ONLINE MINIMAX OPTIMAL DENSITY ESTIMATION AND ANOMALY DETECTION IN NONSTATIONARY ENVIRONMENTS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

By
Kaan Gökcesu
July 2017

ONLINE MINIMAX OPTIMAL DENSITY ESTIMATION AND ANOMALY DETECTION IN NONSTATIONARY ENVIRONMENTS

By Kaan Gökcesu

July 2017

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Süleyman Serdar Kozat(Advisor)

Ezhan Karaşan

Aydın Alatan

Approved for the Graduate School of Engineering and Science:

Ezhan Karaşan
Director of the Graduate School

ABSTRACT

ONLINE MINIMAX OPTIMAL DENSITY ESTIMATION AND ANOMALY DETECTION IN NONSTATIONARY ENVIRONMENTS

Kaan Gökcesu
M.S. in Electrical and Electronics Engineering
Advisor: Süleyman Serdar Kozat
July 2017

Online anomaly detection has attracted significant attention in recent years due to its applications in network monitoring, cybersecurity, surveillance and sensor failure. To this end, we introduce an algorithm that sequentially processes data to detect anomalies in time series. Our algorithm consists of two stages: density estimation and anomaly detection. First, we construct a probability density function to model the normal data. Then, we threshold the density of the newly observed data to detect anomalies. We approach this problem from an information theoretic perspective and, for the first time in the literature, propose minimax optimal schemes for both stages to create an optimal anomaly detection algorithm in a strong deterministic sense. For the first stage, we introduce an online density estimator that is minimax optimal for general nonstationary exponential-family of distributions without any assumptions on the observation sequence. Our algorithm does not require a priori knowledge of the time horizon, the drift of the underlying distribution or the time instances the parameters of the source changes. Our results are guaranteed to hold in an individual sequence manner. For the second stage, we propose an online threshold selection scheme that has logarithmic performance bounds against the best threshold chosen in hindsight. Our complete algorithm adaptively updates its parameters in a truly sequential manner to achieve log-linear regrets in both stages. Because of its universal prediction perspective on its density estimation, our anomaly detection algorithm can be used in unsupervised, semi-supervised and supervised manner. Through synthetic and real life experiments, we demonstrate substantial performance gains with respect to the state-of-the-art.

Keywords: Anomaly Detection, Time Series, Online Learning, Density Estimation, Minimax Optimal, Nonstationary.

ÖZET

DURAĞAN OLMAYAN ORTAMLARDA ÇEVRİMİÇİ MİNİMAKS OPTİMAL YOĞUNLUK TAHMİNİ VE ANOMALİ TESPİTİ

Kaan Gökcesu
Elektrik ve Elektronik Mühendisliği, Yüksek Lisans
Tez Danışmanı: Süleyman Serdar Kozat
Temmuz 2017

Çevrimiçi anomali tespiti, ağ izleme, siber güvenlik, gözetleme ve sensör arızalarındaki uygulamaları nedeniyle son yıllarda büyük ilgi gördü. Bu amaçla, zaman serilerindeki anomalileri saptamak için verileri sıralı olarak işleyen bir algoritma sunmaktayız. Algoritmamız iki aşamadan oluşmaktadır: yoğunluk tahmini ve anomali tespiti. İlk olarak, normal veriyi modellemek için bir olasılık yoğunluk fonksiyonu oluşturuyoruz. Daha sonra, anomalileri tespit etmek için yeni gözlemlerin yoğunluğunu bir eşik değeriyle karşılaştırıyoruz. Bu problemi çözerken bilişim teorisi yaklaşımı kullanıyoruz ve literatürde ilk defa, güçlü ve deterministik olarak optimal bir anomali tespit algoritması yaratmak için her iki aşamada da minimaks optimal yöntemler önermekteyiz. İlk aşamada, gözlemlenen verilerin üzerinde herhangi bir varsayımda bulunmaksızın durağan olmayan ve üstel aileye ait olan olasılıksal dağılımlar için minimaks optimal olan çevrimiçi yoğunluk tahmin algoritmasını sunuyoruz. Algoritmamızın, zaman ufkunu, temel dağılımın değişimini veya kaynak parametrelerinin değiştiği zamanları önceden bilmesine gerek yoktur. Sonuçlarımızın herhangi bir gözlem sekansı için tutulması garanti edilmektedir. Algoritmamızın ikinci aşaması için, seçilebilecek en iyi eşik değerine karşı logaritmik performans sınırlarına sahip bir çevrimiçi eşik belirleme yöntemi önermekteyiz. Algoritmamız her iki aşamada da log-lineer pişmanlıklar elde etmek için parametrelerini çevrimiçi bir şekilde günceller. Yoğunluk tahmini aşamasında evrensel tahmin perspektifi kullandığımız için, anomali tespit algoritmamız gözetimsiz, yarı gözetimli veya gözetimli şekilde kullanılabilir. Sentetik ve gerçek verilerdeki deneylerle önemli performans kazanımları gösteriyoruz.

Anahtar sözcükler: Anomali Tespiti, Zaman Serileri, Çevrimiçi Öğrenme, Yoğunluk Tahmini, Minimaks Optimal, Durağan Olmayan.

Acknowledgement

First and foremost, I would like to express my deepest gratitude to my supervisor Assoc. Prof. Süleyman Serdar Kozat for his continuous support, patience and encouragement for the research and my self improvement. His invaluable experience and guidance have been of great importance to me.

I would also like to thank Prof. Ezhan Karaşan and Prof. Aydın Alatan as my examining committee members and for providing constructive comments.

I would like to extend my deepest gratitude to my beloved family who have always supported me.

I would like to thank my friends for their fruitful discussions and help in the progress of my research.

I would like to acknowledge the financial support of the Scientific and Technological Research Council of Turkey (TUBITAK) during my M.Sc. studies under the BIDEB 2210-A Scholarship Programme with sincere gratitudes.

Contents

List of Figures

Chapter 1

Introduction

1.1 Preliminaries

We study anomaly detection [2], which has attracted significant attention in recent years due to its applications in network monitoring [3], cybersecurity [4], surveillance [5] and sensor failure [6]. Particularly, we study the sequential anomaly detection problem, where at each time t, we sequentially observe a vector $\boldsymbol{x}_t \in \mathcal{X}$ such that $\mathcal{X} \subset \mathbb{R}^m$ and our aim is to decide whether this new observation is anomalous or not, based on the past observations $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{t-1}\}$.

A common approach to this problem is to find a typical set containing the most likely instances in an unknown measure of probability, where "the normal data" is commonly assumed to be generated from an independent and identically distributed (i.i.d) random variable sequence belonging to this unknown measure [2]. In the density level set estimation framework [7], the theoretically optimal nominal set would consist of the samples whose probability under the unknown measure is greater than some density level set parameter in accordance with the fraction of outliers in the model [7]. Therefore, we first investigate the sequential density estimation (or learning) problem, which arise in several different machine learning applications such as pattern recognition [8, 9], novelty detection [10],

unsupervised classification [11], prediction [12], source coding [13], data mining [14,15], big data [16], feature selection [17], feature extraction [18], dimensionality reduction [19] and anomaly detection [20].

However, in general, the observations $\boldsymbol{x}_t$ are prone to distortion (because of a less than ideal communication channel and noise contamination) and may not be the outputs of a stochastic process yet alone independent and identically distributed since the environment possibly exhibits nonstationary behavior and may even be chaotic or adversarial [20]. Therefore, to solve this otherwise difficult modeling of the nominal data, we approach the density estimation problem from a competitive algorithm perspective [21–25] and design algorithms that perform as well as the best density estimator chosen in hindsight (with possibly changing statistics) for any observation sequence. We show that our approach has strong performance guarantees in an individual sequence manner [25]. Thus, our algorithm is able to perform as well as the optimal nonstationary density in our competition class even if there is no underlying stochastic process. Since these algorithms work in an individual sequence manner, they are robust against distorted and dependent observations generated in a dynamically evolving environment as opposed to the explicit design and modeling of an evolving system [26, 27].

Even though there exist several nonparametric approaches to model the distribution of the nominal data [28–30], parametric models are usually more practical because of their faster learning behavior and high modeling powers [29]. The parametric models can suffer if only the assumed model is incapable of continuously modeling the data [2] in a robust manner. Therefore, we model the source that generates the nominal data as an exponential-family distribution [31, 32], since the exponential-family of distributions cover a wide range of parametric statistical models [20] and accurately approximates many nonparametric classes of probability densities [33]. By using online convex optimization [34] to estimate the natural parameter of this exponential-family distribution, we can achieve logarithmic regret bounds under log-loss, i.e., $O(\log T)$ (minimax optimal) [20, 35]. However, most real life applications such as in cybersecurity [4] or surveillance applications [5], the underlying data stream is nearly always nonstationary [2, 36], i.e., has varying statistics over time [37]. Hence, we model the nonstationary

source model (i.e., the changing natural parameter) as piecewise stationary models. We emphasize that there are no restrictions on the number of regions or the length of these regions that are needed for accurate modeling. By combining these stationary models in a switching experts setting [38–41], we achieve the minimax optimal performance bound $O(C \log T)$, when the statistics of the source, i.e., the natural parameter, changes $C-1$ number of times, i.e., the nonstationary model consists of C stationary models (C time segments with piecewise constant parameters).

After the minimax optimal modeling of the distribution of the nominal data in the time series, we produce our anomaly decision by thresholding the density estimation, which optimally minimizes Type-1 errors in certain environments [28,42]. Such anomaly detection methods with two stages, i.e., scoring the time series samples and thresholding them, are extensively studied in the literature [2, 20, 43]. Even though the detection of anomalous samples when their likelihood (probability or score) falls below a certain threshold is an efficient and popular strategy, the selection of this threshold is a notoriously challenging problem [20]. Therefore, we implement a dynamic thresholding scheme, which updates the threshold whenever a feedback about the observed sample is available (i.e., whether it is anomalous or not). We update our threshold using Online Newton Step [35] and achieve again logarithmic performance bounds (minimax optimal) with respect to the best threshold selected in hindsight.

Even though there exists various methods to detect anomalies in a time series, we, for the first time in literature, propose a truly minimax optimal (both in density estimation and threshold selection) online anomaly detection algorithm in nonstationary settings. Through synthetic and real-life experiments, we demonstrate significant performance gains with respect to the state-of-the-art methods in the literature [20, 34, 42, 44].

1.2 Prior Art and Comparisons

Various anomaly detection methods have been proposed in the literature that utilizes Support Vector Machines (SVM) [45, 46], nearest neighbors approach [47, 48], clustering [49] and density estimation [50, 51]. However, techniques based on probability density estimation are demonstrated to provide superior performance when "the normal data" conforms to a nominal distribution and the unknown probability measure can be estimated near perfect [52].

Moreover, even though there exist several anomaly detection methods [2] proposed for supervised [53], semi-supervised [54] and unsupervised [7] settings, none of them has performance bounds in nonstationary or adversarial settings [20].

For these reasons, we adopt the probability density based approach in an individual sequence perspective [25] to provide strong deterministic bounds both in density estimation and anomaly detection stages of our algorithm.

In the literature, there are various methods that can achieve sublinear performance bounds when estimating the density in nonstationary environments.

- In [34], authors propose an approach that can achieve a regret bound of $O(\sqrt{CT})$ when the time horizon T and the total change C is known a priori.
- Without the prior knowledge of the time horizon T, the algorithm in [34] can still achieve $\sqrt{CT}$ regret if it is run in accordance with the doubling trick [24].
- One can also modify the algorithm of [34] to achieve a regret bound of $O(\sqrt{C_{\max}T})$ if an upper bound $C_{\max}$ on C is known instead such that $C_{\max} \geq C$.
- For the case of no prior knowledge about C, an algorithm that achieves a regret bound of $O(C\sqrt{T})$ is proposed in [20].

Nonetheless, none of these methods achieve the minimax regret bound $O(C \log T)$ [55]. In [40] and [41], the authors propose methods to achieve this minimax optimal regret bound only in binary sources and discrete sources respectively.

Achieving the minimax optimal regret bound $O(C \log T)$ is not possible with the state-of-the-art methods when the source belongs to a general exponential-family.

To this end, we introduce a truly online density estimation algorithm that can achieve the minimax optimal regret bound $O(C \log T)$ without any knowledge of C and T beforehand. The computational complexity and storage demand of our minimax optimal algorithm is linear in time.

In anomaly detection with thresholding some likelihood function (such as in the probability density based methods), the optimal selection of the threshold is intrinsically difficult [20]. Therefore, instead of committing to a fixed threshold, we adopt a dynamically selected thresholding scheme and update our threshold in accordance with the feedback.

Most of the algorithms in literature do not provide guaranteed regret bounds in this setting. In the literature [20], performance bounds of only $O(\sqrt{T})$ are achieved with respect to the best threshold selected in hindsight. However, such a regret bound for the anomaly detection performance invalidates the purpose of estimating the density function of the normal data with minimax optimal, i.e., log-linear, ($O(C \log T)$) regret.

Therefore, we propose a thresholding scheme that achieves logarithmic (minimax) regret bound, i.e., $O(\log T)$ regret, to provide a truly minimax optimal anomaly detection algorithm.

1.3 Contributions

Our contributions are as follows:

1. For the first time in the literature, we propose a truly minimax optimal anomaly detection algorithm for nonstationary sources with logarithmic performance bounds by optimizing both the density estimation and threshold selection in an individual sequence manner.

2. Our algorithm can be used in unsupervised, semi-supervised and supervised manner because of its individual sequence (i.e., universal prediction) perspective on its density estimation.

3. Our algorithm can assign distinct costs to making a prediction error on normal and anomalous data since in general their importance are not the same in various applications.

4. For the first time in literature, we propose a density estimation algorithm that achieves the minimax optimal regret $O(C \log T)$ for general exponential-family of sources. Our density estimation algorithm improves upon the source coding literature and asymptotically achieves Merhav's lower bound [55] for any piecewise stationary exponential-family source.

5. Our proposed adaptive thresholding scheme achieves log-linear regret against the best threshold chosen in hindsight and improves greatly upon the $\sqrt{T}$ regret scheme in the literature [20].

6. Our results are uniformly guaranteed to hold in a strong deterministic sense in an individual sequence manner for all possible observation sequences.

7. Our algorithm is strongly sequential such that neither the time horizon T nor the number of changes C of the source statistics are known.

8. Through extensive set of experiments involving synthetic and real datasets, we demonstrate significant performance gains achieved by the proposed algorithm for both density estimation and anomaly detection with respect to the state-of-the-art methods.

1.4 Organization

First, we formally define our problem setting in Chapter 2.

In Chapter 3, we propose a density estimation method that tracks the natural parameter of an exponential-family distribution. In Chapter 3.1, we prove how we can achieve sublinear regret bounds using convex optimization. In Chapter 3.2, we show that by optimizing the learning rate with a universal mixture approach, we can achieve the performance of the optimal learning rate. In Chapter 3.3, we provide some numerical results, which demonstrates that such a universal mixture outperforms the state-of-the-art adaptive methods.

In Chapter 4, we construct a minimax optimal density estimation algorithm. In Chapter 4.1, we show that by using dynamic learning rate decaying with time, we can achieve logarithmic regret bounds for stationary memoryless sources. In Chapter 4.2, we eliminate the requirement of any a priori knowledge of the statistics of the source. In Chapter 4.3, we introduce a minimax optimal density estimator for nonstationary sources.

In Chapter 5, we propose a completely online anomaly detection algorithm that adaptively thresholds the density estimation. We achieve this by first substituting the binary loss with the logistic loss function in Chapter 5.1. Then, in Chapter 5.2, we propose our threshold update scheme which utilizes Online Newton Step.

In Chapter 6, we illustrate significant performance gains in density estimation and anomaly detection over both real and synthetic data and finish with concluding remarks in Chapter 7.

Chapter 2

Problem Description

In this chapter, we formally define and outline the problem.

We introduce an online algorithm for anomaly detection in time series that sequentially observes some $\{\boldsymbol{x}_t\}_{t\geq 1}$ such that $\boldsymbol{x}_t \in \mathbb{R}^m$, at each time t, and then, estimates the probability of the unseen data (observations) based on our previous observations.

Based on this probability estimate, we decide whether the newly observed data is anomalous or not by comparing this estimated density with a specific threshold [2, 28].

We assume that the normal samples that we observe in the time series are generated from a nonstationary source model with piecewise constant source parameters.

We can represent such a density function as $f_t(\cdot)$ whose source parameters are given by the vector $\boldsymbol{\alpha}_t$. Since the source at hand has piecewise constant parameters, the source parameters $\boldsymbol{\alpha}_t$ remains unchanged for some segment of time indices, i.e., $\boldsymbol{\alpha}_t = \boldsymbol{\alpha}_{t+1}$, if both t and $t+1$ are in the same time segment where the source statistics do not change.

We represent the number of changes of the source statistics in a T length observation sequence by C which is given by

$$C \triangleq 1 + \sum_{t=2}^{T} \mathbb{1}_{\boldsymbol{\alpha}_t \neq \boldsymbol{\alpha}_{t-1}}, \tag{2.1}$$

where the beginning is also counted as a change and $\mathbb{1}_x$ is the indicator function that outputs 1 if the statement x is true and 0 otherwise. As an example, for stationary sources, i.e., distributions with unchanging natural parameter, the number of changes C is 1.

We denote the total drift of $\boldsymbol{\alpha}_t$ in T rounds by C_α such that

$$C_\alpha \triangleq \sum_{t=2}^{T} \|\boldsymbol{\alpha}_t - \boldsymbol{\alpha}_{t-1}\|, \tag{2.2}$$

where $\|\cdot\|$ is the L^2-norm. As an example, for stationary sources, i.e., distributions with unchanging natural parameter, the drift C_α is 0.

In our two stage algorithm, we first introduce a pdf estimation algorithm, which estimates the density of the nonstationary source that generates the normal data in the time series.

In the online setting, we observe a sample vector $\boldsymbol{x}_t$ at each time t, which is distributed according to some unknown density function $f_t(\cdot)$. We estimate this unknown density $f_t(\boldsymbol{x}_t)$ based on our past observations $\{\boldsymbol{x}_r\}_{r=1}^{t-1}$ and produce an estimate $\hat{f}_t(\boldsymbol{x}_t)$.

We approach this problem of estimating the density of the nonstationary source from a competitive algorithm perspective where the competing strategy is naturally given by the true density function (if such a distribution exists) or the density function that best represents the data.

To this end, as the performance measure (i.e., the loss function in our competitive framework), we use the log-loss of the density functions, i.e., $l(\hat{f}_t(\boldsymbol{x}_t)) = -\log(\hat{f}_t(\boldsymbol{x}_t))$, since it is the most widely used loss function for probability distributions [56].

We use the notion of "regret" on the log-losses to define our performance in an individual sequence manner [24], such that the regret at time t is

$$r_t = -\log(\hat{f}_t(\boldsymbol{x}_t)) + \log(f_t(\boldsymbol{x}_t)), \tag{2.3}$$

and the cumulative regret up to time T is

$$R_T = \sum_{t=1}^{T} \Big(-\log(\hat{f}_t(\boldsymbol{x}_t)) + \log(f_t(\boldsymbol{x}_t)) \Big). \tag{2.4}$$

This regret formulation is motivated by the Kullback-Leibler divergence (relative entropy) [57], which is often used for measuring the distance between probability distributions. An alternative form of the regret is given by

$$r_t = \log(\frac{f_t(\boldsymbol{x}_t)}{\hat{f}_t(\boldsymbol{x}_t)}), \tag{2.5}$$

which is the logarithm of the likelihood ratio. Hence, the cumulative regret at time T is given by

$$\begin{aligned} R_T &= \sum_{t=1}^{T} r_t \\ &= \sum_{t=1}^{T} \log(\frac{f_t(\boldsymbol{x}_t)}{\hat{f}_t(\boldsymbol{x}_t)}). \end{aligned} \tag{2.6}$$

Thus, the average regret asymptotically converges to the KL divergence by law of large numbers. The instantaneous regret definition in (2.5) can either be positive or negative in a specific round just like in any other expert competition settings [58]. However, the cumulative regret in (2.6) is bound to be positive since the competition (i.e., true distribution) minimizes the cumulative log-loss.

Our goal is to achieve the performance of the best distribution with piecewise constant parameters where each segment is generated by an exponential-family distribution, since the exponential-family of distributions cover a wide range of parametric statistical models [20] and accurately approximates many nonparametric classes of probability densities [33]. For exponential-family of sources, the source statistics $\boldsymbol{\alpha}_t$ is called the natural parameter of the density function. Hence, the true density function $f_t(\cdot)$ is given by an exponential-family of distribution with possible changing natural parameter $\boldsymbol{\alpha}_t$.

