DYNAMICAL SYSTEMS PERSPECTIVES IN MACHINE LEARNING

BY

SIDDHARTHA SATPATHI

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois Urbana-Champaign, 2021

Urbana, Illinois

Doctoral Committee:

Professor R Srikant, Chair
Professor Carolyn L. Beck
Assistant Professor Sabyasachi Chatterjee
Assistant Professor Bin Hu

ABSTRACT

We look at two facets of machine learning from a perspective of dynamical systems, that is, the data generated from a dynamical system and the iterative inference algorithm posed as a dynamical system. In the former, we look at time series data which is generated from a mixture of processes. Each process exists for a fixed duration and generates i.i.d categorical data points during that duration. More than one process can coexist at a particular time. The goal is to find the number of such hidden processes and the characteristic categorical distribution of each. This model is motivated by the problem of finding error *events* in error-logs from a mobile communication network.

In the second direction, we consider the problem of regression using a shallow overparameterized neural network. Broadly, we look at training the neural network with the gradient descent algorithm on the squared loss function and discuss the generalization properties of the output of the gradient descent algorithm on an unseen data point. We look at two problems in this setting. First, we discuss the effect of ℓ_2 regularization on the squared loss and discuss how different strength of regularization provides a trade-off on the generalization of the neural network. Second, we look at squared loss without regularization and discuss the generalization properties when the true function we are trying to learn belongs to the class of polynomials in the presence of noisy samples. In both the problems, we consider the gradient descent algorithm as a dynamical system and use tools from control theory to analyze this dynamical system.

ACKNOWLEDGMENTS

I would like to thank my advisor, Prof. R Srikant, for helping me in my academic endeavors for the past six years and extending support for my mental health in graduate school. I would like to thank my friends and colleagues from graduate school who have helped me through difficult courses and challenging times in graduate school: Harsh Gupta, Zeyu Zhou, Joseph Lubars, Amish Goel, Aditya Deshmukh, Akshayaa Magesh, and many more. I want to thank my dissertation committee members for providing feedback on the initial stages of my work. I would also like to thank Prof. Sabyasachi Chatterjee for valuable feedback on Chapter 4. I appreciate the careful corrections and suggestions offered by Jan Progen form the departmental Editorial Services office for this dissertation. Lastly, I would like to thank my parents and elder brother for their love and support especially during the pandemic last year.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

In machine learning systems, either the data is generated from a dynamical system or the learning algorithm itself behaves like a dynamical system or both. In problems like clustering or classification, we often assume the data to be given apriori, generated from an i.i.d distribution. Although the data is static, one infers patterns from the data using iterative algorithms optimizing a certain loss function. For example, it is possibly either a greedy algorithm like K-means for clustering or the gradient descent algorithm and its variants. The path of the variables changing through each iteration can be viewed as a dynamical system in discrete time. For example, in gradient descent over a loss function $l(w)$ of variable w, $w_{t+1} = w_t - \epsilon \frac{\partial}{\partial w_t} l(w_t)$. To further simplify the dynamical system, one can approximate it with a continuous-time ordinary differential equation in the limit $\epsilon \to 0$, $\frac{d}{dt} w = -\frac{\partial}{\partial w_t} l(w_t)$. In this dissertation, we look at the continuous-time equivalent of gradient descent and analyze the dynamical system from the perspective of Lyapunov theory [1]. This intuition is carried over to the analysis of gradient descent in discrete time.

In many online systems, the data is modeled like it is generated from a dynamical system. For example, in reinforcement learning [2], the data points in the current time (state reward pairs) are generated based on the past states, actions, and rewards. This generation process of the data can be modeled as a dynamical system that is intrinsic to the environment. Another example is the problem of change-point detection in the context of statistics and signal processing. In its vanilla version, data points are assumed to be generated at regular time intervals from a distribution that changes with time, and one has to infer the time instants that mark the change. An application would be detecting anomalous behavior in a router network [3]. In

this dissertation, we consider time series data from a communication network with millions of data points which is assumed to be generated from a dynamical system with a large number of unknown parameters.

We consider the following two problems in the dissertation.

- **Inference on data generated from a dynamical system:** This application deals with the error-logs from a mobile network. We infer changes in the pattern over time in the error-log. The log data consists of time annotated messages. A message can be thought of as a short phrase or string. An *event* is a hidden construct introduced to model the generation of the time series data. Events can be considered as a generator of messages with timestamps. An event is characterized by a categorical distribution over the set of all messages and an unknown process to generate the timestamps. Each event exists during an unknown time interval and generates i.i.d messages from its characteristic distribution with timestamps from the unknown process. The entire error-log is modeled to be generated from an unknown number of events which can co-exist in time. This is a complex dynamical system with many unknown parameters. We only care to learn the number of events, the time duration in which these events exist and the parameters of the categorical distribution for each event. We connect this problem to topic modeling in natural language processing and also use change-point detection.

- **Gradient descent as a dynamical system:** We consider the problem of regression between data points $f(x_i; w)$ and $y_i, i \in [n]$. Function $f(x_i; w)$ is a two-layer overparameterized neural network. We answer two questions in this setting. Firstly, suppose the weight w is initialized to w_0 and the objective function is the least squares loss with addition of a regularizer $\lambda|w - w_0|^2$. For varying strengths of λ, we show different generalization bounds which can improve upon the generalization performance without addition of regularization. We model gradient descent as a continuous-time dynamical system and use tools like Lyapunov arguments from control theory to analyze the dynamical system. Secondly, we look at the squared error loss without addition of regularization. In this setting we characterize the convergence of weight w

when initialized with w_0. This leads to a generalization result for the neural network at the end of gradient descent iterations. Using Lyapunov theory, we provide results for gradient descent and its continuous-time equivalent.

Organization: In Chapter 2 we present the problem on the inference of error-logs. We provide a scalable algorithm and theoretical analysis for the change detection part of the algorithm. In Chapter 3 we analyze the gradient descent algorithm for an overparameterized neural network with regularization. Our main result in this chapter is to show generalization guarantees as a function of the regularization strength. In Chapter 4, we look at the generalization guarantees for an overparameterized neural network in the absence of regularization. We use the Neural Tangent Kernel approach (described in Chapters 3 and 4) to show convergence of weights under gradient descent and this leads to the generalization results. Chapter 5 provides conclusions and highlights some future directions.

CHAPTER 2

LEARNING LATENT EVENTS FROM NETWORK MESSAGE LOGS

2.1 Introduction

In modern data and web services, such as cellular data/voice services, there is a vast number of network elements, like routers or virtual machines (VMs), which communicate with one another. Efficient management and operations of these network elements are of paramount importance as the network size is growing increasingly complex with new technologies like 5G. An integral component of network management is the ability to identify and understand *error events*. We use error event to describe any failures that occur in the hardware and/or software components of the network. However, the complex interdependence between different network elements poses a significant challenge in characterizing an error event because error messages can be generated in network elements beyond the actual source of error. An error log contains all error messages with timestamps generated from different error events occurring at different network elements. In this dissertation, we are interested in the problem of mining latent error event information from messages in the error log. The mined events are useful for troubleshooting purposes. Also, the correlations captured through each learned event could subsequently provide useful on-line detection of potential errors. While our methodology is broadly applicable to any type of data center network, we validate our algorithms by applying them to a large dataset provided by a major wireless network service provider.

While mining error logs have been studied extensively in different contexts, (see [4, 5] for excellent surveys; also see Section 2.1.2 in [6] for a detailed liter-

ature review) there are some fundamental differences in our setting.

Motivating example: Suppose Alice makes a cellphone call to Bob. This call is first routed through a base station which is attached to a data center verifying the caller credentials. If Alice is not at her home location, a VM at this data center must contact a database at her home location to verify her credentials. Once the credentials are verified, the caller's cellular base station connects to the base station near Bob through a complicated network spanning many geographical locations. Consider two potential error scenarios: (i) an error occurs at a router in the path from Bob to Alice's base station, (ii) an error occurs at a router connecting the data centers verifying the caller's credentials. In either scenario, the call will fail to be established leading to the generation of error messages not only at the failed routers but also at network elements responsible for the call establishment which can be in a different geographical location than the router. Additionally, depending on the vendor of a given network element, the timing and content of the error messages could be different.

Based on the motivating example, we now note the following fundamental characteristics which make our error event mining problem challenging:

- In our setting, the source of an error is usually not known. Furthermore, the same type of error log message could be generated due to many different errors. From a data modeling point of view, each (latent) event can be viewed as a probabilistic-mixture of multiple log-messages and also, the set of log messages generated by different events could have non-zero intersection.

- Each error event can produce a sequence of messages, including the same type of message multiple times, and the temporal order between distinct messages from the same event could vary based on the latency between network elements, network-load, co-occurrence of other uncorrelated events, etc. Thus, the temporal pattern of messages may also contain useful information for our purpose. In our model, the message occurrence times are modeled as a stochastic process.

- These messages could correspond to multiple simultaneous events without any further information on the start-time and end-time of each event.

- An additional challenge arises because the network topology information is unknown, because modern networks are very complicated and are constantly evolving due to the churn (addition or deletion) of routers and switches. Third-party vendor software and hardware have no way of providing information to localize and understand the errors. Thus, topological information cannot be used for event mining purposes.

The practical novelty of our work comes from modeling for all of the above factors and proposing scalable algorithms that learn the latent event signatures (the notion of signature is made precise in Section 2.2 along with their occurrence times.

2.1.1 Contributions

We model each error event as a probabilistic mixture of messages from different sources.[1] In other words, the probability distribution over messages characterizes an event, and thus acts as the signature of the event. Each occurrence of an event also has a start/end time and several messages can be generated during the occurrence of an event. We only observe the messages and their timestamps while the event signatures and duration window is unknown; also there could be multiple simultaneous events occurring in the network. Given this setting, we study the following unsupervised learning problem: *given a collection of timestamped log-messages, learn the latent event signatures and event start/end times.*

The main contributions in this dissertation are as follows:

- *Novel algorithmic framework:* One of the main contributions of dissertation is a novel mapping of our problem which transforms it into a problem of topic discovery in documents. Events in our problem correspond to topics and messages in our problem correspond to words in the topic discovery problem. However, there is no direct analog of documents. Therefore, we use a non-parametric change-point detection algorithm, which has linear computational

[1] It is more precise to use the terminology event-class to refer to a specific fault-type; each occurrence can be referred to as an instance of some event class. However, for simplicity, we simply refer to event-class as event and we just say occurrence of the event to mean instance of this class.

complexity in the number of messages, to divide the message log into smaller subsets called episodes, which serve as the equivalents of documents. After this mapping has been done, we use a well-known algorithm for topic discovery, called Latent Dirichlet Allocation (LDA), to solve our problem. We call our algorithm CD-LDA.

- *Scalable change-point detection:* While the details of the LDA algorithm itself are standard, nonparametric change-point detection as we have used in dissertation is not as well studied. We adapt an idea from [7] to design an $O(n)$ algorithm where n is the number of messages in the message log. Our change detection algorithm uses an easy to compute total-variation (TV) distance. We analyze the sample complexity (i.e., the number of samples required to detect change points with a high-degree of accuracy) of our change-point detection algorithm using the method of types and Pinsker's inequality from information theory. To the best of our knowledge, no such sample complexity results exist for the algorithm in [7].

- *Experimental validation:*We compare our algorithm to two existing approaches adapted to our setting: a Bayesian inference-based algorithm and graph-based clustering algorithm. We show the benefits of our approach compared to these methods in terms of scalability and performance, by applying it to small samples extracted from a large dataset consisting of 97 million messages. We also validate our method against two real-world events by comparing the event signature learned by our method with a domain expert validated event signature for a dataset consisting of 700K messages.[2] Finally, we also show results to indicate scalability of our method by applying to the entire 97 million message dataset.

[2]Note that manual inference of event signatures is not scalable; we did this for the purpose of validation.

2.1.2 Context and Related Work

Data-driven techniques have been very useful in extracting meaningful information out of system-logs and alarms for large and complex systems. The primary goal of this "knowledge" extraction is to assist in diagnosing the underlying problems responsible for log-messages and events. Two excellent resources for the large body of work done in the area are [4, 5]. Next, we outline some of the key challenges in this knowledge extraction, associated research in the area, and our problem in the context of existing work.

Mining and clustering unstructured logs: Log-messages are unstructured textual data without any annotation for the underlying fault. A significant amount of research has focused on converting unstructured logs to common *semantic events* [5]. Note that the notion of *semantic events* is different from the actual real-world events responsible for generating the messages, nevertheless, such a conversion helps in providing a canonical description of the log-messages that enables subsequent correlation analysis. These works exploit the structural similarity among different messages to either compute an intelligent log-parser or cluster the messages based on message texts [8, 9, 10, 5]. Each cluster can be viewed as a semantic event which can help in diagnosing the underlying root-cause. One work closely related to ours is [11], in which the authors mine network log messages to first extract templates and then learn pairwise *implication* rules between template pairs. Our setting and objective are somewhat different in that we model events as message distributions from different elements with each event occurrence having certain start and end times; the messages belonging to an event and the associated occurrence time windows are hidden (to be learned). A more recent work [12] develops algorithms to mine an underlying structural event as a workflow graph. The main differences are that, each transaction is a fixed sequence of messages unlike our setting where each message could be generated multiple times based on some hidden stochastic process, and furthermore, in our setting, there could be multiple events manifested in the centralized log-server.

Mining temporal patterns: Log-messages are time series data and thus the temporal patterns contain useful information. Considerable research has gone into learning latent patterns, trends and relationship between events based on timing in-

formation in the messages [13, 14, 15]. We refer to [16, 5, 17] for a survey of these approaches. Extracted event patterns could be used to construct event correlation graphs that could be mined using techniques such as graph clustering. Specifically, these approaches are useful when event streams are available as time series. We are interested in scenarios where each event is manifested in terms of time series of unstructured messages and furthermore, same message could arise from multiple events. Nevertheless, certain techniques developed for temporal event mining could be adapted to our setting as we describe in Section 2.4.1.2; our results indicate that such an adaptation works well under certain conditions. Note that our goal is to also learn the event-occurrence times.

Event-summarization: In large dynamic systems, messages could be generated from multiple components due to reasons ranging from software bugs, system faults, operational activities, security alerts etc. Thus it is very useful to have a global summarized snapshot of messages based on logs. Most works in this area exploit the inter-arrival distribution and co-occurrence of events [18, 19, 20, 21, 5] to produce summarized correlation between events. These methods are useful when the event stream is available and possible event types are known in advance. This limits the applicability to large systems like ours where event types are unknown along with their generation time window.

The body of work closest to our work is research on event summarization. However, there are some fundamental differences in our system: (i) we do not have a readily available event stream, instead, our observables are log-messages, (ii) the event types are latent variables not known in advance and all we observe are message streams, (iii) the time boundaries of different latent events are based on a learning objective, and (iv) since we are dealing with a large system with multiple components where different fault types are correlated, the same message could be generated for different root causes (real-world events).

Apart from the above, a recent paper [22] which uses deep learning models for anomaly detection in message logs by modeling logs as a natural language sequence is also worth a mention.

2.2 Problem Statement and Preliminaries

Problem statement: We are given a dataset $\mathcal{D}$ consisting of messages generated by error events in a large distributed data-center network. We assume that the messages are generated in the time interval $[0, T]$. The set of messages in the dataset come from discrete and finite set $\mathcal{M}$.

We use the term message to mean either a template extracted from a message or an alarm-id. Each message has a timestamp associated with it, which indicates when the message was generated. Suppose that an event e started occurring at time S_e and finished at time F_e. In the interval of time $[S_e, F_e]$, event e will generate a mixture of messages from a subset of $\mathcal{M}$, which we will denote by $\mathcal{M}_e$. In general, an event can occur multiple times in the dataset. If an event e occurs multiple times in the dataset, then each occurrence of the event will have start and finish times associated with it.

As noted before, for simplicity, we will say event to mean an event-class and occurrence of an event to mean an instance from the class. An event e is characterized by its message set $\mathcal{M}_e$ and the probability distribution with which messages are chosen from $\mathcal{M}_e$, which we will denote by $p^{(e)}$, i.e., $p_m^{(e)}$ denotes the probability that event e will generate a message $m \in \mathcal{M}_e$. For compactness of notation, we can simply define $p^{(e)}$ over the entire set of messages $\mathcal{M}$, with $p_m^{(e)} = 0$ if $m \notin \mathcal{M}_e$. Thus, $p^{(e)}$ fully characterizes the event e and can be viewed as the signature of the event. We assume that the support sets of messages for two different events are not identical.

It is important to note that the dataset simply consists of messages from the set $\mathcal{M}$; there is no explicit information about the events in the dataset, i.e., the event information is latent. The goal of this chapter is to solve the following inference problem: from the given dataset $\mathcal{D}$, identify the set of events that generated the messages in the dataset, and for each instance of an event, identify when it started and finished. In other words, the output of the inference algorithm should contain the following information:

- The number of E events which generated the dataset.
- The signatures of these events: $p^{(1)}, p^{(2)}, \ldots, p^{(E)}$.

- For each event $e \in \{1, 2, \ldots, E\}$, the number of times it occurred in the dataset and, for each occurrence, its start and finish times.

Notations: We use the notation $X_i \in \mathcal{M}$, for the i^{th} message. Also, let t_i be the timestamp associated with the i^{th} message. Thus the dataset $\mathcal{D}$ can be characterized by tuples $(X_1, t_1), (X_2, t_2), \ldots (X_n, t_n)$ of n data points.

Before we describe our machine-learning pipeline, we first explain the notion of messages in the context of our work.

Messages: In our work, messages generated by different network elements are one of two types: *syslog texts* in the form of raw-texts, and *alarms*.

1. *Syslog texts:* These are raw-textual messages sent by software components from different elements to a logging server. Raw syslog data fields include timestamp, source, and message text. Since the number of distinct messages are very large and many of them have common patterns, it is often useful [8, 9, 10, 5] to decompose the message text into two parts: an *invariant* part called template, and *parameters* associated with the template. For example, a syslog message "`service wqffv failed due to connection failure to IP address a.b.c.d using port 8231`" would reduce to template "`service wqffv failed due to connection failure to IP address * using port *`." There are many existing methods to extract such templates [4, 5], ranging from tree-based methods to NLP-based methods. In our work, we have a template-extraction pre-processing step before applying our methods. We also say *message* to simply mean the extracted templates.

2. *Alarms:* Network alarms are indications of faults and each alarm type refers to the specific fault condition in a network element. Each alarm has a unique name and the occurrences are also tagged with timestamps. In this work, we view each alarm as a message. Note that since each alarm has a unique name/id associated with it, we do not pre-process alarms before applying our methods. Examples of alarms are `mmscRunTImeError, mmscEAIFUnavailable` sent from a network service named MMSC.

Machine-learning pipeline: In Figure 2.1, we show the machine-learning pipeline for completeness. This dissertation focuses on the module "Latent Event Learner"

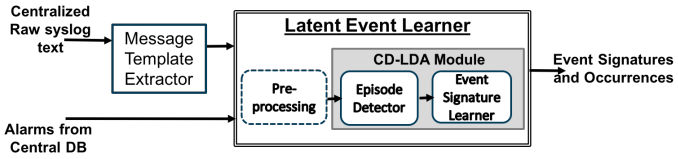


Figure 2.1: Figure showing the machine-learning pipeline. Our main contribution is in "Latent Event Learner" module, specifically proposing the CD-LDA algorithm.

which has a data-processing step followed by the key proposed algorithm in this dissertation, namely the CD-LDA algorithm which we describe in Section 2.3. Syslog texts require more pre-processing while alarms do not. We have shown the two types of messages in Figure 2.1, but for the purposes for developing an algorithm, in the rest of this dissertation, we only refer to messages without distinguishing between them.

2.3 Algorithm CD-LDA

We now present our solution to this problem which we call CD-LDA (Change-point Detection-Latent Dirichlet Allocation). The key novelty in this dissertation is the connection that we identify between event identification in our problem and topic modeling in large document datasets, a problem that has been widely studied in the natural language processing literature. In particular, we process our dataset into a form that allows us to use a widely used algorithm called LDA to solve our problem. In standard LDA, we are given multiple documents, with many words in each document. The goal is to identify the mixture of latent topics that generated the documents, where each topic is identified with a collection of words and a probability distribution over the words. Our dataset has similar features: we have events (which are the equivalents of topics) and messages (which are the equivalents of words) which are generated by the events. However, we do not have a concept of documents. A key idea in our dissertation is to divide the dataset into smaller datasets, each of which will be called an episode. The episodes will be the equivalents of documents in our problem. We do this using a technique called

nonparametric change-point detection.

Now we describe the concept of an episode. An episode is an interval of time over which the same set of events occur i.e., there is no event-churn, and at time instants on either side of the interval, the set of events that occur are different from the set of events in the episode. Thus, we can divide our dataset of events such that no two consecutive episodes have the same set of events. We present an example to clarify the concept of an episode. Suppose the duration of the message dataset $T = 10$. Suppose event one occurred from time 0 to time 5, event two occurred from time 4 to time 8, and event three occurred from time 5 to time 10. Then there are four episodes in this dataset: one in the time interval $[0, 4]$ where only one event occurs, one in the time interval $[4, 5]$ where events one and two occur, one in the time interval $[5, 8]$ where events two and three occur and finally one in $[8, 10]$ where only event three occurs. We assume that between successive episodes, at most one new event starts or one existing event ends.

We use change-point detection to identify episodes. To understand how the change-point detection algorithm works, we first summarize the characteristics of an episode:

- An episode consists of a mixture of events, and each event consists of a mixture of messages.
- Since neighboring episodes consist of different mixtures of events, neighboring episodes also contain different mixtures of messages (due to our assumption that different events do not generate the same set of messages).
- Thus, successive episodes contain different message distributions and therefore, the time instances where these distributions change are the episode boundaries, which we will call *change points*.
- In our dataset, the messages contain timestamps. In general, the inter-arrival time distributions of messages are different in successive episodes, because the episodes represent different mixtures of events. This can be further exploited to improve the identification of change points.

Based on our discussion so far in this section, CD-LDA has two phases as follows:

1. *Change-point detection:* In this phase, we detect the start and end time of each episode. In other words, we identify the time-points where a new event started or an existing event ended. This phase is described in detail in Section 2.3.1.

2. *Applying LDA:* In this phase, we show that, once episodes are known, LDA based techniques can be used to solve the problem of computing message distribution for each event. Subsequently, we can also infer the occurrence times for each event. This phase along with the complete algorithm is described in Section 2.3.2.

2.3.1 Change-Point Detection

Suppose we have n data points and a known number of change-points k. The data points between two consecutive change points are drawn i.i.d from the same distribution.[3] In the inference problem, each data point could be a possible change point. A naive exhaustive search to find the k best locations would have a computational complexity of $O(n^k)$. Nonparametric approaches to change-point detection aim to solve this problem with much lower complexity even when the number of change points is unknown and there are few assumptions on the family of distributions, [23], [7], [24].

The change-point detection algorithm we use is hierarchical in nature. This is inspired by the work in [7]. Nevertheless our algorithm has certain key differences as discussed in Section 2.3.3.1. It is easier to understand the algorithm in the setting of only one change point, i.e., two episodes. Suppose that τ is a candidate change point among the n points. The idea is to measure the change in distribution between the points to the left and right of τ. We use the TV distance between the empirical distributions estimated from the points to the left and right of the candidate change-point τ. In our context the TV distance between two probability mass functions p

[3]The i.i.d. assumption is not always true in practice as messages could be sparser in time in the beginning of an event. Indeed, the algorithms developed in this work does not rely on the i.i.d. assumption, however, the assumption allows us to prove useful theoretical guarantees

and q is given by one half the $L1$ distance $0.5||p-q||_1$. This is maximized over all values of τ to estimate the location of the change point. If the distributions are sufficiently different in the two episodes the TV distance between the empirical distributions is expected to be highest for the correct location of the change point in comparison to any other candidate point τ (we rigorously prove this in the proof Theorem 1, 2).

Further, we also have different inter-arrival times for messages in different episodes. Hence we use a combination of TV distance and mean inter-arrival time as the metric to differentiate the two distributions.[4] We denote this metric by $\widehat{D}(l)$.

$$\widehat{D}(l) = \|\widehat{p}_L(l) - \widehat{p}_R(l)\|_1 + |\widehat{\mathbb{E}}S_L(l) - \widehat{\mathbb{E}}S_R(l)| \tag{2.1}$$

where $\widehat{p}_L(l)$, $\widehat{p}_R(l)$ are empirical estimates of message distributions to the left and right of l and $\widehat{\mathbb{E}}S_L(l)$, $\widehat{\mathbb{E}}S_R(l)$ are empirical estimates of the mean inter-arrival time to the left and right of l, respectively. The empirical distributions $\widehat{p}_L(l)$, $\widehat{p}_R(l)$ have M components. For each $m \in \mathcal{M}$, we can write

$$\widehat{p}_{L,m}(l) = \frac{\sum_{i=1}^{l-1} \mathbb{1}\{X_i = m\}}{l} \tag{2.2}$$

$$\widehat{p}_{R,m}(l) = \frac{\sum_{i=l}^{n} \mathbb{1}\{X_i = m\}}{n-l} \tag{2.3}$$

The mean inter-arrival time $\widehat{\mathbb{E}}S_L(l)$ and $\widehat{\mathbb{E}}S_L(l)$ are defined as

$$\widehat{\mathbb{E}}S_L(l) = \frac{\sum_{i=1}^{l-1} \Delta t_i}{l} \tag{2.4}$$

$$\widehat{\mathbb{E}}S_R(l) = \frac{\sum_{i=l}^{n} \Delta t_i}{n-l} \tag{2.5}$$

We sometimes write $\widehat{D}(l)$ as $\widehat{D}(\widetilde{\gamma}n)$, where the argument $l = \widetilde{\gamma}n$. Symbol $\widetilde{\gamma}$ denotes the index l as a fraction of n and it can take n discrete values between zero to one.

[4] One can potentially use a weighted combination of the TV distance and mean inter-arrival time as a metric with the weight being a hyperparameter. While the unweighted metric performs well in out real-life datasets, it is an interesting future direction of research to understand how to optimally choose a weighted combination in general.

The indicator function $\mathbb{1}\{A\}$ takes value one only when event A occurs and zero otherwise.

