures," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2395–2399, 2014.

[88] J. Zhu, N. Chen, and E. P. Xing, "Infinite svm: a dirichlet process mixture of large-margin kernel machines," in *28th International Conference on Machine Learning (ICML-11)*, pp. 617–624, 2011.

[89] J. Zhu and E. P. Xing, "Maximum entropy discrimination markov networks," *Journal of Machine Learning Research (JMLR)*, vol. 10, pp. 2531–2569, 2009.

[90] M. Opper and O. Winther, "Gaussian process classification and svm: Mean field results and leave-one-out estimator," 1999.

[91] V. Sindhwani, P. Niyogi, and M. Belkin, "Beyond the point cloud: from transductive to semi-supervised learning," in *22nd International Conference on Machine Learning (ICML-05)*, pp. 824–831, 2005.

[92] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011. http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[93] M. Grant and S. Boyd, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*, pp. 95–110, Springer-Verlag Limited, 2008.

[94] M. Grant and S. Boyd, "Cvx: Matlab software for disciplined convex programming, version 2.1," 2014.

[95] V. Franc, "Library for quadratic programming," 2009. http://cmp.felk.cvut.cz/~xfrancv/pages/libqp.html.

[96] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *23th International Conference on Machine Learning (ICML-06)*, pp. 233–240, 2006.

[97] Stuck_In_the_Matrix, "I have every publicly available reddit comment for research. $\sim$ 1.7 billion comments @ 250 gb compressed. any interest in this?." Reddit, 2015. Full Dataset: https://www.reddit.com/r/datasets/comments/3bxlg7/i_have_every_publicly_available_reddit_comment, Kaggle: https://www.kaggle.com/c/reddit-comments-may-2015.

[98] S. Garcia, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," *Computers & Security*, vol. 45, pp. 100–123, 2014. http://mcfp.weebly.com/the-ctu-13-dataset-a-labeled-dataset-with-botnet-normal-and-background-traffic.html.

[99] Y. Vardi, "Network tomography: Estimating source-destination traffic intensities from link data," *Journal of the American Statistical Association*, vol. 91, no. 433, pp. 365–377, 1996.

[100] A. Coates, A. O. H. III, R. Nowak, and B. Yu, "Internet tomography," *IEEE Signal processing magazine*, vol. 19, no. 3, pp. 47–65, 2002.

[101] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot, "Traffic matrix estimation: Existing techniques and new directions," in *Proceedings of the 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pp. 161–174, 2002.

[102] R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu, "Network tomography: Recent developments," *Statistical science*, pp. 499–517, 2004.

[103] E. Lawrence, G. Michailidis, V. N. Nair, and B. Xi, "Network tomography: A review and recent developments," in *Frontiers in statistics*, pp. 345–366, 2006.

[104] M. Coates, R. Castro, R. Nowak, M. Gadhiok, R. King, and Y. Tsang, "Maximum likelihood network topology identification from edge-based unicast measurements," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 30, pp. 11–20, 2002.

[105] R. Cáceres, N. G. Duffield, J. Horowitz, and D. F. Towsley, "Multicast-based inference of network-internal loss characteristics," *IEEE Transactions on Information theory*, vol. 45, no. 7, pp. 2462–2480, 1999.

[106] M. Rabbat, R. Nowak, and M. Coates, "Multiple source, multiple destination network tomography," in *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, pp. 1628–1639, 2004.

[107] Y. Tsang, M. Coates, and R. D. Nowak, "Network delay tomography," *IEEE Transactions on Signal Processing*, vol. 51, no. 8, pp. 2125–2136, 2003.

[108] M.-F. Shih and A. O. Hero, "Unicast-based inference of network link delay distributions with finite mixture models," *IEEE Transactions on Signal Processing*, vol. 51, no. 8, pp. 2219–2228, 2003.

[109] M.-F. Shih and A. O. Hero, "Hierarchical inference of unicast network topologies based on end-to-end measurements," *IEEE Transactions on Signal Processing*, vol. 55, no. 5, pp. 1708–1718, 2007.

[110] N. Duffield, "Network tomography of binary network performance characteristics," *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5373–5388, 2006.

[111] C. Tebaldi and M. West, "Bayesian inference on network traffic using link count data," *Journal of the American Statistical Association*, vol. 93, no. 442, pp. 557–573, 1998.

[112] J. Cao, S. V. Wiel, B. Yu, and Z. Zhu, "A scalable method for estimating network traffic matrices from link counts," tech. rep., 2000.

[113] J. Cao, D. Davis, S. V. Wiel, and B. Yu, "Time-varying network tomography: Router link data," *Journal of the American Statistical Association*, vol. 95, no. 452, pp. 1063–1075, 2000.

[114] J. Zhang and I. C. Paschalidis, "Statistical anomaly detection via composite hypothesis testing for markov models," *IEEE Transactions on Signal Processing*, vol. 66, pp. 589–602, Feb 2018.

[115] R. J. Vanderbei and J. Iannone, "An EM approach to OD matrix estimation," tech. rep., 1994.

[116] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast accurate computation of large-scale ip traffic matrices from link loads," in *Proceedings of the 2003 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '03, pp. 206–217, 2003.