After modeling the normal data in the time series by creating a minimax optimal density estimation $\hat{f}_t(\boldsymbol{x}_t)$, we produce our anomaly decision by thresholding $\hat{f}_t(\boldsymbol{x}_t)$, which optimally minimizes Type-1 errors in certain environments [28, 42]. Hence, the binary anomaly decision $\hat{d}_t$ is constructed as

$$\hat{d}_t = \begin{cases} +1, & \hat{f}_t(\boldsymbol{x}_t) < \tau_t \text{ (anomalous)} \\ -1, & \hat{f}_t(\boldsymbol{x}_t) \geq \tau_t \text{ (not anomalous)} \end{cases}, \tag{2.7}$$

where τ_t is a time varying threshold. We compete against the best threshold chosen in hindsight τ^*. Thus, the regret incurred up to time T by our anomaly detector is given by

$$R_{A,T} = \sum_{t=1}^{T} l_A(\hat{f}_t(\boldsymbol{x}_t), \tau_t, d_t) - \sum_{t=1}^{T} l_A(\hat{f}_t(\boldsymbol{x}_t), \tau^*, d_t), \tag{2.8}$$

where d_t are the true labels (anomaly or not) of the observations and the loss function $l_A(\cdot)$ is defined as

$$l_A(\hat{f}_t(\boldsymbol{x}_t), \tau, d_t) = \mathbb{1}_{\text{sign}(\tau - \hat{f}_t(\boldsymbol{x}_t)) \neq d_t}. \tag{2.9}$$

Our aim is to achieve minimax optimal regret bounds for both (2.4) and (2.8). To this end, in Chapter 4.1, we introduce a minimax optimal density estimator that achieves logarithmic regret bound for stationary memoryless sources. In Chapter 4.2, we use these density estimators as our building blocks to eliminate the requirement of any a priori knowledge of the statistics of the source. In Chapter 4.3, we introduce a minimax optimal density estimator for nonstationary sources and achieve the minimax optimal regret bound $O(C \log T)$ for (2.4). In Chapter 5, we propose a completely online anomaly detection algorithm that adaptively thresholds the density estimation, which achieves the minimax optimal regret bound for (2.8), i.e., $O(\log T)$ regret against the best fixed threshold selected in hindsight.

Chapter 3

Density Estimation with Convex Optimization

In this chapter, we seek to achieve the performance of the best nonstationary distribution from an exponential family by estimating (tracking) its natural parameter.

In this sense, we assume that there exists a density function $f_t(\boldsymbol{x}_t)$ that exactly or most closely represents the true distribution that generates the observation sequence.

We assume that the function $f_t(\boldsymbol{x}_t)$ belongs to an exponential-family [31] with a possibly changing natural parameter vector $\boldsymbol{\alpha}_t \in \mathbb{R}^d$ (which cumulatively representing the mean, sufficient statistics, normalization, etc.) at each time instance t.

Here, at each time t, we observe $\boldsymbol{x}_t \in \mathbb{R}^{d_x}$, which is distributed according to a memoryless (i.e., independent from the past samples) exponential-family distribution

$$f_t(\boldsymbol{x}_t) = \exp\left(-\langle \boldsymbol{\alpha}_t, \boldsymbol{z}_t \rangle - A(\boldsymbol{\alpha}_t)\right). \tag{3.1}$$

Here $\boldsymbol{\alpha}_t \in \mathbb{R}^d$ is the natural parameter of the distribution and $\langle \cdot, \cdot \rangle$ is the inner product.

The natural parameter $\boldsymbol{\alpha}_t$ belongs to a bounded convex feasible set S such that

$$D = \max_{\alpha \in S} \|\alpha\|. \tag{3.2}$$

The function $A(\cdot)$ in (3.1) is the normalization or log-partition function such that

$$A(\alpha) = \log \left(\int_{\mathcal{X}} \exp(-\langle \alpha, \mathcal{T}(x) \rangle) dx \right), \tag{3.3}$$

$\boldsymbol{z}_t$ in (3.1) is the d-dimensional sufficient statistic of $\boldsymbol{x}_t$ [31]. The sufficient statistics $\boldsymbol{z}_t$ is given after a transformation (i.e., $\mathcal{T}(\cdot)$) of the observation $\boldsymbol{x}_t$ such that

$$\boldsymbol{z}_t = \mathcal{T}(\boldsymbol{x}_t). \tag{3.4}$$

In Chapter 3.1, we first introduce the basic density estimator, which tracks the natural parameter $\boldsymbol{\alpha}_t$ using convex optimization. Unfortunately, the performance of this basic density estimator is highly dependent on its learning rate (i.e., step size).

Hence, in Chapter 3.2, we introduce a universal density estimator that merges the beliefs of these basic density estimators to achieve the performance of the optimal learning rate.

In Chapter 3.3, we illustrate significant performance gains over both real and synthetic data in comparison to the other adaptive methods and density estimators.

3.1 Tracking the Natural Parameter with Convex Optimization

In this section, we first construct basic density estimators that can only achieve good regret bounds by optimizing its learning rate with a priori information (e.g., C_α, T). These estimators are used in Chapter 3.2 to construct the algorithm, which achieves these regrets without requiring any information for optimization.

Instead of directly estimating the density $f_t(x)$, we estimate the natural parameter $\boldsymbol{\alpha}_t$ at each time t according to $\{\boldsymbol{x}_\tau\}_{\tau=1}^{t-1}$. The estimated density is

$$\hat{f}_t(\boldsymbol{x}_t) = \exp(-\langle \hat{\boldsymbol{\alpha}}_t, \boldsymbol{z}_t \rangle - A(\hat{\boldsymbol{\alpha}}_t)). \tag{3.5}$$

We use online gradient descent [34] to sequentially produce our estimation $\hat{\boldsymbol{\alpha}}_t$, where we start from an initial estimate $\hat{\boldsymbol{\alpha}}_1$, and update $\hat{\boldsymbol{\alpha}}_t$ based on $\boldsymbol{x}_t$. To update $\hat{\boldsymbol{\alpha}}_t$, we first observe a sample $\boldsymbol{x}_t$ and incur the loss $l(\hat{\boldsymbol{\alpha}}_t, \boldsymbol{x}_t)$ according to our estimation $\hat{\boldsymbol{\alpha}}_t$, which is $-\log(\hat{f}_t(\boldsymbol{x}_t))$ (log-loss). From (3.5), the loss is

$$l(\hat{\boldsymbol{\alpha}}_t, \boldsymbol{x}_t) = \langle \hat{\boldsymbol{\alpha}}_t, \boldsymbol{z}_t \rangle + A(\hat{\boldsymbol{\alpha}}_t). \tag{3.6}$$

Then, we calculate the gradient of the loss with respect to $\hat{\boldsymbol{\alpha}}_t$,

$$\begin{aligned} \nabla_{\boldsymbol{\alpha}} l(\hat{\boldsymbol{\alpha}}_t, \boldsymbol{x}_t) &= \boldsymbol{z}_t + \nabla_{\boldsymbol{\alpha}} A(\hat{\boldsymbol{\alpha}}_t), \\ &= \boldsymbol{z}_t + \frac{\int_{\mathcal{X}} -\mathcal{T}(x) \exp(-\langle \hat{\boldsymbol{\alpha}}_t, \mathcal{T}(x) \rangle) dx}{\int_{\mathcal{X}} \exp(-\langle \hat{\boldsymbol{\alpha}}_t, \mathcal{T}(x) \rangle) dx} \\ &= \boldsymbol{z}_t - \mu_{\hat{\boldsymbol{\alpha}}_t}, \end{aligned} \tag{3.7}$$

where we used the definition in (3.3) in the second equality and $\mu_{\hat{\boldsymbol{\alpha}}_t}$ is the mean of $\mathcal{T}(\boldsymbol{x}_t)$ if $\boldsymbol{x}_t$ were distributed according to $\hat{f}_t(\boldsymbol{x}_t)$ as in (3.5). We update $\hat{\boldsymbol{\alpha}}_t$ as

$$\hat{\boldsymbol{\alpha}}_{t+1} = P_S(\hat{\boldsymbol{\alpha}}_t - \eta(\boldsymbol{z}_t - \mu_{\hat{\boldsymbol{\alpha}}_t})), \tag{3.8}$$

where $P_S(\cdot)$ is the projection onto the set S and is defined as

$$P_S(x) = \arg\min_{y \in S} \|x - y\| \tag{3.9}$$

The complete algorithm is provided in Alg. 1.

Algorithm 1 Basic Density Estimator

Initialize learning rate $\eta \in \mathbb{R}^+$
Select initial parameter $\hat{\boldsymbol{\alpha}}_1$
Calculate the mean $\mu_{\hat{\boldsymbol{\alpha}}_1}$
for $t = 1$ **to** T **do**
    Declare estimation $\hat{\boldsymbol{\alpha}}_t$
    Observe $\boldsymbol{x}_t$
    Calculate $\boldsymbol{z}_t = \mathcal{T}(\boldsymbol{x}_t)$
    Update parameter: $\tilde{\alpha}_{t+1} = \hat{\boldsymbol{\alpha}}_t - \eta(\boldsymbol{z}_t - \mu_{\hat{\boldsymbol{\alpha}}_t})$
    Project onto convex set: $\hat{\boldsymbol{\alpha}}_{t+1} = P_S(\tilde{\alpha}_{t+1})$
    Calculate the mean $\mu_{\hat{\boldsymbol{\alpha}}_{t+1}}$
end for

Next, we provide performance bounds of Alg. 1. Theorem 1 shows that Alg. 1 can achieve the minimum regret bound $O(\sqrt{C_\alpha T})$ if C_α is known a priori to optimize η.

Theorem 1. *When Alg. 1 is used with parameter η to estimate the distribution $f_t(\boldsymbol{x}_t)$, its regret is upper bounded by*

$$R_T \leq \frac{1}{\eta} DC + \eta TG,$$

where D is defined as in (3.2), $C = 2.5D + C_\alpha$ such that C_α is defined as in (2.2), and $G = (\phi_2 + 2\phi_1 M + M^2)/2$ such that $M = \max_{\alpha \in S} \mu_\alpha$, $\phi_1 = \sum_{t=1}^{T} \|\boldsymbol{z}_t\|/T$, $\phi_2 = \sum_{t=1}^{T} \|\boldsymbol{z}_t\|^2/T$.

Proof of Theorem 1. The regret at time t is defined as

$$r_t = l(\hat{\boldsymbol{\alpha}}_t, \boldsymbol{x}_t) - l(\boldsymbol{\alpha}_t, \boldsymbol{x}_t), \tag{3.10}$$

where $l(\alpha, x)$ is as in (3.6). Since the loss function is convex

$$r_t \leq \langle \nabla_{\boldsymbol{\alpha}} l(\hat{\boldsymbol{\alpha}}_t, \boldsymbol{x}_t), (\hat{\boldsymbol{\alpha}}_t - \boldsymbol{\alpha}_t) \rangle. \tag{3.11}$$

We bound the right hand side of (3.11) using the update rule (3.8). By definition of projection in (3.9), we have

$$\|P_S(\hat{\boldsymbol{\alpha}}_t - \eta \nabla_{\boldsymbol{\alpha}} l(\hat{\boldsymbol{\alpha}}_t, \boldsymbol{x}_t)) - \boldsymbol{\alpha}_t\| \leq \|\hat{\boldsymbol{\alpha}}_t - \eta \nabla_{\boldsymbol{\alpha}} l(\hat{\boldsymbol{\alpha}}_t, \boldsymbol{x}_t) - \boldsymbol{\alpha}_t\|. \tag{3.12}$$

Substituting (3.8) in the left hand side provides

$$\|\hat{\boldsymbol{\alpha}}_{t+1} - \boldsymbol{\alpha}_t\| \leq \|\hat{\boldsymbol{\alpha}}_t - \eta \nabla_{\boldsymbol{\alpha}} l(\hat{\boldsymbol{\alpha}}_t, \boldsymbol{x}_t) - \boldsymbol{\alpha}_t\|. \tag{3.13}$$

Hence, we get

$$\begin{aligned}\|\hat{\boldsymbol{\alpha}}_{t+1} - \boldsymbol{\alpha}_t\|^2 \leq & \|\hat{\boldsymbol{\alpha}}_t - \boldsymbol{\alpha}_t\|^2 - 2\eta \langle \nabla_{\boldsymbol{\alpha}} l(\hat{\boldsymbol{\alpha}}_t, \boldsymbol{x}_t), (\hat{\boldsymbol{\alpha}}_t - \boldsymbol{\alpha}_t) \rangle \\ & + \eta^2 \|\nabla_{\boldsymbol{\alpha}} l(\hat{\boldsymbol{\alpha}}_t, \boldsymbol{x}_t)\|^2. \end{aligned} \tag{3.14}$$

Combining (3.11) and (3.14) results in

$$\begin{aligned} r_t &\leq \frac{1}{2\eta} \left(\|\hat{\boldsymbol{\alpha}}_t - \boldsymbol{\alpha}_t\|^2 - \|\hat{\boldsymbol{\alpha}}_{t+1} - \boldsymbol{\alpha}_t\|^2 \right) + \frac{\eta}{2} \|\nabla_{\boldsymbol{\alpha}} l(\hat{\boldsymbol{\alpha}}_t, \boldsymbol{x}_t)\|^2, \\ &\leq \frac{1}{2\eta} (\|\hat{\boldsymbol{\alpha}}_t\|^2 - \|\hat{\boldsymbol{\alpha}}_{t+1}\|^2 - 2\langle \hat{\boldsymbol{\alpha}}_t - \hat{\boldsymbol{\alpha}}_{t+1}, \boldsymbol{\alpha}_t \rangle) + \frac{\eta}{2} \|\nabla_{\boldsymbol{\alpha}} l(\hat{\boldsymbol{\alpha}}_t, \boldsymbol{x}_t)\|^2, \end{aligned} \tag{3.15}$$

since $\eta > 0$. Using (3.7) in the right hand side yields

$$r_t \leq \frac{1}{2\eta} (\|\hat{\boldsymbol{\alpha}}_t\|^2 - \|\hat{\boldsymbol{\alpha}}_{t+1}\|^2) - \frac{1}{\eta} \langle \hat{\boldsymbol{\alpha}}_t - \hat{\boldsymbol{\alpha}}_{t+1}, \boldsymbol{\alpha}_t \rangle + \frac{\eta}{2} \|\boldsymbol{z}_t - \mu_{\hat{\boldsymbol{\alpha}}_t}\|^2. \tag{3.16}$$

Thus, summing (3.16) from $t = 1$ to T, we have the cumulative regret up to time T, which is given by

$$\begin{aligned} R_T \leq & \frac{1}{2\eta} (\|\hat{\boldsymbol{\alpha}}_1\|^2 - \|\hat{\boldsymbol{\alpha}}_{T+1}\|^2) + \frac{\eta}{2} \sum_{t=1}^{T} \|\boldsymbol{z}_t - \mu_{\hat{\boldsymbol{\alpha}}_t}\|^2 \\ & - \frac{1}{\eta} \left(\langle \hat{\boldsymbol{\alpha}}_1, \boldsymbol{\alpha}_1 \rangle + \sum_{t=2}^{T} \langle \hat{\boldsymbol{\alpha}}_t, \boldsymbol{\alpha}_t - \boldsymbol{\alpha}_{t-1} \rangle - \langle \hat{\boldsymbol{\alpha}}_{T+1}, \boldsymbol{\alpha}_T \rangle \right). \end{aligned} \tag{3.17}$$

Using (2.2) and (3.2), we get

$$\begin{aligned} R_T &\leq \frac{1}{\eta} (2.5D^2 + DC_\alpha) + \frac{\eta}{2} \sum_{t=1}^{T} \left(\|z_t + \mu_{\hat{\boldsymbol{\alpha}}_t}\|^2 \right) \\ &\leq \frac{1}{\eta} (2.5D^2 + DC_\alpha) + \frac{\eta T}{2} \left(\phi_2 + 2\phi_1 M + M^2 \right), \end{aligned} \tag{3.18}$$

where, M, ϕ_1 and ϕ_2 are given by

$$M = \max_{\alpha \in S} \mu_\alpha, \;\; \phi_1 = \sum_{t=1}^{T} \frac{\|\boldsymbol{z}_t\|}{T}, \;\; \phi_2 = \sum_{t=1}^{T} \frac{\|\boldsymbol{z}_t\|^2}{T}.$$

We denote $G = (\phi_2 + 2\phi_1 M + M^2)/2$, which is related to the gradient of the log-loss and $C = C_\alpha + 2.5D$, which is the effective change parameter. Hence,

$$R_T \leq \frac{1}{\eta} DC + \eta TG, \tag{3.19}$$

which concludes the proof of the theorem.

□

The result in Theorem 1 is for an estimator that uses fixed learning rate, which will be used to prove the performance bound of the universal estimator in the next section.

Corollary 1. *If Alg. 1 is run with parameter* $\eta = \sqrt{(DC_0)/(G_0 T)}$*, it produces the regret*

$$R_T \leq \sqrt{DTG_0C_0}\left(\frac{C}{C_0} + \frac{G}{G_0}\right)$$

Remark 1. *If* $G_0 = G$ *and* $C_0 = C$ *in Corollary 1, then the regret bound is*

$$R_T \leq 2\sqrt{DCGT}.$$

Remark 2. *The construction of* $\boldsymbol{z}_t$ *requires the knowledge of sufficient statistics mapping* $\mathcal{T}(\cdot)$ *beforehand. Since the sufficient statistics of different kinds of exponential family distributions may differ,* $\mathcal{T}(\cdot)$ *requires the knowledge of the exact distribution kind, e.g. whether the distribution is normal, exponential, gamma etc. This requirement can be easily bypassed by creating an extended statistics vector* $\tilde{\boldsymbol{z}}_t = \tilde{\mathcal{T}}(\boldsymbol{x}_t)$ *such that* $\tilde{\boldsymbol{z}}_t$ *encompasses all sufficient statistics of different distributions that the true density may belong to. This would also solve the problem of estimating a distribution that changes types, e.g., from Gaussian to gamma, gamma to Bernoulli, etc. However, extending the sufficient statistics* $\tilde{\boldsymbol{z}}_t$ *effectively enlarges* S*, hence,* D*,* C*,* G *in Theorem 1 as well.*

Remark 3. *Suppose instead of* $\boldsymbol{x}_t$*, we observe a distorted version such that* $\boldsymbol{y}_t = Q(\boldsymbol{x}_t)$*, where* $Q(\cdot)$ *is the distortion channel, e.g., an additive noise channel. Then, using an unbiased estimator* $\bar{\boldsymbol{z}}_t = \bar{\mathcal{T}}(\boldsymbol{y}_t)$ *such that* $\mathbf{E}[\bar{\boldsymbol{z}}_t] = \mathcal{T}(\boldsymbol{x}_t)$ *produces the same results for expected regret.*

3.2 Aggregating the Estimators to Optimize the Learning Rate

In Chapter 3.1, we constructed the basic estimators that can only achieve the minimum regret bound with a priori information. In this section, we construct a universal density estimator (UDE) that achieves this regret with no a priori information by mixing the beliefs of different basic density estimators.

When used with parameter η, Alg. 1 achieves the regret

$$R_T \leq \sqrt{DCGT}\left(\frac{\eta_*}{\eta}+\frac{\eta}{\eta_*}\right), \tag{3.20}$$

where $\eta_* \triangleq \sqrt{(DC)/(GT)}$, which is the bound in Theorem 1. To achieve the minimum regret with Alg. 1, one must optimize η with some knowledge of η_*. However, with limited or no prior information, it is not possible to achieve the minimum regret using Alg. 1. Therefore, instead of just using Alg. 1 with a fixed learning rate, we combine different runs of Alg. 1 with different learning rates, which will approximate η_* to a sufficient degree to achieve the minimum regret.