Algorithm 1 describes the algorithm in the one change-point case. To make the algorithm more robust, we declare a change point only when the episode length is at least αn and the maximum value of the metric (2.1) is at least δ.

Let us consider a simple example to illustrate the idea of change-point detection with one change point. Suppose we have a sequence of messages with unequal inter-arrival times as shown in Fig. 2.2. All the messages are the same, but the first half of the messages arrive at a rate higher than the second half of the messages. In this scenario, our metric reduces to the difference in the mean inter-arrival times between the two episodes. So, $\widehat{D}(l) = |\widehat{\mathbb{E}}S_L(l) - \widehat{\mathbb{E}}S_R(l)|$. The function $\widehat{D}$ in terms of data point l for this example is shown in Fig. 2.2. As we show later in Section 2.3.3, the shape of $\widehat{D}$ will be close to the following when the number of samples is large: $\widehat{D}$ will be increasing to the left of change-point $\tau = \gamma n$, attain its maximum at the change point and decrease to the right.

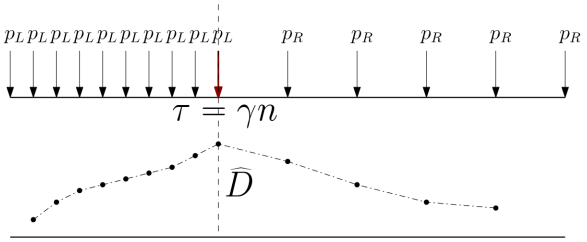


Figure 2.2: Example change point with two episodes.

Algorithm 1 Change-point detection with one change point

1: **Input**: parameter $\delta > 0, \alpha > 0$.
2: **Output**: *changept* denoting whether a change point exists and the location of the change-point τ.
3: Find $\tau \in \arg\max_l \widehat{D}(l)$
4: **if** $\widehat{D}(\tau) > \delta$ and $\alpha n < \tau < 1 - \alpha n$ **then**
5: **return** $changept = 1, \tau$.
6: **else**
7: **return** $changept = 0$.

Next, we consider the case of multiple change points. When we have multiple

change points, we apply Algorithm 1 hierarchically until we cannot find a change point. Algorithm 2 CD$(\mathcal{D}, \alpha, \delta)$ is presented below.

Algorithm 2 CD$(\mathcal{D}, \alpha, \delta)$

Input: data points $\mathcal{D}$, minimum value of TV distance δ, minimum episode length α.
Output: Change-points $\tau_1, \ldots, \tau_k$.
Run FINDCHANGEPT$(1, n)$.
procedure FINDCHANGEPT(L, H)
    changept, $\tau \leftarrow$ ALGORITHM 1 $(X_L, X_{L+1}, \ldots, X_H, \alpha, \delta)$.
    if *changept exists* **then**
        $\tau_l \leftarrow$ FINDCHANGEPT(L, τ),
        $\tau_h \leftarrow$ FINDCHANGEPT(τ, H).
        return $\{\tau_l, \tau, \tau_h\}$
    else
        return

Algorithm 2 tries to detect a single change point first, and if such a change point is found, it divides the dataset into two parts, one consisting of messages to the left of the change point and the other consisting of messages to the right of the change point. The single change-point detection algorithm is now applied to each of the two smaller datasets. This is repeated recursively till no more change points are detected.

2.3.1.1 Discussion: What metric to use for change-point detection in categorical data?

We have used the TV distance between two distributions to estimate the change point in metric (2.1). One can also use other distance measures like the ℓ_2 distance, the Jensen-Shannon (J-S) distance, the Hellinger distance, or the metric used in [7]. The metric used in [7] is shown to be an unbiased estimator of the ℓ_2 distance for categorical data in Appendix A. We argue that for our dataset, all of the above distances give similar performance. Our dataset has 97 m points and 39330 types of messages. In the region where the number of data points is much more than the dimension of the distribution, estimating a change point through all of the above

Table 2.1: Comparison between different metrics for change point.

$\|\|p-q\|\|_1 = 0.1$					
Metric	TV	ℓ_2	Unbiased ℓ_2, [7]	J-S	Hellinger
$\|\widehat{\tau}/n - 0.5\|$	0.021	0.030	0.025	0.030	0.030

metrics gives order wise similar error rate. We show this through synthetic data experiments since we do not know the ground truth to compute the error in estimating the change point in the real dataset.

We present one such experiment with a synthetic dataset here. Consider two distributions p and q whose support set consists of ten points. We assume that p is the uniform distribution, while $q[1] = q[2] = \ldots = q[5] = 0.09$, and $q[6] = q[7] = \ldots = q[10] = 0.11$. There are $n = 25000$ data points. The first half of the data points are independently drawn from p and the second half of the data points are drawn from q. Table 2.1 shows the absolute error in estimating the change point at $0.5n$ to be of the order of 10^{-2} for all the distance metrics.

We test the l_1 distance metric on real data and we show in Section 2.4.2 that it is satisfactory. Since we do not know the ground truth, we take a small part of the real dataset where we can visually identify the approximate location of the major change points. The change-point algorithm with ℓ_1 metric correctly estimates these locations.

A graph-based change-point detection algorithm in [25] can be adapted to our problem such that the metric computation is linear in the number of messages. We can do this if we consider a graph with nodes as the messages and edges connecting message of the same type. But, one can show that the metric in [25] is not consistent for this adaptation.

2.3.2 Latent Dirichlet Allocation

In the problem considered in this dissertation, each episode can be thought of as a document and each message can be thought of as a word. Like in the LDA model where each topic is latent, in our problem, each event is latent and can be thought

of as a distribution over messages. Unlike LDA-based document modeling, we have timestamps associated with messages, which we have already used to extract episodes from our dataset. Additionally, this temporal information can also be used in a Bayesian inference formulation to extract events and their signatures. However, to make the algorithm simple and computationally tractable, as in the original LDA model, we assume that there is no temporal ordering to the episodes or messages within the episodes. Our experiments suggest that this choice is reasonable and leads to very good practical results. However, one can potentially use the temporal information too as in [26, 27], and this is left for future work.

If we apply the LDA algorithm to our episodes, the output will be the event signatures $p^{(e)}$ and episode signatures $\theta^{(\mathcal{E})}$, where an episode signature is a probability distribution of the events in the episode. In other words, LDA assumes that each message in an episode is generated by first picking an event within an episode from the episode signature and then picking a message from the event based on the event signature.

For our event mining problem, we are interested in event signatures and finding the start and finish times of each occurrence of an event. Therefore, the final step (which we describe next) is to extract the start and finish times from the episode signatures.

Putting it all together: In order to detect all the episodes in which the event e occurs prominently, we proceed as follows. We collect all episodes $\mathcal{E}$ for which the event occurrence probability $\theta_e^{(\mathcal{E})}$ is greater than a certain threshold $\eta > 0$. We declare the start and finish times of the collected episodes as the start and finish times of the various occurrences of the event e. If an event spans many contiguous episodes, then the start time of the first episode and the end time of the last contiguous episode can be used as the start and finish time of this occurrence of the event. However, for simplicity, this straightforward step in not presented in the detailed description of the algorithm in Algorithm 3.

Remark 1. *There are many inference techniques for the LDA model, [28, 29, 30, 31, 32, 33]. We use the Gibbs sampling based inference from [28] on the LDA model. For a discussion on the comparison between the above methods, see Appendix A.*

Remark 2. CD-LDA algorithm works without knowledge of topology graph of

Algorithm 3 CD-LDA($\mathcal{D}, \alpha, \delta, \eta$)

1: **Input**: Data points $\mathcal{D}$, threshold of occurrence of an event in an episode η, the minimum value of TV distance δ, minimum episode length α.
2: **Output**: Event signatures $p^{(1)}, p^{(2)}, \ldots, p^{(E)}$, Start and finish time S_e, F_e for each event e.
3: Change-points $\tau_1, \ldots, \tau_k \leftarrow$ CD($\mathcal{D}, \alpha, \delta$). Episode $\mathcal{E}_i \leftarrow \{X_{\tau_{i-1}}, \ldots, X_{\tau_i}\}$ for $i = 1$ to $k + 1$.
4: $p^{(1)}, \ldots, p^{(E)}; \theta^{(\mathcal{E}_1)}, \ldots, \theta^{(\mathcal{E}_{k+1})} \leftarrow$LDA($\mathcal{E}_1, \ldots, \mathcal{E}_{k+1}$)
5: Consider event e. $\mathcal{G}_e \leftarrow$ Set of all episodes $\mathcal{E}$ such that $\theta_e^{(\mathcal{E})} > \eta$. $S_e, F_e \leftarrow$ start and finish times of all episodes in set $\mathcal{G}_e$.

message-generating elements. If the topology graph is known, then the algorithm can be improved as follows. We can run a change-detection phase separately for messages restricted to each element and its graph neighbors (either single-hop or two-hop neighbors). The union of change points could be used in the subsequent LDA phase. Since impact of an event is usually restricted to few hops within the topology, such an approach detects change points better by eliminating several messages far from event source.

Note that the LDA algorithm requires an input for the number of events E. However, one can run LDA for different values of E and choose the one with maximum likelihood [29]. Hence E need not be assumed to be an input to CD-LDA. One can also use the Hierarchical Dirichlet Process (HDP) algorithm [34] which is an extension of LDA and figure out the number of topics from the data. In our experiments, we use the maximum likelihood approach to estimate the number of events.

2.3.3 Results for Change Detection (CD)

As mentioned earlier, the novelty in the CD-LDA algorithm lies in the connection we make to topic modeling in document analysis. In this context, our key contribution is an efficient algorithm to divide the dataset of messages into episodes (documents). Once this is done, the application of the LDA of episodes (documents), consisting of messages (words) generated by events (topics) is standard. Therefore, the correctness and efficiency of the CD part of the algorithm will determine the

correctness and efficiency of CD-LDA as a whole. We focus on analyzing the CD part of the algorithm in this section. We only present the main results here, and the proofs can be found in the Appendix A.

Section 2.3.3.1 shows that the computational complexity of CD algorithm is linear in the number of data points. Section 2.3.3.2 contains the asymptotic analysis of the CD algorithm while Section 2.3.3.3 has the finite sample results.

2.3.3.1 Computational complexity of CD

In this section we discuss the computational complexities of Algorithm 1 and Algorithm 2. We will first discuss the computational complexity of detecting a change point in case of one change point. Algorithm 1 requires us to compute $\arg\max_l \widehat{D}(l)$ for $1 \leq l \leq n$. From the definition of $\widehat{D}(l)$ in (2.1), we only need to compute the empirical probability estimates $\widehat{p}_L(l)$, $\widehat{p}_R(l)$, and the empirical mean of the inter arrival time $\widehat{\mathbb{E}}S_L(l)$, $\widehat{\mathbb{E}}S_R(l)$ for every value of l between 1 to n.

We focus on the computation of $\widehat{p}_L(l)$, $\widehat{p}_R(l)$. Consider any message m in the distribution. For each m, we can compute $\widehat{p}_{L,m}(l)$, $\widehat{p}_{R,m}(l)$ in $O(n)$ for every value of l by using neighboring values of $\widehat{p}_{L,m}(l-1)$, $\widehat{p}_{R,m}(l-1)$.

$$\begin{aligned}\widehat{p}_{L,m}(l) &= \frac{(l-1)\widehat{p}_{L,m}(l-1) + \mathbb{1}\{X_{l-1} = m\}}{l} \\ \widehat{p}_{R,m}(l) &= \frac{(n-l+1)\widehat{p}_{R,m}(l-1) - \mathbb{1}\{X_{l-1} = m\}}{n-l}\end{aligned} \tag{2.6}$$

The computation of $\widehat{\mathbb{E}}S_L(l)$, $\widehat{\mathbb{E}}S_R(l)$ for every value of l from 1 to n is similar.

Performing the above computations for all M messages, results in a computational complexity of $O(nM)$. In the case of k change points, it is straightforward to see that we require $O(nMk)$ computations. In much of our discussion, we assume M and k are constants and therefore, we present the computational complexity results in terms of n only.

Related work: Algorithm 2 executes the process of determining change points hierarchically. This idea was inspired by the work in [7]. However, the metric $\widehat{D}$ we use to detect change points is different from that of [7]. The change in metric

necessitates a new analysis of the consistency of the CD algorithm which we present in the next subsection. Further, for our metric, we are also able to derive sample complexity results which are presented in Section 2.3.3.3.

2.3.3.2 The consistency of change-point detection

In this section we discuss the consistency of the change-point detection algorithm, i.e., when the number of data points n goes to infinity one can accurately detect the location of the change points. In both this subsection and the next, we assume that the inter-arrival times of messages within each episode are i.i.d., and are independent (with possibly different distributions) across episodes.

Theorem 1. *For $\widetilde{\gamma} \in (0,1)$, $D(\widetilde{\gamma}) = \lim_{n\to\infty} \widehat{D}(\widetilde{\gamma}n)$ is well-defined and $D(\widetilde{\gamma})$ attains its maximum at one of the change points if there is at least one change point.*

Remark 3. *The proof of Theorem 1 for the single change-point case is relatively easy, but the proof in the case of multiple change points is rather involved. So we only provide a proof of the single change-point case and refer the interested reader to Appendix A for the proof of the multiple change-point case.*

Proof. **Proof for single change-point case:** We first discuss the single change-point case. Let the change point be at index τ. The location of the change point is determined by the point where $\widehat{D}(l)$ maximizes over $1 < l < n$. We will show that when n is large the argument where $\widehat{D}(l)$ maximizes converges to the change-point τ.

Suppose all the points X to the left of the change-point τ are chosen i.i.d from a distribution F and all the points from the right of τ are chosen from a distribution G, where $F \neq G$. Also, say the inter-arrival times Δt_i's are chosen i.i.d from distribution F_t and G_t to the left and right of change-point τ, respectively. Let $l = \widetilde{\gamma}n$, $0 < \widetilde{\gamma} < 1$ be the index of any data point and $\tau = \gamma n$, the index of the change point.

Case 1 $\widetilde{\gamma} \leq \gamma$: Suppose we consider the value of $\widehat{D}(l) = \widehat{D}(\gamma n)$ to the left of the actual change point, i.e., $l < \tau$ or $\widetilde{\gamma} < \gamma$. The distribution to the left of $\widetilde{\gamma}n$, $\widehat{p}_L(\widetilde{\gamma}n)$,

has all the data points chosen from the distribution F. So $\widehat{p}_L(\widetilde{\gamma}n)$ is the empirical estimate for F. On the other hand, the data points to the right of $\widetilde{\gamma}n$ come from a mixture of distribution F and G. $\widehat{p}_R(\widetilde{\gamma}n)$ has $\frac{\gamma-\widetilde{\gamma}}{1-\widetilde{\gamma}}$ fraction of samples from F and $\frac{1-\gamma}{1-\widetilde{\gamma}}$ fraction of samples from G. Figure 2.3 explains it pictorially.

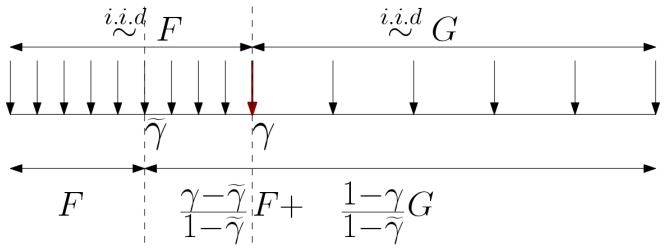


Figure 2.3: Consistency with two change points.

So $\widehat{p}_L(l)$ and $\widehat{p}_R(l)$ defined in (2.3) converges to

$$\widehat{p}_L(l) \to F, \quad \widehat{p}_R(l) \to \frac{\gamma-\widetilde{\gamma}}{1-\widetilde{\gamma}}F + \frac{1-\gamma}{1-\widetilde{\gamma}}G \tag{2.7}$$

Similarly, we can say that the empirical mean estimates $\widehat{E}S_L(l)$ and $\widehat{E}S_R(l)$ converge to

$$\widehat{E}S_L(l) \to \mathbb{E}F_t, \quad \widehat{\mathbb{E}}S_R(l) \to \frac{\gamma-\widetilde{\gamma}}{1-\widetilde{\gamma}}\mathbb{E}F_t + \frac{1-\gamma}{1-\widetilde{\gamma}}\mathbb{E}G_t \tag{2.8}$$

We can combine (2.7) and (2.8) to say that $\widehat{D}(\widetilde{\gamma}n) \to D(\widetilde{\gamma})$ where

$$\begin{aligned}\widehat{D}(\widetilde{\gamma}n) &= \|\widehat{p}_L(\widetilde{\gamma}n) - \widehat{p}_R(\widetilde{\gamma}n)\| + |\mathbb{E}S_L(\widetilde{\gamma}n) - \mathbb{E}S_R(\widetilde{\gamma}n)| \\ &\to D(\widetilde{\gamma}) := \frac{1-\gamma}{1-\widetilde{\gamma}}(\|F-G\|_1 + |\mathbb{E}F_t - \mathbb{E}G_t|)\end{aligned} \tag{2.9}$$

Note that from the definition of D, $D(\gamma) = \|F-G\|_1 + |\mathbb{E}F_t - \mathbb{E}G_t|$.
Case 2 $\widetilde{\gamma} > \gamma$: Proceeding in a similar way to Case 1, we can show

$$\widehat{D}(\widetilde{\gamma}n) \to D(\widetilde{\gamma}) := \frac{\gamma}{\widetilde{\gamma}}(\|F-G\|_1 + |\mathbb{E}F_t - \mathbb{E}G_t|) \tag{2.10}$$

From Case 1 and Case 2, we have

$$\begin{aligned} \widetilde{\gamma} \leq \gamma, \qquad & \widehat{D}(\widetilde{\gamma} n) \to D(\widetilde{\gamma}) = \frac{1-\gamma}{1-\widetilde{\gamma}} D(\gamma) \\ \widetilde{\gamma} > \gamma, \qquad & \widehat{D}(\widetilde{\gamma} n) \to D(\widetilde{\gamma}) = \frac{\gamma}{\widetilde{\gamma}} D(\gamma) \end{aligned} \tag{2.11}$$

Equation (2.11) shows that the maximum of $D(\widetilde{\gamma})$ is obtained at $\widetilde{\gamma} = \gamma$.

□

2.3.3.3 The sample complexity of change-point detection

In the previous subsection, we studied the CD algorithm in the limit as $n \to \infty$. In this section, we analyze the algorithm when there are only a finite number of samples. For this purpose, we assume that the inter-arrival distribution of messages have sub-Gaussian tails.

We say that Algorithm CD is correct if the following conditions are satisfied. Let $\epsilon > 0$ be a desired accuracy in estimation of the change point.

Definition 1. *Given $\epsilon > 0$, Algorithm CD is correct if*

- *there are change-points $0 < \frac{\tau_1}{n} = \gamma_1, \ldots, \frac{\tau_k}{n} = \gamma_k < 1$ and the algorithm gives $\widehat{\gamma}_1, \ldots, \widehat{\gamma}_k$ such that $\max_i |\widehat{\gamma}_i - \gamma_i| < \epsilon$.*
- *there is no change point and $\widehat{D}(\gamma n) < \delta, \forall \gamma \in \{\gamma_1, \ldots, \gamma_k\}$.*

Now we can state the correctness theorem for Algorithm 2. The sample complexity is shown to scale logarithmically with the number of change points.

Theorem 2. *Algorithm 2 is correct in the sense of Definition 1 with probability $(1-\beta)$ if*

$$n = \Omega\left(\max\left(\frac{\log\left(\frac{2k+1}{\beta}\right)}{\epsilon^2}, \frac{M^{1+c}}{\epsilon^{2(1+c)}}\right)\right)$$

for sufficiently small α, δ, ϵ and for any $c > 0$.

Remark 4. *The proof of this theorem uses the method of types and Pinsker's inequality. We present the proof for the single change-point case for the sake of clarity. We move the proof for multiple change points to Appendix A.4.*

Proof. We first characterize the single change-point case in finite sample setting. In order to get the sample complexity, we prove the correctness for Algorithm 1 as per Definition 1 with high probability. Before we go into the proof, we state the assumptions on α, δ, ϵ under which the proof is valid.

- Suppose a change point exists at index γn and the metric $\widehat{D}(\gamma n)$ converges to $D(\gamma)$ at the change point. Then ϵ can only be chosen in following region: ϵ has to be less than the value of the metric at the change point, $\epsilon < D(\gamma)$; ϵ has to be less than the minimum episode length, $\epsilon < \min(\gamma, 1-\gamma)$.
- If a change point exists at index γn, α has to chosen less than the minimum episode length minus ϵ, $\alpha < \min(\gamma, 1-\gamma) - \epsilon$.
- The threshold $\delta < D(\gamma) - \epsilon$.

Under the above assumptions we show that Algorithm 1 is correct as per the Definition 1 with probability at least

$$1 - (6n+4)\exp\left(-\frac{\min(\delta,1)^2\epsilon^2\alpha^2}{512\max(\sigma^2,1)}n + M\log(n)\right)$$

Suppose

$$\widehat{\gamma}n = \arg\max_{\widetilde{\gamma}n} \widehat{D}(\widetilde{\gamma}n)$$

The idea is to upper bound the probability when Algorithm 1 is not correct. From Definition 1 this happens when,

- Given a change point exists at $\gamma \in (0,1)$,

  $$(\widehat{D}(\widehat{\gamma}n) > \delta, |\gamma - \widehat{\gamma}| < \epsilon, \alpha < \widehat{\gamma} < 1-\alpha)^c$$

  occurs. Say the event E_1 denotes $E_1 = \{\widehat{D}(\widehat{\gamma}) > \delta, |\gamma - \widehat{\gamma}| < \epsilon, \alpha < \widehat{\gamma} < 1-\alpha\}$.

- Given a change point does not exist,

$$\widehat{D}(\widehat{\gamma}) > \delta, \alpha < \widehat{\gamma} < 1 - \alpha$$

When a change point does not exist we write $\gamma = 0$. Say the event E_2 denotes $E_2 = \{\gamma = 0, \alpha < \widehat{\gamma} < 1 - \alpha\}$.

So

$$\begin{aligned} &P(\text{Algorithm 1 is NOT correct}) \\ &\leq P(E_1^c | 0 < \gamma < 1) + P(\widehat{D}(\widehat{\gamma}) > \delta | E_2) \end{aligned} \tag{2.12}$$

We analyze each part in (2.12) separately.

Case 1: Suppose no change point exists and say all the data points are drawn from the same multinomial distribution F and all inter-arrival times are generated i.i.d from a distribution F_t. Given event E_2, if $\|\widehat{p}_L(\widehat{\gamma}n) - F\|$, $\|\widehat{p}_R(\widehat{\gamma}n) - F\|$, $|\widehat{\mathbb{E}}S_L(\widehat{\gamma}n) - \mathbb{E}F_t|$ and $|\widehat{\mathbb{E}}S_R(\widehat{\gamma}n) - \mathbb{E}F_t|$ are all less than $\delta/4$, then $\widehat{D}(\widehat{\gamma}) < \delta$. So $P(\widehat{D}(\widehat{\gamma}) > \delta | E_2) \leq P(\|\widehat{p}_L(\widehat{\gamma}n) - F\| > \delta/4 | E_2) + P(\|\widehat{p}_R(\widehat{\gamma}n) - F\| > \delta/4 | E_2) + P(|\widehat{\mathbb{E}}S_L(\widehat{\gamma}n) - \mathbb{E}F_t| > \delta/4 | E_2) + P(|\widehat{\mathbb{E}}S_R(\widehat{\gamma}n) - \mathbb{E}F_t| > \delta/4 | E_2)$. Now, we can use Sanov's theorem followed by Pinsker's inequality to upper bound each of the above terms as

$$\begin{aligned} &P(\widehat{D}(\widehat{\gamma}) > \delta | E_2) \leq (n\widehat{\gamma} + 1)^M \exp(-n\delta^2/16) \\ &+ ((1 - \widehat{\gamma})n + 1)^M \exp(-n\delta^2/16) + 2\exp(-\alpha n\delta^2/32\sigma^2) \\ &+ 2\exp(-\alpha n\delta^2/32\sigma^2) \\ &\leq 4(n+2)^M \exp\left(-n\frac{\alpha\delta^2}{32\max(\sigma^2, 1)}\right) \end{aligned} \tag{2.13}$$

Case 2: Next, we look at the case when a change point exists at γn. Say the messages are drawn from a distribution F to the left of the change point and G to the right of the change point. Also, suppose the inter-arrival time distribution to the left of the change point is F_t and the inter-arrival time distribution to the right is G_t. According to our assumptions, α is chosen such that $\alpha + \epsilon < \gamma < 1 - (\alpha + \epsilon)$.

Hence

$$
\begin{aligned}
&P(E_1^c|0<\gamma<1) \leq P(\widehat{D}(\widehat{\gamma}n)<\delta|0<\gamma<1) \\
&+P(|\widehat{\gamma}-\gamma|>\epsilon|\widehat{D}(\gamma)>\delta, 0<\gamma<1) \\
&+P(\alpha<\widehat{\gamma}<1-\alpha|\widehat{D}(\gamma)>\delta, |\widehat{\gamma}-\gamma|<\epsilon, 0<\gamma<1) \qquad (2.14)
\end{aligned}
$$

Given the assumption on α, $P(\alpha<\widehat{\gamma}<1-\alpha|\widehat{D}(\gamma)>\delta, |\widehat{\gamma}-\gamma|<\epsilon, 0<\gamma<1)=0$. The rest of the proof deals with upper bounding $P(\widehat{D}(\widehat{\gamma}n)<\delta|0<\gamma<1)$ and $P(|\widehat{\gamma}-\gamma|>\epsilon|\widehat{D}(\gamma)>\delta, 0<\gamma<1)$.

In Lemmas 1-3 we develop the characteristics of $\widehat{\gamma}$ and $D(\widehat{\gamma})$ when a change point exists at γn. Lemmas 1-3 are proved in Appendix A.5 and Appendix A.6. First, we analyze the concentration of $\widehat{D}(\widetilde{\gamma}n)$ for any value of $\widetilde{\gamma}$ in the Lemma 1.