[117] Y. Zhang, M. Roughan, C. Lund, and D. Donoho, "An information-theoretic approach to traffic matrix estimation," in *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '03, pp. 301–312, 2003.

[118] M. Roughan, Y. Zhang, W. Willinger, and L. Qiu, "Spatio-temporal compressive sensing and internet traffic matrices (extended version)," *Networking, IEEE/ACM Transactions on*, vol. 20, pp. 662–676, June 2012.

[119] M. Mardani, G. Mateos, and G. Giannakis, "Recovery of low-rank plus compressed sparse matrices with application to unveiling traffic anomalies," *Information Theory, IEEE Transactions on*, vol. 59, pp. 5186–5205, Aug 2013.

[120] M. Mardani and G. Giannakis, "Estimating traffic and anomaly maps via network tomography," *Networking, IEEE/ACM Transactions on*, vol. 24, pp. 1–15, June 2016.

[121] A. Lakhina, M. Crovella, and C. Diot, "Characterization of network-wide anomalies in traffic flows," in *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pp. 201–206, 2004.

[122] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," in *ACM SIGCOMM Computer Communication Review*, vol. 34, pp. 219–230, 2004.

[123] Y. Zhang, Z. Ge, A. Greenberg, and M. Roughan, "Network anomography," in *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*, pp. 30–30, 2005.

[124] H. Ringberg, A. Soule, J. Rexford, and C. Diot, "Sensitivity of pca for traffic anomaly detection," in *Proceedings of the 2007 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pp. 109–120, 2007.

[125] H. Kasai, W. Kellerer, and M. Kleinsteuber, "Network volume anomaly detection and identification in large-scale networks based on online time-structured traffic tensor tracking," *IEEE Transactions on Network and Service Management*, vol. 13, pp. 636–650, Sept 2016.

[126] X. Li, F. Bian, M. Crovella, C. Diot, R. Govindan, G. Iannaccone, and A. Lakhina, "Detection and identification of network anomalies using sketch subspaces," in *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, pp. 147–152, 2006.

[127] B. Krishnamurthy, S. Sen, Y. Zhang, and Y. Chen, "Sketch-based change detection: methods, evaluation, and applications," in *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, pp. 234–247, 2003.

[128] M. Thottan and C. Ji, "Anomaly detection in ip networks," *IEEE Transactions on signal processing*, vol. 51, no. 8, pp. 2191–2204, 2003.

[129] Y. Gu, A. McCallum, and D. Towsley, "Detecting anomalies in network traffic using maximum entropy estimation," in *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*, pp. 32–32, USENIX Association, 2005.

[130] S. Ranshous, S. Shen, D. Koutra, S. Harenberg, C. Faloutsos, and N. F. Samatova, "Anomaly detection in dynamic networks: a survey," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 7, no. 3, pp. 223–247, 2015.

[131] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: a survey," *Data mining and knowledge discovery*, vol. 29, no. 3, pp. 626–688, 2015.

[132] H. E. Egilmez and A. Ortega, "Spectral anomaly detection using graph-based filtering for wireless sensor networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pp. 1085–1089, IEEE, 2014.

[133] M. Khatua, S. H. Safavi, and N. Cheung, "Sparse laplacian component analysis for internet traffic anomalies detection," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, pp. 697–711, Dec 2018.

[134] Y.-J. Lee, Y.-R. Yeh, and Y.-C. F. Wang, "Anomaly detection via online oversampling principal component analysis," *IEEE transactions on knowledge and data engineering*, vol. 25, no. 7, pp. 1460–1470, 2013.

[135] K. P. Murphy, *Machine Learning: A Pobabilistic Perspective*. MIT press, 2012.

[136] X. Yi and C. Caramanis, "Regularized em algorithms: A unified framework and statistical guarantees," in *Advances in Neural Information Processing Systems*, pp. 1567–1575, 2015.

[137] S. Kullback, *Information theory and statistics*. 1997.

[138] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. 2006.

[139] S. Basu, "Bayesian hypotheses testing using posterior density ratios," *Statistics & probability letters*, vol. 30, no. 1, pp. 79–86, 1996.

[140] G. Casella and R. Berger, *Statistical Inference*. Duxbury Thomson Learning, 2005.

[141] NYC Taxi & Limousine Commission, "TLC trip record data." http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml.

[142] T. W. Schneider, "Unified new york city taxi and uber data." Github, 2017. https://github.com/toddwschneider/nyc-taxi-data.

[143] R. Harshbarger, "Tlc cracking down on drivers who illegally pick up street hails," *New York Post*, June 10 2014.

[144] S. Watanabe, *Algebraic Geometry and Statistical Learning Theory*, vol. 25. Cambridge University Press, 2009.

[145] S. S. Wilks, "The large-sample distribution of the likelihood ratio for testing composite hypotheses," *The Annals of Mathematical Statistics*, vol. 9, no. 1, pp. 60–62, 1938.

[146] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Found. Trends Mach. Learn.*, vol. 1, pp. 1–305, Jan. 2008.

[147] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh, "Clustering with bregman divergences," *J. Mach. Learn. Res.*, vol. 6, pp. 1705–1749, Dec. 2005.