To this end, we first construct a parameter vector $\boldsymbol{\eta}$ of size N such that $\boldsymbol{\eta}[r] = \eta_r$, for $r \in \{1, 2, \ldots, N\}$. We construct N experts each of which runs Alg. 1 with parameter η_r, i.e., r^{th} element of the parameter vector $\boldsymbol{\eta}$. As illustrated in Fig. 3.1, each one of the N experts takes the input $\boldsymbol{x}_t$ and outputs a belief $\hat{f}_t^r(x_t)$ at each round t (prediction stage). Then, we mix all of the beliefs as

$$\hat{f}_t^u(x_t) = \sum_{r=1}^{N} w_t^r \hat{f}_t^r(x_t), \tag{3.21}$$

where w_t^r is the combination weight of the belief of the r^{th} expert at time t (mixture stage). Initially we assign uniform weights to all expert outputs such that their combination weights are given by $w_1^r = 1/N$. Then, at each time t, we update their weights according to the rule

$$w_{t+1}^r = w_t^r \hat{f}_t^r(x_t)/\hat{f}_t^u(x_t), \tag{3.22}$$

where $\hat{f}_t^u(x_t)$ acts as the normalizer.

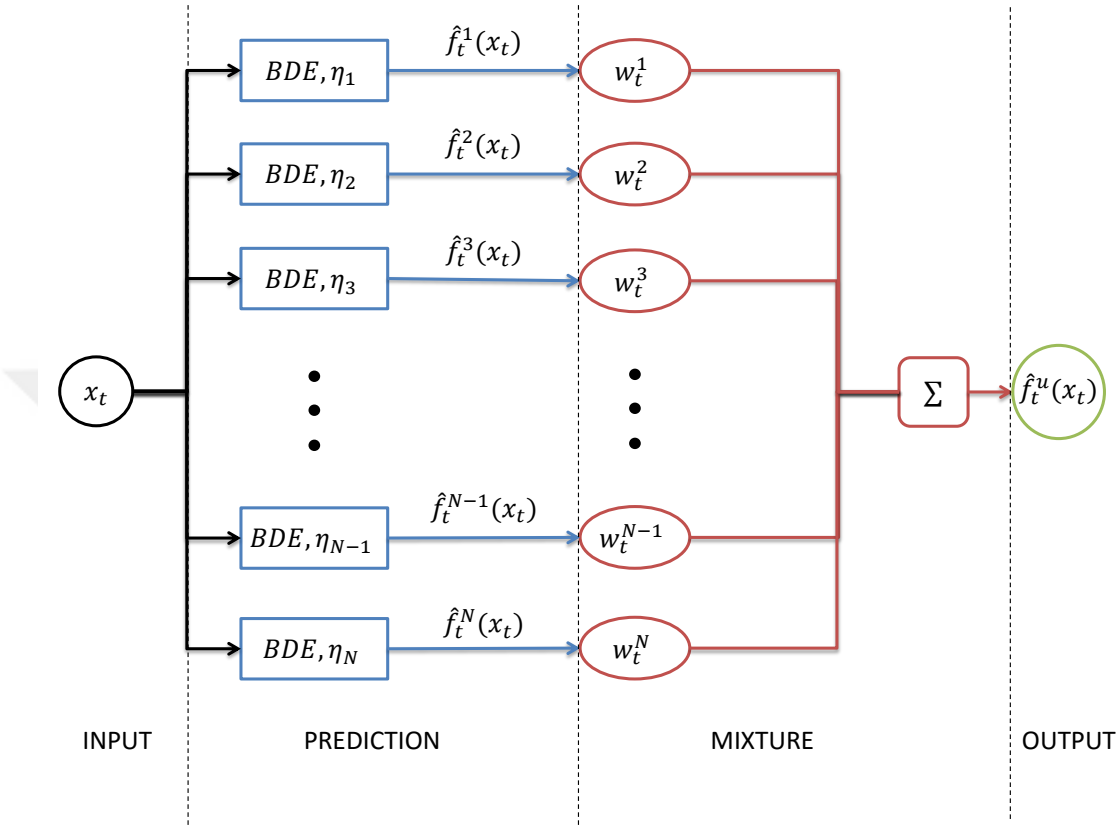

Figure 3.1: Illustration of Combining Basic (Component) Density Estimators to Create a Universal Density Estimator

Instead of our weighting, different methods [58–62] can also be used.

Our combination in (3.21), (3.22) makes use of the mixability property of density functions under log-loss (it is 1-mixable) [58], where $\log N$ redundancy is optimal [62–64].

We have provided a complete description of the universal algorithm in Alg. 2.

Next, we provide the performance bounds of the universal density estimator, i.e., Alg. 2.

The results of Theorem 2 and Corollary 2 show that the optimal regret bound $O(\sqrt{CT})$ is achieved without any prior information on C.

Algorithm 2 Universal Density Estimator

Initialize constants η_r, for $r \in \{1, 2, \ldots, N\}$
Create N copies Alg. 1, where the r^{th} algorithm runs with the parameter η_r and its belief is given by $\hat{f}_t^r(x)$.
Initialize weights $w_1^r = 1/N$
for $t = 1$ **to** T **do**
  Receive the beliefs $\hat{f}_t^r(x)$ for $r \in \{1, 2, \ldots, N\}$
  Declare estimation $\hat{f}_t^u(x) = \sum_{r=1}^N w_t^r \hat{f}_t^r(x)$
  Observe $\boldsymbol{x}_t$
  Calculate $\boldsymbol{z}_t = \mathcal{T}(\boldsymbol{x}_t)$
  for $r = 1$ **to** N **do**
    Update parameters $\hat{\boldsymbol{\alpha}}_t^r$ of r^{th} algorithm according to Alg. 1
    $w_{t+1}^r = w_t^r \hat{f}_t^r(x_t) / \hat{f}_t^u(x_t)$
  end for
end for

Theorem 2. *Alg. 2 has the regret bound*

$$R_T \leq \log(N) + \sqrt{DCGT} \left[\min_{i \in \{1,2,\ldots,N\}} \left(\frac{\eta_*}{\eta_i} + \frac{\eta_i}{\eta_*} \right) \right],$$

where D is defined as in (3.2), $C = 2.5D + C_\alpha$ such that C_α is defined as in (2.2), $G = (\phi_2 + 2\phi_1 M + M^2)/2$ such that $M = \max_{\alpha \in S} \mu_\alpha$, $\phi_1 = \sum_{t=1}^T \|\boldsymbol{z}_t\|/T$, $\phi_2 = \sum_{t=1}^T \|\boldsymbol{z}_t\|^2/T$, $\eta_* = \sqrt{DCGT}$ and η_i is the parameter of the i^{th} expert.

Proof of Theorem 2. The regret at time t is given by

$$r_t = -\log(\hat{f}_t^u(x_t)) + \log(f_t(\boldsymbol{x}_t)) \tag{3.23}$$

Summing (3.23) from $t = 1$ to T gives

$$R_T = -\log(\prod_{t=1}^T \hat{f}_t^u(x_t)) + \sum_{t=1}^T \log(f_t(\boldsymbol{x}_t)). \tag{3.24}$$

Using (3.21), we have

$$R_T = -\log(\prod_{t=1}^T (\sum_{r=1}^N w_t^r \hat{f}_t^r(x_t))) + \sum_{t=1}^T \log(f_t(\boldsymbol{x}_t)). \tag{3.25}$$

From (3.22), we can infer that the weights are given by

$$w_t^r = \frac{\prod_{\tau=1}^{t-1} \hat{f}_\tau^r(x_\tau)}{\sum_{r=1}^N \prod_{\tau=1}^{t-1} \hat{f}_\tau^r(x_\tau)}. \tag{3.26}$$

Hence, putting (3.26) in (3.25) produces,

$$\begin{aligned}
R_T &= -\log(\prod_{t=1}^{T}(\frac{\sum_{r=1}^{N}\prod_{\tau=1}^{t}\hat{f}_t^r(x_t)}{\sum_{r=1}^{N}\prod_{\tau=1}^{t-1}\hat{f}_t^r(x_t)})) + \sum_{t=1}^{T}\log(f_t(\boldsymbol{x}_t)) \\
&= -\log(\sum_{r=1}^{N}\prod_{\tau=1}^{T}\hat{f}_t^r(x_t)) + \log(N) + \sum_{t=1}^{T}\log(f_t(\boldsymbol{x}_t)) \\
&\leq \log(N) - \max_r(\sum_{t=1}^{T}\log(\hat{f}_t^r(x_t))) + \sum_{t=1}^{T}\log(f_t(\boldsymbol{x}_t)) \qquad (3.27) \\
&\leq \log(N) + \sqrt{DCGT}\left[\min_{i\in\{1,2,\ldots,N\}}\left(\frac{\eta_*}{\eta_i}+\frac{\eta_i}{\eta_*}\right)\right], \qquad (3.28)
\end{aligned}$$

and concludes the proof. □

The result of Theorem 2 shows that the performance bound is dependent on the set of learning rates used in the algorithm. In Corollary 2, we show that we can achieve the minimum regret bound with log-linear complexity.

Corollary 2. *Suppose we run the experts with parameters between η' and η''. We denote $K = \eta''/\eta'$ and $N = \lceil \log_2 K \rceil + 1$. Then, running Alg. 2 with parameter vector $\eta_i = 2^{i-1}\eta'$ for $i \in \{1, 2, \ldots, N\}$ gives the following regret bounds.*

1. *If $\eta' \leq \eta_* \leq \eta''$:*

$$R_T \leq \log(\lceil \log_2 \eta''/\eta' \rceil + 1) + \frac{3\sqrt{2}}{2}\sqrt{DCGT},$$

   since $(\frac{\eta_*}{\eta_i} + \frac{\eta_i}{\eta_*})$ is maximum if $\eta_* = 2^a\sqrt{2}$ for some a.

2. *If $\eta_* \geq \eta''$*

$$R_T \leq \log(\lceil \log_2 \eta''/\eta' \rceil + 1) + (1 + \frac{\eta_*}{\eta''})\sqrt{DCGT}.$$

   Since $\eta_* \leq \sqrt{(4 + 1/T)D^2M^{-2}}$, by letting $\eta'' \geq \sqrt{(4 + 1/T)D^2M^{-2}}$, we can make this case invalid.

3. *If $\eta_* \leq \eta'$*

$$R_T \leq \log(\lceil \log_2 \eta''/\eta' \rceil + 1) + (1 + \frac{\eta'}{\eta_*})\sqrt{DCGT}.$$

Since $\eta_* \geq \sqrt{2.5D^2/(TG)}$*, setting* $\eta' \leq \sqrt{2.5D^2/T}$ *gives*

$$R_T \leq \log(\lceil \log_2 \eta''/\eta' \rceil + 1) + (1+\sqrt{G})\sqrt{DCGT}.$$

Note that, we may not be able to make $\eta' \leq \sqrt{2.5D^2/T}$ *to make this case invalid, since we may not be able to bound* G*. However, we may be able to bound* G *with high probability, which will in turn create the regret bound in 1 with high probability.*

Hence, by running the Alg. 2 with an appropriate parameter vector, we achieve $O(\sqrt{CT})$ regret with $O(\log T)$ computational complexity, since the separation between η' and η'' is mainly dependent on C, which is bounded as $2.5D \leq C \leq (2T+0.5)D$.

Remark 4. *If* $\eta_* = \sqrt{(DC)/(GT)}$ *is known completely beforehand, then running Alg. 2 with the parameter vector* $\boldsymbol{\eta} = \{\eta_*\}$*, i.e.,* $N = 1$*, produces the regret bound*

$$R_T \leq 2\sqrt{DCGT},$$

which is equivalent to achieving the optimal regret bound using Alg. 1 with a priori information about source.

Remark 5. *For unknown time horizon* T*, we may use the doubling trick to get* $O(\sqrt{CT})$ *regret.*

Remark 6. *Note that, we have only included probability density estimators of Alg. 1 as experts for Alg. 2. However, the result in* (3.27) *is general and is true for any expert used in the mixture. Therefore, incorporating various different density estimators (parametric or nonparametric) in Algorithm 2, we can achieve the optimum performance in the mixture.*

Remark 7. *Applications that use density estimation such as anomaly detection algorithms will consequently perform better since UDE have substantially better performance bounds than the state-of-the-art methods.*

3.3 Numerical Results

In this section, we demonstrate the performance of our algorithm both on real and synthesized data in comparison to the-state-of-art techniques Maximum Likelihood (ML) [65], Online Convex Programming with static (OCP.static) [34] and dynamic learning rate (OCP.dynamic) [20], Gradient Descent (GD) [66], Momentum [67], Nesterov Accelerated Gradient (NAG) [68], Adagrad [69], Adadelta [70] and Adam [71].

All of the algorithms are implemented just as instructed in their respective papers.

OCP.static uses the doubling trick [24] to run in an online manner. We have implemented OCP.static with the fixed learning rate of $1/\sqrt{T}$ for an epoch of length T.

OCP.dynamic, on the other hand, uses a dynamic learning rate of $1/\sqrt{t}$ at time t.

For an online behavior, the ML algorithm uses a sliding window to determine its estimations.

In our implementation, ML uses the doubling trick and is run in epochs of durations $\{1, 2, 4, 8, \ldots\}$. Before the start of each epoch at time t, we run ML for the past $t-1$ observations with window lengths of $\{1, 2, 4, 8, \ldots, t-1\}$. Then, in its current epoch, we use the window length that provides the minimum log-loss. Hence, ML has a time complexity of $O(\log T)$ per round.

The other algorithms are also run with the doubling trick and used a similar approach to search their parameter spaces (like the window size for the ML algorithm) since implementing them with their default parameters provided poor performance.

We have run GD, Momentum, NAG, Adagrad, Adam in the past observations

for the step sizes $\eta = \{\frac{1}{T}, \frac{2}{T}, \frac{4}{T} \ldots, \frac{1}{4}, \frac{1}{2}, 1, 2, 4, \ldots, \frac{T}{4}, \frac{T}{2}, T\}$, and selected the step size that provided the minimum log-loss (i.e., best performance) and use it in their next epoch.

We have optimized only the step size and left the momentum term in Momentum, NAG to be its default value, i.e., $\gamma = 0.9$, since the step size is the main parameter that effects the performance and additional optimization of the γ parameter would have increased the time complexity to $O(\log^2 T)$, which would have been unfair.

We have set the smoothing term of Adagrad to its default value $\epsilon = 10^{-8}$.

In Adam, we have set the parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$ as instructed in its paper.

For Adadelta, the parameter that most influences the performance is the exponential update parameter γ. Therefore, we optimize γ among the set $\{0, \frac{1}{T}, \frac{2}{T}, \frac{4}{T} \ldots, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, \frac{7}{8}, \ldots, 1 - \frac{4}{T}, 1 - \frac{2}{T}, 1 - \frac{1}{T}, 1\}$ and set the smoothing parameter to its default value $\epsilon = 10^{-8}$.

We have run our algorithm, UDE, with learning rates in the range $1/T \leq \eta \leq T$ for a T length epoch of the algorithm.

We have also created a variant UDE.all that combines not only the subroutines of UDE but also all of the competing algorithms to demonstrate the option of using various density estimators in combination.

The experiments[1] consists of two parts, which are performance comparison in synthetic and real datasets.

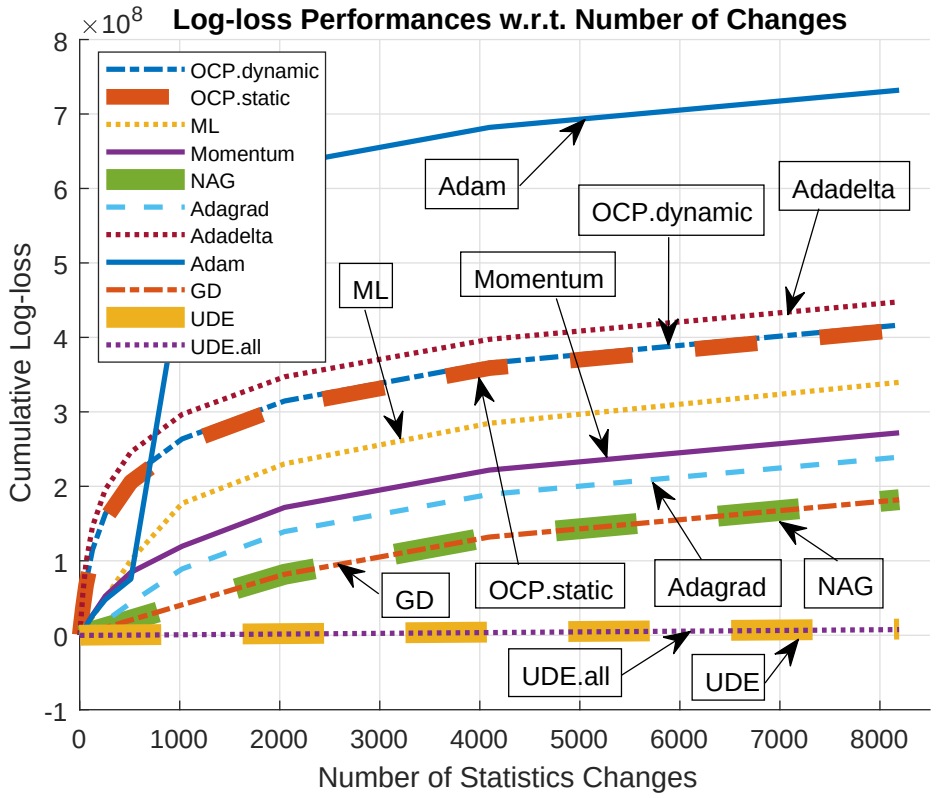

Figure 3.2: Cumulative log-loss regret performances of the density estimation algorithms with respect to the number of changes in the statistics of a nonstationary Gaussian process.

3.3.1 Synthetic Dataset

In this section, we compare the cumulative log-loss regrets of the algorithms with respect to the number of changes in the statistics of the source. To this end, we compare the algorithms' performances when the source has $C \in \{1, 2, 4, 8, \ldots\}$ changes in its statistics. For each value of C, we synthesize a dataset of size 10000 from a univariate Gaussian process with a unit standard deviation, i.e., $\sigma = 1$ and mean value alternating between 100 and -100 in every T/C samples (equal length time segments) such that in the first segment $\mu = 100$, $\mu = -100$ in the next and alternates as such. No information about the data is given to the algorithms except the variance. They all start from an initial mean 0.

[1]The codes used in the experiments are made publicly available at "http://www.ee.bilkent.edu.tr/~gokcesu/density_codes.zip"

In Fig. 3.2, we have illustrated the regret performance of the algorithms. We observe in Fig. 3.2 that the Adam algorithm performs substantially worse in high number of changes in the statistics. Since Adam gets rid of the step size for a completely adaptive behavior, its convergence rate is simply not good enough in this setting. OCP.static, OCP.dynamic and Adadelta have similar performances but still perform worse than ML. Even though, Momentum and Adagrad have better performance than ML, they are still outperformed by GD and NAG. Nonetheless, our algorithm, UDE, outperforms all of the other algorithms by huge margins, because it does not try to optimize its parameters but rather combines them to ensure the optimal one will survive. UDE.all performs basically the same as UDE since UDE has substantially greater performance.

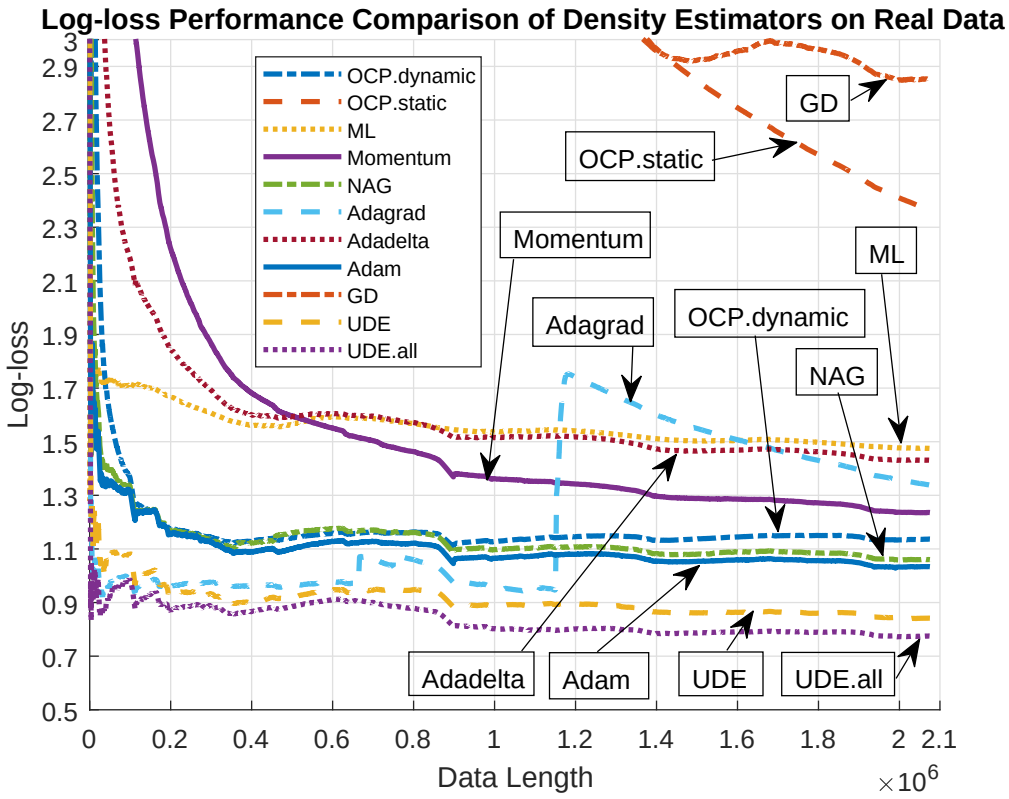

Figure 3.3: Average log-loss performance of the density estimators over the Individual Household Electric Power Consumption Dataset [1].