Lemma 1. *The difference* $|\widehat{D}(\widetilde{\gamma}n) - D(\widetilde{\gamma})| \leq \epsilon$ *w.p. at least* $1 - 3n\exp\left(-\frac{\epsilon^2\alpha^2}{128\sigma^2}n + M\log(n)\right)$ *for all values of* $\widetilde{\gamma}$ *when* $\widehat{D}(\widetilde{\gamma}n)$ *is defined.*

Lemma 1 shows that the empirical estimate $\widehat{D}(\widetilde{\gamma}n)$ is very close to the asymptotic value $D(\widetilde{\gamma})$ with high probability. Recall that the argument at which $\widehat{D}$ maximizes is $\widehat{\gamma}n$. we next show in Lemma 3 that the value of metric D at $\widehat{\gamma}$ is very close to the value of the D at the change-point γ.

Lemma 2. *The difference* $|D(\gamma) - D(\widehat{\gamma})| < 2\epsilon$ *w.p.* $1 - 3n\exp\left(-\frac{\epsilon^2\alpha^2}{128\sigma^2}n + M\log(n)\right)$.

Finally, in Lemma 3 we show that $\widehat{\gamma}$ is close to the change-point γ with high probability.

Lemma 3. *The absolute error* $|\widehat{\gamma} - \gamma| < \epsilon$ *w.p.* $1 - 3n\exp\left(-\frac{\epsilon^2 D^2(\gamma)\alpha^2}{512\sigma^2}n + M\log(n)\right)$.

Also, using Lemma 2 and assuming that δ is chosen such that $\delta < D(\gamma) - \epsilon$,

$$\begin{aligned}
&P(\widehat{D}(\widehat{\gamma}n) < \delta | 0 < \gamma < 1) \\
&\leq P(\widehat{D}(\widehat{\gamma}n) < \delta | 0 < \gamma < 1, |\widehat{D}(\widehat{\gamma}n) - D(\gamma)| < \epsilon) \\
&+ P(|\widehat{D}(\widehat{\gamma}n) - D(\gamma)| > \epsilon) \\
&\leq 0 + 3n \exp\left(-\frac{\epsilon^2\alpha^2}{128\sigma^2}n + M\log(n)\right) \qquad (2.15)
\end{aligned}$$

Lemma 3 gives a bound on $P(|\widehat{\gamma} - \gamma| > \epsilon | \widehat{D}(\gamma) > \delta, 0 < \gamma < 1)$. Using this along with (2.15) in (2.14) we have

$$\begin{aligned}
P(E_1^c | 0 < \gamma < 1) &\leq 3n \exp\left(-\frac{\epsilon^2\alpha^2}{128\sigma^2}n + M\log(n)\right) \\
&+ 3n \exp\left(-\frac{\epsilon^2 D^2(\gamma)\alpha^2}{512\sigma^2}n + M\log(n)\right) \qquad (2.16)
\end{aligned}$$

Finally, putting together (2.13) and (2.16) into (2.12), we have

$$\begin{aligned}
&P(\text{Algorithm 1 is NOT correct}) \\
&\leq (6n+4)\exp(-\frac{\min(\delta,1)^2\epsilon^2\alpha^2}{512\max(\sigma^2,1)} + M\log(n+2)) \qquad (2.17)
\end{aligned}$$

Ignoring the constants in (2.17), we can derive the sample complexity result for the one change-point case. □

2.4 Experiments

We now present our experimental results with real datasets from a large operational network. The purpose of experiments is threefold. First, we wish to compare our proposed CD-LDA algorithm with other techniques proposed (adapted to our setting) in the literature. Second, we want to validate our results against manual expert-derived event signature for a prominent event. Third, we want to understand the scalability of our method with respect to very large datasets.

Datasets used: We use two datasets: one from a legacy network of physical elements like routers, switches etc., and another from a recently deployed virtual network function (VNF). The VNF dataset is used to validate our algorithm by comparing with expert knowledge. The other one is used to show that our algorithm is scalable, i.e., it can handle large datasets and it is less sensitive to the hyperparameters.

- **Dataset-1:** This dataset consists of around 97 million raw syslog messages collected from 3500 distinct physical network elements (mostly routers) from a nationwide operational network over a 15-day period in 2017. There are 39330 types of messages.
- **Dataset-2:** The second dataset consists of around $728,000$ messages collected from 285 distinct physical/virtual network elements over three months from a newly deployed *virtual network function* (VNF) which is implemented on a data-center using multiple VMs.

We implemented the machine-learning pipeline as shown in Figure 2.1. The main algorithmic component in the figure shows the CD-LDA algorithm; however, for the purpose of comparison, we also implemented two additional algorithms described shortly. Before the data is applied to any of the algorithms, there are two steps, namely, template-extraction (in case of textual syslog data) and pre-processing (for both syslog and alarms). These steps are described in Appendix A.9.

2.4.1 Benchmark Algorithms

We compare CD-LDA with the following algorithms.

2.4.1.1 Algorithm B: A Bayesian inference based algorithm

We consider a fully Bayesian inference algorithm to solve the problem. A Bayesian inference algorithm requires some assumptions on the statistical generative model by which the messages are generated. Our model here is inspired by

topic modeling across documents generated over multiple eras [26]. Suppose that there are E events which generated our dataset, and event e has a signature $p^{(e)}$ as mentioned earlier. The generative model for generating each message is assumed to be as follows:

- To generate a message, we first assume that an event $e \in [1, 2, \ldots, E]$ is chosen with probability P_e.
- Next, a message m is chosen with probability $p_m^{(e)}$.
- Finally, a timestamp is associated with the message which is chosen according to a beta distribution $\beta(a_e, b_e)$, where the parameters of the beta distribution are distinct for different events.

The parameters of the generative model $P_e, p_m^{(e)}, a_e, b_e$ are unknown. As in standard in such models, we assume a prior on some of these parameters. Here, as in [26], we assume that there is a prior distribution on q over the space of all possible P and a prior r over the space of all possible $p^{(e)}$. The prior r is assumed to be independent of e. Given these priors, the Bayesian inference problem becomes a maximum likelihood estimation problem, i.e.,

$$\max_{a_e, b_e, {p^{(e)}}_e, P} \mathbb{P}_{q,r}(\mathcal{D}|P, \{p^{(e)}\}_e)$$

We use Gibbs sampling to solve the above maximization problem. There are two key differences between Algorithm B and proposed CD-LDA. CD-LDA first breaks up the datasets into smaller episodes whereas Algorithm-B uses prior distributions (the beta distributions) to model that different events happen at different times. We show that, such an algorithm works, but the inference procedure is dramatically slow due to additional parameters to infer $\{a_e, b_e\}_e$.

2.4.1.2 Algorithm C: A graph-clustering-based algorithm

For the purposes of comparison, we will also consider a very simple graph-based clustering-based algorithm to identify events. This algorithm is inspired from

graph-based clustering used in event log data in [35]. The basic idea behind the algorithm is as follows: we construct a graph whose nodes are the messages in the set $\mathcal{M}$. We divide the continuous-time interval $[0, T]$ into T/w timeslots, where each timeslot is of duration w. For simplicity, we will assume that T is divisible by w. We draw an edge between a pair of nodes (messages) and label the edge by a distance metric between the messages, which roughly indicates the likelihood with which two messages are generated by the same event. Then, any standard distance-based clustering algorithm on the graphs will cluster the messages into clusters, and one can interpret each cluster as an event. Clearly, the algorithm has the following major limitation: it can detect $\mathcal{M}_e$ for an event e and not $p^{(e)}$. In some applications, this may be sufficient. Therefore, we consider this simple algorithm as a possible candidate algorithm for our real dataset.

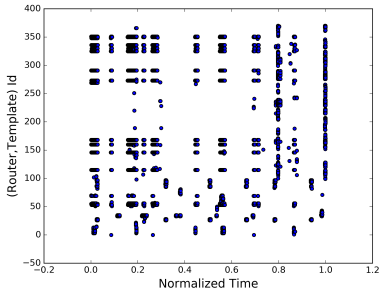
We now describe how the similarity metric is computed for two nodes i and j. Let N_i be the number of timeslots during which a message i occurs and let N_{ij} be the number of timeslots during which both i and j appear in the same timeslot. Then, the distance metric between nodes i and j is defined as

$$\rho_{ij} = 1 - \frac{N_{ij}}{N_i + N_j - N_{ij}}$$

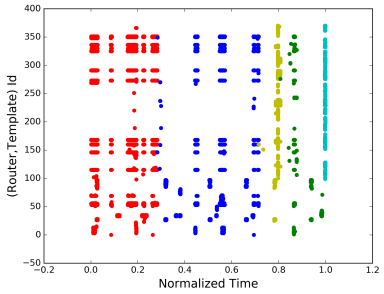
Thus, a smaller ρ_{ij} indicates that i and j co-occur frequently. The idea behind choosing this metric is as follows: messages generated by the same event are likely to occur closer together in time. Since ρ_{ij} is small it indicates that the messages are more likely to have been generated by the same event, and thus are closer together in distance.

2.4.2 Results: Comparison with Benchmark Algorithms

For the purposes of this section only, we consider a smaller slice of data from Dataset-1. Instead of considering all the 97 million messages, we take a small slice of 10,000 messages over a three hour duration from 135 distinct routers. Let us call this dataset $\mathcal{D}_s$. There are two reasons for considering this smaller slice. Firstly, it is easier to visually observe the ground truth in this small dataset and verify visually



(a)



(b)

Figure 2.4: Top panel shows scatter plot of different message-ids over the period of comparison and bottom panel shows the episodes detected by the CD phase of Algorithm CD-LDA.

if CD-LDA is giving us the ground truth. We can also compare the results from different methods with this smaller dataset. Secondly, as we show later in this section, the Bayesian inference Algorithm B is dramatically slow and so running it over the full dataset is not feasible. Nevertheless, the smaller dataset allows us to validate the key premise behind our main algorithm, i.e., the decomposition of the algorithm into the CD and LDA parts.

Applying CD-LDA on this dataset slice: Figure 2.4 (a) shows the data points in the x-axis and the message-ids on the y-axis. Figure 2.4 (b) shows the five episodes after the CD part of CD-LDA, where we chose $\alpha = 0.1$ and $\delta = 0.5$. For the LDA part, instead of specifying the number of events, we use maximum likelihood to find the optimal number of events and based on this, the number of events was found to be two.

We next compare event signatures produced by CD-LDA with Algorithm B and Algorithm C.

CD-LDA versus Algorithm B: For all unknown distributions, we assume a uniform prior in Algorithm B. Algorithm B is run with an input number of events from two to five. It turns out that, with three events the algorithm converges to a solution which has maximum likelihood. However, upon clustering the event signatures $p^{(e)}$ based on TV-distance between the event signatures, we find only two events. *The maximum TV-distance between the events signatures found from the two algorithms is* 0.068. Hence, we can conclude that the event signatures found by both the algorithms are very similar.

Despite Algorithm B using fewer hyperparameters, it is not fast enough to run on large datasets. Figure 2.5 shows the time taken by CD-LDA and Algorithm B as we increase the size of the dataset from $10,000$ to $40,000$ points. With $40,000$ data points and 12 events as input Algorithm B takes three hours whereas CD-LDA only takes 26.57 seconds. Clearly, we cannot practically run Algorithm B on large datasets with millions of points.

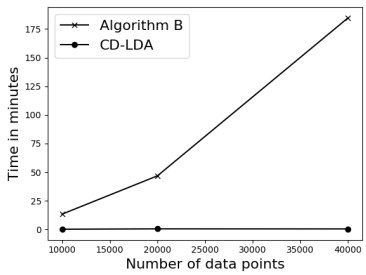


Figure 2.5: Time performance: CD-LDA vs Algorithm B.

CD-LDA versus Algorithm C: In this section we compare CD-LDA versus algorithm C on dataset $\mathcal{D}_s$. Algorithm C can produce the major event clusters as CD-LDA, but does not provide the start and end times for the events. We form the co-occurrence graph for Algorithm C with edge weight as described in Section 2.4.1.2 and nodes as messages which occur more than at least five times in the dataset $\mathcal{D}_s$. All the edges with weight more than 0.6 are discarded and we run a clique detection algorithm in the resulting graph.

Table 2.2: Events generated by CD-LDA and the constituent messages in decreasing order of probability. Event 8 matches with expert provided event signature.

Event one	Event two	...	Event eight
mmscRuntimeError	ISCSI_multipath		SNMP_sshd
SUDBConnectionDown	Logmon_contrail		SNMP_crond
SocketConnectionDown	VRouter-Vrouter		SNMP_AgentCheck
SUDBConnectionUp	LogFile_nova		SNMP_ntpd
SocketConnectionUp	SUDBConnectionDown		SNMP_CPU
mmscEAIFUnavailable	IPMI		SNMP_Swap
bigipServiceUp	bigipServiceDown		SNMP_Mem
bigipServiceDown	bigipServiceUp		SNMP_Filespace
SNMP_Mem	HW_IPMI		Ping_vm

We quantitatively compare the event signature $\mathcal{M}_e$ of the top two cliques found by Algorithm C with those found by CD-LDA. Suppose that message sets identified by Algorithm C for the two events are $\mathcal{M}_{e1}$ and $\mathcal{M}_{e2}$ respectively. Message sets (messages with probability more than 0.007) identified by CD-LDA for the two events are denoted by $\mathcal{S}_{e1}$ and $\mathcal{S}_{e2}$. We can now compute the Jaccard Index between the two sets.

$$\frac{|\mathcal{M}_{e1} \cap \mathcal{S}_{e1}|}{|\mathcal{M}_{e1} \cup \mathcal{S}_{e1}|} = 0.73 \quad \frac{|\mathcal{M}_{e2} \cap \mathcal{S}_{e2}|}{|\mathcal{M}_{e2} \cup \mathcal{S}_{e2}|} = 0.68$$

Since the full Bayesian inference (Algorithm B) agrees with CD-LDA closely, we can conclude that Algorithm C gets a large fraction of the messages associated with the event correctly. However, it also misses a significant fraction of the messages, and additionally Algorithm C does not provide any information about start and end times of the events. Also, the events found are sensitive to the threshold for choosing the graph edges, something we have carefully chosen for this small dataset.

2.4.3 Results: Comparison with Expert Knowledge and Scalability

Validation by comparing with manual event signature: The intended use-case of our methodology is for learning events where the scale of data and system does not allow for manual identification of event signatures. However, we still wanted to validate our output against a handful of event signatures inferred man-

ually by domain experts. For the purpose of this section, we ran CD-LDA for Dataset-2 which is for an operational VNF. For this dataset, an expert had identified that a known service issue had occurred on two dates: 11-Oct and 26-Nov, 2017. This event generated messages with ids `Ping_vm`, `SNMP_AgentCheck`, `SNMP_ntpd`, `SNMP_sshd`, `SNMP_crond`, `SNMP_Swap`, `SNMP_CPU`, `SNMP_Mem`, `SNMP_Filespace`.

We ran CD-LDA on this dataset with parameters $\alpha = 0.01$ and $\delta = 0.1$. We chose ten events for the LDA phase by looking at the likelihood computed using cross validation for different number of topics. See Section 2.4.3.1 for details of the maximium likelihood approach. Table 2.2 shows the events detected by CD-LDA in decreasing order of probability. Also, top nine messages are listed for each event. Indeed, we note that *event eight resembles the expert provided event. CD-LDA detected this event as having occurred from 2017-10-08 17:35 to 2017-10-17 15:55 and 2017-11-25 13:45 to 2017-11-26 03:10.* The longer than usual detection window for 11-Oct is because there were other events occurring simultaneously in the network and the event eight contributed to a small fraction of messages generated during this time window. Finally, as shown in Table 2.2, our method also discovered several event signatures not previously known.

Scalability and sensitivity: To understand the scalability of CD-LDA with data size, we ran it on Dataset-1 with about 97 million data points. CD-LDA was run with the following input: $\alpha = 0.01$, $\delta = 0.1$, and the number of events equal to 20. The CD part of the algorithm detects 57 change points. The sensitivity of this output with respect to α and δ is discussed next. The event signatures are quite robust to this parameter choice, but as expected, the accuracy of the start and finish time estimates of the events will be poorer for large values of α and δ. Overall, CD-LDA takes about six hours to run, which is quite reasonable for a dataset of this size. Reducing the running time by using other methods for implementing LDA, such as variational inference, is a topic for future work.

Parameter α specifies the minimum duration of episode that can be detected in the change detection. By increasing δ we can control to detect the more sharp change points (change points across which the change in distribution is large), and decreasing δ helps us detect the soft change points as well. So α and δ control the

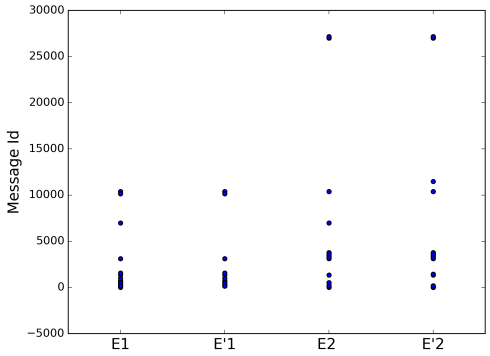


Figure 2.6: Comparison of event signatures for first two events with α_1, δ_1(E1,E2) vs α_2, δ_2(E'1,E'2).

granularity of the change-point detection algorithm. Parameter η is a user defined parameter to detect the episodes in which a particular event occurs. We demonstrate the sensitivity of CD-LDA to α and δ. We run CD-LDA with $\alpha_2 = 10\%$ and $\delta_2 = 0.5$ on Dataset-2 and compare it with results when run with parameters $\alpha_1 = 1\%$ and $\delta_1 = 0.1$. Tables 2.3 and 2.4 show the first two events for parameters α_1 and δ_1 when compared to the first two events for parameter α_2 and δ_2. CD-LDA detects 57 change points with α_1 and δ_1 whereas it only detects 19 change points with α_2 and δ_2. Despite this, Figure 2.6 and Table 2.5 show that the event signatures for the first two events are almost the same. But, since the episodes are larger in duration with α_2 and δ_2, the start and end times of the first two events are less accurate than α_1 and δ_1. In particular, event two is shown to occur from 2-10 05:00 to 2-14 00:00 with α_2 and δ_2 in Table 2.3 whereas it is broken into two episodes, 2-10 05:00 to 2-10 13:33 and 2-10 15:27 to 2-14 00:00, with α_1 and δ_1 in Table 2.4.

2.4.3.1 Selection of the number of topics in LDA

For Dataset-1, we do tenfold cross validation. We group the 58 documents found by change detection into ten sets randomly. We compute the likelihood on one group with a model trained using documents in the remaining nine groups. We plot the average likelihood in Figure 2.7 vs the number of topics. There is a decrease in

Table 2.3: Results of CD-LDA on Dataset-2 with $\alpha_2 = 10\%$ and $\delta_2 = 0.5$.

Event one	Event two
2017-02-14 00:00 to 2017-02-15 23:59	2017-02-06 19:29 to 2017-02-07 16:42
	2017-02-08 00:00 to 2017-02-08 06:25
	2017-02-08 23:59 to 2017-02-10 04:07
	2017-02-10 05:00 to 2017-02-14 00:00

Table 2.4: Results of CD-LDA on Dataset-2 with $\alpha_1 = 1\%$ and $\delta_1 = 0.1$.

Event one	Event two
2017-02-14 00:00 to 2017-02-15 23:59	2017-02-05 06:21 to 2017-02-07 16:42
	2017-02-08 00:00 to 2017-02-10 00:00
	2017-02-10 03:07 to 2017-02-10 04:07
	2017-02-10 05:00 to 2017-02-10 13:33
	2017-02-10 15:27 to 2017-02-14 00:00

Table 2.5: Comparing results of CD-LDA for different values of α and δ.

$\alpha_1 = 1\%, \delta_1 = 0.1$ vs $\alpha_2 = 10\%, \delta_2 = 0.5$			
$\frac{\lvert\mathcal{M}_1 \Delta \mathcal{M}_1'\rvert}{\lvert\mathcal{M}_1 \cup \mathcal{M}_1'\rvert}$	$\frac{\lvert\mathcal{M}_2 \Delta \mathcal{M}_2'\rvert}{\lvert\mathcal{M}_2 \cup \mathcal{M}_2'\rvert}$	TV dist in $p^{(1)}$	TV dist in $p^{(2)}$
0.046	0.077	0.036	0.08

likelihood around 20 and hence, we choose the number of topics as 20.

For Dataset-2, we do tenfold cross validation and choose the number of topics as ten from the Figure 2.8. In this case, we create the ten groups of documents in the following way. Out of 58 documents, group one has document number $1, 11, 21 \ldots$, group two has documents $2, 22, 32, \ldots$, etc. Subsampling in this fashion respects the ordering in the documents.

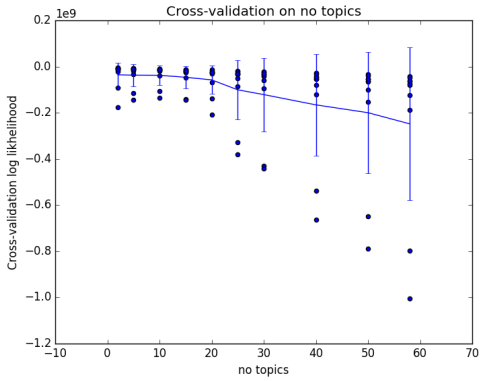


Figure 2.7: Likelihood vs number of topics in Dataset-1.

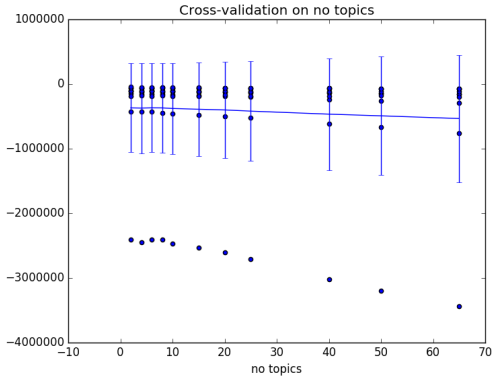


Figure 2.8: Likelihood vs number of topics in Dataset-2.

CHAPTER 3

THE ROLE OF REGULARIZATION IN OVERPARAMETERIZED NEURAL NETWORKS

3.1 Introduction

Neural networks have proved to be remarkably successful in achieving outstanding performance in many classification and regression tasks. Such networks are trained with a large amount of training data to yield good performance on test data. There are many design choices that lead to good test error performance (also called generalization performance), including the network architecture, the number of model parameters, use of appropriate regularizers or weight decay to control the values of the network weights, and the choice of good training algorithms (such as stochastic gradient descent). One of the surprising aspects of neural network research is the empirical finding that overparameterized networks (i.e., networks in which the number of parameters is larger than the number of training points used) are easy to train and have very good test error performance. This observation appears to be counterintuitive since one would expect overparameterization to lead to overfitting the training data and poor generalization performance; however, recent theoretical work in [36, 37, 38, 39, 40] supports the claim that overparameterization can lead to nice loss function landscapes over which it is easy for gradient descent to find a good minimum. In this dissertation, we add to this growing literature by studying the impact of regularization on the performance of a neural network. Our motivation is twofold:

- We show that, when appropriately initialized gradient descent is performed on the loss function of an overparameterized network, ℓ_2 regularization provides a knob to control the trade-off between training and test errors. In particu-

lar, instead of only relying on good initialization, we show that the use of a regularizer provides a tighter control over the optimization-generalization trade-off.

- The amount of overparameterization needed for good training and generalization in prior work depends on the data distribution through the minimum eigenvalue of the Neural Tangent Kernel (NTK) matrix [41]. We show that by adding a regularizer, one does not need the NTK matrix to be positive definite and hence the amount of overparameterization only depends on user-defined parameters like number of data points and strength of regularization.

3.1.1 Related Work and Our Contributions

- **Mean-squared loss:** The standard mean-squared regression loss without any regularization is considered in [41, 42]. In both these papers, the generalization performance is obtained as follows:

$$\mathbb{E}_{(x,y)}|y - f^*(x)| \leq \sqrt{\frac{2y^T H^{-1} y}{n}} \;\; w.h.p.$$

Here $f^*(x)$ is the neural network function with weights obtained from minimizing the least squared loss function. An informal statement of Theorem 3.4.6 in this chapter (Section 3.4.2) would show that upon addition of $\lambda\|w-w(0)\|^2$ as a regularizer, the test error $\mathbb{E}_{(x,y)}|y-f^*(x)|$ is upper bounded by,

$$\frac{\lambda\|(H+\lambda)^{-1}y\|}{\sqrt{n}} + \sqrt{\frac{2c_\lambda}{n^\epsilon}}$$

with high probability where $y^T(H+\lambda I)^{-1}H(H+\lambda I)^{-1}y \leq n^{1-\epsilon}c_\lambda$ for some $\epsilon > 0$. In the above bound, the first term reflects the training error and the second term reflects the complexity of the function class (via Rademacher complexity). For small values of λ, the second term dominates the first, and vice versa for large values of λ. Hence, λ provides a handle to leverage the

training vs generalization error trade-off. This can help us tune the training algorithm as desired. As an example, we compute the above test error bound for different values of λ for the first two classes of the MNIST dataset (containing about $12k$ data points). The results are presented in Table 3.1. We can see that there is sweet spot for the choice of $\lambda = 2.4$ and it is a tighter bound than $\lambda = 0$ in [41].

Table 3.1: Test/Train error bound for the first two classes of the MNIST dataset.

λ	0 ([41])	0.001	2.4	50
test error	0.481	0.476	0.226	0.293
train error	0	0.001	0.123	0.236

- **Mean-squared loss with regularization:** The authors in [43] also consider the distance to initialization as the regularization (similar to what we do in (3.2)). But, the authors consider a noisy setting in which the observed label is the true label with sub-Gaussian noise (with variance proxy σ^2) added to it. With noisy labels, the test error performance is shown to be

$$\mathbb{E}_{(x,y)}|y - f^*(x)| \leq O(\lambda + 1)\sqrt{\frac{y^T H^{-1} y}{n}} + \frac{\sigma}{\lambda}$$

with high probability. If we were to set $\sigma = 0$, this suggests that our test error bound is sharper. Further, the analysis in [43] is limited to the linearized approximation of a neural network, whereas we also prove the neural network model to be close to its linearized approximation in our setting. We focus on the traditional statistical learning theory model where the test and training data come from the same distribution. The extension to the case of $\sigma > 0$, where the training data is sampled from a noisy version of the original data distribution is straightforward, and is therefore deferred to Appendix B.12.