3.3.2 Real Dataset

We use "Individual Household Electric Power Consumption Dataset[2]" for real big data benchmark purposes, which is readily accessible online [1]. This dataset includes measurements of electric power consumption in one household with a one-minute sampling rate over a period of almost 4 years [1]. We have assumed a possibly nonstationary multivariate Gaussian process for this dataset and run the algorithms to estimate its distribution. Since the true distribution is not known, we have compared the performances of the algorithms directly with their log-losses instead of their regrets. All of the algorithms are initialized to zero-mean unit variance.

In Fig. 3.3, we have illustrated the log-loss performances of all of the algorithms. Interestingly, GD performed the worst with OCP.static a close second, even though it was one of the best performing competing algorithms in the previous experiments. ML provided an average performance similar to the first set of experiments. However, this time performed worse than the OCP.dynamic and the Adadelta algorithms. NAG performed similarly by being one of the best performing competing algorithms. Even though Adagrad was able to outperform all the other competing algorithms up to the middle of the dataset, it encountered a sudden increase in its log-loss and was unable to recuperate fast enough. The most interesting result was for the Adam algorithm, which performed the best among the competing algorithms, even though it performed the worst in the first set of experiments. From the distinct performances of the competing algorithms in real and synthetic datasets, we can infer that our algorithm UDE is able to outperform them in various different environments. In Fig. 3.3, we again observe a substantial performance gain in comparison to the other algorithms. Additionally, we are also able to observe a small performance increase in UDE.all on top of UDE, which supports the notion that combining different estimators would lead to better performance because of the low regret redundancy of the mixture.

[2]This dataset is the most popular ($\approx$ 125000 hits) large dataset (greater than 10^5 samples) in UCI Machine Learning Repository, which is publicly available at "https://archive.ics.uci.edu/ml/datasets/Individual+household+electric+power+consumption"

3.4 Chapter Summary

We have introduced a truly sequential and online algorithm, which estimates the density of a nonstationary memoryless exponential-family source with Hannan consistency. Our algorithms are truly sequential such that neither the time horizon T nor the total drift of the natural parameter are required to optimize its parameters. Here, the regret of our algorithm is increasing with only the square-root of time horizon T and the total drift of the natural parameter C_α. The results we provide are uniformly guaranteed to hold in a strong deterministic sense in an individual sequence manner for all possible observation sequences since we refrain from making any assumptions on the observations. We achieve this performance with a computational complexity only log-linear in the data length by carefully designing different probability density estimators and combining them in a mixture-of-experts setting. Due to such efficient performance and storage need, our algorithm can be effectively used in big data applications.

Our aim is to design a minimax optimal density estimator. However, for nonstationary sources, logarithmic regret bound is infeasible under low computational complexity [20]. Therefore, in Chapter 4, we propose a linear complexity algorithm that can achieve the minimax optimal regret bound $O(C \log T)$.

Chapter 4

Minimax Optimal Density Estimation

In this chapter, we propose a density estimation algorithm that can achieve the minimax optimal regret bound $O(C \log T)$. In Chapter 4.1, we introduce a minimax optimal density estimator that achieves logarithmic regret bound for stationary memoryless sources. We show that by selecting the learning rate of the gradient descent method in the order $O(1/t)$ (decreasing with time), we can achieve logarithmic regret bounds against the best stationary distribution. However, in the selection of the learning rate, we need a parameter that is dependent on the statistics of the data, which is an information we do not have a priori. Hence, in Chapter 4.2, we combine different versions of the estimators in Chapter 4.1 (each run with a different parameter) to achieve the performance of the optimal selection. Finally, in Chapter 4.3, we introduce a minimax optimal density estimator for nonstationary sources. To do this, at each time instance, we start another one of the estimator given in Chapter 3.2 and universally combine them to achieve the performance of the best piecewise stationary model. This way, we achieve the minimax optimal regret bound $O(C \log T)$.

4.1 Stationary Density Estimator

In this section, we first construct a density estimator that achieves minimax regret bound for stationary sources (unchanging statistics, hence, natural parameter). Here, at each time t, we observe $\boldsymbol{x}_t \in \mathbb{R}^m$ distributed according to a memoryless exponential-family distribution

$$f(\boldsymbol{x}_t) = \exp\left(-\langle \boldsymbol{\alpha}, \boldsymbol{z}_t \rangle - A(\boldsymbol{\alpha})\right), \tag{4.1}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^d$ is the unknown natural parameter of the exponential-family distribution and belongs to a convex feasible set S, $A(\cdot)$ is a known function of the parameter $\boldsymbol{\alpha}$ (normalization factor or log-partition function), $\langle \cdot, \cdot \rangle$ is the inner product and $\boldsymbol{z}_t$ is the d-dimensional known sufficient statistic of $\boldsymbol{x}_t$ [31], i.e.,

$$\boldsymbol{z}_t = \mathcal{T}(\boldsymbol{x}_t). \tag{4.2}$$

Remark 8. *In general, a distribution belonging to an exponential family has the form* $f(x) = \exp(-\langle \boldsymbol{\alpha}, \mathcal{T}(x) \rangle - A(\boldsymbol{\alpha}) - B(x))$*, where* $B(x)$ *is only a function of the observation* x*. However, this function can simply be included inside of* $\mathcal{T}(x)$ *whose corresponding parameter in the inner product will simply be* 1 *in the true probability density.*

Instead of directly estimating the density function $f(\cdot)$, we cast the problem as a convex optimization problem and estimate the natural parameter $\boldsymbol{\alpha}$ according to our past observations $\{\boldsymbol{x}_r\}_{r=1}^{t-1}$ ($\boldsymbol{\alpha}$ completely describes $f(\cdot)$ and estimating $\boldsymbol{\alpha}$ instead of $f(\cdot)$ still provides logarithmic performance bounds). Hence, our density estimation is

$$\hat{f}_t(\boldsymbol{x}_t) = \exp(-\langle \hat{\boldsymbol{\alpha}}_t, \boldsymbol{z}_t \rangle - A(\hat{\boldsymbol{\alpha}}_t)). \tag{4.3}$$

We use Online Gradient Descent (OGD) [35] to sequentially produce our estimate $\hat{\boldsymbol{\alpha}}_t$, where we first start from an initial estimate $\hat{\boldsymbol{\alpha}}_1$, and update our recent estimation $\hat{\boldsymbol{\alpha}}_t$ based on our new observation $\boldsymbol{x}_t$. To update $\hat{\boldsymbol{\alpha}}_t$, we first observe a sample $\boldsymbol{x}_t$ and incur the loss $l(\hat{\boldsymbol{\alpha}}_t, \boldsymbol{x}_t)$ according to our estimation $\hat{\boldsymbol{\alpha}}_t$, which is

Algorithm 3 Stationary Density Estimator

Set H
Initialize learning rates $\eta_t = (Ht)^{-1}$ for $t \in \{1, 2, \ldots\}$
Select initial parameter $\hat{\boldsymbol{\alpha}}_1$
Calculate the mean $\mu_{\hat{\boldsymbol{\alpha}}_1}$
for $t = 1, 2, \ldots$ **do**
    Declare estimation $\hat{\boldsymbol{\alpha}}_t$
    Observe $\boldsymbol{x}_t$
    Calculate $\boldsymbol{z}_t = \mathcal{T}(\boldsymbol{x}_t)$
    Update parameter: $\tilde{\boldsymbol{\alpha}}_{t+1} = \hat{\boldsymbol{\alpha}}_t - \eta_t(\boldsymbol{z}_t - \mu_{\hat{\boldsymbol{\alpha}}_t})$
    Project onto convex set: $\hat{\boldsymbol{\alpha}}_{t+1} = P_S(\tilde{\boldsymbol{\alpha}}_{t+1})$
    Calculate the mean $\mu_{\hat{\boldsymbol{\alpha}}_{t+1}}$
end for

the log-loss, i.e., $-\log(\hat{f}_t(\boldsymbol{x}_t))$. From (4.3), we receive the loss

$$l(\hat{\boldsymbol{\alpha}}_t, \boldsymbol{x}_t) = \langle \hat{\boldsymbol{\alpha}}_t, \boldsymbol{z}_t \rangle + A(\hat{\boldsymbol{\alpha}}_t). \tag{4.4}$$

Then, we calculate the gradient of the loss with respect to $\hat{\boldsymbol{\alpha}}_t$,

$$\begin{aligned} \nabla_{\hat{\boldsymbol{\alpha}}_t} l(\hat{\boldsymbol{\alpha}}_t, \boldsymbol{x}_t) =& \boldsymbol{z}_t + \nabla_{\hat{\boldsymbol{\alpha}}_t} A(\hat{\boldsymbol{\alpha}}_t), \\ =& \boldsymbol{z}_t - \mu_{\hat{\boldsymbol{\alpha}}_t}, \end{aligned} \tag{4.5}$$

where $\mu_{\hat{\boldsymbol{\alpha}}_t}$ is the mean of $\boldsymbol{z}_t$ if $\boldsymbol{x}_t$ were distributed according to $\hat{f}_t(\cdot)$. We update the parameter $\hat{\boldsymbol{\alpha}}_t$ such that

$$\hat{\boldsymbol{\alpha}}_{t+1} = P_S(\hat{\boldsymbol{\alpha}}_t - \eta_t(\boldsymbol{z}_t - \mu_{\hat{\boldsymbol{\alpha}}_t})), \tag{4.6}$$

where η_t is the learning rate and $P_S(\cdot)$ is the projection onto the convex feasible set S and is defined as

$$P_S(x) = \arg\min_{y \in S} \|x - y\|. \tag{4.7}$$

The complete algorithm is provided in Alg. 3 where the learning rates are given by $\eta_t = (Ht)^{-1}$ where $H > 0$ is an input to the algorithm. The optimal value for H and the instructions on its selection are given in Theorem 3. Next, we provide performance bounds of Alg. 3. Theorem 3 shows that using Alg. 3 with a suitable parameter H, we can achieve minimax regret $O(\log T)$.

Theorem 3. *When Alg. 3 is used with parameter H to estimate the distribution $f_t(\boldsymbol{x}_t)$, its regret is upper bounded by*

$$R_T \leq \frac{D}{2H}(\log T + 1), \tag{4.8}$$

if H is such that $\Sigma_{\hat{\boldsymbol{\alpha}}_t} \succeq H I_{d\times d}$ for all t, where $\Sigma_{\hat{\boldsymbol{\alpha}}_t}$ is the covariance of $\boldsymbol{z}_t$ when $\boldsymbol{x}_t$ is distributed with natural parameter $\hat{\boldsymbol{\alpha}}_t$, $I_{d\times d}$ is the d-by-d identity matrix and D is defined as

$$D \triangleq \frac{\sum_{t=1}^{T} t^{-1}\|\boldsymbol{z}_t - \mu_{\hat{\boldsymbol{\alpha}}_t}\|^2}{\sum_{t=1}^{T} t^{-1}}.$$

The result of Theorem 3 shows that the regret is reciprocally dependent on our input parameter H. Therefore, choosing H lower than necessary may result in a high regret. In Chapter 4.2, we mitigate this problem by adopting a mixture of experts approach [72, 73].

Proof of Theorem 3. The regret at time t is defined as

$$r_t(\hat{\boldsymbol{\alpha}}_t) = l(\hat{\boldsymbol{\alpha}}_t, \boldsymbol{x}_t) - l(\boldsymbol{\alpha}, \boldsymbol{x}_t), \tag{4.9}$$

where $l(\boldsymbol{\alpha}, x)$ is as in (4.4).

By H-strong convexity, the regret becomes

$$r_t \leq \langle \nabla_{\boldsymbol{\alpha}} l(\hat{\boldsymbol{\alpha}}_t, \boldsymbol{x}_t), (\hat{\boldsymbol{\alpha}}_t - \boldsymbol{\alpha})\rangle - \frac{H}{2}\|\hat{\boldsymbol{\alpha}}_t - \boldsymbol{\alpha}\|^2. \tag{4.10}$$

We bound the first term in the right hand side of (4.10) using the update rule (4.6). By definition of projection in (4.7), we have

$$\|P_S(\hat{\boldsymbol{\alpha}}_t - \eta_t \nabla_{\boldsymbol{\alpha}} l(\hat{\boldsymbol{\alpha}}_t, \boldsymbol{x}_t)) - \boldsymbol{\alpha}\| \leq \|\hat{\boldsymbol{\alpha}}_t - \eta_t \nabla_{\boldsymbol{\alpha}} l(\hat{\boldsymbol{\alpha}}_t, \boldsymbol{x}_t) - \boldsymbol{\alpha}\|.$$

Substituting (4.6) in the left hand side provides

$$\|\hat{\boldsymbol{\alpha}}_{t+1} - \boldsymbol{\alpha}\| \leq \|\hat{\boldsymbol{\alpha}}_t - \eta_t \nabla_{\boldsymbol{\alpha}} l(\hat{\boldsymbol{\alpha}}_t, \boldsymbol{x}_t) - \boldsymbol{\alpha}\|. \tag{4.11}$$

Hence, we get

$$\begin{aligned}\|\hat{\boldsymbol{\alpha}}_{t+1} - \boldsymbol{\alpha}\|^2 &\leq \|\hat{\boldsymbol{\alpha}}_t - \eta_t \nabla_{\boldsymbol{\alpha}} l(\hat{\boldsymbol{\alpha}}_t, \boldsymbol{x}_t) - \boldsymbol{\alpha}\|^2, \\ &\leq \|\hat{\boldsymbol{\alpha}}_t - \boldsymbol{\alpha}\|^2 - 2\eta_t \langle \nabla_{\boldsymbol{\alpha}} l(\hat{\boldsymbol{\alpha}}_t, \boldsymbol{x}_t), (\hat{\boldsymbol{\alpha}}_t - \boldsymbol{\alpha})\rangle \\ &\quad + \eta_t^2 \|\nabla_{\boldsymbol{\alpha}} l(\hat{\boldsymbol{\alpha}}_t, \boldsymbol{x}_t)\|^2. \end{aligned} \tag{4.12}$$

Since $\eta_t > 0$ for all t, rearranging (4.12) results in

$$\begin{aligned}
&\langle \nabla_{\boldsymbol{\alpha}} l(\hat{\boldsymbol{\alpha}}_t, \boldsymbol{x}_t), (\hat{\boldsymbol{\alpha}}_t - \boldsymbol{\alpha}) \rangle \\
\leq &\frac{1}{2\eta_t} \left(\|\hat{\boldsymbol{\alpha}}_t - \boldsymbol{\alpha}\|^2 - \|\hat{\boldsymbol{\alpha}}_{t+1} - \boldsymbol{\alpha}\|^2 \right) + \frac{\eta_t}{2} \|\nabla_{\boldsymbol{\alpha}} l(\hat{\boldsymbol{\alpha}}_t, \boldsymbol{x}_t)\|^2, \\
\leq &\frac{1}{2\eta_t} \left(\|\hat{\boldsymbol{\alpha}}_t - \boldsymbol{\alpha}\|^2 - \|\hat{\boldsymbol{\alpha}}_{t+1} - \boldsymbol{\alpha}\|^2 \right) + \frac{\eta_t}{2} \|\boldsymbol{z}_t - \mu_{\hat{\boldsymbol{\alpha}}_t}\|^2, \qquad (4.13)
\end{aligned}$$

where we used (4.5) in the last step. Putting (4.13) in the right hand side of (4.10) yields

$$\begin{aligned}
r_t \leq &\frac{1}{2\eta_t} \left(\|\hat{\boldsymbol{\alpha}}_t - \boldsymbol{\alpha}\|^2 - \|\hat{\boldsymbol{\alpha}}_{t+1} - \boldsymbol{\alpha}\|^2 \right) \\
&+ \frac{\eta_t}{2} \|\boldsymbol{z}_t - \mu_{\hat{\boldsymbol{\alpha}}_t}\|^2 - \frac{H}{2} \|\hat{\boldsymbol{\alpha}}_t - \boldsymbol{\alpha}\|^2. \qquad (4.14)
\end{aligned}$$

Thus, summing (4.14) from $t = 1$ to T, we have the cumulative regret up to time T, which is given by

$$\begin{aligned}
R_T \leq &\frac{1}{2} \sum_{t=2}^{T} \left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} - H \right) \|\hat{\boldsymbol{\alpha}}_t - \boldsymbol{\alpha}\|^2 + \frac{1}{2} \left(\frac{1}{\eta_1} - H \right) \|\hat{\boldsymbol{\alpha}}_1 - \boldsymbol{\alpha}\|^2 \\
&- \frac{1}{2\eta_T} \|\hat{\boldsymbol{\alpha}}_{T+1} - \boldsymbol{\alpha}\|^2 + \frac{1}{2} \sum_{t=1}^{T} \eta_t \|\boldsymbol{z}_t - \mu_{\hat{\boldsymbol{\alpha}}_t}\|^2, \\
\leq &\sum_{t=1}^{T} \frac{\|\boldsymbol{z}_t - \mu_{\hat{\boldsymbol{\alpha}}_t}\|^2}{2Ht}, \\
\leq &\frac{D}{2H} (\log T + 1),
\end{aligned}$$

where we used $\eta_t = (Ht)^{-1}$, and

$$D = \frac{\sum_{t=1}^{T} t^{-1} \|\boldsymbol{z}_t - \mu_{\hat{\boldsymbol{\alpha}}_t}\|^2}{\sum_{t=1}^{T} t^{-1}}, \qquad (4.15)$$

which concludes the proof of the theorem. □

Remark 9. *Instead of OGD, any other gradient based approach such as Momentum [67], Nesterov's Accelerated Gradient [68], Adam [71], Adagrad [69], Adadelta [70] can also be used. Similar derivations will also hold for them as well.*

Algorithm 4 Universal Density Estimator

INPUTS:
Set $H_{\min}$ and $H_{\max}$
Set $N = \left\lfloor \log \frac{H_{max}}{H_{min}} \right\rfloor + 1$
Start N density estimators $\hat{f}_t^r(\cdot)$ each running Alg. 1 with parameter $H_r = H_{\min} \cdot 2^{r-1}$ for $r = \{1, 2, \ldots, N\}$
Initialize weights $w_1^r = 1/N$, $\forall r$
ALGORITHM:
for $t = 1$ **to** T **do**
  OUTPUT:
  Declare estimation $\hat{f}_t^u(x) = \sum_{r=1}^{N} w_t^r \hat{f}_t^r(x)$
  UPDATE:
  Observe $\boldsymbol{x}_t$
  Calculate $\boldsymbol{z}_t = \mathcal{T}(\boldsymbol{x}_t)$
  for $r = 1$ **to** N **do**
    Update density estimations $\hat{f}_t^r()\cdot$ according to "Stationary Density Estimator"
    $w_{t+1}^r = w_t^r \hat{f}_t^r(x_t) / \hat{f}_t^u(x_t)$
  end for
end for

4.2 Universal Density Estimator

The Algorithm in Chapter 4.1 needs an input H which needs a priori knowledge we do not have on the underlying process to be set appropriately. Therefore, in this section, we propose an algorithm to estimate the density of a stationary source when we do not know H a priori.

Suppose, we run N separate Alg. 3 and label each of the resulting estimators as $\hat{f}_t^r(x), r \in \{1, 2, \ldots, N\}$. We mix all of the estimators in a weighted manner such that our density estimation is given by

$$\hat{f}_t^u(x) = \sum_{r=1}^{N} w_t^r \hat{f}_t^r(x), \tag{4.16}$$

where w_t^r is the mixture weight of the algorithm labeled r at time t. These weights are normalized versions of the algorithms' performance weights $\hat{w}_t^r$ such that

$$w_t^r = \frac{\hat{w}_t^r}{\sum_{r'=1}^{N} \hat{w}_t^{r'}}. \tag{4.17}$$

The performance weights of the algorithms are given by

$$\hat{w}_t^r = \prod_{\tau=1}^{t-1} \hat{f}_\tau^r(\boldsymbol{x}_\tau), \tag{4.18}$$

for $t > 1$ and $\hat{w}_1^r = 1$ at the start for $r \in \{1, 2, \ldots, N\}$. We point out that the mixture weights w_t^r can be recursively updated as

$$w_{t+1}^r = w_t^r \hat{f}_t^r / \hat{f}_t^u, \tag{4.19}$$

from (4.16), (4.17), (4.18), where $w_1^r = 1/N$. We next provide the performance bound of this mixture approach.

Theorem 4. *When we combine a total of N estimators with the weighting scheme in* (4.16) *and* (4.19)*, the regret incurred by our universal density estimator ($\hat{f}_t^u(\cdot)$), $R_{T,U}$, is upper bounded by*

$$R_{T,U} \leq \log N + \min_{r \in \{1,\ldots,N\}} R_{T,r},$$

where $R_{T,r}$ is the regret incurred by the estimator labeled r.

Theorem 4 implies that even by mixing exponential number of estimators we can still achieve sublinear regret since the additional regret of mixing is only logarithmically dependent on the number of density estimators mixed together.

Proof of Theorem 4. We receive the accumulated log-loss $L_{T,U}$ such that

$$L_{T,U} = \sum_{t=1}^{T} -\log \hat{f}_t^u(\boldsymbol{x}_t) = \sum_{t=1}^{T} -\log \left(\sum_{r=1}^{N} w_t^r \hat{f}_t^r(\boldsymbol{x}_t) \right),$$

where we used (4.16). Using (4.17) and (4.18), we get

$$\begin{aligned} L_{T,U} &= -\log \frac{\sum_{r=1}^{N} \hat{f}_1^r(\boldsymbol{x}_1)}{N} - \sum_{t=2}^{T} \log \left(\frac{\sum_{r=1}^{N} \prod_{\tau=1}^{t} \hat{f}_\tau^r(\boldsymbol{x}_\tau)}{\sum_{r=1}^{N} \prod_{\tau=1}^{t-1} \hat{f}_\tau^r(\boldsymbol{x}_\tau)} \right), \\ &= -\log \frac{1}{N} \sum_{r=1}^{N} \prod_{\tau=1}^{T} \hat{f}_\tau^r(\boldsymbol{x}_\tau) \\ &\leq -\log \frac{1}{N} \prod_{\tau=1}^{T} \hat{f}_\tau^r(\boldsymbol{x}_\tau), \end{aligned}$$

for all $r = \{1, 2, \ldots, N\}$. Hence,

$$L_{T,U} \leq \log N + \min_{r \in \{1,2,\ldots,N\}} \sum_{t=1}^{T} - \log \hat{f}_t^r(\boldsymbol{x}_t),$$
$$= \log N + \min_r L_{T,r},$$

where $L_{T,r}$ is the cumulative log-loss of the estimator labeled r.

Subtracting the loss incurred by the true density function from both sides we get,

$$R_{T,U} \leq \log N + \min_r R_{T,r},$$

which concludes the proof. □

Corollary 3. *Suppose we do not know H exactly but know H_{min} and H_{max} such that $H_{min} \leq H \leq H_{max}$. For each of the estimators mixed, i.e. $\hat{f}_t^r(\cdot)$, we set the input parameter as $H_r = H_{min} \cdot 2^{r-1}$ for $r = \{1, 2, \ldots, \left\lfloor \log_2 \frac{H_{max}}{H_{min}} \right\rfloor + 1\}$. We incur $R_{T,U}$ such that,*

$$R_{T,U} \leq \log \left(\left\lfloor \log_2 \frac{H_{max}}{H_{min}} \right\rfloor + 1 \right) + \frac{D}{H}(\log T + 1).$$

Proof of Corollary 3. In Theorem 3, we showed that by knowing H a priori and using it as the input in Alg. 3, we can achieve the regret $R_T \leq \frac{D}{2H}(\log T + 1)$. Running Alg. 3 with some $H_r \leq H$ incurs regret $\frac{D}{2H_r}(\log T + 1)$ as all the derivations of Theorem 3 would still hold. Since, in the mixture, there is one $H_{r'}$ such that $H/2 < H_{r'} \leq H$, the regret incurred by the corresponding $f^{r'}$ is

$$R_{T,\hat{f}^{r'}} \leq \frac{D}{2H_{r'}}(\log T + 1) \leq \frac{D}{H}(\log T + 1).$$

Since we mix $N = \left(\left\lfloor \log_2 \frac{H_{max}}{H_{min}} \right\rfloor + 1 \right)$ number of algorithms, our total regret is given by

$$R_{T,U} \leq \log \left(\left\lfloor \log_2 \frac{H_{max}}{H_{min}} \right\rfloor + 1 \right) + \frac{D}{H}(\log T + 1),$$

from Theorem 4. □

The complete algorithm is given in Alg. 4.

4.3 Nonstationary Density Estimator

In Chapter 4.2, we have introduced an algorithm that achieves the minimax optimal regret when estimating a stationary exponential-family source without any a priori knowledge. In this section, we extend that result to nonstationary sources, i.e., dynamic natural parameter $\boldsymbol{\alpha}_t$.

To achieve the minimax regret in nonstationary sources, we can run Alg. 4 on our observations and reset it at each instance the source statistics change. While this approach achieves the minimax optimal regret, it requires the knowledge of exact time instances the source statistics change.

Since we do not know these time instances, we create a new Alg. 4 at each time t and mix them instead of resetting the algorithm.

We point out that these newly created Alg. 4's train their estimators with only the observations after the time they were created. Hence, at each time τ, we create a new Alg. 4 and label the estimation of this algorithm at time t ($\tau \leq t$) as $\hat{g}_t^\tau(\cdot)$.

We point out that $\hat{g}_t^\tau(\cdot)$ only uses the observations $\{\boldsymbol{x}_\tau, \boldsymbol{x}_{\tau+1}, \boldsymbol{x}_{\tau+2}, \ldots, \boldsymbol{x}_{t-1}\}$ to construct its density estimation at time t. Since $\hat{g}_t^\tau(\cdot)$ is the output of Alg. 4, it has already gone through a mixture.

We adopt a double mixture approach and combine these $\hat{g}_t^\tau(\cdot)$ again with carefully selected weights as

$$\hat{g}_t^u(x) = \sum_{\tau=1}^{t} v_t^\tau \hat{g}_t^\tau(x), \tag{4.20}$$

where v_t^τ is the combination weight of the estimator that has started at time instance τ at time t.

Algorithm 5 Nonstationary Density Estimator

Initialize $\hat{v}_1^1 = 1$
Set H_{min}, H_{max}
Initialize estimator $\hat{g}_1^1(\cdot)$ using Alg. 4
with inputs $\{H_{\min}, H_{\max}\}$.
for $t = 1$ **to** T **do**
  Declare estimation $\hat{g}_t^u(x) = \sum_{\tau=1}^{t} v_t^\tau \hat{g}_t^\tau(x)$
  Observe $\boldsymbol{x}_t$
  for $\tau = 1$ **to** t **do**
    Update $\hat{v}_{t+1}^\tau = \dfrac{t+1-\tau}{t-\tau+2} \cdot \hat{v}_t^\tau \cdot \hat{g}_t^\tau(\boldsymbol{x}_t)$
    Update $\hat{g}_{t+1}^\tau(\cdot)$ according to Alg. 4
  end for
  Create $\hat{v}_{t+1}^{t+1} = \displaystyle\sum_{\tau=1}^{t} \frac{1}{t-\tau+2} \cdot \hat{v}_t^\tau \cdot \hat{g}_t^\tau(\boldsymbol{x}_t)$
  Start estimator $\hat{g}_{t+1}^{t+1}(\cdot)$ according to Alg. 4
  for $\tau = 1$ **to** $t+1$ **do**
    $v_t^\tau = \hat{v}_t^\tau / \left(\sum_{\tau=1}^{t} \hat{v}_t^\tau\right)$
  end for
end for

These combination weights are normalized versions of the algorithms' performance weights $\hat{v}_t^\tau$ such that

$$v_t^\tau = \frac{\hat{v}_t^\tau}{\sum_{\tau=1}^{t} \hat{v}_t^\tau}, \tag{4.21}$$

for $1 \leq \tau \leq t$.

These performance weights are recursively calculated as

$$\hat{v}_t^\tau = \begin{cases} \dfrac{t-\tau}{t-\tau+1} \;\; \hat{v}_{t-1}^\tau \;\; \hat{g}_{t-1}^\tau(\boldsymbol{x}_{t-1}), & \tau < t \\ \\ \displaystyle\sum_{\tau=1}^{t-1} \frac{1}{t-\tau+1} \;\; \hat{v}_{t-1}^\tau \;\; \hat{g}_{t-1}^\tau(\boldsymbol{x}_{t-1}), & \tau = t \end{cases}, \tag{4.22}$$

where $\hat{v}_1^1 = 1$ at the start.

The complete description of the algorithm is given in Alg. 5. Next, we provide the performance bound of Alg. 5.

Theorem 5. *Algorithm 5 has the following regret bound, $R_{T,NS}$,*

$$R_{T,NS} \leq \sum_{i=1}^{C} \log t_i + \sum_{i=1}^{C-1} \log(t_i + 1) + \sum_{i=1}^{C} R_{i,U}, \tag{4.23}$$

where C is the total number of change in the source statistics as in (2.1) and the lengths of the piecewise stationary segments are t_i for $i \in \{1, 2, \ldots, C\}$ ($\sum_{i=1}^{C} t_i = T$). Initializing $t_0 = 0$, $R_{i,U}$ is the regret incurred by the estimator $\hat{g}_t^{\sum_{j=1}^{i-1} t_j + 1}$ in time interval $t \in [\sum_{j=1}^{i-1} t_j + 1, \sum_{j=1}^{i} t_j]$.

The result of Theorem 5 shows that the additional regret we incur from not knowing the time instances when the source statistics change is linearly dependent on the number of such changes.

Proof of Theorem 5. Combining (4.20) and (4.21), we have,

$$\hat{g}_t^u(x) = \frac{\sum_{\tau=1}^{t} \hat{v}_t^\tau \hat{g}_t^\tau(x)}{\hat{v}_t^t + \sum_{\tau=1}^{t-1} \hat{v}_t^\tau} = \frac{\sum_{\tau=1}^{t} \hat{v}_t^\tau \hat{g}_t^\tau(x)}{\sum_{\tau=1}^{t-1} \hat{v}_{t-1}^\tau \hat{g}_{t-1}^\tau(\boldsymbol{x}_{t-1})},$$

for $t > 1$ and $\hat{g}_{t=1}^u(x) = \hat{g}_1^1(x)$. Thus, we receive the following accumulated log-loss, L_{NS},

$$\begin{aligned} L_{NS} &= \sum_{t=1}^{T} -\log \hat{g}_t^u(\boldsymbol{x}_t), \\ &= -\log \hat{g}_1^1(\boldsymbol{x}_1) - \sum_{t=2}^{T} \log \left(\frac{\sum_{\tau=1}^{t} \hat{v}_t^\tau \hat{g}_t^\tau(\boldsymbol{x}_t)}{\sum_{\tau=1}^{t-1} \hat{v}_{t-1}^\tau \hat{g}_{t-1}^\tau(\boldsymbol{x}_{t-1})} \right), \\ &= -\log \sum_{\tau=1}^{T} \hat{v}_T^\tau \hat{g}_T^\tau(\boldsymbol{x}_T). \end{aligned}$$

We see that this expression can be written as the log-loss of a weighted sum of losses incurred from various strategies belonging to a set S_T after we recursively substitute $\hat{v}_T^\tau$ using (4.22) and consider the distributive property of multiplication over addition.

A strategy s in S_T, uses $\hat{g}_{t-1}^\tau(\cdot)$ at time $t-1$ and it is only allowed to continue along its last estimator to $\hat{g}_t^\tau(\cdot)$ or switch to $\hat{g}_t^t(\cdot)$ at time t.

Let s_t denote the superscript (τ) of whichever estimator is utilized by s at time t.

Consequently, our loss becomes,

$$\begin{aligned} L_{NS} &= -\log \sum_{s \in S_T} \Big(Q(s) \prod_{t=1}^{T} \hat{g}_t^{s_t}(\boldsymbol{x}_t) \Big), \\ &\leq -\log Q(s) \prod_{t=1}^{T} \hat{g}_t^{s_t}(\boldsymbol{x}_t), \end{aligned} \tag{4.24}$$

for all $s \in S_T$, where $Q(s)$ denotes the prior weight determined by how long each of the different estimators $\hat{g}_t^{\tau}(\cdot)$'s are used by s.

In (4.22), the prior components $\frac{t-\tau}{t-\tau+1}$ and $\frac{1}{t-\tau+1}$ in the calculation of $\hat{v}_t^{\tau}$ result in the following $Q(s)$ for an s which utilizes a total of C_s estimators.

$$Q(s) = \begin{cases} \dfrac{1}{T}, & C_s = 1 \\ \displaystyle\prod_{i=1}^{C_s} \frac{1}{t_i} \prod_{i=1}^{C_s - 1} \frac{1}{t_i + 1}, & C_s > 1 \end{cases},$$

where t_i's are the durations in which fixed $\hat{g}_t^{\tau}(\cdot)$'s are used, i.e., the same estimator that started at τ.

To upper-bound L_{NS}, we use $s^* \in S_T$ in (4.24), which changes its estimator at the exact time instances where the true density changes (for a total of $C-1$ times). Hence,

$$\begin{aligned} L_{NS} &\leq -\log Q(s^*) \prod_{t=1}^{T} \hat{g}_t^{s_t^*}(\boldsymbol{x}_t), \\ &\leq -\log \Bigg(\prod_{t=1}^{C} \frac{1}{t_i} \prod_{i=1}^{C-1} \frac{1}{t_i + 1} \prod_{i=1}^{T} \hat{g}_T^{s_i}(\boldsymbol{x}_t) \Bigg), \\ &\leq \sum_{i=1}^{C} \log t_i + \sum_{i=1}^{C-1} \log(t_i + 1) + \sum_{t=1}^{T} -\log \hat{g}_t^{s_t}(\boldsymbol{x}_t). \end{aligned}$$

After subtracting the log-loss incurred by the true density from both sides, we end up with the bound given in (4.23). □

Corollary 4. *Combining Theorem 4 and Theorem 5, we can achieve the following regret bound where* $\sum_{i=1}^{C} t_i = T$.

$$R_{T,NS} \leq \sum_{i=1}^{C} \log t_i + \sum_{i=1}^{C-1} \log(t_i + 1) + C \log N + \sum_{i=1}^{C} \frac{D_i}{H_i} (\log(t_i) + 1).$$

If there exists a $D \geq D_i$ and $H \leq H_i$ for all $i \in \{1, 2, \dots C\}$, the bound can be written in a more compact form as

$$R_{T,NS} \leq C \left(\left(2 + \frac{D}{H} \right) \log \left(\frac{T}{C} \right) + 1 + \frac{D}{H} + \log N \right). \tag{4.25}$$

Therefore, our density estimation algorithm achieves Merhav's lower bound $O(C \log T)$ [55] hence achieving the minimax optimal regret.

Remark 10. *In [55], the authors show the achievability of logarithmic regret bounds using the Minimum Description Length (MDL) principle. They show that the total number of bits needed for universal coding of a source with piecewise constant parameters is at least* $(1.5C - 1) \log T$. *Hence, our algorithm has a multiplicative redundancy factor of* D/H, *where* D *increases with the set of sufficient statistics and* H *increases with stronger convexity. Note that the lower bounds in [55] are proven for a finite size alphabet and their extension to the continuous distributions are not straightforward. However, similar logarithmic redundancy bounds should hold for different continuous distributions albeit with different constants. Our multiplicative redundancy* D/H *increases with the hardness of the problem, so it is reasonable to assume that a similar constant exists for the lower bound as well, which is different for different distributions and different estimation problems. Nonetheless, logarithmic performance bounds are always welcomed in both machine learning, signal processing and information theory literature since logarithmic bounds decay exponentially fast.*

4.4 Chapter Summary

We have introduced a minimax optimal density estimation algorithm for general exponential-family of sources. To do this, we first achieved logarithmic regret bounds under log-loss, i.e., $O(\log T)$, for stationary distributions by using online convex optimization with a decaying learning rate in Chapter 4.1.

However, the optimal setting of the learning rate requires some a priori information about the source statistics. Hence, to eliminate this requirement, we have adopted a universal mixture approach in Chapter 3.2 and achieved the performance of the optimal learning rate without any a priori information.

To achieve the minimax optimal regret bounds in nonstationary setting, we modeled the nonstationary source model (i.e., the changing natural parameter) as piecewise stationary models. By combining these stationary models in a switching experts setting, we achieved the minimax optimal performance bound $O(C \log T)$, when the statistics of the source, i.e., the natural parameter, changes $C-1$ number of times, i.e., the nonstationary model consists of C stationary models (C time segments with piecewise constant parameters).

In Chapter 5, we propose a dynamic threshold selection scheme with logarithmic performance bounds to achieve minimax optimality in both stages of our anomaly detector.

Chapter 5

Anomaly Detection with Logarithmic Performance Bounds

In this chapter, we propose a minimax optimal dynamic threshold selection scheme to be used with our minimax optimal density estimator. The selection of the threshold for anomaly detection is a notoriously difficult problem. Specifically, optimization of the threshold in a binary loss setting makes the updating of the threshold challenging.

To this end, in Chapter 5.1, we first substitute this binary loss with the logistic loss function, which is convex in its argument. By changing the loss definition (and subsequently the regret definition), we can optimize the threshold a lot more efficiently.

To optimize the threshold, we utilize the Online Newton Step [35] in Chapter 5.2. Since the logistic loss is exp-concave in the threshold parameter, we can achieve logarithmic performance bounds.

5.1 Using Logistic Loss for Convex Analysis

To decide whether the data $\boldsymbol{x}_t$ is anomalous or not, we threshold the output of the nonstationary density estimation $\hat{g}_t^u(\boldsymbol{x}_t)$ of Alg. 5 with τ_t such that

$$\hat{d}_t = \begin{cases} +1, & \hat{g}_t^u(\boldsymbol{x}_t) < \tau_t \text{ (anomaly)} \\ -1, & \hat{g}_t^u(\boldsymbol{x}_t) \geq \tau_t \text{ (normal)} \end{cases} . \tag{5.1}$$

This thresholding is equivalent to thresholding a general monotonically increasing function of $\hat{g}_t^u$. Hence, in a more general form, we have

$$\hat{d}_t = \begin{cases} +1, & \hat{p}_t(\boldsymbol{x}_t) < \tau_t \text{ (anomaly)} \\ -1, & \hat{p}_t(\boldsymbol{x}_t) \geq \tau_t \text{ (normal)} \end{cases} , \tag{5.2}$$

where $\hat{p}_t = \Phi(\hat{g}_t^u)$ and $\Phi(\cdot)$ is a monotonically increasing function. We define our error with $0-1$ loss, i.e., $\mathbb{1}_{\hat{d}_t \neq d_t}$. However, we emphasize that, in general, the cost of making a prediction error on normal and anomalous data may not be the same. To this end, our algorithm can assign different costs to these distinct errors. Let J_{d_t} be the cost of misclassification of the data with label d_t ($d_t \in \{-1, 1\}$). Then, we can rewrite the error as $J_{d_t}\mathbb{1}_{\hat{d}_t \neq d_t}$. However, the $0-1$ loss definition is not convex in τ_t, which makes optimization difficult. Therefore, we substitute this $0-1$ loss with the widely used logistic loss function $\log(1+\exp(-(\tau_t - \hat{p}_t)d_t))$, which completely upper bounds the $0-1$ loss $\mathbb{1}_{\hat{d}_t \neq d_t}$. Hence, the loss function is given by

$$l(\tau_t, \hat{p}_t, d_t) = J_{d_t} \log(1+\exp(-(\tau_t - \hat{p}_t)d_t)).$$

We compete against a best fixed threshold, thus, regret in a time horizon T is defined as

$$R_{A,T} = \sum_{t=1}^{T} l(\tau_t, \hat{p}_t, d_t) - \min_{\tau \in V} \sum_{t=1}^{T} l(\tau, \hat{p}_t, d_t). \tag{5.3}$$

Let the time indices where we make a prediction error be defined by the set T_e. Then, the actual regret that approximates the $0-1$ loss regret is given by

$$R_{A,T} = \sum_{t \in T_e} l(\tau_t, \hat{p}_t, d_t) - \min_{\tau \in V} \sum_{t \in T_e} l(\tau, \hat{p}_t, d_t). \tag{5.4}$$

5.2 Online Newton Step

We use Online Newton Step [35] to update our threshold τ_t to minimize the regret in (5.4). Since the game is now defined only for the time instances we make a prediction error, the threshold is updated only in those time instances. Therefore, the gradient of our loss is given by

$$l'(\tau_t, \hat{p}_t, d_t) = \frac{-J_{d_t} d_t}{1 + \exp((\tau_t - \hat{p}_t) d_t)} \mathbb{1}_{\hat{d}_t \neq d_t}. \tag{5.5}$$

We define variables B_{t-1} and K_{t-1} as follows

$$B_{t-1} = \sum_{r=1}^{t-1} (l'(\tau_r, \hat{p}_t, d_t))^2,$$
$$K_{t-1} = \sum_{r=1}^{t-1} ((l'(\tau_r, \hat{p}_t, d_t))^2 \tau_r - \frac{1}{\boldsymbol{\alpha}} l'(\tau_r, \hat{p}_t, d_t)),$$

for input parameter $\boldsymbol{\alpha}$. Let $\hat{p}_t$ be in a feasible subset $\mathbb{V}$ of $\mathbb{R}$. Hence, τ_t belongs to the feasible set $\mathbb{V}$, thus, the prediction τ_t is given by

$$\tau_t = \arg\min_{v \in V} \left(v - \frac{K_{t-1}}{B_{t-1}} \right)^2. \tag{5.6}$$

The complete algorithm is given in Alg. 6. Next, we provide regret bounds on the anomaly detector.