- **Logistic loss:** The authors in [44] consider the task of minimizing the logistic loss function. Under assumptions on the joint distribution of labels and data, they show that the test error goes to zero when the width of the network

$O(\text{poly}\log n)$, where n is the number of training points. This is further extended to deep networks in [45, 46]. An interesting problem is to quantify the effect of adding regularization to logistic loss.

- **Deep networks:** In this dissertation, we consider a shallow two-layer network. However, we can extend our results to deep neural networks, using techniques from recent work [47, 48, 49, 50]. We leave this extension for future work.

3.2 Neural Network Model

We consider a single hidden-layer neural network with m neurons of the form

$$f(x) = \frac{1}{\sqrt{m}} \sum_{i=1}^{m} a_i \sigma(w_i^T x) \tag{3.1}$$

where $f(x) \in \Re$ denotes the output of the neural network when the input is x, and σ denotes an activation function or a neuron. It is assumed that x is a d-dimensional vector, w_i are also vectors and a_i are scalars. We note that a bias term is often used in the input to each neuron, i.e., the output of each neuron is written as $\sigma(w_i^T x + b_i)$. But we omit b_i without loss of generality by assuming the last element of the input vector is always one so that b_i can be subsumed in w_i.

We are given n data points $(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)$ in a training dataset and the goal is to choose the neural network parameters to minimize the regularized loss function

$$\ell(w) = \frac{1}{n} \sum_{j=1}^{n} (y_i - f(x_i))^2 + \frac{\lambda}{2} \sum_{i=1}^{m} \|w_i - w_i(0)\|^2 \tag{3.2}$$

where the first term on the right-hand side above is the standard regression loss, the second term is a regularizer added to the loss function to ensure that the weights do not deviate from some value $w_i(0)$ which will be chosen appropriately as will be explained shortly, and $\lambda > 0$ is a regularizer parameter. This regularization has also been previously studied in [43].

As in [41], we will assume that training takes place in the following manner:

- Each a_i is chosen to be a Rademacher random variable (i.e., takes values ± 1 with equal probability) and fixed throughout the training process. This is the reason that the loss function is defined only as a function of $w = (w_1, w_2, \ldots, w_m)^T$ in (3.2).

- Each w_i is independently initialized by taking a random sample from $N(0, \kappa^2 I)$, and then gradient descent is used to update it, i.e.,

$$w_i(k+1) = w_i(k) - \eta \nabla_{w_i} \ell(w), \quad w_i(0) \sim N(0, \kappa^2 I)$$

  where $\eta > 0$ is a step size and κ is appropriately chosen later.

The above dynamics are intended to capture some features of practical neural network training: (i) gradient descent is usually performed from a randomly initialized point [51, 52] and (ii) some heuristic either in the form of regularization as above or by using a more general procedure called *weight decay* is used to ensure that weights do not become very large [53]. We note that regularization is typically imposed on w_i and not on $w_i - w_i(0)$ as we have done here, but this form is more convenient for our mathematical analysis.

In the rest of the dissertation, we will assume that the neuron used in our neural network is a rectified linear unit (ReLU), i.e., $\sigma(z) = (z)^+$. It has been shown in [41, 54] that w_i does not change very much from its initial condition and that one can approximate the neural network by a function which is linear in w_i. In particular, the intuition behind the approximation of f is as follows: we linearize f using Taylor's series as

$$\begin{aligned} f(x; w) \approx & \frac{1}{\sqrt{m}} \sum_i a_i \sigma(w_i^T(0)x) \\ & + \frac{1}{\sqrt{m}} \sum_i a_i \sigma'(w_i^T(0)x)(w_i - w_i(0))^T x \end{aligned}$$

While the ReLU activation function is strictly speaking not differentiable, one can use the natural formula $\sigma'(z) = I_{z \geq 0}$ and the relationship $(z)^+ = z I_{z \geq 0} =$

$z\sigma'(z)$ to further simplify the above expression as

$$f(x;w) \approx \frac{1}{\sqrt{m}} \sum_i a_i I_{w_i^T(0)x \geq 0} w_i^T x$$

If we interpret the term $\frac{1}{\sqrt{m}} x I_{w_i^T(0)x \geq 0}, i = 1, 2, \ldots m$ as feature vectors $\phi_i(x)$ associated with the input x, then the above approximation can be rewritten as

$$f(x;w) \approx \sum_{i=1}^{m} a_i w_i^T \phi_i(x)$$

We note that the above approximation to the original neural network is only linear in w_i but is still a nonlinear function of x and has universal approximation power as shown in [55]. Further, the approximations above can be justified precisely as in [41]. The feature vectors $\phi_i(x), i \in [n]$ can be seen as a kernel transformation on the space of x with the kernel,

$$H_{ij} := \mathbb{E}\phi_i^T(x)\phi_j(x) = x_i^T x_j \left(\frac{\pi - \arccos x_i^T x_j}{2\pi} \right)$$

as shown in [56]. For the generalization results to make sense, we make the following assumption on the dataset, which is also implicit in [41].

Assumption 3.2.1. *The joint distribution of labels $y = [y_1 \quad y_2 \ldots y_n]^T$ and data points $x_i, i \in [n]$ satisfies*

$$\frac{y^T(H + \lambda I)^{-1} H (H + \lambda I)^{-1} y}{n^{1-\epsilon}} \leq c_\lambda \; \forall \lambda > 0 \tag{3.3}$$

for some $\epsilon > 0$. Here, c_λ is a function of λ. An upper bound to the LHS in (3.3) *is $\forall \lambda$,*

$$\sum_{i:\lambda_i(H)>0} \frac{1}{\lambda_i(H)} (U_i^T y)^2 \geq y^T(H + \lambda I)^{-1} H (H + \lambda I)^{-1} y \tag{3.4}$$

where $H = U \Lambda U^{-1}$ is the eigen decomposition of H. The matrix $U = [U_1 U_2 \ldots U_n]$

and $\Lambda = diag(\{\lambda_i(H)\}_{i=1}^n)$. *Note that LHS in* (3.4) *simplifies to* $y^T H^{-1} y$ *when* H *is invertible, which is same as the expression of generalization in [41]. Hence, Assumption* (3.3) *is weaker than the condition in [41].*

Assumption (3.3) *can be verified in real datasets. For example, consider the first two classes in the MNIST dataset. One can observe that* $\frac{y^T H^{-1} y}{n}$ *decreases with* n *as shown in Figure 3.1.*

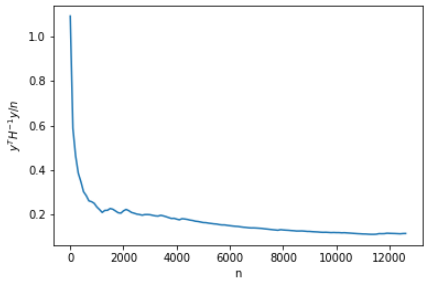


Figure 3.1: Plot of $\frac{y^T H^{-1} y}{n}$ vs n for the first two classes in MNIST.

Assumption 3.2.2. *The labels and data points are bounded, i.e.,* $|y_i| \leq 1$ *and* $\|x_i\| = 1, i \in [n]$. *This can be achieved by simply normalizing the dataset.*

Note that, unlike [41], we do not require that H be positive definite. In order to get an intuition of the main proof, we will first analyze the linear model in Section 3.3. We defer the analysis of the neural network model to Section 3.4.

3.3 Analysis of Linearized Model

For the purposes of this section, we will assume that the linear approximation is accurate and use the following expression for the neural network:

$$f(x; w) = \frac{1}{\sqrt{m}} \sum_{i=1}^{m} a_i w_i^T \phi_i(x) \tag{3.5}$$

Our final modeling choice in this chapter is to approximate the discrete-time

gradient descent algorithm by its continuous-time counterpart, as in [54]:

$$\frac{dw_i}{dt} = -\nabla_{w_i}\ell(w)$$

with $w_i(0) \sim N(0, \kappa^2 I)$ where the variance κ will be initialized appropriately later. By using the expressions for the loss function in (3.2) and the form of the neural network in (3.5), the gradient descent ODE can be rewritten as

$$\frac{dw_i}{dt} = -\sum_{j=1}^{n}(f(x_j; w) - y_j)a_i\phi_i(x_j) + \lambda(w_i - w_i(0))$$

Define ∇f_0 to be the matrix whose $(i,j)^{\text{th}}$ element is $a_i\phi_i(x_j)$, f to be the vector whose j^{th} element is $f(x_j; w)$, and y to be the vector whose j^{th} element is y_j. Then the gradient descent ODE can be compactly written as

$$\frac{dw}{dt} = -\nabla f_0(f - y) - \lambda(w - w(0)) \tag{3.6}$$

In order to analyze the training error, we work with the dynamics of f rather than the dynamics of w. By differentiating (3.5) and using $\nabla f_0^T w(0) = f_0$, where f_0 is a vector whose j^{th} element is $f(x_j; w(0))$, we get

$$\dot{f} = -\nabla f_0^T \nabla f_0(f - y) - \lambda(f - f_0) \tag{3.7}$$

We can now conclude the following theorem on the training loss.

Theorem 3.3.1. *Define*

$$f_\infty = y + \lambda(\nabla f_0^T \nabla f_0 + \lambda I)^{-1}(f_0 - y)$$

Then,

$$f(t) - f_\infty = \exp\left(-(\nabla f_0^T \nabla f_0 + \lambda I)t\right)(f_0 - f_\infty)$$

and

$$\lim_{t\to\infty} \|f(t) - y\| \leq \lambda\|(\nabla f_0^T \nabla f_0 + \lambda I)^{-1}(f_0 - y)\|$$

where $\|\cdot\|$ denotes the standard ℓ_2 norm.

Proof. Since $\lambda > 0$, $(\nabla f_0^T \nabla f_0 + \lambda I)$ is positive definite, and hence (3.7) is a stable system. The result follows by examining the solution of the differential equation, $\dot{f} = -(\nabla f_0^T \nabla f_0 + \lambda I)(f - f_\infty)$. □

Corollary 1. *Since $\mathbb{E} f(x, w(0)) = 0$, f_0 can be shown to be close to zero w.h.p. Further $\nabla f_0^T \nabla f_0$ can be shown to be close to $H = \mathbb{E} \nabla f_0^T \nabla f_0$ and thus,*

$$\lim_{t \to \infty} \|f(t) - y\| \lessapprox \lambda \|(H + \lambda I)^{-1} y\|$$

Additionally we can show that the output of gradient descent $w(\infty)$ remains close to the initialization of weights $w(0)$. This is stated in Lemma 3.3.2 and it will be used to provide generalization bounds in Theorem 3.3.3.

Lemma 3.3.2. *Given $w(0), a$ and data points x_i, y_i, we can bound*

$$\lim_{t \to \infty} \|w(t) - w(0)\| \leq \big\{ (f_0 - y)^T (\nabla f_0^T \nabla f_0 + \lambda I)^{-1} (f_0 - y) \\ -\lambda (f_0 - y)^T (\nabla f_0^T \nabla f_0 + \lambda I)^{-2} (f_0 - y) \big\}^{1/2}$$

Proof. Rearranging (3.6),

$$\dot{w} = -(\nabla f_0 \nabla^T f_0 + \lambda)(w - w_\infty) \tag{3.8}$$

where

$$w_\infty = w(0) + (\nabla f_0 \nabla^T f_0 + \lambda I)^{-1} \nabla f_0 (y - f_0)$$

Since $\lambda > 0$, (3.8) is a stable system, and

$$w(t) - w_\infty = \exp\left(-(\nabla f_0 \nabla^T f_0 + \lambda I)t\right)(w(0) - w_\infty)$$

Upon taking the limit $t \to \infty$ and using the definition of w_∞,

$$w(\infty) - w(0) = (\nabla f_0 \nabla^T f_0 + \lambda I)^{-1} \nabla f_0 (y - f_0)$$

We arrive at the result after using Woodbury matrix identity. □

Corollary 2. *Assuming f_0 to be close to zero and $\nabla f_0^T \nabla f_0$ to be close to H,*

$$\lim_{t\to\infty} \|w(t) - w(0)\| \lessapprox \sqrt{y^T(H+\lambda I)^{-1}y - \lambda y^T(H+\lambda I)^{-2}y}$$

Finally, the generalization bound depends on the Rademacher complexity of the function class containing all possible outputs of the ODE (3.6) (see [57] for details):

$$\mathcal{F}_w = \{w : \|w - w(0)\|^2 \le B; \\ y^T(H+\lambda I)^{-1}y - \lambda y^T(H+\lambda I)^{-2}y \le n^{1-\epsilon}c_\lambda\}$$

Note that $\lim_{t\to\infty} w(t) \in \mathcal{F}_w$. Define the Rademacher complexity of class $\mathcal{F}_w$ as,

$$Rad(\mathcal{F}_w) := \frac{1}{n}\mathbb{E}_\epsilon \sup_{w\in\mathcal{F}_w}\left(\sum_i \epsilon_i f(x_i; w)\right)$$

Subtracting a constant ($f(x_i, w(0))$) inside the supremum does not change the expectation, therefore:

$$Rad(\mathcal{F}_w) := \frac{1}{n}\mathbb{E}_\epsilon \sup_{w\in\mathcal{F}_w}\left(\sum_i \epsilon_i (f(x_i; w) - f(x_i, w(0)))\right)$$

Also, $f(x_i; w) - f(x_i, w(0)) = \nabla^T f_0(x_i)(w - w(0))$. Using Cauchy-Schwartz inequality followed by the bound on $\|w - w(0)\|$ in $\mathcal{F}_w$ and $\|\nabla f_0(x_i)\|^2 \approx \mathbb{E}\|\nabla f_0(x_i)\|^2 = 1/2$,

$$Rad(\mathcal{F}_w) \le \sqrt{\frac{c_\lambda}{2n^\epsilon}}$$

Combining the above result with the training error from Theorem 3.3.1, we get the final result for the linearized model.

Theorem 3.3.3. *With high probability,*

$$\mathbb{E}_{x,y}|y - f(x, w(\infty))| \lessapprox \frac{\lambda\|(H+\lambda I)^{-1}y\|}{n} + 2Rad(\mathcal{F}_w)$$

3.4 Analysis of Neural Network Model

In this section we analyze the non-linear neural network and show that the result is close to its linearized model analyzed in Section 3.3. But the analysis is more involved than the linearized case. We also justify all the approximations made in Section 3.3. For simplicity, we break the analysis into two parts, Section 3.4.1 provides all the probabilistic concentration results, and Section 3.4.2 presents the analysis under satisfaction of all the conditions in Section 3.4.1.

Notation:

- $f(x) = \sum_k \frac{1}{\sqrt{m}} a_k \sigma(w_k^T x)$.
- $\nabla_k f(x) = \frac{\partial}{\partial w_k} f(x) = \frac{a_k}{\sqrt{m}} \sigma'(w_k^T x) x$.
- $\nabla f(x) = [\nabla_1^T f(x) \;\; \nabla_2^T f(x) \dots \nabla_m^T f(x)]^T$.
- $\nabla_k f = [\nabla_k^T f(x_1) \;\; \nabla_k^T f(x_2) \dots \nabla_k^T f(x_n)]$.
- $\nabla f = [\nabla f(x_1) \;\; \nabla f(x_2) \dots \nabla f(x_m)]$.
- R is a constant such that $\max_k \|w_k - w_k(0)\| \leq R$.
- $H = \mathbb{E} \nabla f_0^T \nabla f_0$, $\lambda_{\min}(H) := \lambda_0 \geq 0$.
- $f_0(x) = \sum_k \frac{1}{\sqrt{m}} a_k \sigma(w_k^T(0) x)$.

3.4.1 Probability Conditions

Lemma 3.4.1. *With probability more than $1 - \delta$ over initialization $a, w(0)$ and i.i.d samples $x_i, y_i, i \in [n]$ the following conditions are true.*

1. $|w_k(0)| \leq 2\kappa \sqrt{d \log\left(\frac{20md}{\delta}\right)}$ *for all* $k \in [m]$

2. *For $f(x)$ defined on any weights $w \in S_w := \{w : |w_k - w_k(0)| \leq R \forall k \in [m]\}$,*

$$\sup_{w \in S_w} |\nabla^T f(x_i) w(0)| \leq 4\sqrt{d \log(m) \kappa^2} + \sqrt{\frac{40\kappa^2 n}{\delta}}$$
$$\sup_{w \in S_w} |\nabla^T f w(0)| \leq 4\sqrt{dn \log(m) \kappa^2} + \sqrt{\frac{40\kappa^2 n^2}{\delta}}$$
$$|f_0(x_i)| \leq 4\sqrt{d \log(m) \kappa^2} + \sqrt{\frac{40\kappa^2 n}{\delta}}$$
$$|f_0| \leq 4\sqrt{dn \log(m) \kappa^2} + \sqrt{\frac{40\kappa^2 n^2}{\delta}}$$

3. *Suppose f is computed on weights w such that $\max_k |w_k - w_k(0)| \leq R$. Then,*

$$\sup_w |\nabla f - \nabla f_0|_F^2 \leq \frac{8nR}{\sqrt{2\pi}\kappa} + 4\sqrt{2}n\sqrt{\frac{\log\left(\frac{20}{\delta}\right)}{m}}$$

4. *Suppose f is computed on w such that $\max_k |w_k - w_k(0)| \leq R$, then $\sup_w |\nabla^T f \nabla f - \nabla^T f_0 \nabla f_0| \leq \frac{8n^2 R}{\kappa\sqrt{2\pi}} + 4\sqrt{2}n^2\sqrt{\frac{\log\left(\frac{20}{\delta}\right)}{m}}$*

5. *Let the $d-$dimensional vector $h(a_{-k})$ be defined as,*

$$h(a_{-k}) := \sum_{i \neq k} \frac{a_i}{m} v_i$$

where $a_{-k} = \{a_j, j \neq k\}$ and $v_i \in \mathbb{R}^d$ in any vector that satisfies $|v_i| \leq B, \forall i \neq k$ for some constant $B > 0$. Then for all $k \in [m]$, $|h(a_{-k})| \leq \frac{2B\sqrt{d \log\left(\frac{20md}{\delta}\right)}}{\sqrt{m}}$

6. $|\nabla^T f_0 \nabla f_0 - H| \leq \sqrt{\frac{32n^2}{m} \log\left(\frac{20n}{\delta}\right)}$, $|(\nabla^T f_0 \nabla f_0 + \lambda)^{-1}| \leq \frac{2}{\lambda + \lambda_0}$, *when* $m \geq \frac{128n^2 \log\left(\frac{20n}{\delta}\right)}{(\lambda + \lambda_0)^2}$

7. $\sup_{|x_i| \leq 1} \sum_k \frac{2R}{\sqrt{m}} \mathbb{1}\{|w_k^T(0) x_i| \leq R\} \leq 32R\sqrt{d \log(m+1)} + \frac{2R^2\sqrt{m}}{\kappa\sqrt{2\pi}} + \sqrt{32R^2 \log\left(\frac{10}{\delta}\right)}$

8. $\sup_{|x|\leq 1} f_0(x) \leq \mathbb{E}\sup_{|x|\leq 1} \sum_k \frac{a_k}{\sqrt{m}}\sigma(w_k^T(0)x) + \sqrt{\frac{40d\kappa^2}{\delta}}$

9. *For all* $i \in [n]$, $|\nabla f_0(x_i)|^2 \leq \frac{1}{2} + \sqrt{\frac{1}{m}\log\left(\frac{20n}{\delta}\right)}$

10. *Suppose* $\sup_{|y|\leq 1,|x|\leq 1} |y - f(x)| \leq M$. *Then,*

$$\mathbb{E}_{x,y}|y - f(x)| \leq \frac{1}{n}\sum_i |y_i - f(x_i)| + 2Rad(\mathcal{F}_w) + 3M\sqrt{\frac{\log\left(\frac{20}{\delta}\right)}{2n}}$$

The proof of the Lemma 3.4.1 is application of various concentration inequalities and is deferred to Appendix B.1.

3.4.2 Deterministic Analysis

The results in this section assume the conditions in Lemma 3.4.1 and hence hold with probability $1 - \delta$. We will make the following assumptions on variance κ and the number of neurons m.

Assumption 3.4.1. $\kappa = \frac{\delta}{n^{1+\epsilon}\log(m)}$ *for any* $\epsilon > 0$ *and* $m = poly(n, \frac{1}{\delta}, \frac{1}{\lambda})$.

The amount of overparameterization we need in the assumption above only depends on λ and not the minimum eigenvalue of H unlike [41].

In [56], the authors analyze the trajectory of f, and substitute it into the trajectory of $\dot{w}$ to further show that $|w_i(t) - w_i(0)|$ is bounded for all $t > 0$. Instead, we directly prove that $|w_i(t) - w_i(0)|$ is bounded, by expressing the RHS of the ODE $\dot{w}_k$ in terms of w_k and bounded error terms. The regularizer term $-\lambda(w_i(t) - w_i(0))$ inherently help us make this conclusion since $\dot{w}_k$ can be easily shown to be stable for $\lambda > 0$. For $\lambda = 0$, this approach might not work since the coefficient of the linear term of w_k for ODE $\dot{w}_k$ is $\nabla_k f \nabla_k^T f$ and it is not a positive definite matrix for $m > n$.

We know that gradient descent satisfies the following ODE for the weights w,

$$\dot{w} = -\nabla f(\nabla^T f w - y) - \lambda(w - w(0)) \tag{3.9}$$

Taking $w(\infty) = w(0) + \nabla f_0(\nabla^T f_0 \nabla f_0 + \lambda)^{-1} y$, we can rewrite (3.9) as,

$$\begin{aligned}\dot{w} = &-(\nabla f_0 \nabla^T f_0 + \lambda)(w - w(\infty)) + (\nabla f - \nabla f_0) y \\ &+ (\nabla f_0 \nabla^T f_0 - \nabla f \nabla^T f) w - \nabla f_0 \nabla^T f_0 w(0)\end{aligned} \tag{3.10}$$

Consider each coordinate of $w_k, k \in [m]$. It satisfies the following ODE,

$$\dot{w}_k = -(\nabla_k f_0 \nabla_k^T f_0 + \lambda)(w_k - w_k(\infty)) + err_k \tag{3.11}$$

where err_k can be written as

$$\begin{aligned}err_k := &(\nabla_k f - \nabla_k f_0) y + (\nabla_k f_0 \nabla^T f_0 - \nabla_k f \nabla^T f) w \\ &- \nabla_k f_0 \nabla^T f_0 w(0) - \sum_{i \neq k} \nabla_k f_0 \nabla_i^T f_0 (w_i - w_i(\infty))\end{aligned}$$

We bound $|err_k|$ by $O(\frac{1}{\sqrt{m}})$ in Appendix B.2. This can be used to bound on $\|w_k(t) - w_k(\infty)\|$ in the Lemma 3.4.2. The proof of Lemma 3.4.2 is relegated to Appendix B.3.

Lemma 3.4.2. $|w_k(t) - w_k(\infty)| \leq O(\frac{1}{\sqrt{m}})\ \forall t > 0 \forall k \in [m]$.

Next we look at the proof for training error.

3.4.2.1 Proof for training error

From the gradient descent dynamics (3.9), we can compute the dynamics of $\dot{f}$,

$$\dot{f} = -\nabla^T f \nabla f (f - y) - \lambda(f - \nabla^T f w(0)) \tag{3.12}$$

which can be re-written as,

$$\begin{aligned}\dot{f} &= -(H+\lambda)(f-y+\lambda(H+\lambda)^{-1}y) \\ &\quad + \lambda\nabla^T f w(0) + (H-\nabla^T f\nabla f)(f-y)\end{aligned}$$

Suppose, $f_\infty := y-\lambda(H+\lambda)^{-1}y$ and $err_f := \lambda\nabla^T f w(0)+(H-\nabla^T f\nabla f)(f-y)$ then $\dot{f}$ reduces to

$$\dot{f} = -(H+\lambda)(f-f_\infty) + err_f \tag{3.13}$$

We will show in Appendix B.4 that $|err_f| \leq O(\frac{1}{\sqrt{n}}) + O(\frac{1}{\sqrt{m}})|f-f_\infty|$.

Theorem 3.4.3. *The training error reduces exponentially fast to*

$$\frac{|f(\infty)-y|^2}{n} \leq \frac{\lambda^2 y^T(H+\lambda)^{-2}y}{n} + O\left(\frac{1}{\sqrt{n}}\right)$$

The proof for Theorem 3.4.3 is provided in Appendix B.5.

3.4.2.2 Proof for test error

To get a bound on the test error we need a bound on $|w-w(0)|$.
Consider again the dynamics of w in (3.9). We can rewrite it as,

$$\begin{aligned}\dot{w} &= -(\nabla f\nabla^T f+\lambda)(w-w(\infty)) - \nabla f\nabla^T f w(0) \\ &\quad + (\nabla f\nabla^T f+\lambda)((\nabla f\nabla^T f+\lambda)^{-1}\nabla f \\ &\quad - (\nabla f_0\nabla^T f_0+\lambda)^{-1}\nabla f_0)y\end{aligned}$$

Again, consider the error term,

$$\begin{aligned}err &:= -\nabla f\nabla^T f w(0) \\ &\quad + (\nabla f\nabla^T f+\lambda)((\nabla f\nabla^T f+\lambda)^{-1}\nabla f \\ &\quad - (\nabla f_0\nabla^T f_0+\lambda)^{-1}\nabla f_0)y\end{aligned}$$

So, the dynamics of w can be simplified into

$$\dot{w} = -(\nabla f \nabla^T f + \lambda)(w - w(\infty)) + err \tag{3.14}$$

We will prove that $|err|$ is $o(1)$ in Appendix B.6.

Lemma 3.4.4. *For all time $t > 0$,*

$$|w(t) - w(0)| \leq 2\sqrt{y^T(H+\lambda)^{-1}H(H+\lambda)^{-1}y} + o(1)$$

and at steady state $t = \infty$,

$$|w(\infty) - w(0)| \leq \sqrt{y^T(H+\lambda)^{-1}H(H+\lambda)^{-1}y} + o(1)$$

The proof is presented in Appendix B.7. Next, we get bounds on the Rademacher complexity and the test error. Lemma 3.4.5 is proved in Appendix B.9.