Theorem 6. *Using Alg. 6 with parameter*

$$\boldsymbol{\alpha} = \frac{1}{2} \min \left(\exp(-A), \frac{1 + \exp(-A)}{4AJ} \right),$$

we achieve the following anomaly detection regret bound for (5.3)

$$R_{A,T} \leq 3 \left(\exp(A) + \frac{4AJ}{1 + \exp(-A)} \right) \log T, \tag{5.7}$$

where A is the diameter of the feasible set $\mathbb{V}$ such that $A \triangleq \sup_{x,y \in V} \|x - y\|$, $J_{max} \triangleq \max(J_{-1}, J_{+1})$ and $J_{min} \triangleq \min(J_{-1}, J_{+1})$.

Proof. The loss function is given by

$$l(\tau_t, \hat{p}_t, d_t) = J_{d_t} \log \left(1 + \exp(-(\tau_t - \hat{p}_t) d_t)\right). \tag{5.8}$$

Algorithm 6 Anomaly Detector

Determine error weights J_{d_t} for $d_t \in \{-1, +1\}$
Select scoring mapping $\Phi(\cdot)$
Select feasible set $\mathbb{V}$
Initialize $\boldsymbol{\alpha}$
Initialize τ_1
for $t = 1$ **to** T **do**
    Observe $\boldsymbol{x}_t$
    Get density estimation $\hat{g}_t^u(\boldsymbol{x}_t)$ according to Alg. 5
    Produce score $\hat{p}_t = \Phi(\hat{g}_t^u(\boldsymbol{x}_t))$.
    Declare prediction $\hat{d}_t = \text{sign}(\tau_t - \hat{p}_t)$.
    Set $l' = -J_{d_t} d_t (1 + \exp(\tau_t - \hat{p}_t) d_t)^{-1} \mathbb{1}_{\hat{d}_t \neq d_t}$
    $B_t = B_{t-1} + (l')^2$
    $K_t = K_{t-1} + ((l')^2 \tau_t - l'/\boldsymbol{\alpha})$
    $\tau_{t+1} = \arg\min_{v \in V} (v - K_t/B_t)^2$
end for

We take the first derivative of the loss function as

$$l'(\tau_t, \hat{p}_t, d_t) = \frac{-J_{d_t} d_t}{1 + \exp((\tau_t - \hat{p}_t) d_t)}, \tag{5.9}$$

and second derivative as

$$l''(\tau_t, \hat{p}_t, d_t) = \frac{J_{d_t} \exp((\tau_t - \hat{p}_t) d_t)}{(1 + \exp((\tau_t - \hat{p}_t) d_t))^2}, \tag{5.10}$$

since $d_t^2 = 1$ for $d_t \in \{-1, +1\}$. Since $\hat{p}_t$ and τ_t are in the feasible set $\mathbb{V}$, the loss function is λ-exp-concave for $\lambda = J_{min} \exp(-A)$ since

$$\begin{aligned} \frac{(l'(\tau_t, \hat{p}_t, d_t))^2}{l''(\tau_t, \hat{p}_t, d_t)} &= J_{d_t} \exp(-(\tau_t - \hat{p}_t) d_t), \\ &\geq J_{min} \exp(-A). \end{aligned}$$

The gradient of the loss function is upper bounded as

$$\| l'(\tau_t, \hat{p}_t, d_t) \| \leq Y, \tag{5.11}$$

where

$$Y = \frac{J_{max}}{(1 + \exp(-A))}. \tag{5.12}$$

Using Online Newton Step [35] with $\boldsymbol{\alpha} = 0.5 \min(\lambda, 1/4AY)$, we achieve a regret bound of $3(\lambda^{-1} + 4AY) \log T$. Therefore

$$R_{A,T} \leq 3 \left(\frac{\exp(A)}{J_{min}} + \frac{4AJ_{max}}{1 + \exp(-A)} \right) \log T. \tag{5.13}$$

□

The result of Theorem 6 shows that our misclassification regret is logarithmically dependent on the time horizon T. However, its dependence on the diameter of the feasible set is exponential. To circumvent this problem, we can use suitable transformations such as $\hat{p}_t(\boldsymbol{x}_t) = \log(1 + \hat{g}_t^u(\boldsymbol{x}_t))$. However, attenuating the density estimation output too much may decrease robustness.

Corollary 5. *Running Alg. 6 with transformation $\Phi(x) = \log(1 + x)$ results in the following regret if density estimation is upper-bounded by $P \geq \hat{g}_t^u(\boldsymbol{x}_t)$.*

$$R_{A,T} \leq 3 \left((1 + P) \left(1 + \frac{4 \log(1 + P) J}{2 + P} \right) \right) \log T. \tag{5.14}$$

Proof of Corollary 5. Putting $A = \log(1 + P)$ directly in Theorem 6 concludes the proof. □

Remark 11. *In a semi-supervised setting, the threshold can be updated at times when a feedback is received. In this setting, the same performance bounds will hold if we define the regret for only the times feedback is received.*

Remark 12. *In unsupervised setting, we can use a fixed step threshold update instead of the Newton Step approach. Suppose, we want to achieve a false alarm rate of α. If we increment the threshold by δ, when we observe a sample with density estimate above the threshold and decrement it by $\delta(1 - \alpha)/\alpha$ when the observed sample has a density estimate below the threshold, we can converge to the optimal threshold with false alarm rate of α for sufficiently small δ and sufficiently large number of observations since our density estimation has logarithmic performance guarantees.*

5.3 Chapter Summary

We have introduced a dynamic threshold selection scheme, which updates the threshold whenever a feedback about the observed sample is available (i.e., whether it is anomalous or not).

In Chapter 5.1, we have adopted the logistic loss function as our surrogate loss for convex analysis.

In Chapter 5.2, we proposed our algorithm which updates the threshold using Online Newton Step to achieve logarithmic performance bounds (minimax optimal) with respect to the best threshold selected in hindsight.

Chapter 6

Experiments

In this chapter, we demonstrate the performance of our algorithm both on synthetic and real data. We use two synthetic datasets and two real datasets to show how our algorithm performs individually and in comparison to the state-of-the art algorithms.

In Chapter 6.1, we are detecting outliers in a synthesized dataset which is generated by a nonstationary Gaussian process. This experiment compares the performances of the algorithms when the anomalies consists of outliers.

In Chapter 6.2, we compare the algorithms when detecting change anomalies in a nonstationary environment, where the anomalies are caused by the leakage of normal data before the source statistics changes. This experiment, on the other hand, is adversarial since the anomalies consist of the recently modeled normal data (because of the leakage).

In Chapter 6.3, we show the performance of our algorithm in comparison to the state-of-the-art in a real world dataset. We use the famous Iris dataset [74].

In Chapter 6.4, we compare the algorithms with a real dataset [75] in an unsupervised setting.

The non-density based anomaly detection approaches such as SVM [45, 46], nearest neighbors [47, 48] and clustering [49] can be thought of as the fitting of appropriate kernels to decide the outliers (e.g., SVM fits appropriate kernels to determine its boundaries and nearest neighbors fits the appropriate kernels on the training sample points respectively). Therefore, we use the Kernel Density Estimator algorithm (KDE) [44] in our performance comparisons for both the density estimation and the anomaly detection. For the benefit of KDE, we use the optimal kernel selection in accordance with the dataset and optimally tune its bandwidth parameter with Silverman's rule [76]. Moreover, we compare our algorithm against the Maximum Likelihood algorithm (ML) [42], which optimally fits an appropriate parametric model to the dataset. Additionally, we compare our method against the adaptive algorithms Filtering and Hedging for Time-varying Anomaly Recognition (FHTAGN) [20] and Online Convex Programming (OCP) [34]. All of these algorithms first estimate a density function for the normal data and thresholds that function to detect anomalies. We used the same thresholding scheme in Alg. 6 in all of the algorithms for a fair comparison. In the unsupervised real data experiment, the algorithms use a fixed step threshold update for fixed false positive rate as in Remark 12. In all of the experiments, we showed the log-loss performances of the density estimators for normal data to compare their modeling power capabilities. The density estimation part of our algorithm is called Minimax Optimal Density Estimator (MODE) and the anomaly detector is called Anomaly Detection with Minimax Optimal Density Estimation (ADMODE).

To compare the anomaly detection performances of the algorithms, we used the Area Under Curve (AUC) parameter [77]. We used the AUC parameter in [78,79], and approximated it by using an approach similar to [80], where we have sampled the ROC curve at multiple points by varying the discrimination threshold [80] of each method. This sampling of the ROC curve provided different True Positive Rate (TPR) and False Positive Rate (FPR) pairs. Suppose TPR_i, FPR_i for $i = 1, 2, 3, \ldots, n$ are the sampled points of the ROC curve at different values including the trivial points $(0, 0)$ and $(1, 1)$. Suppose further, these pairs are sorted in an ascending manner according to FPR_i values. Then, we can fit a

piecewise linear function onto these samples to approximate the ROC curve. The area under this plot is given by

$$AUC = \sum_{i=1}^{n-1} 0.5\,(TPR_{i+1} + TPR_i)\,(FPR_{i+1} - FPR_i). \tag{6.1}$$

We use this AUC metric [78] to evaluate the performances of the anomaly detectors. We emphasize that the online setting is different than the batch learning setting. During the course of the algorithms' run, we do not have fixed TPR/FPR pairs, they evolve through time, hence, provide an AUC metric evolving with time. We sample the ROC curve by varying the threshold that determines the behavior of the anomaly detector [80]. However, our online algorithm, i.e., Alg. 6, dynamically updates the threshold. Hence, to sample the ROC curve at different points, we vary the cost metrics J_{-1} and J_{+1}. These cost metrics are varied one at a time, where we fix one of them to 1 and exponentially vary the other one in 100 steps from 2^{-10} to 1 (i.e., the other cost metric is selected from the set $\{1, 2^{-0.1}, 2^{-0.2}, \ldots, 2^{-9.8}, 2^{-9.9}, 2^{-10}\}$). Since the update speed of the threshold is different at false positives and false negatives, this approach provides us with multiple samples on the ROC curve, which we use to approximate AUC.

We set $H_{\min} = T^{-2}$, $H_{\max} = T$ and $H_{\max} = T^2$ for the density estimation of our algorithm in supervised and unsupervised experiments respectively. The algorithms use $\Phi(x) = \log(1 + x)$ and $\Phi(x) = \log(x)$ transformation on the density estimations for the threshold update scheme in supervised and unsupervised experiments respectively. The diameter of the feasible set A is set to the maximum value of these score estimations for the algorithms. Following [34], OCP is run with the parameter $T_r^{-1/2}$ and $10T_r^{-1/2}$ in supervised and unsupervised experiments respectively for each epoch with length T_r. Following [20], FHTAGN is run with the learning rate $1/\sqrt{t}$ and $10/\sqrt{t}$ at time t in supervised and unsupervised experiments respectively. ML and KDE uses sliding windows of length $10 \log t$, $5 \log t$ and $100 \log T$ at time t in synthetic experiments, Iris Dataset and Occupancy Detection respectively.

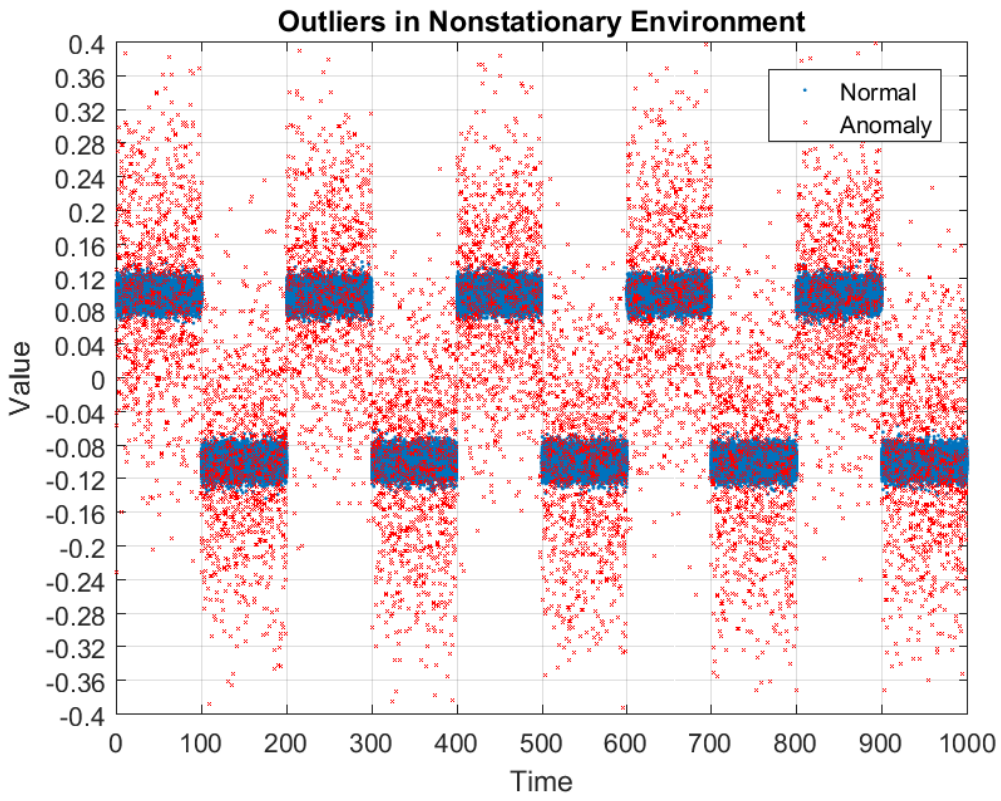

Figure 6.1: Visualization of the Outliers in Nonstationary Environment Dataset (normal and anomalous sample points)

6.1 Outliers in Nonstationary Environment

In the first part of the experiments, we compare the performances of the algorithms in detecting outliers in a nonstationary environment. In each trial of the experiments we create a length 1000, i.e., $T = 1000$, dataset that is generated from a Gaussian process with mean changing between 0.1 and -0.1 every 100 samples and variance of 10^{-4}. Then, we randomly select 100 indices that will contain the anomalous data. The samples at the indices containing anomalies are generated from Gaussian processes with mean equal to the mean of the normal data at that index and variance equal to 10^{-2}. We feed this dataset to all of the algorithms and repeat this experiment for 100 trials. The log-loss performances and AUC performances are illustrated in Fig. 6.2 and 6.3 respectively. The normal and anomalous data points generated in all 100 trials are plotted in

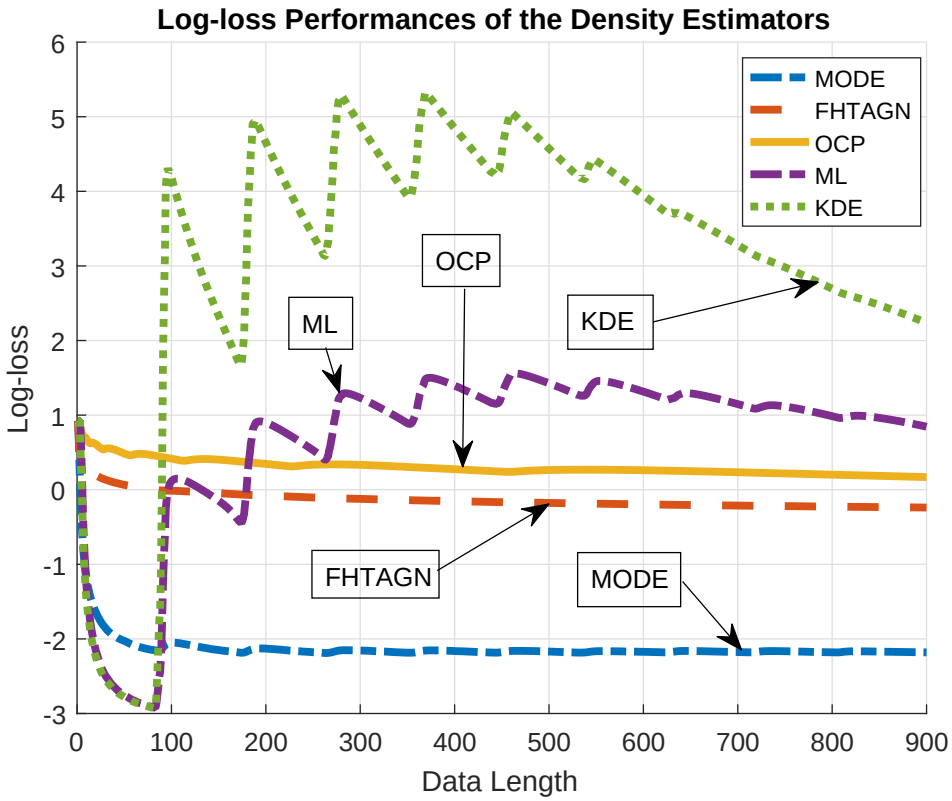

Figure 6.2: Average log-loss performances of the density estimation algorithms in the Outliers in Nonstationary Environment Dataset

Fig. 6.1 for illustration of the dataset. As can be seen in Fig. 6.1, the respective position of the outliers to the normal data stays the same.

As can be seen in Fig. 6.2, the algorithms OCP, FHTAGN and MODE are more robust since their log-loss performances are not affected from the environment changes. However, ML and KDE algorithm suffer greatly from the change of source statistics since their tolerance to the nonstationary environment are quite lower. Nonetheless, our algorithm, i.e., MODE achieves a much lower log-loss since its modeling capabilities greatly surpass OCP and FHTAGN.

From Fig. 6.3, we again see that FHTAGN outperforms OCP as in the log-loss performances. Similarly, ML outperforms OCP and FHTAGN at the beginning but its performance deteriorates over time. However, the deterioration speed of its AUC performance is slower than in its log-loss so that it is only

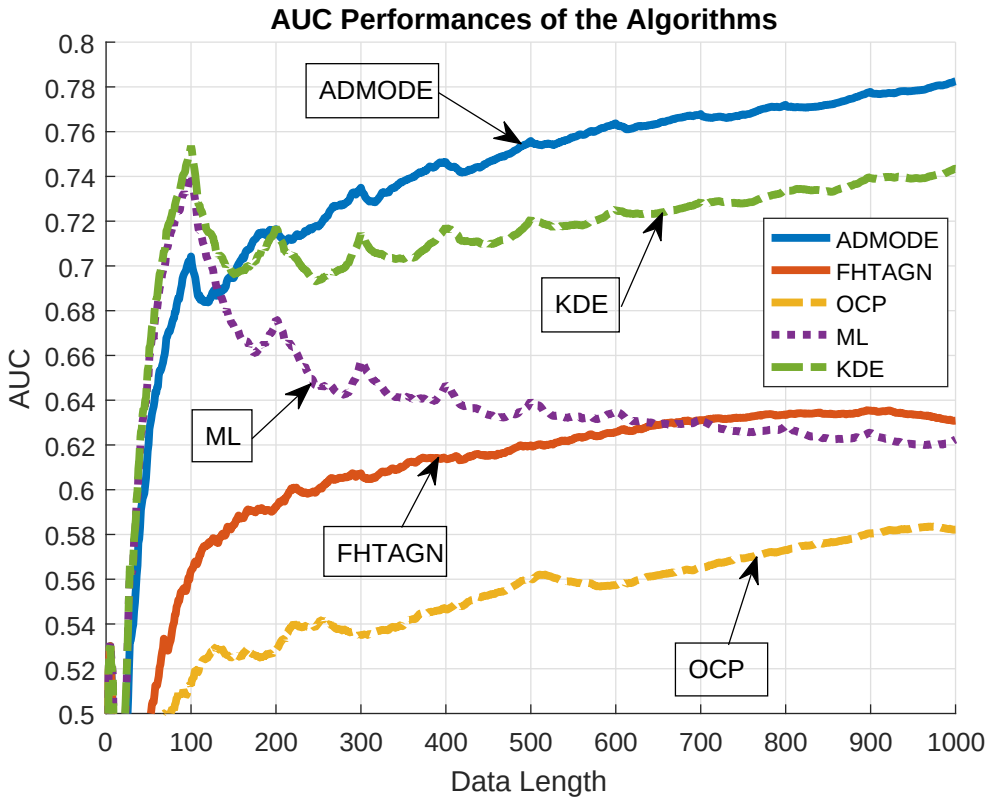

Figure 6.3: Average AUC performances of the anomaly detection algorithms in the Outliers in Nonstationary Environment Dataset

outperformed by FHTAGN while still having better performance than OCP. The most interesting performance is maybe achieved by KDE. While KDE performed the worst in terms of log-loss, its AUC performance is significantly better than the other competitors. While its log-loss behavior, similar to ML, deteriorated over time, its AUC performance increases ever so slightly. This perhaps shows that even though KDE may not necessarily model the normal data in the best way, its modeling distinguishes outliers quite well. Nonetheless, our algorithm, ADMODE, significantly outperforms all of the other algorithms in a stable and robust way by keeping its AUC performance nearly unaffected to the changes in the source statistics. ADMODE is able to achieve this superior performance because of its individual sequence approach and superior modeling due to achieving the performance of the best model with the optimal parameters tuned to the underlying sequence.

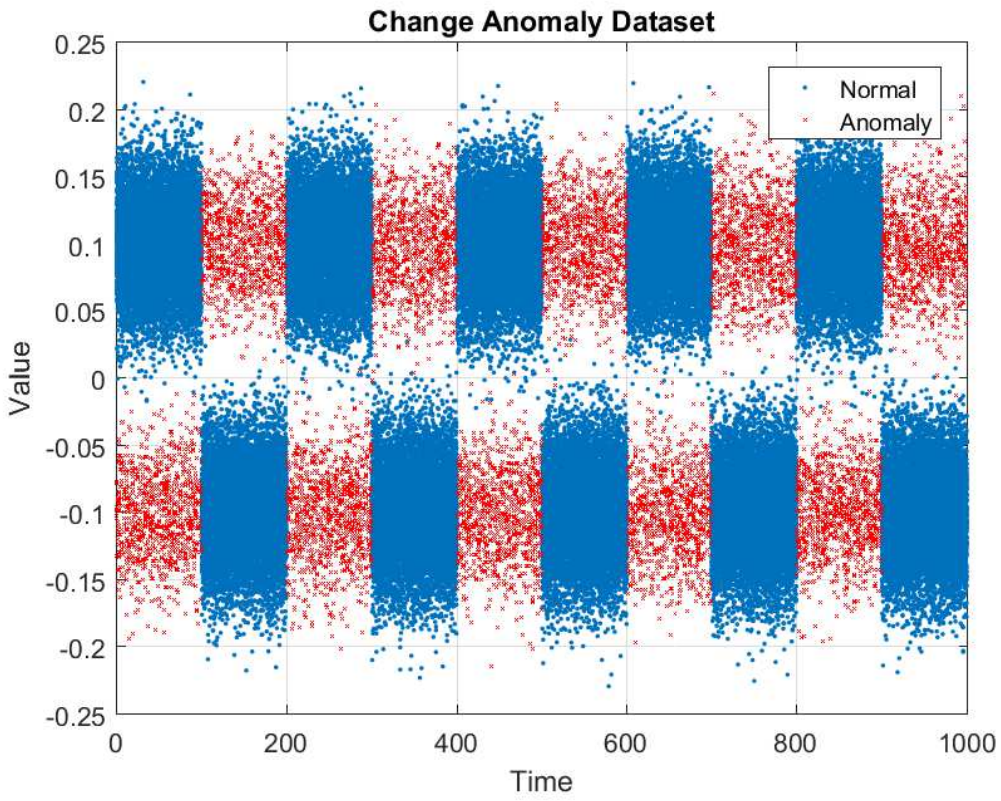

Figure 6.4: Visualization of the Change Point Anomaly Dataset (normal and anomalous sample points)

6.2 Change Point Anomalies

In the second part of the experiments, we compare the performances of the algorithms in detecting change anomalies in a nonstationary environment, i.e., the anomalous samples in a given time segment are caused by the leakage of normal data from the previous time segment. In each trial of the experiments, we create a length 1000, i.e., $T = 1000$ dataset that is generated from a Gaussian process with mean changing between 0.1 and -0.1 every 100 samples and variance of 10^{-3}. Then, we randomly select 100 indices that will contain the anomalous data. The samples at the indices containing anomalies have their means changed with the mean of the normal data of the previous section. We feed this dataset to all of the algorithms and repeat this experiment for 100 trials. The log-loss performances

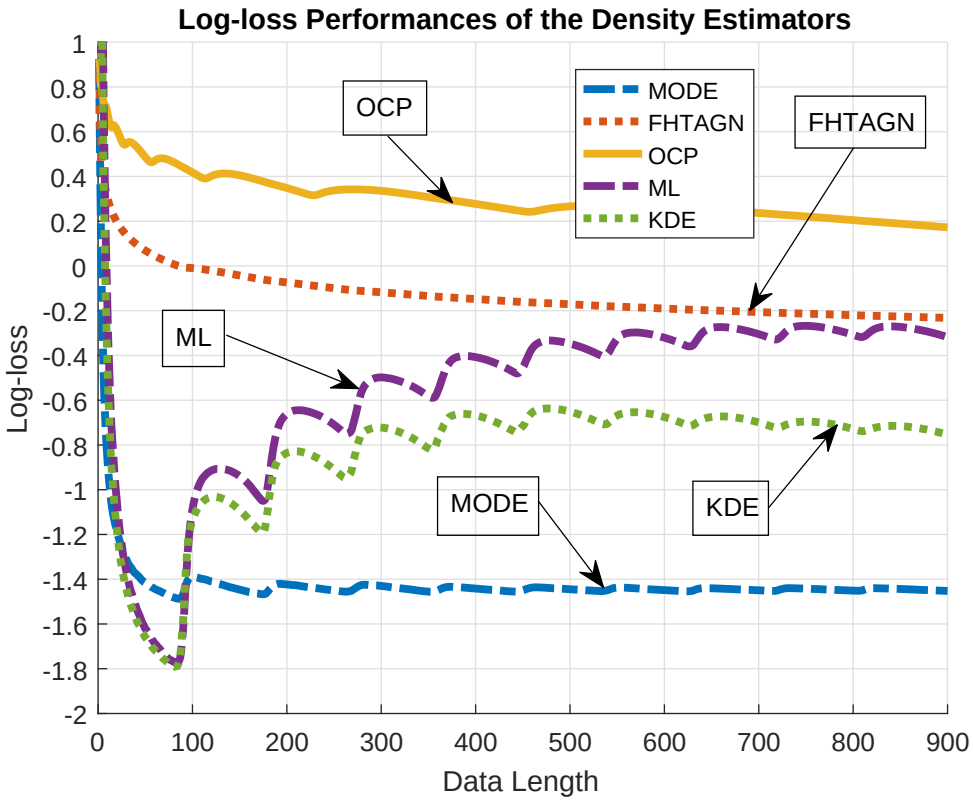

Figure 6.5: Average log-loss performances of the density estimation algorithms in the Change Point Anomaly Dataset

and AUC performances are illustrated in Fig. 6.5 and 6.6 respectively. The normal and anomalous data points generated in all 100 trials are plotted in Fig. 6.4 for illustration of the dataset. As can be seen in Fig. 6.4, the position of the anomalies in a given time segment, e.g., $t \in \{401, 402, 403, \ldots, 498, 499, 500\}$), is similar to the position of the normal data in the previous time segment, e.g., $t \in \{301, 302, 303, \ldots, 398, 399, 400\}$), since the anomalies in this dataset are created from the leakage of normal data to the successive time segment.

As can be seen in Fig. 6.5, the algorithms OCP, FHTAGN and MODE are more robust since their log-loss performances are not really affected from the environment changes. However, ML and KDE algorithm suffer from the change of source statistics albeit with smaller margins compared to the previous experiment since this dataset has a higher variance. Nonetheless, our algorithm, i.e., MODE achieves a much lower log-loss since its modeling capabilities greatly surpass all

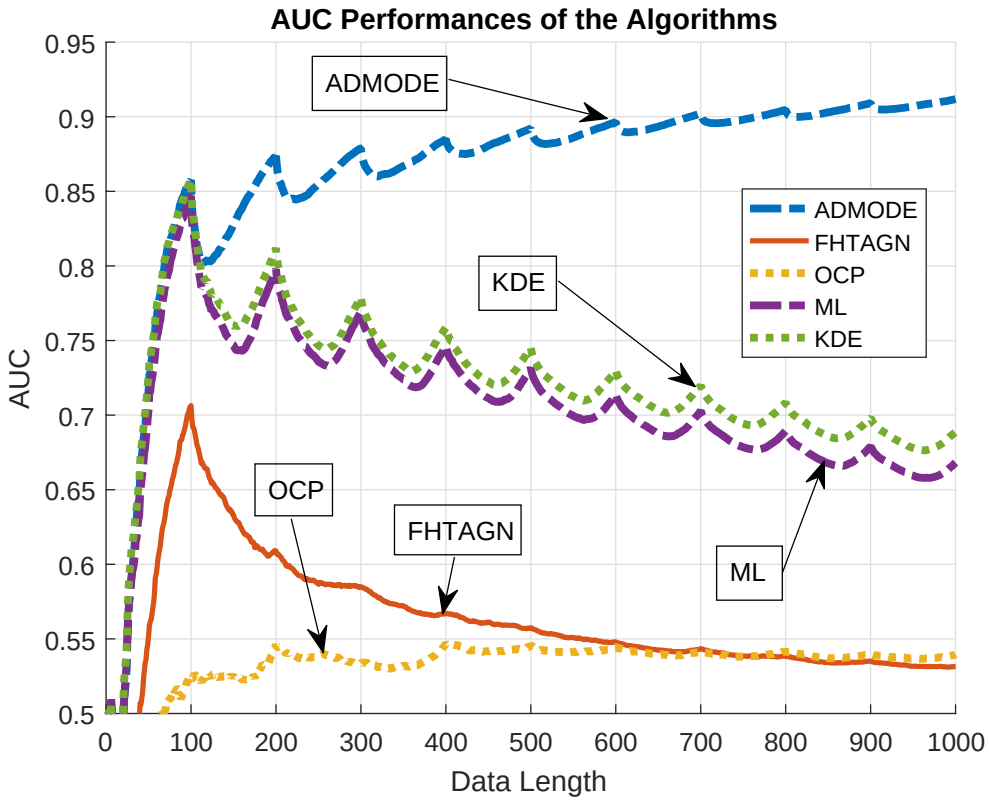

Figure 6.6: Average AUC performances of the anomaly detection algorithms in the Change Point Anomaly Dataset

of the algorithms because of its individual sequence approach.

From Fig. 6.6, we see that FHTAGN outperforms OCP at the beginning but its performance starts declining with the first environment change until its performance is lower than OCP at the end of the dataset. While ML and KDE outperform OCP and FHTAGN similarly in log-loss, their performance seriously decline with each change in environment. Nonetheless, our algorithm, ADMODE, significantly outperforms all of the other algorithms in a stable and robust way because of its superior modeling capabilities. Its AUC performance remains unaffected by the changes in the source statistics, and continues to increase as its learning continues.

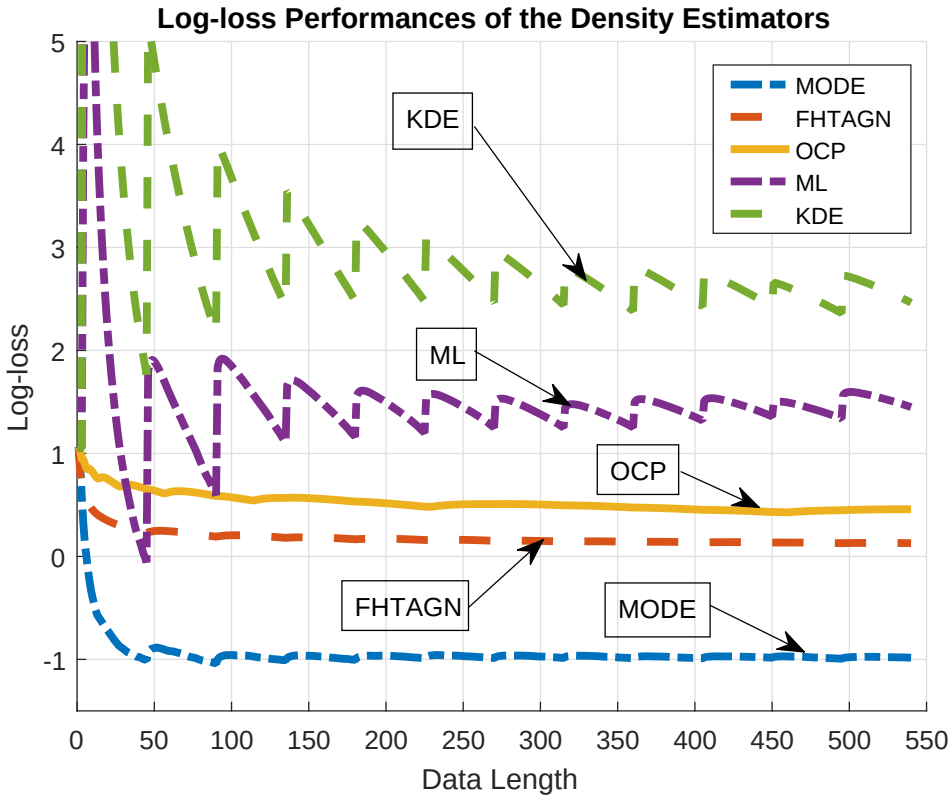

Figure 6.7: Average log-loss performances of the density estimation algorithms in the Iris Dataset

6.3 Iris Dataset

To compare the algorithms in a real life dataset we use the famous Iris dataset [74]. Iris dataset contains 3 classes with 50 instances each. Each instance contains 4 features. We preprocess the dataset so that we separate each feature and treat each feature of each class as separate classes. This creates a dataset of 12 classes with 50 instances each. To get a sequential dataset, we randomly decide on an order of appearance for 12 classes and concatenate them to get a time series of length 600. Then, each 50 length section is shuffled randomly in itself. Lastly, we randomly select 5 samples for each section and mark them as anomalous. These 5 samples in each section are substituted according to a cyclic order of sections we randomly decide. We average the results of 100 trials. We illustrate the log-loss and AUC performances in Fig. 6.7 and 6.8 respectively.

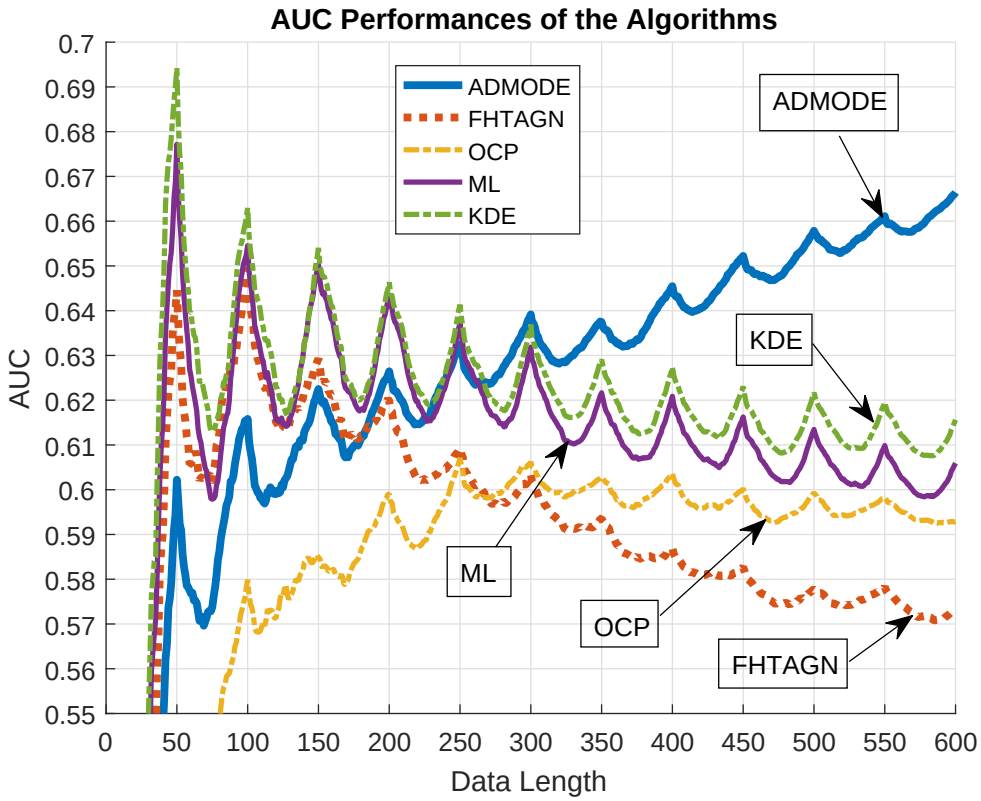

Figure 6.8: Average AUC performances of the anomaly detection algorithms in the Iris Dataset

As can be seen in Fig. 6.7, the algorithms OCP, FHTAGN and MODE are more robust since their log-loss performances are not really affected from the environment changes. However, the performances of ML and KDE suffer with each change of the source statistics. Nonetheless, our algorithm, i.e., MODE, again achieves a much lower log-loss since its modeling capabilities greatly surpass all of the algorithms. From Fig. 6.8, we see that FHTAGN shows better performance than OCP at the beginning. However, its performance rapidly declines and it is finally outperformed by OCP approximately at the 250^{th} sample. ML and KDE have better performances than OCP and FHTAGN even though they were outperformed in the log-loss performances. Moreover, their performances greatly decline with each change in the environment. Nonetheless, our algorithm, ADMODE, significantly outperforms all of the other algorithms in a stable and robust way because of its superior modeling capabilities. ADMODE is the only algorithm whose AUC performance increases in a decisive manner.