Lemma 3.4.5. *The Rademacher complexity of $f(x_i), i \in [n]$ with samples $x_i, |x_i| \leq 1$ over class, $\mathcal{F}_w = \{w : |w - w(0)| \leq B, |w_k - w_k(0)| \leq R \forall k \in [m]\}$ is given by*

$$Rad(\mathcal{F}_w) := \frac{1}{n}\mathbb{E}_\epsilon \sup_{w \in \mathcal{F}_w} \left(\sum_i \epsilon_i f(x_i) \right) \leq \frac{B}{\sqrt{2n}} + o\left(\frac{1}{\sqrt{n}}\right)$$

Since

$$|w(\infty) - w(0)| \leq \sqrt{y^T(H+\lambda)^{-1}H(H+\lambda)^{-1}y} + o(1)$$

from Lemma 3.4.4, the Rademacher complexity of $w(\infty)$ can be deduced from Assumption (3.3) as, $Rad(\mathcal{F}_{w(\infty)}) \leq \sqrt{\frac{c_\lambda}{2n^{-\epsilon}}} + o\left(\frac{1}{\sqrt{n}}\right)$.

Using Lemma 3.4.5 we can compute the test error for loss function $|y - f(x)|$. Theorem 3.4.6 is proved in Appendix B.10.

Theorem 3.4.6. *The test error* $\mathbb{E}_{x,y}|y - f(x)|$ *is bounded by,*

$$\mathbb{E}_{x,y}|y - f(x)| \leq \frac{\lambda|(H+\lambda)^{-1}y|}{\sqrt{n}} + \sqrt{\frac{2c_\lambda}{n^{-\epsilon}}} + 100\sqrt{\log\left(\frac{20}{\delta}\right)}o(n^{-\epsilon/2})$$

where $y^T(H+\lambda)^{-1}H(H+\lambda)^{-1}y \leq n^{1-\epsilon}c_\lambda$ *from Assumption* (3.3).

3.5 Experiments

Setup: We run experiments on the MNIST dataset only containing the first two classes for digits zero and one. It contains $10k$ training data points and about $2k$ test data points. The labels are converted to $+1, -1$ to consider the problem of regression loss. We use a two-layer neural network of width $10k$. During the training of the two-layer neural network we only run gradient descent on the weights of the first layer with learning rate 0.01.

Generalization: In order to see a significant improvement in test error, we add noise to the dataset as follows: if the label is $+1$, then we subtract an i.i.d sample from $U[0,1]$ and if the label is -1, we add an i.i.d sample from $U[0,1]$. Under this setting, the test loss is shown to vary between $\lambda = 0$ to $\lambda = 10$ in the Fig. 3.2.

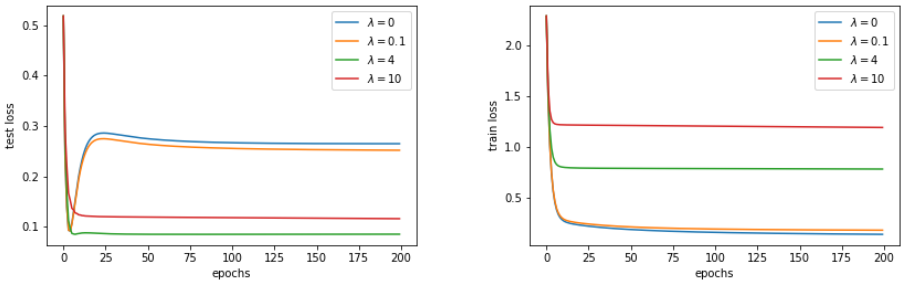


Figure 3.2: Test loss and training loss in MNIST dataset during training epochs.

CHAPTER 4

THE DYNAMICS OF GRADIENT DESCENT FOR OVERPARAMETERIZED NEURAL NETWORKS

4.1 Introduction

Neural networks have shown promise in many supervised learning tasks like image classification [58] and semi-supervised learning like reinforcement learning [59]. A key reason for the success of neural networks is that they can approximate any continuous function with arbitrary accuracy [60, 61]. Further, choosing the neural network parameters by optimizing a loss function over this complex non-convex function class using simple first-order methods, like gradient descent, leads to good generalization for unseen data points. We address the optimization and generalization aspect of neural network training in this chapter. Recently it has been shown that when the network is overparameterized, i.e., the number of neurons in a layer is much larger than the number of training points, one can achieve zero training loss by running gradient descent on the squared loss function [56]. This line of research is motivated by the fact: when one runs gradient descent on an overparameterized network initialized appropriately, the network behaves as though it is linear function of its weights [55, 62]. Although one can achieve zero loss in an overparameterized network, there may be more than one set of network weights which achieve zero loss. In this chapter, we study the dynamics of gradient descent for a single hidden layer, overparameterized neural network, and show that gradient descent converges to a certain minimum norm solution. We present an application of such a characterization of the limit behavior of the network weights by providing an alternative proof of a generalization result in [41].

We are motivated by the results in [56] where it was shown that if the network

width is polynomial in the number of data points, gradient descent converges to the global optimum of the squared loss function. The main idea behind their proof is to show that an overparameterized neural network behaves similarly to a linear function (linear in the weights). The linear approximation can be described as follows: the input is mapped to a high-dimensional feature vector and the linear approximation is an inner product of the network weights and this high-dimensional feature vector. In the limit as the number of neurons goes to infinity, this mapping of the input to a high-dimensional space can also be viewed in terms of a kernel corresponding to a Reproducing Kernel Hilbert Space (RKHS). Such a kernel is known as the Neural Tangent Kernel (NTK) and was introduced in [55]. In this chapter, we prove that the weights of an overparameterized neural network converge close to the point where the weights of the linearized neural network would have converged had we used the linearized network in the optimization process. To the best of our knowledge, the results in earlier papers do not address such convergence properties.

4.1.1 Related Work

- **Convergence of gradient descent for squared loss:** For the squared loss function and ReLU activation function, the convergence of gradient descent was proved in [56]. This is further extended to deep networks in [63, 64], but the dependence of width grows exponentially with the depth of the network; see [62, 54] for analysis in case of general differentiable activation functions. The dependence of the width of the network in terms of the data points has been further improved in [65] by careful choice of Lyapunov functions.

- **Convergence of gradient descent for logistic loss:** Overparameterized neural networks with logistic loss function were shown to be in the linear regime for a finite time in [66]. Under assumptions that the data distribution is linearly separable by the neural tangent kernel, it is shown that gradient descent reaches good test accuracy in this finite time. Unlike the squared loss, optimizing the logistic loss only requires a poly-logarithmic dependence on the number of data points. This is further extended to deep networks in [67].

- **Implicit regularization:** This line of work is closest to our work (see [68] for an overview). It is known that for least squares linear regression, the iterates of gradient descent converge to the minimum norm solution subject to zero loss. This is also true for certain nonlinear models as shown in [69]. We further add to this literature by showing that the iterates of the neural network weights under gradient descent stay close to the minimum norm solution of the linearized model and this distance decreases with the increase in the width of the network. Since gradient descent chooses a particular characterization of weights from all possible solutions achieving zero training error, this phenomenon is often referred to as the "implicit bias" or "implicit regularization". For logistic loss, the weights diverge to infinity, but implicit regularization still occurs. Specifically, with linear classification, the direction along which the weights diverge to infinity matches the direction of the hard margin SVM solution when the data is separable [70, 71]. This result has been extended to some classes of neural networks recently in [72].

- **Generalization results for squared loss:** Shallow neural networks are shown to generalize well for a large class of functions in [41]. We provide an alternative proof of their result for the case of the squared loss function. In [73], the authors consider the generalization properties of kernel ridge regression for NTK under different activation functions, but their results are not directly applicable to finite-width networks as is the case in this work.

4.2 Problem Statement and Contribution

Consider a neural network which takes x as its input and produces an output $f(x, w, b)$ where w, b are certain weight parameters to be chosen. We suppose that $f : \mathbb{R}^d \times \mathbb{R}^{md} \times \mathbb{R}^m \to \mathbb{R}$ is of the form

$$f(x, w, b) := \sum_{k=1}^{m} \frac{a_k}{\sqrt{m}} \sigma(w_k^T x + b_k)$$

where $\sigma(.) = \max(0, .)$ is the Rectified Linear Unit (ReLU) activation function and $x \in \mathbb{R}^d$. This describes a neural network with one hidden layer where the input weights are denoted by w's, input biases are b's and output layer weights are a's. We absorb the biases b's as an extra dimension in the weight vector w's. Likewise, let $\tilde{x} = \{x, 1\}$. So we can compactly write the neural network function as $f : \mathbb{R}^d \times \mathbb{R}^{m(d+1)} \to \mathbb{R}$.

$$f(x, w) := \sum_{k=1}^{m} \frac{a_k}{\sqrt{m}} \sigma(w_k^T \tilde{x})$$

We are given n data points $\{x_i, y_i\}_{i=1}^n$ drawn i.i.d. from a distribution $\mathcal{X} \times \mathcal{Y}$ and we would like to minimize the mean-square error

$$L(w) = \sum_{i=1}^{n} (y_i - f(x_i, w))^2$$

over all w. One of the ways to perform the above minimization is to use the Gradient Descent (GD) algorithm. GD is an iterative algorithm where in each step w is updated in the direction of the negative gradient of $L(w)$. Given appropriate initialization of $w(0)$, and step size η for $k = 1, 2, \ldots$

$$w(k+1) = w(k) - \eta \frac{\partial}{\partial w} L(w)\Big|_{w=w(k)}$$

GD is known to converge to a global optimum if $L(w)$ were a convex function of w for small enough values of η. In our case, even if $L(w)$ is not convex, [56] show that GD initialized appropriately converges to a global optimum, $i.e.$, $L(w(k))$ goes to zero as $k \to \infty$.

We would like to characterize the performance of $w(k)$ given a new data point sampled from the same distribution $\mathcal{X} \times \mathcal{Y}$. In particular, we are interested in the generalization error for $w(k)$ which is defined as

$$E_{x \times y \sim \mathcal{X} \times \mathcal{Y}} (y - f(x, w(k)))^2$$

We assume that the samples of x are chosen from $\mathcal{X}$. Given x, y is conditionally

chosen as $y = f^*(x) + \zeta$, where ζ is drawn from a mean zero distribution independent of $\mathcal{X}$ and f^* is a continuous function to be specified later.

For the noise-less case, i.e., $\zeta = 0$, the generalization error has been characterized in [41] for certain choices of f^*. For the noisy case, to the best of our knowledge the generalization error has not been characterized. But, [73] consider the generalization error for the linearized version of the neural network. We first explain this linearization.

Consider the first-order Taylor approximation of $f(x, w)$ around $w(0)$. Define

$$\begin{aligned} f_L(x, w) &:= f(x, w(0)) + \nabla^\top f(x, w(0))(w - w(0)) \\ &= \frac{1}{\sqrt{m}} \sum_{k=0}^{m} a_k \mathbb{1}\{w_k^\top(0)\tilde{x} \geq 0\} w_k^\top \tilde{x} \end{aligned}$$

It was shown in [56] that w stays close to $w(0)$ during the training iterations of gradient descent when then neural network is overparameterized, $i.e., m > poly(n)$. As a result, $f(x_i, w)$ is close to $f_L(x_i, w)$ for $i \in [n]$ during the GD iterations. This makes us consider the following problem. Instead of minimizing $L(w)$, consider the loss function $L_{lin}(w) = \sum_i (y_i - f_L(x_i, w))^2$. Minimizing $L_{lin}(w)$ using GD is expected to achieve zero loss since $L_{lin}(w)$ is a convex function. Since the network is overparameterized, there are infinitely many values of w achieving zero loss. However, it is known that GD finds the solution which has the lowest ℓ_2 norm. So, minimizing $L_{lin}(w)$ using gradient descent would lead to a solution

$$w_L^* := \arg\min_w \|w - w(0)\| \;\; s.t. \;\; f_L(x_i, w) = y_i, i = \{1, 2, \ldots, n\} \tag{4.1}$$

For appropriately chosen $w(0)$, the prediction function $f_L(x, w_L^*)$ can be shown to be close to $\sum_{i=1}^n c_i K^{(m)}(x, x_i)$, for some constants c_i and the kernel function

$$K^{(m)}(x, x_i) = \frac{1}{m} \sum_{k=1}^{m} x^\top x_i \mathbb{1}\{w_k^\top(0)x \geq 0\} \mathbb{1}\{w_k^\top(0)x_i \geq 0\}$$

In the infinite width limit, $i.e., m \to \infty$, the kernel $K^{(m)}$ converges to a kernel K

which is independent of the initialization $w(0)$ [55]. If $K(x,y) = \phi^\top(x)\phi(y)$, then by Moore-Aronszajn theorem, the RKHS $\mathcal{H}$ induced by K is given by this: A function $g \in \mathcal{H}$ can be defined by $g(.) = \phi^\top(.)w_g$ and inner product between functions f and g in $\mathcal{H}$ is given by $\langle f, g\rangle_{\mathcal{H}} = w_g^\top w_f$. Further, by Representer theorem, the solution to kernel regression in $\mathcal{H}$ would be given by

$$\begin{aligned} f_{KR}(.) &= \min_{g\in\mathcal{H}} \|g\|_{\mathcal{H}} \text{ s.t. } y_i = g(x_i) \\ &= \sum_i^n c_i K(., x_i) \quad \text{s.t.} \quad \sum_{i=1}^n c_i K(x_j, x_i) = y_j \text{ for } j = \{1, 2, \ldots, n\} \end{aligned} \tag{4.2}$$

Hence, in the infinite width limit, the prediction function $f_L(x, w_L^*)$ is close to the solution of kernel regression on the RKHS induced by K. Now we discuss our contributions below.

- The results in [73] analyze the generalization properties of f_{KR} in the presence of noise. They would apply to our work in the following manner.

$$E_{x\sim\mathcal{X},\zeta}(y - f(x, w(k)))^2 \leq 2E_{x\sim\mathcal{X},\zeta}(y - f_{KR}(x))^2 \tag{4.3}$$

$$+ 2E_{x\sim\mathcal{X}}(f(x, w(k)) - f_{KR}(x))^2 \tag{4.4}$$

  If we show that the second term in the RHS of (4.4) is small, then the generalization results in [73] applies to our setting. Our goal is to show that the second term is small and in order to do so we require to show a relation between $w(k)$ and w_L^*. Informally, we show that

$$\|w(k) - w_L^*\| = O\left(\frac{1}{m^{0.125}}\right) \text{ for } m \geq poly(n) \text{ w.h.p for large enough } k \tag{4.5}$$

  It was observed that in [56, 41] that the weights stay in a ball around initialization, i.e., $\|w(k) - w(0)\| = O(poly(n))$. Equation (4.5) is a finer characterization of $w(k)$ than [56].

- In the noiseless case, [41] provided generalization bounds for the prediction

function $f(x, w(k))$. We provide an alternate derivation motivated by the results in [74] in the noiseless case instead of the Rademacher complexity approach in [63]. The results in [74] are derived from the point of view of linear regression. Thus it naturally applies to providing bounds to kernel regression in a RKHS. In the noise-free setting, ($\zeta = 0$) we derive a bound on the first term in (4.4), i.e., $E_{x\sim\mathcal{X}}(y - f_{KR}(x))^2$.

The structure of this chapter is as follows. We establish the result (4.5) for a continuous-time version of gradient descent in Section 4.3. This develops an intuitive understanding of the proof. In Section 4.4 this is extended to the discrete-time GD. In Section 4.5, we use the results developed on the convergence of weights under GD in Sections 4.3-4.4, to show that $E_{x\sim\mathcal{X}}(f(x, w(k)) - f_{KR}(x))^2$ is small. In addition, $E_{x\sim\mathcal{X}}(y - f(x, w(k)))^2$ is shown to be small for the noise-free case.

4.2.1 Notation

Let $w(0)$ denote the point at which gradient descent is initialized. The observation vector is $Y = [y_1 \quad y_2 \dots y_n]^T$ and the neural network function for the inputs $\{x_i\}_{i=1}^n$ is $f = [f(x_1, w) \dots f(x_n, w)]^T$. At initialization $f_0 = [f(x_1, w(0)) \dots f(x_n, w(0))]^T$. Let the ReLU activation function be $\sigma(x) = \max(x, 0)$. The gradient of $f(x, w)$ w.r.t w is $\nabla f(x, w) = [\frac{a_1}{\sqrt{m}}\sigma'(w_1^T\tilde{x})\tilde{x}^T \dots \frac{a_m}{\sqrt{m}}\sigma'(w_m^T\tilde{x})\tilde{x}^T]^T$. Denote the $m(d+1) \times n$ matrix $\nabla f = [\nabla f(x_1, w) \dots \nabla f(x_n, w)]$ and the gradient matrix at initialization $\nabla f_0 = [\nabla f(x_1, w(0)) \dots \nabla f(x_n, w(0))]$. Also $H = \mathbb{E}_{w(0)}\left(\nabla^T f_0 \nabla f_0\right)$. Let the projection matrix onto the column space of ∇f_0 be $P_0 = \nabla f_0(\nabla^T f_0 \nabla f_0)^{-1}\nabla^T f_0$ and the projection matrix onto the null space be $P_0^\perp := I - P_0$. When we write $\|.\|$ it means the ℓ_2 norm if . is a vector or the operator norm if . is a matrix.

4.2.2 Assumptions

- The data points are bounded, i.e., $|y_i| \leq C_y, \|x_i\| = \sqrt{d}, \forall i = 1, \dots, n$.
- No two data points x_i, x_j are parallel to each other, $x_i \nparallel x_j$.

The assumptions stated above are mild and the same as in the literature [56]. The second assumption essentially ensures that the matrix $H = \mathbb{E}_{w(0)}[\nabla^T f_0 \nabla f_0]$ is positive definite [56, Theorem 3.1]. We reprove this in Lemma 4. Note that H does not depend on the width of the network.

Lemma 4. *Define θ_{min} by $\cos\theta_{\min} := \frac{\max_{i\neq j} x_i^T x_j + 1}{d+1}$. Under the above assumptions, the smallest eigenvalue of $H = \mathbb{E}_{w(0)}[\nabla^T f_0 \nabla f_0]$ is strictly positive, $H \succeq cI, c > 0$ and*

$$\Omega\left(\frac{(d+1)\theta_{\min}}{\sqrt{\log(2n)+1}}\right) = c \leq \lambda_{\min}(H) = O((d+1)\theta_{\min})$$

when $\theta_{\min} < 1$. More complicated expressions for the lower and upper bounds which do not require the condition $\theta_{\min} < 1$ can be found in the proof of this lemma in the Appendix C.2.

4.3 Continuous-Time Gradient Descent Algorithm

First we describe the continuous-time gradient descent algorithm below.

- Initialize $w_k(0) \sim \mathcal{N}(0, \kappa^2 I_d)$; a_k's are initialized as 1 with probability $1/2$ and -1 with probability $1/2$.
- Run gradient descent in continuous time, $\dot{w} = \frac{\partial L(w)}{\partial w} := -\nabla f(f - Y)$.

4.3.1 Training Loss and Bound on $\|w_k - w_k(0)\|$

The training loss goes to zero exponentially fast. This is shown in Theorem 3.2, [56].

Lemma 5 ([56])**.** *The continuous-time gradient descent algorithm achieves zero loss with probability greater than $1 - \delta$ (where the randomness is due to the initialization) when $m = \Omega\left(\frac{n^6 d^4 C_y^2}{c^4 \delta^3}\right)$ and $\kappa = 1$. Further, the rate of convergence can be characterized as*

$$\|f(t) - Y\| \leq \exp(-ct/4)\|f_0 - Y\| \tag{4.6}$$

Moreover, the weights w_k's remain in a small ball around the initialization $w_k(0)$ in the following sense:

$$\|w_k(t) - w_k(0)\| = O\left(\frac{\sqrt{dn}}{c\sqrt{m}}\right)\|f_0 - Y\| \tag{4.7}$$

4.3.2 Bound on $\|w - w_L^*\|$

We can show that w_L^* from (4.1) is equal to

$$w_L^* = P_0^{\perp} w(0) + \nabla f_0(\nabla^T f_0 \nabla f_0)^{-1} Y$$

since $\nabla^T f_0 \nabla f_0$ would be a positive definite matrix. We prove that the weights $w(t)$ converge to a point close to w_L^* and the distance decreases when the number of neurons m increases.

Theorem 3. *If the number of neurons $m = \Omega\left(\frac{n^6 d^4 C_y^2}{c^4 \delta^3}\right)$, then under assumptions from Section 4.2.2, with probability greater than $1-\delta$ over initialization and $\kappa = 1$*

$$\|w(t) - w_L^*\| \leq \exp(-c/2t)\|w(0) - w_L^*\| + O\left(\frac{(dn)^{2.5} C_y^{1.5}}{c^{2.5}\delta^{1.5} m^{0.25}}\right)$$

Proof idea: Consider the dynamics of w,

$$\dot{w} = \frac{\partial L(w)}{\partial w} := -\nabla f(f - Y)$$

Since $w(t)$ is close to $w(0)$ from Lemma 5, we can show that $\nabla f \approx \nabla f_0$ throughout the dynamics of w. Hence $\dot{w}$ approximately lies in the column space of ∇f_0. So $P_0^{\perp}\dot{w}$ can be expected to be small. To capture this intuition we choose the Lyapunov functions

$$V_{\perp} := \|P_0^{\perp}(w - w_L^*)\|^2 \text{ and } V_{\|} := \|P_0(w - w_L^*)\|^2 \tag{4.8}$$

The proof of this theorem is deferred to the Appendix C.1.

4.4 Discrete-Time Gradient Descent Algorithm

In this section we present the discrete-time gradient descent algorithm, and show results similar to the continuous-time version.

- Initialize $w_k(0) \sim \mathcal{N}(0, \kappa^2 I_d)$; a_k's are chosen to be 1 with probability $1/2$ and -1 with probability $1/2$. Choose a step size $\eta > 0$.
- Run gradient descent, i.e., for $k = 0, 1, \ldots$, update the weights as

$$w_{k+1} = w_k - \eta \nabla f(f - Y)$$

4.4.1 Bound on $\|w - w_L^*\|$ for Gradient Descent

The analysis of the gradient descent in discrete time is similar in spirit to that in continuous time. We relegate the proof of Theorem 4 to Appendix C.3. Unlike in continuous time, Theorem 4 requires more neurons for the result to hold, $O\left(\frac{1}{m^{1/8}}\right)$ as opposed to $O\left(\frac{1}{m^{1/4}}\right)$.

Theorem 4. *If the number of neurons* $m = \Omega\left(\frac{(dn)^{10} C_y^6}{c^{10}\delta^6}\right), \kappa = 1$ *and* $\eta = O\left(\frac{c}{(dn)^2}\right)$*, then under assumptions from Section 4.2.2, with probability greater than* $1 - \delta$ *over initialization*

$$\|w(k) - w_L^*\| \leq \left(1 - \frac{c\eta}{2}\right)^{k/2} \|w(0) - w_L^*\| + O\left(\frac{(dnC_y)^{1.5}}{c^{1.5}\delta^{1.5} m^{0.125}}\right)$$

for $k = 0, 1, \ldots,$*. Also,* $w(k)$ *converges to some point* w^* *as* $k \to \infty$*.*

4.5 Generalization

In this section we provide generalization results for the output of gradient descent. The analysis depends on the results from Theorem 4 from optimization. Suppose the data points $\{x_i\}_{i=1}^n$ are sampled from a distribution $\mathcal{X}$. In Lemma 6

we show that the prediction function $f(x, w(k))$ at the end of iteration k of gradient descent is close to the minimum norm interpolator for kernel regression from (4.2). We can show that the solution to (4.2) is given by $f_{KR}(x)$ given below.

$$f_{KR}(x) = h^T(x)H^{-1}Y,\ h(x) := [K(x, x_i), i = 1 \text{ to } n] \tag{4.9}$$

$$H_{ij} = K(x_i, x_j) = \tilde{x}_i^T \tilde{x}_j \frac{\pi - \arccos\left(\frac{\tilde{x}_i^T \tilde{x}_j}{d+1}\right)}{2\pi} \tag{4.10}$$

Lemma 6. *Suppose at the end of iteration k, the prediction function is $f(x, w(k))$ for a new data point sampled i.i.d from the distribution $\mathcal{X}$. Then*

$$\mathbb{E}_x(f(x, w(k)) - f_{KR}(x))^2 \leq O\left(\frac{1}{\sqrt{n}}\right) + \left(1 - \frac{c\eta}{2}\right)^k (Y^T H^{-1} Y + O(1)) \text{ w.p. } 1-\delta$$

when $\kappa^2 = O\left(\frac{c\delta}{d^2 n^{1.5}}\right)$, $\eta = O\left(\frac{c}{(dn)^2}\right)$, $m \geq poly(d, n, C_y, 1/c, 1/\delta)$.

Outline of the proof: We will prove this lemma using various concentration inequalities. The first step is to show that $f(x, w(k))$ is close to its linear approximation $f_L(x, w(k)) = \nabla^\top f(x, w(0))w(k)$ around $w(0)$. The second step is to show that $f_L(x, w(k))$ is close to the linear prediction function at w_L^*, $f_L(x, w_L^*)$. The final step is to show that $f_L(x, w_L^*)$ is close to $f_{KR}(x)$ which close to the limit of $f_L(x, w_L^*)$ as $m \to \infty$.

The characterization in Lemma 6 is essential in showing the generalization result in Theorem 5. The kernel $K(x_1, x_2)$ can be expressed as the inner product of infinite-dimensional feature vectors $\phi(x_1)$ and $\phi(x_2)$. Such a feature vector $\phi(.)$ exists because the kernel is positive definite and symmetric [57]. Indeed it is easy to characterize the feature vector for $K(., .)$. It is computed in Lemma 7.