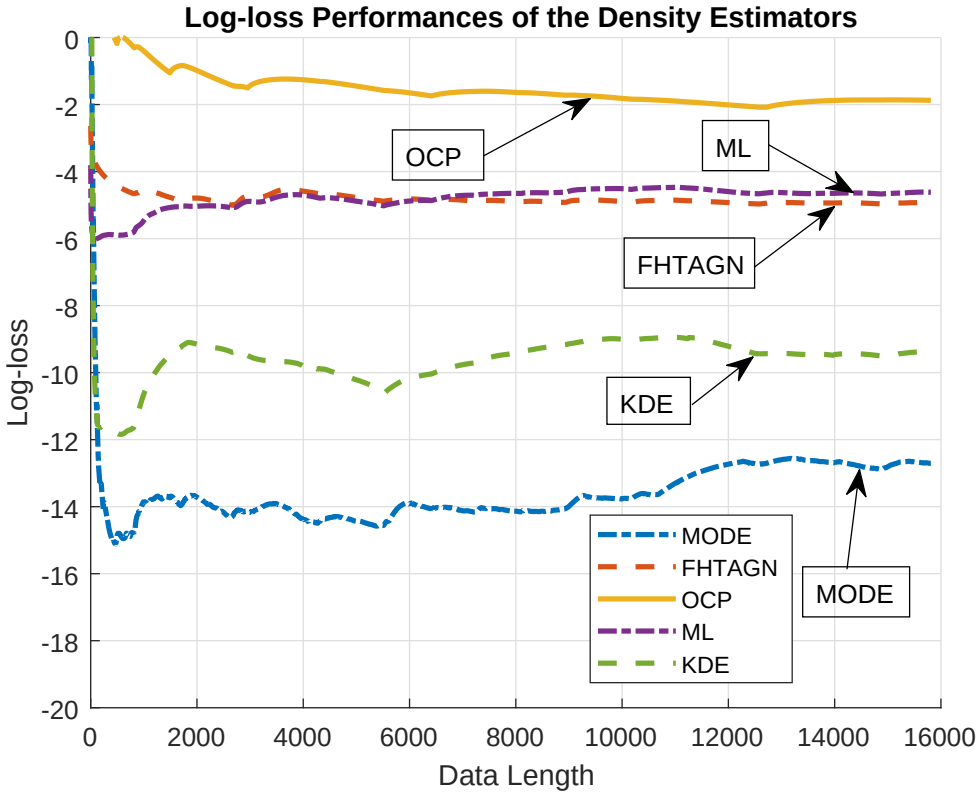

Figure 6.9: Average log-loss performances of the density estimation algorithms in the Occupancy Detection Dataset

6.4 Occupancy Detection

In the last experiment, we have compared the algorithms with a real dataset in an unsupervised setting. We have used the Occupancy Detection dataset [75], which contains 2 classes (occupied or not) and 5 features, which are Temperature (in Celsius), Relative Humidity (%), Light (in Lux), CO2 (in ppm) and Humidity Ratio. This dataset is divided into three time series. We have concatenated these sets to create a single time series of length 20560, where 15810 are normal and 4750 are anomalies. We have normalized all the features to $[0, 1]$. Since we are working in unsupervised setting, the algorithms update their density with each data sample. The log-loss is capped at 300. For the learning rate of the unsupervised update of the threshold, we have exponentially searched over $[\delta, 1/\delta]$ (with multiplicative increments of $2^{0.1}$), where $\delta = 300/T$ ($T = 20560$), since it

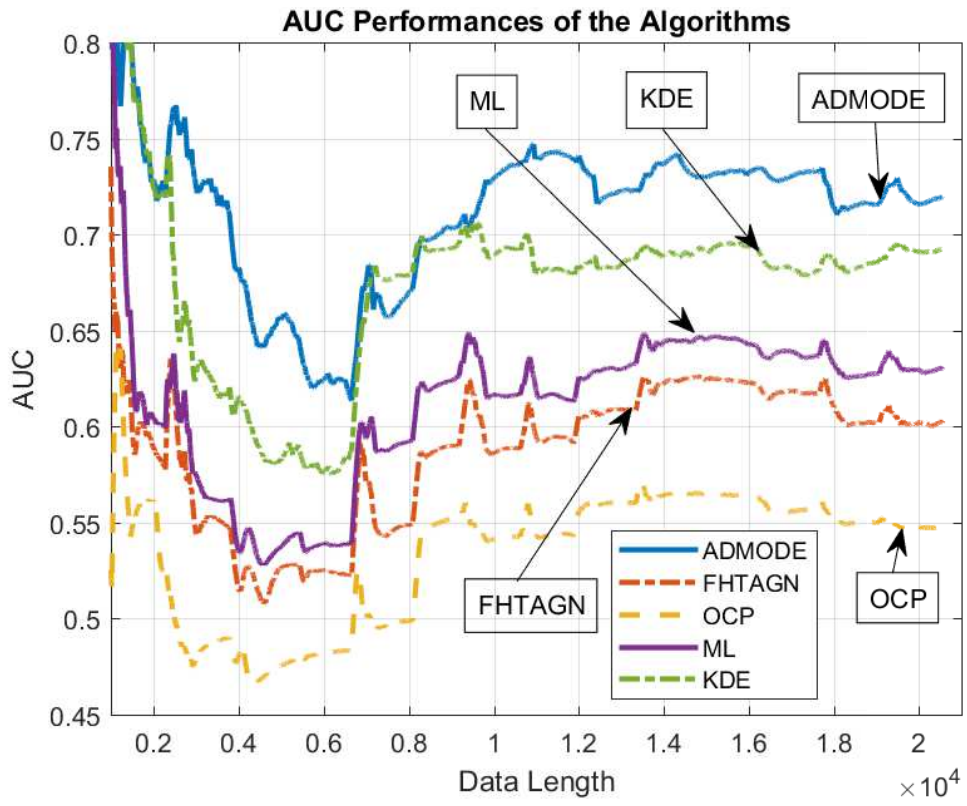

Figure 6.10: Average AUC performances of the anomaly detection algorithms in the Occupancy Detection Dataset.

is the minimum learning increment needed to transverse over the whole log-loss space. We illustrate the log-loss and AUC performances in Fig. 6.9 and 6.10. For each algorithm, the best AUC from the optimal learning rate is plotted.

As can be seen in Fig. 6.9, OCP has the worst log-loss performance. FHTAGN and ML have similar performances (better than OCP). Even though KDE is able to outperform them, it still performs significantly worse than MODE. From Fig. 6.10, we see that the performance order did not change much. The only difference is between ML and FHTAGN. Although FHTAGN had better log-loss performance, ML had showed better AUC performance. Because of the characteristics of the dataset, all of the algorithms have shown a similar behavior, where their AUC performance gradually decreases until $\approx 7000^{th}$ sample and then increases until convergence. For our algorithm, the performance decrease by these changes are significantly less. Thus, ADMODE outperforms the other algorithms.

6.5 Chapter Summary

In Chapter 6.1, we compared the performances of the algorithms when the anomalies consists of outliers. Our algorithm, i.e., MODE achieved a much lower log-loss since its modeling capabilities greatly surpass the others. In anomaly detection, our algorithm, ADMODE, significantly outperformed all of the other algorithms in a stable and robust way by keeping its AUC performance nearly unaffected to the changes in the source statistics. ADMODE is able to achieve this superior performance because of its individual sequence approach and superior modeling due to achieving the performance of the best model with the optimal parameters tuned to the underlying sequence.

In Chapter 6.2, we compared the algorithms when detecting change anomalies. Similarly, our algorithms were able to outperform the others. The performance of our algorithm remained unaffected by the changes in the source statistics and continued to increase as its learning continues.

In Chapter 6.3, we further demonstrated our algorithms' superior performances in comparison to the state-of-the-art in a famous real world dataset.

In Chapter 6.4, we compared the algorithms with a real dataset in an unsupervised setting. Since our algorithm is more robust to the changes in the statistics in the dataset, the performance decrease caused by these changes were significantly less than the others.

Chapter 7

Conclusions

In this thesis, we have introduced a truly sequential anomaly detection algorithm that detects anomalies in a time series. Our algorithm has two stage and consists of first estimating the density of the nominal data in an online manner and then thresholding this density estimation to detect anomalies. We approach this problem from an information theoretic perspective and propose minimax optimal schemes for both stages to create an optimal anomaly detection algorithm in a strong deterministic sense. For the first stage of our algorithm, we, for the first time in literature, have introduced a truly sequential minimax optimal density estimation algorithm for general nonstationary exponential-family of sources without any assumptions on the observation sequence. Moreover, for the second stage, we have proposed a threshold selection scheme that is minimax optimal with respect to the best threshold selected in hindsight. In both stages of our algorithm, we achieve logarithmic regret bounds by adaptively updating its parameters in a completely sequential manner. Our algorithm showed significant performance gains against the state-of-the-art methods in both synthetic and real datasets.

Bibliography

[1] G. Hebrail and A. Berard, "UCI machine learning repository," 2012.

[2] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection : A survey," *ACM Computing Surveys*, vol. 11, pp. 1–72, 2009.

[3] B. Baingana and G. B. Giannakis, "Joint community and anomaly tracking in dynamic networks," *IEEE Transactions on Signal Processing*, vol. 64, pp. 2013–2025, April 2016.

[4] K. Cohen, Q. Zhao, and A. Swami, "Optimal index policies for anomaly localization in resource-constrained cyber systems," *IEEE Transactions on Signal Processing*, vol. 62, pp. 4224–4236, Aug 2014.

[5] J. Sharpnack, A. Rinaldo, and A. Singh, "Detecting anomalous activity on networks with the graph fourier scan statistic," *IEEE Transactions on Signal Processing*, vol. 64, pp. 364–379, Jan 2016.

[6] T. Xie, N. M. Nasrabadi, and A. O. Hero, "Learning to classify with possible sensor failures," *IEEE Transactions on Signal Processing*, vol. 65, pp. 836–849, Feb 2017.

[7] A. B. Tsybakov *et al.*, "On nonparametric estimation of density level sets," *The Annals of Statistics*, vol. 25, no. 3, pp. 948–969, 1997.

[8] C. M. Bishop, *Neural Networks for Pattern Recognition*. New York, NY, USA: Oxford University Press, Inc., 1995.

[9] A. Penalver and F. Escolano, "Entropy-based incremental variational bayes learning of gaussian mixtures," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, pp. 534–540, March 2012.

[10] X. Ding, Y. Li, A. Belatreche, and L. P. Maguire, "Novelty detection using level set methods," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, pp. 576–588, March 2015.

[11] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New Yotk: John Willey & Sons, 1973.

[12] A. S. Weigend and A. N. Srivastava, "Predicting conditional probability distributions: a connectionist approach," *International Journal of Neural Systems*, vol. 6, no. 2, pp. 109–118, 1995.

[13] A. R. Barron and T. M. Cover, "Minimum complexity density estimation," *IEEE Transactions on Information Theory*, vol. 37, pp. 1034–1054, Jul 1991.

[14] E. Müller, I. Assent, R. Krieger, S. Günnemann, and T. Seidl, "Dens-est: Density estimation for data mining in high dimensional spaces," in *Proc. SIAM International Conference on Data Mining (SDM 2009), Sparks, Nevada, USA.*, pp. 173–184, SIAM, 2009.

[15] Y. Cao, H. He, and H. Man, "Somke: Kernel density estimation over data streams by sequences of self-organizing maps," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, pp. 1254–1268, Aug 2012.

[16] Y. Nakamura and O. Hasegawa, "Nonparametric density estimation based on self-organizing incremental neural network for large noisy data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, no. 99, pp. 1–10, 2016.

[17] P. Padungweang, C. Lursinsap, and K. Sunat, "A discrimination analysis for unsupervised feature selection via optic diffraction principle," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, pp. 1587–1600, Oct 2012.

[18] E. Izquierdo-Verdiguier, V. Laparra, R. Jenssen, L. Gómez-Chova, and G. Camps-Valls, "Optimized kernel entropy components," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, no. 99, pp. 1–7, 2016.

[19] W. Bian, T. Zhou, A. M. Martinez, G. Baciu, and D. Tao, "Minimizing nearest neighbor classification error for nonparametric dimension reduction," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, pp. 1588–1594, Aug 2014.

[20] M. Raginsky, R. M. Willett, C. Horn, J. Silva, and R. F. Marcia, "Sequential anomaly detection in the presence of noise and limited feedback," *IEEE Transactions on Information Theory*, vol. 58, pp. 5544–5562, Aug 2012.

[21] A. Gyorgy, T. Linder, and G. Lugosi, "Efficient tracking of large classes of experts," *IEEE Transactions on Information Theory*, vol. 58, pp. 6709–6725, Nov 2012.

[22] A. György, T. Linder, and G. Lugosi, "Efficient tracking of large classes of experts," in *2012 IEEE International Symposium on Information Theory Proceedings*, pp. 885–889, July 2012.

[23] A. Gyorgy, T. Linder, and G. Lugosi, "Tracking the best quantizer," *IEEE Transactions on Information Theory*, vol. 54, pp. 1604–1625, April 2008.

[24] N. Cesa-Bianchi, Y. Freund, D. Haussler, D. P. Helmbold, R. E. Schapire, and M. K. Warmuth, "How to use expert advice," *J. ACM*, vol. 44, pp. 427–485, May 1997.

[25] N. Merhav and M. Feder, "Universal prediction," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2124–2147, 1998.

[26] A. Bain, D. Crisan, D. Dawson, D. Geman, G. Grimmett, I. Karatzas, F. Kelly, Y. Le Jan, B. Øksendal, G. Papanicolaou, *et al.*, *Fundamentals of stochastic filtering*, vol. 3. Springer, 2009.

[27] C. R. Shalizi *et al.*, "Dynamics of bayesian updating with dependent data and misspecified models," *Electronic Journal of Statistics*, vol. 3, pp. 1039–1074, 2009.

[28] H. Ozkan, F. Ozkan, and S. S. Kozat, "Online anomaly detection under markov statistics with controllable type-i error," *IEEE Transactions on Signal Processing*, vol. 64, no. 6, pp. 1435–1445, 2016.

[29] H. Ozkan, F. Ozkan, I. Delibalta, and S. S. Kozat, "Efficient NP tests for anomaly detection over birth-death type DTMCs," *Journal ofSignal Processing Systems*, vol. 1, pp. 1–10, June 2016.

[30] V. Vapnik and S. Mukherjee, "Support vector method for multivariate density estimation," in *Advances in neural information processing systems*, pp. 659–665, 2000.

[31] B. O. Koopman, "On distributions admitting a sufficient statistic," *Transactions of the American Mathematical Society*, vol. 39, no. 3, pp. 399–409, 1936.

[32] K. S. Azoury and M. K. Warmuth, "Relative loss bounds for on-line density estimation with the exponential family of distributions," *Machine Learning*, vol. 43, pp. 211–246, Jun 2001.

[33] A. R. Barron and C.-H. Sheu, "Approximation of density functions by sequences of exponential families," *Ann. Statist.*, vol. 19, pp. 1347–1369, 09 1991.

[34] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pp. 928–936, 2003.

[35] E. Hazan, A. Agarwal, and S. Kale, "Logarithmic regret algorithms for online convex optimization," *Machine Learning*, vol. 69, no. 2, pp. 169–192, 2007.

[36] C. W. Ten, J. Hong, and C. C. Liu, "Anomaly detection for cybersecurity of the substations," *IEEE Transactions on Smart Grid*, vol. 2, pp. 865–873, Dec 2011.

[37] K. B. Dyer, R. Capo, and R. Polikar, "Compose: A semisupervised learning framework for initially labeled nonstationary streaming data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, pp. 12–26, Jan 2014.

[38] M. Herbster and M. K. Warmuth, "Tracking the best linear predictor," *J. Mach. Learn. Res.*, vol. 1, pp. 281–309, 2001.

[39] M. Herbster and M. K. Warmuth, "Tracking the best expert," *Machine Learning*, vol. 32, no. 2, pp. 151–178, 1998.

[40] F. M. J. Willems, "Coding for a binary independent piecewise-identically-distributed source," *IEEE Transactions on Information Theory*, vol. 42, pp. 2210–2217, Nov 1996.

[41] G. I. Shamir and N. Merhav, "Low-complexity sequential lossless coding for piecewise-stationary memoryless sources," *IEEE transactions on information theory*, vol. 45, no. 5, pp. 1498–1519, 1999.

[42] H. V. Poor, *An Introduction to Signal Detection and Estimation*. NJ: Springer, 1994.

[43] C. Horn and R. M. Willett, "Online anomaly detection with expert system feedback in social networks," in *ICASSP*, pp. 1936–1939, 2011.

[44] J. S. Simonoff, *Smoothing methods in statistics*. Springer Science & Business Media, 2012.

[45] D. M. Tax and R. P. Duin, "Support vector data description," *Machine learning*, vol. 54, no. 1, pp. 45–66, 2004.

[46] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[47] K. Zhang, M. Hutter, and H. Jin, "A new local distance-based outlier detection approach for scattered real-world data," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 813–822, Springer, 2009.

[48] G. G. Cabral, A. L. Oliveira, and C. B. Cahú, "Combining nearest neighbor data description and structural risk minimization for one-class classification," *Neural Computing and Applications*, vol. 18, no. 2, pp. 175–183, 2009.

[49] M. Moshtaghi, T. C. Havens, J. C. Bezdek, L. Park, C. Leckie, S. Rajasegarar, J. M. Keller, and M. Palaniswami, "Clustering ellipses for anomaly detection," *Pattern Recognition*, vol. 44, no. 1, pp. 55–69, 2011.

[50] K.-R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, "An introduction to kernel-based learning algorithms," *IEEE transactions on neural networks*, vol. 12, no. 2, pp. 181–201, 2001.

[51] C. Park, J. Z. Huang, and Y. Ding, "A computable plug-in estimator of minimum volume sets for novelty detection," *Operations Research*, vol. 58, no. 5, pp. 1469–1480, 2010.

[52] X. Ding, Y. Li, A. Belatreche, and L. P. Maguire, "An experimental evaluation of novelty detection methods," *Neurocomputing*, vol. 135, pp. 313 – 327, 2014.

[53] I. Steinwart, D. Hush, and C. Scovel, "A classification framework for anomaly detection," *Journal of Machine Learning Research*, vol. 6, no. Feb, pp. 211–232, 2005.

[54] C. Scott and G. Blanchard, "Novelty detection: Unlabeled data definitely help.," in *AISTATS*, pp. 464–471, 2009.

[55] N. Merhav, "On the minimum description length principle for sources with piecewise constant parameters," *IEEE Transactions on Information Theory*, vol. 39, pp. 1962–1967, Nov 1993.

[56] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[57] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, pp. 79–86, 03 1951.

[58] N. Cesa-Bianchi and G. Lugosi, *Prediction, learning, and games*. Cambridge university press, 2006.

[59] V. Vovk, "Aggregating strategies," in *Proceedings of COLT*, pp. 371–383, 1990.

[60] N. Cesa-Bianchi, Y. Freund, D. Helmbold, D. Haussler, R. Schapire, and M. Warmuth, "How to use expert advice," *Annual ACM Symposium on Theory of Computing*, pp. 382–391, 1993.

[61] A. C. Singer and M. Feder, "Universal linear prediction by model order weighting," *IEEE Transactions on Signal Processing*, vol. 47, no. 10, pp. 2685–2699, 1999.

[62] V. Vovk and C. Watkins, "Universal portfolio selection," in *Proceedings of the eleventh annual conference on Computational learning theory*, pp. 12–23, ACM, 1998.

[63] Y. M. Shtar'kov, "Universal sequential coding of single messages," *Problemy Peredachi Informatsii*, vol. 23, no. 3, pp. 3–17, 1987.

[64] A. Orlitsky, N. P. Santhanam, and J. Zhang, "Universal compression of memoryless sources over unknown alphabets," *IEEE Transactions on Information Theory*, vol. 50, no. 7, pp. 1469–1481, 2004.

[65] I. J. Myung, "Tutorial on maximum likelihood estimation," *J. Math. Psychol.*, vol. 47, pp. 90–100, Feb. 2003.

[66] L. Bottou, "Online learning and stochastic approximations," *On-line learning in neural networks*, vol. 17, no. 9, p. 142, 1998.

[67] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural Networks*, vol. 12, no. 1, pp. 145 – 151, 1999.

[68] Y. Nesterov, "A method of solving a convex programming problem with convergence rate o (1/k2)," in *Soviet Mathematics Doklady*, vol. 27, pp. 372–376, 1983.

[69] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[70] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[71] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[72] V. Vovk, "Aggregating strategies," in *Proceedings of COLT*, pp. 371–383, 1990.

[73] N. Cesa-Bianchi, Y. Freund, D. Haussler, D. P. Helmbold, R. E. Schapire, and M. K. Warmuth, "How to use expert advice," *J. ACM*, vol. 44, pp. 427–485, May 1997.

[74] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 7, pp. 179–188, 1936.

[75] L. M. Candanedo and V. Feldheim, "Accurate occupancy detection of an office room from light, temperature, humidity and co2 measurements using statistical learning models," *Energy and Buildings*, vol. 112, pp. 28 – 39, 2016.

[76] B. W. Silverman, *Density estimation for statistics and data analysis*, vol. 26. CRC press, 1986.

[77] D. M. W. Powers, "Evaluation: From precision, recall and f-measure to roc., informedness, markedness & correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011.

[78] A. P. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms," *Pattern Recognition*, vol. 30, no. 7, pp. 1145 – 1159, 1997.

[79] T. Fawcett, "An introduction to {ROC} analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861 – 874, 2006. {ROC} Analysis in Pattern Recognition.

[80] X. Ding, Y. Li, A. Belatreche, and L. P. Maguire, "An experimental evaluation of novelty detection methods," *Neurocomputing*, vol. 135, pp. 313 – 327, 2014.