Lemma 7. *Define a feature vector $\phi(x)$ of data point x as*

$$\phi(x) := [\sqrt{d_0}, \sqrt{d_1}x, \sqrt{d_2}x^{\otimes 2}, \sqrt{d_3}x^{\otimes 3}, \ldots]$$

where $x^{\otimes k}$ denotes the k time Kronecker product of vector x with itself and

$$d_0 = 0.25 + \sum_{p\geq 1} \frac{c_{2p}}{(d+1)^{2p}}, \quad d_1 = 0.25 + \sum_{p\geq 1} \frac{2pc_{2p}}{(d+1)^{2p}}$$
$$d_k = \sum_{p\geq \lceil k/2 \rceil} \frac{c_{2p}}{(d+1)^{2p}} \binom{2p}{k} \text{ for } k \geq 2, \quad c_{2p} := \frac{(2p-3)!!(d+1)}{2\pi(2p-2)!!(2p-1)}$$

Then $K(x,y) = \phi(x)^T\phi(y) = \sum_{k\geq 0} d_k(x^\top y)^k = \tilde{x}_i^T\tilde{x}_j \frac{\pi - \arccos\left(\frac{\tilde{x}_i^T\tilde{x}_j}{d+1}\right)}{2\pi}$ for $\tilde{x} = \{x, 1\}$.

In Theorem 5, we show that all functions $y = \phi^T(x)\bar{w}, \|\bar{w}\| < \infty$ which are linear in the feature vector $\phi(x)$ are learnable by a finite width shallow ReLU network. Corollary 1 provides some example functions in this class. This function class can be shown to be learnable from the generalization result in [41]. Their results are presented without using a bias term in each neuron, in which case, the infinite-width NTK RKHS is not a universal approximator. However, it is straightforward to extend their results to allow the addition of a bias term in each neuron, in which case, it is known that the infinite-width NTK RKHS is a universal approximator [75]. In the latter, more general case, their results match the results in this chapter. Our contribution is to provide an alternative proof of the result in [41].

Theorem 5. *Suppose $y = \phi^\top(x)\bar{w}$ and x is sampled from distribution $\mathcal{X}$. Then there exists a constant $C > 0$ such that*

$$E_x(y - f(x, w(k)))^2 \leq C|\bar{w}|^2\sqrt{\frac{d\log(1/\delta)}{n}} \text{ w.p. greater than } 1-\delta$$

over initialization when $\kappa^2 = O\left(\frac{c\delta}{d^2n^{1.5}}\right)$, $\eta = O\left(\frac{c}{(dn)^2}\right), m \geq poly(d, n, C_y, 1/c, 1/\delta)$ and the number of gradient descent iterations $k = \Omega\left(\frac{\log\left(n(\|\bar{w}\|^2+1)\right)}{\eta c}\right)$.

Proof sketch: The argument follows from the proof of the noiseless case in [74]. Denote matrix $\Phi = [\phi(x_1), \phi(x_2), \ldots \phi(x_n)]$. Together with Lemma 6 we can show that $\mathbb{E}_x(y - f(x,w))^2 \approx \mathbb{E}_x(y - f_{KR}(x))^2 = \bar{w}^T P_\phi^\perp \mathbb{E}_x\phi(x)\phi^T(x) P_\phi^\perp \bar{w}$ where

$P_\phi^\perp$ denotes the projection matrix to the null space of matrix Φ. Observe that the sample covariance matrix $n^{-1}\sum_{i=1}^n \phi(x_i)\phi^T(x_i)$ is orthogonal to the column space of Φ. This reduces the problem of bounding $\mathbb{E}_x(y - f_{KR}(x))^2$ to one of bounding the error between the population and sample covariance of $\phi(x)$. The result then follows from use of McDiarmid's inequality.

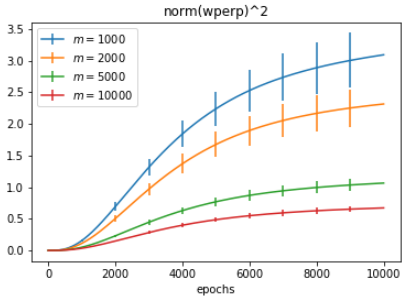
Corollary 1. *If* $y = (x^\top\beta)^p = \phi^\top(x)\bar{w}$ *for* $p = 0, 1, 2, \ldots$ *and* $\|\beta\| \leq 1$*, then*

$$\bar{w} = [0, \ldots, 0, \frac{1}{\sqrt{d_p}}\beta^{\otimes p}, 0, \ldots]$$

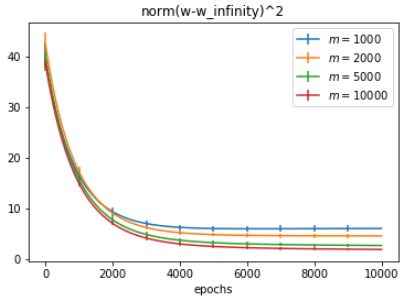
Therefore, by Theorem 5, $\mathbb{E}_x(y - f(x, w(k)))^2 = O\left((d+1)^p(p+1)^{1.5}\sqrt{\frac{\log(1/\delta)}{n}}\right)$ *w.p. more than* $1-\delta$ *for* $p \geq 2$ *and* $\mathbb{E}_x(y - f(x, w(k)))^2 = O\left(\sqrt{\frac{\log(1/\delta)}{n}}\right)$ *w.p. more than* $1-\delta$ *for* $p = 0, 1$ *when* $\kappa^2 = O\left(\frac{c\delta}{d^2 n^{1.5}}\right)$, $\eta = O\left(\frac{c}{(dn)^2}\right), m \geq poly(n, d^{p/2}, 1/c, 1/\delta)$ *and the number of gradient descent iterations* $k = \Omega\left(\frac{\log(n)+p\log(d+1)}{\eta c}\right)$. *Using the Stone–Weierstrass theorem we can show that the set of polynomial functions are dense in the space of continuous functions in the compact input space and hence, the result here can be extended to include all continuous functions with appropriate approximation error.*

4.6 Experiments

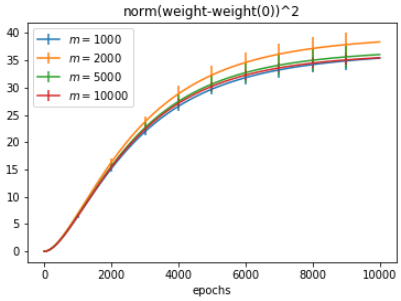
We show an example with a synthetic dataset below. We generate 100 data points from a uniform distribution $[-1, 1]^5$ in $\mathbb{R}^5$ and normalize it to have unit norm. The output is $y = (x^T\beta)^2$, where x is the input and β is chosen from a uniform distribution $[-1, 1]^5$ in $\mathbb{R}^5$. The output points are normalized by subtracting the empirical mean and then dividing by the empirical standard deviation. We fit the data points using a shallow neural network of widths $m = 1000, 2000, 5000, 10000$ and mean-squared loss. We perform full gradient descent with a learning rate of 0.01 and do not train the last layer (i.e., the a_i) to be consistent with the theory presented in this chapter. The experiment is repeated five times with different initialization and the



(a) $\|P_0^{\perp}(w - w_L^*)\|^2$.



(b) $\|w - w_L^*\|^2$.



(c) $\|w - w(0)\|^2$.

Figure 4.1: Gradient descent iterations for different widths of the network.

standard deviation is shown in Figures 4.1 $(a), (b)$, and (c). In Figure 4.1 (b), we plot the different $\|w(t) - w_L^*\|^2$ for different widths across the iterations of gradient descent. Figure 4.1 (a) shows that $\|P_0^{\perp}(w(t) - w_L^*)\|^2$ is upper bounded and the upper bound decreases with increase in width of the network. Figure 4.1 (c) plots distance of the iterates from initialization, $\|w(t) - w(0)\|^2$. We can see that the there is no clear trend on the bound for $\|w(t) - w(0)\|^2$ with increase in m whereas $\|w(t) - w_L^*\|^2$ decreases when m increases.

CHAPTER 5

CONCLUSION AND FUTURE WORK

In this dissertation we look at two types of machine learning problems, one where the data itself is generated from a dynamical system and another where the inference algorithm is modeled as a dynamical system.

In the first case, we look at the problem of detecting events in an error log generated by a distributed data center network. The error log consists of error messages with timestamps. Our goal is to detect latent events which generate these messages and find the distribution of messages for each event. We solve this problem by relating it to the topic modeling problem in documents. We introduce a notion of episodes in the time series data which serves as the equivalent of documents. We also propose a linear time change detection algorithm to detect these episodes. We present consistency and sample complexity results for this change detection algorithm. Further, we demonstrate the performance of our algorithm on a real dataset by comparing it with two benchmark algorithms existing in the literature. We believe our approach is generic enough to be applied to other problem settings where the data has similar characteristics as network logs.

In the second case, we look at the analysis of gradient descent for two loss functions on a shallow neural network. Firstly, we consider the analysis of gradient descent on the squared loss function and show that adding a ℓ_2 regularizer $\lambda\|w - w(0)\|^2$, provides us with control between training and generalization error. Secondly, we analyze the generalization performance for the squared error in absence of any regularization. It is well known that, contrary to traditional wisdom, overparameterized neural networks perform well in training and generalization performance. One intuition behind this is that gradient descent for linear regression performs implicit regularization, i.e., minimizes the network parameter

vector among all parameter vectors which provides zero training loss. Motivated by this intuition, recent works have shown that gradient descent, with good initialization, carried out on the squared loss function of neural networks have good generalization properties. We examine this phenomenon more closely and show that gradient descent with appropriate initialization converges to a point very close to a minimum solution of a linear regression approximation to the neural network training problem. We further use this result to provide generalization guarantees on the prediction function output by gradient descent in the noise-free case.

5.1 Future Work

- In Chapter 2, we only analyze the change detection part of CD-LDA algorithm. Some topic modeling algorithms, such as [32, 33], based on spectral methods can be theoretically analyzed. It would be interesting to combine analysis of the change detection and topic modeling algorithm to generate an end to end analysis.

- In Chapter 3, we analyzed the role of regularization $|w - w_0|^2$ with the mean-squared loss function where w_0 is the initialization of the gradient descent algorithm. In practice, however, weight decay is used [76], which is a regularization of $|w|^2$. The proof technique (NTK analysis) used in Chapter 3 does not extend to this case directly since the weight moves farther away from w_0 upon addition of weight decay. Establishing results for weight decay would potentially require a new framework and therefore, is an interesting direction of future work since it highlights the shortcomings of the NTK analysis. In this spirit, a recent work in [77] analyzes the classification problem using a shallow neural network with weight decay. The analysis relies on showing some characteristics of the landscape of the loss function. But, the inference algorithm used is a variant of gradient descent which is not typically used in practice. Extending the techniques used in [77] for the gradient descent algorithm on the squared loss function would be an interesting direction of future research.

- In Chapter 4, we provide the generalization results in the noiseless case. Extending the generalization results to the non-realizable case, i.e., in the presence of i.i.d noise, using techniques from [73] is a topic for future work.

APPENDIX A

PROOFS FROM CHAPTER 2

We are making a code of our algorithm available at `https://github.com/siddpiku/CD-LDA`. The code also includes the generation of a synthetic dataset on which one can run the algorithm.

A.1 Which Algorithm for Inference in LDA Model?

In order to perform this inference of event and episode signatures using topic modeling, many inference techniques exist: Gibbs sampling on the LDA model [28], variational inference [29], online variational inference [30], stochastic variational inference [31]. There are also provable inference models based on spectral methods, such as the tensor decomposition method in [32] and the SVD based method in [33]. We use one of the popular Python packages based on Gibbs sampling based inference, [28], for the real data experiments. One can also choose to use other more recent methods for inference as mentioned above. We work in the region where the number of messages are much larger that than the number of types of messages. In this region we show that most of the inference algorithms perform the same for our problem through a synthetic data experiment.

So we compare three different inference algorithms, namely, Gibbs sampling on the LDA model [28], online variational inference [30] and the tensor decomposition method in [32]. We build an example with four types of messages and 10000 messages. We generate the time series as follows: There are two events, event one has message distribution $[0.25, 0.25, 0.499, 1e-3]$ and event two has message distribution $[0.25, 0.25, 1e-3, 0.499]$. Episode one starts from message one to message

Table A.1: Comparison between inference methods for topic modeling.

ℓ_1 norm between the estimated and the true event-message distribution maximized over all events		
Gibbs Sampling, [28]	Variational Inference, [30]	Spectral LDA, [32]
0.014	0.021	0.094

3500 and has only event one; episode two starts from message 3501 to 6054 and has half of event one and half of event two. Episode three begins at message 6055 and continues until the end with only event two occurring in this episode. We run change detection based on ℓ_1 metric followed by three different types of topic modeling inference algorithms on the episodes. We compare the inferred event-message distribution to the true event message distribution by computing the ℓ_1 norm between the estimated and the true distribution maximized over all events. Table A.1 summarizes the results. We can see that the error in estimating the event-message distributions is in the same order of magnitude.

A.2 Proof for Multiple Change-Point Case, Theorem 1

To study the multiple change-points case, [7] exploits the fact that their metric for change-point detection is convex between change points. However, the TV distance we use is not convex between two change points. But we work around this problem in the proof of Theorem 1 by showing that $D(\widetilde{\gamma})$ is increasing to the left of the first change point, unimodal/increasing/decreasing between any two change points and decreasing to the right of the last change point. Hence, any global maximum of $D(\widetilde{\gamma})$ for $0 < \widetilde{\gamma} < 1$ is located at a change point.

Suppose we have more than one change point. We plan to show that $D(\widetilde{\gamma}n) \to D(\widetilde{\gamma})$ and $D(\widetilde{\gamma})$ is increasing to the left of the first change point, unimodal/increasing/decreasing between two consecutive change points and decreasing to the right of the last change point. If this happens, then we can conclude that one of the global maximas of the $D(\widetilde{\gamma})$ occurs at a change point. Using similar

techniques from the single change-point case, it is easy to show that $D(\widetilde{\gamma})$ is increasing to the left of the first change point and decreasing to the right of the last change point. (The proof is left to the readers as an exercise.) Hence, it remains to show that $D(\widetilde{\gamma})$ is unimodal/increasing/decreasing between two consecutive change points. Lemma 8 proves this result. The prove of Lemma 8 is relegated to Appendix A.3.

Lemma 8. *$D(\gamma) = \lim_{n\to\infty} \widehat{D}(\gamma n)$ is unimodal or increasing or decreasing between two consecutive change points when there is more than one change point.*

Remark 5. *When we say $D(\gamma)$ is unimodal between two consecutive change-points $\gamma_1 < \gamma_2$, it means that there exists $\widetilde{\gamma}$, $\gamma_1 < \widetilde{\gamma} < \gamma_2$ such that $D'(\gamma) < 0$ for $\gamma < \widetilde{\gamma}$ and $D'(\gamma) > 0$ for $\gamma > \widetilde{\gamma}$.*

A.3 Proof of Lemma 8

Consider any two consecutive change points at index $\tau_1 = \gamma_1 n$ and $\tau_2 = \gamma_2 n$. Suppose the data points X are drawn i.i.d from distribution G between change-points τ_1 and τ_2. The data points to the left of τ_1 are possibly drawn independently from more than one distribution. But, for the asymptotic analysis we can assume that the data points to the left of τ_1 are possibly drawn i.i.d from the mixture of more than one distribution. Let us call this mixture distribution F. Similarly, the data points to the right of τ_2 can be assumed to be drawn i.i.d from a mixture distribution H. Let the inter-arrival time Δt be drawn from a distribution F_t to the left of τ_1, G_t between τ_1 and τ_2 and H_t to the right of τ_2.

Suppose we consider the region $\widetilde{\gamma}$ between change-points γ_1 and γ_2. So $\widehat{p}_L(\widetilde{\gamma} n)$ is a mixture of $\frac{\gamma_1}{\widetilde{\gamma}}$ fraction of samples from F and $\frac{\widetilde{\gamma}-\gamma_1}{\widetilde{\gamma}}$ fraction from G. Note that

$\widehat{p}_R(\widetilde{\gamma}n)$ is a mixture of $\frac{\gamma_2-\widetilde{\gamma}}{1-\widetilde{\gamma}}$ fraction from G and $\frac{1-\gamma_2}{1-\widetilde{\gamma}}$ fraction from H. So

$$\begin{aligned}\widehat{p}_L(\widetilde{\gamma}n) &\to \frac{\gamma_1}{\widetilde{\gamma}}F + \frac{\widetilde{\gamma}-\gamma_1}{\widetilde{\gamma}}G \\ \widehat{p}_R(\widetilde{\gamma}n) &\to \frac{\gamma_2-\widetilde{\gamma}}{1-\widetilde{\gamma}}G + \frac{1-\gamma_2}{1-\widetilde{\gamma}}H \end{aligned} \tag{A.1}$$

Similarly, the mean inter-arrival time of samples to the left of $\widetilde{\gamma}n$ converges to $\frac{\gamma_1}{\widetilde{\gamma}}\mathbb{E}F_t + \frac{\widetilde{\gamma}-\gamma_1}{\widetilde{\gamma}}\mathbb{E}G_t$, and the mean inter-arrival time to the right of $\widetilde{\gamma}n$ converges to $\frac{\gamma_2-\widetilde{\gamma}}{1-\widetilde{\gamma}}\mathbb{E}G_t + \frac{1-\gamma_2}{1-\widetilde{\gamma}}\mathbb{E}H_t$. Combining this with (A.1), we can say that

$$\begin{aligned}\widehat{D}(\widetilde{\gamma}n) \to D(\widetilde{\gamma}) =& \|\frac{\gamma_1}{\widetilde{\gamma}}(F-G) + \frac{1-\gamma_2}{1-\widetilde{\gamma}}(G-H)\|_1 \\ &+ |\frac{\gamma_1}{\widetilde{\gamma}}\mathbb{E}(F_t-G_t) + \frac{1-\gamma_2}{1-\widetilde{\gamma}}\mathbb{E}(G_t-H_t)| \end{aligned} \tag{A.2}$$

If we expand $\|\frac{\gamma_1}{\widetilde{\gamma}}(F-G) + \frac{1-\gamma_2}{1-\widetilde{\gamma}}(G-H)\|_1$ to sum of probabilities of individual messages as $\sum_{m=1}^{M}\frac{\gamma_1}{\widetilde{\gamma}}(F_m-G_m)+\frac{1-\gamma_2}{1-\widetilde{\gamma}}(G_m-H_m)$, we can write $D(\widetilde{\gamma})$ from (A.2) as a function of $\widetilde{\gamma}$ as

$$D(\widetilde{\gamma}) = \sum_{i=1}^{M+1}|\frac{a_i}{\widetilde{\gamma}} + \frac{b_i}{1-\widetilde{\gamma}}| \tag{A.3}$$

for some constants $a_i, b_i \in \mathbb{R}, i \in \{1, 2, \ldots, M+1\}$. Function $D(\widetilde{\gamma})$ from (A.3) is only well defined over $\gamma_1 < \widetilde{\gamma} < \gamma_2$. For the purpose of this proof, with some abuse of notation we assume the function $D(\gamma)$ to have the same definition outside $[\gamma_1, \gamma_2]$. We then show that $D(\widetilde{\gamma})$ defined in (A.3) is unimodal/increasing/decreasing as a function of $\widetilde{\gamma}$ between $(0, 1)$. This would naturally imply that $D(\widetilde{\gamma})$ is unimodal/increasing/decreasing in $[\gamma_1, \gamma_2]$. The rest of the proof deals with this analysis.

Without loss of generality we can assume $a_i \geq 0, \ \ \forall i$. Note that $a_i, b_i \neq 0$ for all

i. We can expand (A.3) as

$$D(\widetilde{\gamma}) = \sum_{a_i>0,b_i>0} |\frac{a_i}{\widetilde{\gamma}} + \frac{b_i}{1-\widetilde{\gamma}}| + \sum_{a_i>0,b_i<0} |\frac{a_i}{\widetilde{\gamma}} - \frac{-b_i}{1-\widetilde{\gamma}}| + \sum_{b_i=0} \frac{a_i}{\widetilde{\gamma}} + \sum_{a_i=0} \frac{|b_i|}{1-\widetilde{\gamma}} \tag{A.4}$$

$$= \frac{a}{\widetilde{\gamma}} + \frac{b}{1-\widetilde{\gamma}} + \sum_{a_i,d_i>0} |\frac{a_i}{\widetilde{\gamma}} - \frac{d_i}{1-\widetilde{\gamma}}| \tag{A.5}$$

where $d_i = -b_i$ when $b_i < 0$, $\sum_{i:b_i\geq 0} a_i = a$ and $\sum_{i:b_i>0} +b_i \sum_{a_i=0} |b_i| = b$. For $\widetilde{\gamma} < \frac{a_i}{a_i+d_i}$, $\frac{a_i}{\widetilde{\gamma}} - \frac{d_i}{1-\widetilde{\gamma}} > 0$. We can assume w.l.o.g. that $\frac{a_i}{a_i+d_i}$ are in increasing order. Suppose $\frac{a_s}{a_s+d_s} < \widetilde{\gamma} < \frac{a_{s+1}}{a_{s+1}+d_{s+1}}$.

$$D(\widetilde{\gamma}) = \frac{a - \sum_{i<s} a_i + \sum_{i\geq s} a_i}{\widetilde{\gamma}} + \frac{b + \sum_{i<s} d_i - \sum_{i\geq s} d_i}{1-\widetilde{\gamma}}$$

Let $a(s) = a - \sum_{i<s} a_i + \sum_{i\geq s} a_i$ and $b(s) = b + \sum_{i<s} d_i - \sum_{i\geq s} d_i$. So for $\frac{a_s}{a_s+b_s} < \widetilde{\gamma} < \frac{a_{s+1}}{a_{s+1}+b_{s+1}}$

$$D(\widetilde{\gamma}) = \frac{a(s)}{\widetilde{\gamma}} + \frac{b(s)}{1-\widetilde{\gamma}}, \quad \frac{a_s}{a_s+b_s} < \widetilde{\gamma} < \frac{a_{s+1}}{a_{s+1}+b_{s+1}} \tag{A.6}$$

See that $a(s) > 0$ for $s = 0$ and it is a *decreasing* function of s. $b(s)$ is a *increasing* function of s. Based on where $a(s)$ changes sign w.r.t $b(s)$ we have the following cases. Note that $a(s)$ and $b(s)$ cannot both be negative for any value of s. $D'(\widetilde{\gamma})$ denotes the derivative of $D(\widetilde{\gamma})$ whereever it is defined.

- Suppose $a(s) > 0, b(s) > 0$ for all values of s. $D(\widetilde{\gamma})$ is a convex function of $\widetilde{\gamma}$ and hence is unimodal.
- Suppose $a(s) > 0$ for all values of s and $b(s)$ changes sign at $s = u$, i.e., $b(u) < 0, b(u+1) > 0$. So for $s \leq u$, $D'(\widetilde{\gamma}) < 0$ and for $s > u$, $D'(\widetilde{\gamma}) > 0$. Hence, $D(\widetilde{\gamma})$ is an unimodal function of $\widetilde{\gamma}$ between zero and one.
- Suppose $a(s)$ changes sign at $s = t$, i.e., $a(t) \geq 0, a(t+1) < 0$ and $b(s) > 0$ for all s. So for $s \leq t$, $D(\widetilde{\gamma}n)$ is convex, and for $s > t$, $D'(\widetilde{\gamma}n)$ is positive. Hence, $D(\widetilde{\gamma}n)$ is either increasing or unimodal between zero and one.

- Suppose $a(t) \geq 0, a(t+1) < 0$ and $b(u) \leq 0, b(u+1) > 0$. Also $t < u$. So for $s \leq t$ $D'(\widetilde{\gamma}n)$ is decreasing, for $t < s \leq u$ $D(\widetilde{\gamma}n)$ is convex and for $s > u$ $D'(\widetilde{\gamma}n)$ is increasing. Hence $D(\widetilde{\gamma}n)$ is unimodal.

A.4 Proof for Multiple Change-Point Case, Theorem 2

Similar to the single change-point case we first characterize the estimated change-points $\widehat{\gamma}_1, \ldots, \widehat{\gamma}_k$ for finite n in Lemmas 9-11.

Lemma 9. $|\widehat{D}(\widetilde{\gamma}n) - D(\widetilde{\gamma})| \leq \epsilon$ *w.p. at least* $1 - 4n \exp\left(-\frac{\epsilon^2\alpha^2}{32\max(\sigma,k+1)^2}n + M\log(n+k)\right)$ *for all values of* $\widetilde{\gamma}$.

Lemma 10 can be proved in a similar way to Lemma 2 in the single change-point case.

Lemma 10. $|D(\gamma_i) - D(\widehat{\gamma}_i)| < 2\epsilon$ *w.p. at least* $1 - 4n \exp\left(-\frac{\epsilon^2\alpha^2}{32\max(\sigma,k+1)^2}n + M\log(n+k)\right)$ *for any change-point* γ_i.

Lemma 11. $|\widehat{\gamma}_i - \gamma_i| < c\epsilon$ *w.p. at least* $1 - 4n \exp\left(-\frac{\epsilon^2\alpha^2}{32\max(\sigma,k+1)^2}n + M\log(n+k)\right)$ *for some constant* $c > 0$ *and any change-point* γ_i.

Now, we can state the correctness result for Algorithm 2. Algorithm 2 is correct given accuracy $\epsilon > 0$ as mentioned in Definition 1 with probability $1 - 7(2k+1)\exp(-\frac{(D^*-\epsilon)^2\epsilon^2\alpha^4}{512\max(\sigma^2,k+1)}n + M\log(n))$.

We upper bound the probability that Algorithm 2 is not correct. From Definition 1, this happens when

- Algorithm 2 is correct every time it calls Algorithm 1.

The maximum number of times Algorithm 1 would be applied is $2k+1$ if it is correct every time it is applied. Out of the $2k+1$ times k number of times should

return a change point and $k+1$ number of times should return no change point. So

$$\begin{aligned}
&P(\text{Algorithm 2 is NOT correct}) \\
&\leq kP(\text{Algorithm 1 does NOT detect a change point} \\
&\text{when one exists for dataset } X_L, \ldots, X_H) \\
&+ (k+1)P(\text{Algorithm 1 returns a change point} \\
&\text{when one does not exist})
\end{aligned} \tag{A.7}$$

We assume that $X_L, \ldots, X_H$ is at least of size αn or an episode is at least αn samples long. Let D^* denote the minimum value of metric D at a global maxima for the reduced problem of $X_L, \ldots, X_H$ over all possible values of L, H for which Algorithm 1 is applied. From the correctness result for one change point, we have that

$$\begin{aligned}
&P(\text{Algorithm 1 does NOT detect a change point} \\
&\text{when one exists for dataset } X_L, \ldots, X_H) \\
&\leq 3n \exp\left(-\frac{\epsilon^2\alpha^2}{128\sigma^2}n + M\log(n)\right) \\
&+ 3n \exp\left(-\frac{\epsilon^2 D^*}{512\sigma^2}n + M\log(n)\right)
\end{aligned} \tag{A.8}$$

and

$$\begin{aligned}
&P(\text{Algorithm 1 returns a change point} \\
&\text{when one does not exist}) \\
&\leq (n+2)^V \exp(-n\delta^2/8)
\end{aligned} \tag{A.9}$$

Combining the above two cases, we get

$$
\begin{aligned}
&P(\text{Algorithm 2 is NOT correct}) \\
&\leq 7(2k+1)\exp(-\frac{(D^*-\epsilon)^2\epsilon^2\alpha^4}{512\max(\sigma^2,k+1)}n + M\log(n)) \qquad \text{(A.10)}
\end{aligned}
$$

Finally, we can derive the sample complexity result for k change points from (A.10) by assuming

- $\epsilon < \frac{D^*}{2}$.
- $\alpha + \epsilon < \min_r |\gamma_r - \gamma_{r-1}|$.
- $\delta < D^* - \epsilon$.

A.5 Proof of Lemma 1

For notational simplicity, for any $\widetilde{\gamma}$, suppose $\widehat{p}_L(\widetilde{\gamma}n) \to p_L(\widetilde{\gamma})$ and $\widehat{p}_R(\widetilde{\gamma}n) \to p_R(\widetilde{\gamma})$.

Since $|\widehat{D}(\widetilde{\gamma}n) - D(\widetilde{\gamma})| \leq |\quad \|\widehat{p}_L(\widetilde{\gamma}n) - \widehat{p}_R(\widetilde{\gamma}n)\| - \|p_L(\widetilde{\gamma}n) - q(\widetilde{\gamma}n)\| \quad| + |\ |\widehat{\mathbb{E}}S_1(\widetilde{\gamma}n) - \widehat{\mathbb{E}}S_2(\widetilde{\gamma}n)| - |\widehat{\mathbb{E}}S_1(\widetilde{\gamma}n) - \widehat{\mathbb{E}}S_2(\widetilde{\gamma}n)|\ |$,

$$
\begin{aligned}
&P(|\widehat{D}(\widetilde{\gamma}n) - D(\widetilde{\gamma}n)| > \epsilon) \\
&\leq P(|\ \ \|\widehat{p}(\widetilde{\gamma}n) - \widehat{q}(\widetilde{\gamma}n)\| - \|p(\widetilde{\gamma}n) - q(\widetilde{\gamma}n)\|\ \ | > \frac{\epsilon}{2}) \\
&+ P(|\ \ |\widehat{m}_1(\widetilde{\gamma}n) - \widehat{m}_2(\widetilde{\gamma}n)| - |m_1(\widetilde{\gamma}n) - m_2(\widetilde{\gamma}n)|\ \ | > \frac{\epsilon}{2}) \qquad \text{(A.11)}
\end{aligned}
$$

First we focus on finding an upper bound to the probability $P(|\ \ \|\widehat{p}(\widetilde{\gamma}n) - \widehat{q}(\widetilde{\gamma}n)\| - \|p(\widetilde{\gamma}n) - q(\widetilde{\gamma}n)\|\ \ | > \frac{\epsilon}{2})$.

$$
\begin{aligned}
&P(|\ \ \|\widehat{p}(\widetilde{\gamma}n) - \widehat{q}(\widetilde{\gamma}n)\| - \|p(\widetilde{\gamma}n) - q(\widetilde{\gamma}n)\|\ \ | > \frac{\epsilon}{2}) \qquad \text{(A.12)} \\
&\leq P(\|\widehat{p}(\widetilde{\gamma}n) - \widehat{q}(\widetilde{\gamma}n)\| > \|p(\widetilde{\gamma}n) - q(\widetilde{\gamma}n)\| + \frac{\epsilon}{2}) \\
&+ P(\|\widehat{p}(\widetilde{\gamma}n) - \widehat{q}(\widetilde{\gamma}n)\| < \|p(\widetilde{\gamma}n) - q(\widetilde{\gamma}n)\| - \frac{\epsilon}{2}) \qquad \text{(A.13)}
\end{aligned}
$$

Using $\|\widehat{p}(\widetilde{\gamma}n) - \widehat{q}(\widetilde{\gamma}n)\| \leq \|\widehat{p}(\widetilde{\gamma}n) - p(\widetilde{\gamma}n)\| + \|\widehat{q}(\widetilde{\gamma}n) - q(\widetilde{\gamma}n)\| + \|p(\widetilde{\gamma}n) - q(\widetilde{\gamma}n)\|$ and $\|\widehat{p}(\widetilde{\gamma}n) - \widehat{q}(\widetilde{\gamma}n)\| \geq \|p(\widetilde{\gamma}n) - q(\widetilde{\gamma}n)\| - \|\widehat{p}(\widetilde{\gamma}n) - p(\widetilde{\gamma}n)\| - \|\widehat{q}(\widetilde{\gamma}n) - q(\widetilde{\gamma}n)\|$ in (A.13) we have,

$$P(|\ \ \|\widehat{p}(\widetilde{\gamma}n) - \widehat{q}(\widetilde{\gamma}n)\| - \|p(\widetilde{\gamma}n) - q(\widetilde{\gamma}n)\|\ \ | > \frac{\epsilon}{2}) \tag{A.14}$$

$$\leq 2P(\|\widehat{p}(\widetilde{\gamma}n) - p(\widetilde{\gamma}n)\| + \|\widehat{q}(\widetilde{\gamma}n) - q(\widetilde{\gamma}n)\| > \frac{\epsilon}{2}) \tag{A.15}$$

$$\leq 2P(\|\widehat{p}(\widetilde{\gamma}n) - p(\widetilde{\gamma}n)\| > \frac{\epsilon}{4}) + 2P(\|\widehat{q}(\widetilde{\gamma}n) - q(\widetilde{\gamma}n)\| > \frac{\epsilon}{4}) \tag{A.16}$$

Let $\widetilde{\gamma} > \gamma$. Now, $\|\widehat{p}(\widetilde{\gamma}n) - p(\widetilde{\gamma}n)\| \leq \frac{\gamma}{\widetilde{\gamma}}\|\widehat{p}(\gamma n) - p\| + \frac{\widetilde{\gamma}-\gamma}{\widetilde{\gamma}}\|\widehat{p}(\gamma n : \widetilde{\gamma}n) - q\|$. So,

$$P(\|\widehat{p}(\widetilde{\gamma}n) - p(\widetilde{\gamma}n)\| > \frac{\epsilon}{4}) \tag{A.17}$$

$$\leq P(\frac{\gamma}{\widetilde{\gamma}}\|\widehat{p}(\gamma n) - p\| > \frac{\epsilon}{8}) + P(\frac{\widetilde{\gamma}-\gamma}{\widetilde{\gamma}}\|\widehat{p}(\gamma n : \widetilde{\gamma}n) - q\| > \frac{\epsilon}{8}) \tag{A.18}$$

We will apply Sanov's theorem to find an upper bound to $P(\frac{\gamma}{\widetilde{\gamma}}\|\widehat{p}(\gamma n) - p\| > \frac{\epsilon}{8})$. Consider the set E of empirical probability distributions from i.i.d samples $X_1, \ldots, X_{\gamma n}$. $E = \{\widehat{p}(\gamma n) : \frac{\gamma}{\widetilde{\gamma}}\|\widehat{p}(\gamma n) - p\| > \frac{\epsilon}{8}\}$. By Sanov's theorem we can say that,

$$P(E) \leq (\gamma n + 1)^V \exp(-n \min_{p^* \in E} D_{KL}(p^*||p)) \tag{A.19}$$

Further, by Pinsker's inequality, we have $D_{KL}(p^*||p) \geq \frac{1}{2}\|p^* - p\|^2$. Using this in (A.19),

$$P(\frac{\gamma}{\widetilde{\gamma}}\|\widehat{p}(\gamma n) - p\| > \frac{\epsilon}{8}) \leq (\gamma n + 1)^V \exp\left(-\frac{n\epsilon^2}{128}\left(\frac{\widetilde{\gamma}}{\gamma}\right)^2\right) \tag{A.20}$$

A similar approach yields

$$P(\frac{\widetilde{\gamma}-\gamma}{\widetilde{\gamma}}\|\widehat{p}(\gamma n:\widetilde{\gamma}n)-q\|>\frac{\epsilon}{8})$$
$$\leq((\widetilde{\gamma}-\gamma)n+1)^V\exp\left(-\frac{n\epsilon^2}{128}\left(\frac{\widetilde{\gamma}}{\widetilde{\gamma}-\gamma}\right)^2\right) \tag{A.21}$$

Combining (A.20) and (A.21) and substituting in (A.18), we get

$$P(\|\widehat{p}(\widetilde{\gamma}n)-p(\widetilde{\gamma}n)\|>\frac{\epsilon}{4})\leq(\gamma n+1)^V\exp\left(-\frac{n\epsilon^2}{128}\left(\frac{\widetilde{\gamma}}{\gamma}\right)^2\right)$$
$$+((\widetilde{\gamma}-\gamma)n+1)^V\exp\left(-\frac{n\epsilon^2}{128}\left(\frac{\widetilde{\gamma}}{\widetilde{\gamma}-\gamma}\right)^2\right) \tag{A.22}$$

Also using Sanov's theorem followed by Pinsker's inequality we have,

$$P(\|\widehat{q}(\widetilde{\gamma}n)-q\|>\frac{\epsilon}{4})\leq((1-\widetilde{\gamma})n+1)^V\exp\left(-\frac{n\epsilon^2}{32}\right) \tag{A.23}$$

Finally, (A.22) and (A.23) yield the following inequality using (A.16),

$$P(|\ \ \|\widehat{p}(\widetilde{\gamma}n)-\widehat{q}(\widetilde{\gamma}n)\|-\|p(\widetilde{\gamma}n)-q(\widetilde{\gamma}n)\|\ \ |>\frac{\epsilon}{2}) \tag{A.24}$$
$$\leq 2(\gamma n+1)^V\exp\left(-\frac{n\epsilon^2}{128}\left(\frac{\widetilde{\gamma}}{\gamma}\right)^2\right)$$
$$+2((\widetilde{\gamma}-\gamma)n+1)^V\exp\left(-\frac{n\epsilon^2}{128}\left(\frac{\widetilde{\gamma}}{\widetilde{\gamma}-\gamma}\right)^2\right) \tag{A.25}$$
$$+2((1-\widetilde{\gamma})n+1)^V\exp\left(-\frac{n\epsilon^2}{32}\right) \tag{A.26}$$
$$\leq 3\exp\left(-\frac{\epsilon^2\alpha^2}{128}n+V\log(n)\right) \tag{A.27}$$

We can get a result that is the same as (A.27) for $\widetilde{\gamma}<\gamma$.

Now, let us prove concentration results for $g(\widetilde{\gamma})$.

$$g(\widetilde{\gamma}) = \widehat{\mathbb{E}}S_1(\widetilde{\gamma}n) - \widehat{\mathbb{E}}S_2(\widetilde{\gamma}n) = \frac{\sum_{j=1}^{\widetilde{\gamma}n} \Delta t_j}{\widetilde{\gamma}n} - \frac{\sum_{j=\widetilde{\gamma}n+1}^{n} \Delta t_j}{(1-\widetilde{\gamma})n}$$

By assumption, Δt_j is sub-Gaussian from $j = 1$ to γn with parameter σ_1^2 and from $j = \gamma n + 1$ to γn with parameter σ_2^2. If Δt_j is sub-Gaussian, so is r.v. $-\Delta t_j$ with the same sub-Gaussian parameter. Sum of sub-Gaussian r.v is also sub-Gaussian with parameter equal to the sum of individual sub-Gaussian parameters. Let $\sigma = \max(\sigma_1, \sigma_2)$. So, the sum of sub-Gaussian parameters for $g(\widetilde{\gamma})$, say σ_g, is upper bounded by

$$\sigma_g^2 \leq \sum_{j=1}^{\widetilde{\gamma}n} \frac{\sigma^2}{\widetilde{\gamma}^2 n^2} + \sum_{j=\widetilde{\gamma}n+1}^{n} \frac{\sigma^2}{(1-\widetilde{\gamma})^2 n^2} \leq \frac{\sigma^2}{\alpha^2 n}$$

$$P(|g(\widetilde{\gamma}n) - \mathbb{E}g(\widetilde{\gamma}n)| > \frac{\epsilon}{2}) \leq 2\exp\left(-\frac{\epsilon^2}{8\sigma_g^2}\right) \tag{A.28}$$

$$\leq 2\exp\left(-n\epsilon^2\frac{\alpha^2}{8\sigma^2}\right) \tag{A.29}$$

Putting together (A.29) and (A.27) with (A.11),

$$P(|\widehat{D}(\widetilde{\gamma}n) - D(\widetilde{\gamma}n)| > \epsilon) \leq 3\exp\left(-\frac{\epsilon^2\alpha^2}{128}n + V\log(n)\right) + 2\exp\left(-n\epsilon^2\frac{\alpha^2}{8\sigma^2}\right) \tag{A.30}$$

$$\leq 3\exp\left(-\frac{\epsilon^2\alpha^2}{128\sigma^2}n + V\log(n)\right) \tag{A.31}$$

For all values of $\widetilde{\gamma}$, we have by union bound,

$$P(|\widehat{D}(\widetilde{\gamma}n) - D(\widetilde{\gamma}n)| < \epsilon, \text{ for all } \alpha < \widetilde{\gamma} < 1-\alpha) \leq 1 - 3n\exp\left(-\frac{\epsilon^2\alpha^2}{128\sigma^2}n + V\log(n)\right) \tag{A.32}$$

A.6 Proof of Lemma 2 and Lemma 3

From (2.11) $\arg\max_{\widetilde{\gamma}} D(\widetilde{\gamma}n) = \gamma$. Let $\arg\max_{\widetilde{\gamma}} \widehat{D}(\widetilde{\gamma}n) = \widehat{\gamma}$.
Now, w.p. $1 - 3n\exp\left(-\frac{\epsilon^2\alpha^2}{128\sigma^2}n + V\log(n)\right)$

$$D(\gamma n) - \widehat{D}(\widehat{\gamma}) < \widehat{D}(\gamma n) - \widehat{D}(\widehat{\gamma}n) + \epsilon < \epsilon$$

Also,

$$D(\gamma n) - \widehat{D}(\widehat{\gamma}) > D(\gamma n) - D(\widehat{\gamma}n) - \epsilon > -\epsilon$$

So we have,

$$|D(\gamma n) - \widehat{D}(\widehat{\gamma}n)| \leq \epsilon$$

w.p. $1 - 3n\exp\left(-\frac{\epsilon^2\alpha^2}{128\sigma^2}n + V\log(n)\right)$.
Now $0 < D(\gamma n) - D(\widehat{\gamma}n) < D(\gamma n) - \widehat{D}(\widehat{\gamma}n) + \epsilon < 2\epsilon$.
Suppose $\widehat{\gamma} > \gamma$. So by (2.11), $|D(\gamma n) - D(\widehat{\gamma}n)| = D(\gamma n)|\frac{\widehat{\gamma}-\gamma}{\widehat{\gamma}}| > \frac{D(\gamma n)}{1-\alpha}|\widehat{\gamma} - \gamma|$.
For $\widehat{\gamma} < \gamma$, $|D(\gamma n) - D(\widehat{\gamma}n)| = D(\gamma n)|\frac{\widehat{\gamma}-\gamma}{1-\widehat{\gamma}}| > \frac{D(\gamma n)}{1-\alpha}|\widehat{\gamma} - \gamma|$.

A.7 Proof of Lemma 9

Since $|\widehat{D}(\widetilde{\gamma}n) - D(\widetilde{\gamma}n)| \leq |\ \ \|\widehat{p}(\widetilde{\gamma}n) - \widehat{q}(\widetilde{\gamma}n)\| - \|p(\widetilde{\gamma}n) - q(\widetilde{\gamma}n)\|\ \ | + |\ \ |\widehat{m}_1(\widetilde{\gamma}n) - \widehat{m}_2(\widetilde{\gamma}n)| - |m_1(\widetilde{\gamma}n) - m_2(\widetilde{\gamma}n)\ \ |$,

$$\begin{aligned}
&P(|\widehat{D}(\widetilde{\gamma}n) - D(\widetilde{\gamma}n)| > \epsilon) \\
&\leq P(|\ \ \|\widehat{p}(\widetilde{\gamma}n) - \widehat{q}(\widetilde{\gamma}n)\| - \|p(\widetilde{\gamma}n) - q(\widetilde{\gamma}n)\|\ \ | > \frac{\epsilon}{2}) \\
&+ P(|\ \ |\widehat{m}_1(\widetilde{\gamma}n) - \widehat{m}_2(\widetilde{\gamma}n)| - |m_1(\widetilde{\gamma}n) - m_2(\widetilde{\gamma}n)|\ \ | > \frac{\epsilon}{2}) \qquad \text{(A.33)}
\end{aligned}$$

First we focus on finding an upper bound to the probability $P(\mid \ \|\widehat{p}(\widetilde{\gamma}n) - \widehat{q}(\widetilde{\gamma}n)\| - \|p(\widetilde{\gamma}n) - q(\widetilde{\gamma}n)\| \ \mid > \frac{\epsilon}{2})$.

$$P(\mid \ \|\widehat{p}(\widetilde{\gamma}n) - \widehat{q}(\widetilde{\gamma}n)\| - \|p(\widetilde{\gamma}n) - q(\widetilde{\gamma}n)\| \ \mid > \frac{\epsilon}{2}) \tag{A.34}$$

$$\leq P(\|\widehat{p}(\widetilde{\gamma}n) - \widehat{q}(\widetilde{\gamma}n)\| > \|p(\widetilde{\gamma}n) - q(\widetilde{\gamma}n)\| + \frac{\epsilon}{2})$$

$$+ P(\|\widehat{p}(\widetilde{\gamma}n) - \widehat{q}(\widetilde{\gamma}n)\| < \|p(\widetilde{\gamma}n) - q(\widetilde{\gamma}n)\| - \frac{\epsilon}{2}) \tag{A.35}$$

Using $\|\widehat{p}(\widetilde{\gamma}n) - \widehat{q}(\widetilde{\gamma}n)\| \leq \|\widehat{p}(\widetilde{\gamma}n) - p(\widetilde{\gamma}n)\| + \|\widehat{q}(\widetilde{\gamma}n) - q(\widetilde{\gamma}n)\| + \|p(\widetilde{\gamma}n) - q(\widetilde{\gamma}n)\|$ and $\|\widehat{p}(\widetilde{\gamma}n) - \widehat{q}(\widetilde{\gamma}n)\| \geq \|p(\widetilde{\gamma}n) - q(\widetilde{\gamma}n)\| - \|\widehat{p}(\widetilde{\gamma}n) - p(\widetilde{\gamma}n)\| - \|\widehat{q}(\widetilde{\gamma}n) - q(\widetilde{\gamma}n)\|$ in (A.35), we have,

$$P(\mid \ \|\widehat{p}(\widetilde{\gamma}n) - \widehat{q}(\widetilde{\gamma}n)\| - \|p(\widetilde{\gamma}n) - q(\widetilde{\gamma}n)\| \ \mid > \frac{\epsilon}{2}) \tag{A.36}$$

$$\leq 2P(\|\widehat{p}(\widetilde{\gamma}n) - p(\widetilde{\gamma}n)\| + \|\widehat{q}(\widetilde{\gamma}n) - q(\widetilde{\gamma}n)\| > \frac{\epsilon}{2}) \tag{A.37}$$

$$\leq 2P(\|\widehat{p}(\widetilde{\gamma}n) - p(\widetilde{\gamma}n)\| > \frac{\epsilon}{4}) + 2P(\|\widehat{q}(\widetilde{\gamma}n) - q(\widetilde{\gamma}n)\| > \frac{\epsilon}{4}) \tag{A.38}$$

Let $\gamma_r < \widetilde{\gamma} < \gamma_{r+1}$. Now, $\|\widehat{p}(\widetilde{\gamma}n) - p(\widetilde{\gamma}n)\| \leq \sum_{j=0}^{r} \frac{\gamma_j - \gamma_{j-1}}{\widetilde{\gamma}} \|\widehat{p}^{j-1} - p^{j-1}\| + \frac{\widetilde{\gamma} - \gamma_r}{\widetilde{\gamma}} \|\widehat{p}(\gamma n : \widetilde{\gamma}n) - p^r\|$. So,

$$P(\|\widehat{p}(\widetilde{\gamma}n) - p(\widetilde{\gamma}n)\| > \frac{\epsilon}{4}) \tag{A.39}$$

$$\leq \sum_{j=0}^{r} P(\frac{\gamma_j - \gamma_{j-1}}{\widetilde{\gamma}} \|\widehat{p}^{j-1} - p^{j-1}\| > \frac{\epsilon}{4(r+1)})$$

$$+ P(\frac{\widetilde{\gamma} - \gamma_r}{\widetilde{\gamma}} \|\widehat{p}(\gamma_r n : \widetilde{\gamma}n) - p^r\| > \frac{\epsilon}{4(r+1)}) \tag{A.40}$$

We will apply Sanov's theorem to find an upper bound to $P(\frac{\gamma_j - \gamma_{j-1}}{\widetilde{\gamma}} \|\widehat{p}^{j-1} - p^{j-1}\| > \frac{\epsilon}{4(r+1)})$. Consider the set E of empirical probability distributions from i.i.d samples $X_{\gamma_{j-1}n}, \ldots, X_{\gamma_j n}$. $E = \{\widehat{p}^{j-1} : \frac{\gamma_j - \gamma_{j-1}}{\widetilde{\gamma}} \|\widehat{p}^{j-1} - p^{j-1}\| > \frac{\epsilon}{4(r+1)}\}$. By Sanov's theorem we can say that,

$$P(E) \leq ((\gamma_j - \gamma_{j-1})n + 1)^V \exp(-n \min_{p^* \in E} D_{KL}(p^* || p^{j-1})) \tag{A.41}$$

Further, by Pinsker's inequality, we have $D_{KL}(p^*||p^{j-1}) \geq \frac{1}{2}\|p^* - p^{j-1}\|^2$. Using this in (A.41),

$$\begin{aligned} &P(\frac{\gamma_j - \gamma_{j-1}}{\widetilde{\gamma}} \|\widehat{p}^{j-1} - p^{j-1}\| > \frac{\epsilon}{4(r+1)}) \\ &\leq ((\gamma_j - \gamma_{j-1})n + 1)^V \exp\left(-\frac{n\epsilon^2}{32(r+1)^2}\left(\frac{\widetilde{\gamma}}{\gamma_j - \gamma_{j-1}}\right)^2\right) \end{aligned} \tag{A.42}$$

A similar approach yields

$$\begin{aligned} &P(\frac{\widetilde{\gamma} - \gamma_r}{\widetilde{\gamma}} \|\widehat{p}(\gamma n : \widetilde{\gamma} n) - p^{j-1}\| > \frac{\epsilon}{4(r+1)}) \\ &\leq ((\widetilde{\gamma} - \gamma_r)n + 1)^V \exp\left(-\frac{n\epsilon^2}{32(r+1)^2}\left(\frac{\widetilde{\gamma}}{\widetilde{\gamma} - \gamma_r}\right)^2\right) \end{aligned} \tag{A.43}$$

Combining (A.42) and (A.43) and substituting in (A.40), we get

$$
\begin{aligned}
&P(\|\widehat{p}(\widetilde{\gamma}n) - p(\widetilde{\gamma}n)\| > \frac{\epsilon}{4}) \\
&\leq \sum_{j=0}^{r}((\gamma_j - \gamma_{j-1})n + 1)^V \times \\
&\exp\left(-\frac{n\epsilon^2}{32(r+1)^2}\left(\frac{\widetilde{\gamma}}{\gamma_j - \gamma_{j-1}}\right)^2\right) \\
&+ ((\widetilde{\gamma} - \gamma_r)n + 1)^V \exp\left(-\frac{n\epsilon^2}{32(r+1)^2}\left(\frac{\widetilde{\gamma}}{\widetilde{\gamma} - \gamma_r}\right)^2\right) \\
&\leq \left(\sum_{j=0}^{r}((\gamma_j - \gamma_{j-1})n + 1)^V + ((\widetilde{\gamma} - \gamma_r)n + 1)^V\right) \times \\
&\exp\left(-\frac{n\epsilon^2\alpha^2}{32(k+1)^2}\right) \qquad \text{(A.44)}
\end{aligned}
$$

Also using Sanov's theorem followed by Pinsker's inequality we have,

$$
\begin{aligned}
&P(\|\widehat{q}(\widetilde{\gamma}n) - q(\widetilde{\gamma}n)\| > \frac{\epsilon}{4}) \\
&\leq \left(\sum_{j=r+1}^{k}((\gamma_j - \gamma_{j-1})n + 1)^V + ((\gamma_{r+1} - \widetilde{\gamma})n + 1)^V\right) \times \\
&\exp\left(-\frac{n\epsilon^2\alpha^2}{32(k+1)^2}\right) \qquad \text{(A.45)}
\end{aligned}
$$

Finally, (A.44) and (A.45) yield the following inequality using (A.38),

$$
\begin{aligned}
&P(|\ \ \|\widehat{p}(\widetilde{\gamma}n) - \widehat{q}(\widetilde{\gamma}n)\| - \|p(\widetilde{\gamma}n) - q(\widetilde{\gamma}n)\|\ \ | > \frac{\epsilon}{2}) \\
&\leq 2(\sum_{j=0}^{r}((\gamma_j - \gamma_{j-1})n + 1)^V + ((\widetilde{\gamma} - \gamma_r)n + 1)^V \\
&+ ((\gamma_{r+1} - \widetilde{\gamma})n + 1)^V \\
&+ \sum_{j=r+1}^{k}((\gamma_j - \gamma_{j-1})n + 1)^V) \exp\left(-\frac{n\epsilon^2\alpha^2}{32(k+1)^2}\right) \qquad \text{(A.46)} \\
&\leq 2\exp\left(-\frac{\epsilon^2\alpha^2}{32(k+1)^2}n + V\log(n+k)\right) \qquad \text{(A.47)}
\end{aligned}
$$

Now, let us prove concentration results for $g(\widetilde{\gamma})$.

$$
g(\widetilde{\gamma}) = \widehat{\mathbb{E}}S_1(\widetilde{\gamma}n) - \widehat{\mathbb{E}}S_2(\widetilde{\gamma}n) = \frac{\sum_{j=1}^{\widetilde{\gamma}n}\Delta t_j}{\widetilde{\gamma}n} - \frac{\sum_{j=\widetilde{\gamma}n+1}^{n}\Delta t_j}{(1-\widetilde{\gamma})n}
$$

By assumption, Δt_j is sub-Gaussian from $j = 1$ to $\gamma_1 n$ with parameter σ_1^2 and from $j = \gamma_1 n + 1$ to $\gamma_2 n$ with parameter σ_2^2 and so on. If Δt_j is sub-Gaussian, so is r.v. $-\Delta t_j$ with the same sub-Gaussian parameter. Sum of sub-Gaussian r.v is also sub-Gaussian with parameter equal to the sum of individual sub-Gaussian parameters. Let $\sigma = \max(\sigma_1, \sigma_2, \ldots, \sigma_k)$. So, the sum of sub-Gaussian parameters for $g(\widetilde{\gamma})$, say σ_g , is upper bounded by

$$
\sigma_g^2 \leq \sum_{j=1}^{\widetilde{\gamma}n}\frac{\sigma^2}{\widetilde{\gamma}^2n^2} + \sum_{j=\widetilde{\gamma}n+1}^{n}\frac{\sigma^2}{(1-\widetilde{\gamma})^2n^2} \leq \frac{\sigma^2}{\alpha^2 n}
$$

$$
\begin{aligned}
P(|g(\widetilde{\gamma}n) - \mathbb{E}g(\widetilde{\gamma}n)| > \frac{\epsilon}{2}) &\leq 2\exp\left(-\frac{\epsilon^2}{8\sigma_g^2}\right) \\
&\leq 2\exp\left(-n\epsilon^2\frac{\alpha^2}{8\sigma^2}\right) \qquad \text{(A.48)}
\end{aligned}
$$

Putting together (A.48) and (A.47) with (A.33),

$$
\begin{aligned}
&P(|\widehat{D}(\widetilde{\gamma}n) - D(\widetilde{\gamma}n)| > \epsilon) \\
&\leq 2\exp\left(-\frac{\epsilon^2\alpha^2}{32(k+1)^2}n + V\log(n+k)\right) \\
&+ 2\exp\left(-n\epsilon^2\frac{\alpha^2}{8\sigma^2}\right) \\
&\leq 4\exp\left(-\frac{\epsilon^2\alpha^2}{32\max(\sigma,k+1)^2}n + V\log(n+k)\right) \qquad \text{(A.49)}
\end{aligned}
$$

For all values of $\widetilde{\gamma}$, we have by union bound,

$$
\begin{aligned}
&P(|\widehat{D}(\widetilde{\gamma}n) - D(\widetilde{\gamma}n)| < \epsilon, \text{ for all } \alpha < \widetilde{\gamma} < 1-\alpha) \\
&\leq 1 - 4n\exp\left(-\frac{\epsilon^2\alpha^2}{32\max(\sigma,k+1)^2}n + V\log(n+k)\right) \qquad \text{(A.50)}
\end{aligned}
$$

A.8 Proof of Lemma 11

From Lemma 10, $|D(\gamma n) - D(\widehat{\gamma}n)| < 2\epsilon$ w.p. at least $1-$ $4n\exp\left(-\frac{\epsilon^2\alpha^2}{32\max(\sigma,k+1)^2}n + V\log(n+k)\right)$. Also, from Lemma 8 we know that γn is a change point, and all local maximas in $D(\widetilde{\gamma}n)$ for $0 < \widetilde{\gamma} < 1$ correspond to a change point. Suppose that γ_r is a change point closest to $\widehat{\gamma}$ such that $|D(\gamma_r n) - D(\widehat{\gamma}n)| < 2\epsilon$. Also, since $D(\widetilde{\gamma}n)$ for $0 < \widetilde{\gamma} < 1$ is unimodal or monotonic between γ_r, γ_{r+1} or γ_{r-1}, γ_r we assume w.l.o.g that $D'(\gamma_r n)$ and $D(\widehat{\gamma}n)$ have the same sign. Hence, $D(\widetilde{\gamma}n)$, $\widetilde{\gamma}$ between γ_r and $\widehat{\gamma}$ is monotonic. We want to lower bound $\frac{|D(\gamma_r n) - D(\widehat{\gamma}n)|}{|\gamma_r n - \widehat{\gamma}n|}$. W.l.o.g we assume that $\gamma_r < \widehat{\gamma}$ and $D(\widetilde{\gamma}n)$, $\widetilde{\gamma}$ between γ_r and $\widehat{\gamma}$, is decreasing.

From (A.5) we know the expression for $D(\widetilde{\gamma}n)$ as

$$
D(\widetilde{\gamma}n) = \frac{a}{\widetilde{\gamma}} + \frac{b}{1-\widetilde{\gamma}} \qquad \text{(A.51)}
$$

for some constants a, b. The constants a and b may change over different ranges of $\widetilde{\gamma}$ between γ_r and $\widehat{\gamma}$. Consider a range of $\widetilde{\gamma}$ between γ_r and $\widehat{\gamma}$ over which a, b are

constant. Now, consider the difference $D(\gamma^1 n) - D(\gamma^2 n)$ for $\gamma^1 < \gamma^2$ belonging to that range. We will lower bound $\frac{D(\gamma^1 n)-D(\gamma^2 n)}{\gamma^2 n-\gamma^1 n}$ for different values of a, b.

- Suppose $a > 0, b > 0$. So,

$$\frac{D(\gamma^1 n) - D(\gamma^2 n)}{(\gamma^2 - \gamma^1)} = \left(\frac{a}{\gamma^1\gamma^2} - \frac{b}{(1-\gamma^2)(1-\gamma^1)}\right) \tag{A.52}$$

  Now (A.52) is a decreasing function of γ^2 since $a, b > 0$. Now $\frac{D(\gamma^1 n)-D(\gamma^2 n)}{(\gamma^2-\gamma^1)}$ is a minimum when $\gamma^2-\gamma^1$ is maximum. $\gamma^2-\gamma^1$ is maximum when $D(\gamma^1 n)-D(\gamma^2 n)$ is 2ϵ. So, $\frac{D(\gamma^1 n)-D(\gamma^2 n)}{(\gamma^2-\gamma^1)}$ is $c(\epsilon, a, b) > 0$ at minimum, where c is some constant as a function of $2\epsilon, a, b$.

- Suppose $a > 0, b < 0$.

$$\begin{aligned}
\frac{D(\gamma^1 n) - D(\gamma^2 n)}{(\gamma^2 - \gamma^1)} &= \left(\frac{a}{\gamma^1\gamma^2} - \frac{b}{(1-\gamma^2)(1-\gamma^1)}\right) && \text{(A.53)}\\
&\geq \left(\frac{a}{\gamma^1\gamma^2} + \frac{b}{(1-\gamma^2)(1-\gamma^1)}\right) && \text{(A.54)}\\
&\geq \left(\frac{a}{\gamma^1} + \frac{b}{(1-\gamma^1)}\right) && \text{(A.55)}\\
&= D(\gamma^1 n) && \text{(A.56)}
\end{aligned}$$

From the above two cases we can conclude that $\frac{D(\gamma^1 n)-D(\gamma^2 n)}{(\gamma^2-\gamma^1)} \geq \min(D(\gamma^1 n), 2\epsilon)$.

Suppose a, b change values at l different places between γ_r and $\widehat{\gamma}$. Let the points be denoted as $\gamma^1, \gamma^2, \ldots, \gamma^l$. So,

$$D(\gamma_r n) - D(\widehat{\gamma} n) \tag{A.57}$$
$$= D(\gamma_r n) - D(\gamma^1 n) + D(\gamma^1 n) - D(\gamma^2 n) \tag{A.58}$$
$$+ \ldots + D(\gamma^l n) - D(\widehat{\gamma} n) \tag{A.59}$$
$$\geq \min(D(\gamma_r n), c(\epsilon, a^1, b^1))(\gamma^1 - \gamma_r) + \ldots$$
$$+ \min(D(\gamma_l n), c(\epsilon, a^l, b^l))(\widehat{\gamma} - \gamma^l) \tag{A.60}$$
$$\geq \min(D(\gamma^l n), \min_{1<i<l} c(\epsilon, a_i, b_i))(\widehat{\gamma} - \gamma_r) \tag{A.61}$$
$$\geq \min(D(\gamma_r n) - 2\epsilon, \min_{1<i<l} c(\epsilon, a_i, b_i))(\widehat{\gamma} - \gamma_r) \tag{A.62}$$

So,

$$(\widehat{\gamma} - \gamma_r) \leq \frac{2\epsilon}{\min(D(\gamma_r n) - 2\epsilon, \min_{1<i<l} c(\epsilon, a_i, b_i))}$$

We can prove similarly when $\widehat{\gamma} < \gamma^r$.

A.9 Setup and Methodology for Experiments

Template extraction: Raw syslog data has three fields: timestamp, router id, and message text. Since the number of distinct messages are very large and many of them have common patterns, it is often useful [8, 9, 10, 5] to decompose the message text into two parts: an *invariant* part called template, and *parameters* associated with template. For example, two different messages in the log can look like:

- `Base SVCMGR-MINOR-sapCemPacketDefectAlarmClear-2212 [CEM SAP Packet Errors]: SAP 124 in service wqffv (customer 1): Alarm b́frUnderrun Ṕort 23.334 Alarm b́frUnderrun Ź2333242 ,22595400`
- `Base SVCMGR-MINOR-sapCemPacketDefectAlarmClear-2212 [CEM SAP Packet Errors]: SAP 231 in service qaazxs (customer 1): Alarm b́frUnderrun Ṕort 3322 Alarm b́frUnderrun Ź2121222 ,22595400`

Ideally, we wish to extract the following template from these identical messages:

- `Base SVCMGR-MINOR-sapCemPacketDefectAlarmClear-2212 [CEM SAP Packet Errors]: SAP * in service * (customer 1): Alarm b́frUnderrun Ṕort * Alarm b́frUnderrun ś ,22595400`

There are many existing methods to extract such templates [4, 5], ranging from tree-based methods to NLP based methods. In our work, we use an NLP based method as follows: (i) We compute the bigram probability of each word in the message corpus, (ii) next, each words above a predetermined empirical probability is declared as a word belonging to a template, (iii) each message is converted into a template by substituting the non-template-words with * as in the preceding paragraph, and (iv) finally, we assign an id to each template-router tuple in every log entry. The last step essentially combines two fields in syslog, namely text message converted to template, and source/router field. The output of this last step is treated as *message* by CD-LDA and the other algorithms. When we applied this steps to our first dataset, we extracted 39,330 distinct template-router combinations.

Note that, when alarms are reported, the template extraction stage is redundant.

Additional pre-processing: Since each event in a real-system has effects that last for several minutes to hours (even days at times), we are only interested in events at the time-scale of several minutes to an hour. Thus, in this step, we round the time-steps from $msec$ granularity to minutes (or fraction of minute) . This temporal rounding helps us to speed-up our algorithms while serving the intended practical benefit. We chose 1 minute rounding for dataset-1 and 5 minute rounding for dataset-2. Note that, upon performing temporal rounding, we do not discard duplicate messages that could result from the rounding.

A.10 The Metric in Matteson et al.

In [7] the data points lie in a continuous real space. We can still apply it to categorical data like ours if we encode a categorical data point $i \in \{1, 2, \ldots, M\}$ as a vector with all zeros except for the location i. If we use this encoding, we can show that the metric used in [7] degenerates to an unbiased estimator of the squared ℓ_2 norm. This encoding also helps us compute the metric in linear cost as opposed

to quadratic computation cost in [7]. The proof follows below.

Suppose $X_1, \ldots, X_n$ are drawn i.i.d from p and $Y_1, \ldots, Y_m$ are drawn i.i.d from q. Then [7] computes the similarity in the two distributions as,

$$\begin{aligned}
\widehat{E}(X, Y, \alpha) &= \frac{2}{mn} \sum_{i,j} |X_i - Y_j|^\alpha (\alpha \in (0, 2)) \\
&- \binom{n}{2}^{-1} \sum_{i<j} |X_i - X_j|^\alpha - \binom{m}{2}^{-1} \sum_{i<j} |Y_i - Y_j|^\alpha \\
&= \frac{2}{mn} \sum_{i,j} \mathbb{1}\{X_i \neq Y_j\} \\
&- \binom{n}{2}^{-1} \sum_{i<j} \mathbb{1}\{X_i \neq X_j\} - \binom{m}{2}^{-1} \sum_{i<j} \mathbb{1}\{Y_i \neq Y_j\} \qquad \text{(A.63)}
\end{aligned}$$

Let n_i denote the number of data points in $X_1, \ldots, X_n$ taking the value i and m_i denote the number of data points in $Y_1, \ldots, Y_m$ taking value i. One can reduce (A.63) to

$$\widehat{E}(X, Y, \alpha) = \sum_i \frac{n_i^2 - n_i}{n^2 - n} + \frac{m_i^2 - m_i}{m^2 - m} - 2\frac{n_i m_i}{nm} \qquad \text{(A.64)}$$

As $n, m \to \infty$, $\widehat{E}(X, Y, \alpha) \to \|p - q\|_2^2$. Also, $\mathbb{E}\widehat{E}(X, Y, \alpha) = \|p - q\|_2^2$. So $\widehat{E}(X, Y, \alpha)$ is both a consistent and unbiased estimator for $\|p - q\|_2^2$.

APPENDIX B

PROOFS FROM CHAPTER 3

B.1 Probability Conditions, Lemma 3.4.1

1. With probability more than $1 - \frac{\delta}{10}$, the Gaussian tail satisfies $|w_k(0)| \leq 2\kappa\sqrt{d \log\left(\frac{20md}{\delta}\right)}$ for all $k \in [m]$.

2. So $\nabla^T f(x_i)w(0) = \sum_k \frac{a_k}{\sqrt{m}}\sigma'(w_k^T x_i)w_k^T(0)x_i$ for any $w \in S_w = \{w : |w_k - w_K(0)| \leq R \forall k \in [m]\}$. We will apply Effron-Stein inequality to prove the concentration on $\sup_{w \in S_w} \nabla^T f(x_i)w(0)$ and then bound its expectation using the VC theory. Denote the random variable

$$h(a, w(0)) := \sup_{w \in S_w} \nabla^T f(x_i)w(0)$$

Consider the set of random variable $(a_k, w_k(0))$ for $k \in [m]$. We create a copy $(a_k^{(l)}, w_k^{(l)}(0)) \forall k \in [m]$ as follows. For one index l, $(a_l^{(l)}, w_l^{(l)}(0))$ is an i.i.d copy of $(a_l, w_l(0))$. For all $k \neq l$, $(a_k^{(l)}, w_k^{(l)}(0)) = (a_k, w_k(0))$. Changing one coordinate l of the random variables $(a, w(0))$ only affects change in one coordinate w_l and leads to a change in $h(\cdot)$ by

$$\begin{aligned}
&|h(a, w(0)) - h(a^{(l)}, w^{(l)}(0))|^2 \\
&\leq \frac{2}{m}(|w_l^T(0)x_i|^2 + |(w_l^{(l)}(0))^T x_i|^2)
\end{aligned}$$

Hence, by Effron-Stein inequality,

$$\begin{aligned} &Var(h(a, w(0))) \\ &\leq \mathbb{E} \sum_{l} \frac{2}{m} (|w_l^T(0) x_i|^2 + |(w_l^{(l)}(0))^T x_i|^2) = 4\kappa^2 \end{aligned}$$

as $w_l^T(0) x_i$ is distributed as $\mathcal{N}(0, \kappa^2)$.

So, applying Chebyshev's inequality,

$$\mathbb{P}(|h(a, w(0)) - \mathbb{E}h(a, w(0))| > t) \leq \frac{4\kappa^2}{t^2}$$

Now, we need to bound $\mathbb{E}h(a, w(0))$. Note that,

$$\mathbb{E}h(a, w(0)) = \mathbb{E}_{w(0)} \mathbb{E}_a \sup_{w \in S_w} \sum_k \frac{a_k}{\sqrt{m}} \sigma'(w_k^T x_i) w_k^T(0) x_i$$

We can consider $\mathbb{1}\{w_k^T x_i \geq 0\}$ as a function which classifies x_i into zero or one. Consider the class of functions $\mathcal{G}_w(x_i) = \{\mathbb{1}\{w^T x_i \leq R\}, w \in S_w\}$. $\mathcal{G}_w$ classifies a point x_i as one when $w^T x_i \geq 0$. Since $\mathcal{G}_w$ is a linear classifier, its VC dimension is $d + 1$ for data points $w_1, \ldots, w_m$. So, by VC theory,

$$\mathbb{E}h(a, w(0)) \qquad \leq \mathbb{E}_{w(0)} 4 \sqrt{d \log(m) \sum_k \frac{1}{m} |w_k^T(0) x_i|^2}$$

Further, by Jensen's inequality,

$$\mathbb{E}h(a, w(0)) \leq 4\kappa \sqrt{d \log(m)}$$

Finally, using union bound for all $i \in [n]$, w.p. more than $1 - \frac{\delta}{10}$

$$\sup_{w \in S_w} |\nabla^T f(x_i) w(0)| \leq 4\sqrt{d \log(m) \kappa^2} + \sqrt{\frac{40\kappa^2 n}{\delta}}$$

$$\sup_{w \in S_w} |\nabla^T f w(0)| \leq 4\sqrt{dn \log(m) \kappa^2} + \sqrt{\frac{40\kappa^2 n^2}{\delta}}$$

$$|f_0(x_i)| = |\nabla^T f_0(x_i) w(0)| \leq 4\sqrt{d \log(m) \kappa^2} + \sqrt{\frac{40\kappa^2 n}{\delta}}$$

$$|f_0| \leq 4\sqrt{dn \log(m) \kappa^2} + \sqrt{\frac{40\kappa^2 n^2}{\delta}}$$

3. Suppose f is computed on weights w such that $\max_k |w_k - w_k(0)| \leq R$. Consider,

$$\begin{aligned}
&\sup_{w: \forall k |w_k - w_k(0)| \leq R} |\nabla f - \nabla f_0|_F^2 \\
&= \sup_{w: \forall k |w_k - w_k(0)| \leq R} \frac{1}{m} \sum_{i,j} (\sigma'(w_i^T x_j) - \sigma'(w_i^T(0) x_j))^2 |x_j^2| \\
&\leq \frac{1}{m} \sum_{i,j} 4 \mathbb{1}\{|w_i^T(0) x_j| \leq R\}
\end{aligned}$$

We can apply McDiarmid's inequality now. See that bounded difference upon changing w_1 to its i.i.d copy w_1' is upper bounded by $\frac{8n}{m}$. Hence, by McDiarmid's inequality,

$$\begin{aligned}
&\mathbb{P}\left(\left| \frac{1}{m} \sum_{i,j} 4 \mathbb{1}\{|w_i^T(0) x_j| \leq R\} - \frac{8nR}{\sqrt{2\pi}\kappa} \right| \geq t \right) \\
&\leq 2 \exp\left(-\frac{mt^2}{32n^2} \right)
\end{aligned}$$

So w.p. greater than $1 - \frac{\delta}{10}$

$$\sup_{w:\forall k|w_k - w_k(0)| \leq R} |\nabla f - \nabla f_0|_F^2$$
$$\leq \frac{8nR}{\sqrt{2\pi}\kappa} + 4\sqrt{2}n\sqrt{\frac{\log\left(\frac{20}{\delta}\right)}{m}}$$

4. In order to find a concentration on $|\nabla^T f \nabla f - \nabla^T f_0 \nabla f_0|$ for any $\{w : |w_k - w_k(0)| \leq R \forall k \in [m]\}$, we follow a similar approach to [56, 41] simplifying the problem into concentration of each element of matrix $\nabla^T f \nabla f - \nabla^T f_0 \nabla f_0$, but we prove sub-Gaussian tail bound instead of using Markov inequality as in [56, 41]. Consider,

$$\begin{aligned}
&\sup_w |\nabla^T f \nabla f - \nabla^T f_0 \nabla f_0| \\
&\leq \sup_w |\nabla^T f \nabla f - \nabla^T f_0 \nabla f_0|_F \\
&\leq \sup_w \sum_{i,j} \frac{|x_i^T x_j|}{m} \sum_k |\sigma'(w_k^T x_i)\sigma'(w_k^T x_j) \\
&- \sigma'(w_k^T(0) x_i)\sigma'(w_k^T(0) x_j)| \\
&\leq \sum_{i,j} \frac{2}{m} \sum_k (\mathbb{1}\{|w_k^T(0) x_i| \leq R\} + \mathbb{1}\{|w_k^T(0) x_j| \leq R\})
\end{aligned}$$

Now, we can apply McDiarmid's inequality on $\sum_{i,j} \frac{2}{m} \sum_k \mathbb{1}\{|w_k^T(0) x_i| \leq R\} + \mathbb{1}\{|w_k^T(0) x_j| \leq R\}$. Changing $w_1(0)$ to its i.i.d copy $w_1'(0)$, one can bound the change in function $\sum_{i,j} \frac{2}{m} \sum_k \mathbb{1}\{|w_k^T(0) x_i| \leq R$ or $|w_k^T(0) x_j| \leq R\}$ by $\frac{8n^2}{m}$. So by McDiarmid's inequality,

$$\mathbb{P}\left(\left|\sum_{i,j} \frac{2}{m} \sum_k \mathbb{1}\{|w_k^T(0) x_i| \leq R\} + \mathbb{1}\{|w_k^T(0) x_j| \leq R\} - \frac{8n^2 R}{\kappa\sqrt{2\pi}}\right|\right) \leq 2\exp\left(\frac{-mt^2}{32n^4}\right)$$

Hence w.p. greater than $1 - \frac{\delta}{10}$,

$$\sup_{w:|w_k - w_k(0)| \leq R \forall k \in [m]} |\nabla^T f \nabla f - \nabla^T f_0 \nabla f_0|$$
$$\leq \frac{8n^2 R}{\kappa\sqrt{2\pi}} + 4\sqrt{2}n^2 \sqrt{\frac{\log\left(\frac{20}{\delta}\right)}{m}}$$

5. Let the $d-$dimensional vector $h(a_{-k})$ be defined as,

$$h(a_{-k}) := \sum_{i \neq k} \frac{a_i}{m} v_i$$

where $a_{-k} = \{a_j, j \neq k\}$ and $v_i \in \mathbb{R}^d$ is any vector that satisfies $|v_i| \leq B, \forall i \neq k$ for some constant $B > 0$. Then for all $k \in [m]$w.p. $\geq 1 - \frac{\delta}{10}$

$$|h(a_{-k})| \leq \frac{2B\sqrt{d \log\left(\frac{20md}{\delta}\right)}}{\sqrt{m}}$$

Proof: We will apply McDiarmid's inequality to $h(a_{-k})$. Since $h(a_{-k})$ is a vector we will compute the bounded difference property for each coordinate of $h(a_{-k})$. Consider a coordinate vector $e_b, b \in [d]$. Suppose we change one of a_l to its i.i.d copy a'_l for some $l \neq k$, then the maximum change in $\langle h(a_{-k}) - h(a'_{-k}), e_m \rangle$ can be computed as,

$$\sup_{a_l, a'_l} |\langle h(a_{-k}) - h(a'_{-k}), e_b \rangle|$$
$$\leq \sup_{a_l, a'_l} |h(a_{-k}) - h(a'_{-k})|$$
$$= |a_l - a'_l| |\frac{1}{m} v_l|$$
$$\leq \frac{2}{m} B$$

Note that $\mathbb{E}_{a_{-k}} h(a_{-k}) = 0$. So, by McDiarmid's inequality,

$$\mathbb{P}(|\langle h(a_{-k}), e_b \rangle| \geq t) \leq 2 \exp\left(\frac{-mt^2}{2B^2}\right)$$

So, for all $k \in [m], b \in [d]$, by using union bound, with probability greater than $1 - \frac{\delta}{10}$,

$$|\langle h(a_{-k}), e_b \rangle| \leq \frac{B\sqrt{2 \log\left(\frac{20md}{\delta}\right)}}{\sqrt{m}}$$

For the entire vector $h(a_{-k})$,

$$|h(a_{-k})| \leq \sqrt{d} \sup_{b \in [d]} |\langle h(a_{-k}), e_b \rangle| \leq \frac{B\sqrt{2d \log\left(\frac{20md}{\delta}\right)}}{\sqrt{m}}$$

6. With probability greater than $1 - \frac{\delta}{10}$, $|\nabla^T f_0 \nabla f_0 - H| \leq \sqrt{\frac{32n^2}{m} \log\left(\frac{20n}{\delta}\right)}$. We apply matrix McDiarmid's inequality from [78] to prove this result. Consider the matrix $\nabla^T f_0 \nabla f_0 - H$ as sum of m i.i.d self ad-joint matrices $\sum_k \nabla_k^T f_0 \nabla_k f_0 - H_k$. Now, changing, say, $w_1(0)$ to its i.i.d copy $w_1'(0)$ leads to a maximum difference,

$$|\nabla_1^T f_0 \nabla_1 f_0 - \nabla_1^T f_0' \nabla_1 f_0'|^2 \leq \frac{4n^2}{m^2}$$

Hence the variance proxy for matrix McDiarmid's is $|\sum_k \frac{4n^2}{m^2}| = \frac{4n^2}{m}$. So,

$$\mathbb{P}(|\nabla^T f_0 \nabla f_0 - H| > t) \leq 2n \exp\left(\frac{-mt^2}{32n^2}\right)$$

Hence, with probability greater than $1 - \frac{\delta}{10}$,

$$|\nabla^T f_0 \nabla f_0 - H| \leq \sqrt{\frac{32n^2}{m} \log\left(\frac{20n}{\delta}\right)}$$

$$\lambda_{\min}(\nabla^T f_0 \nabla f_0 + \lambda) \geq \lambda + \max\left(\lambda_0 - \sqrt{\frac{32n^2}{m} \log\left(\frac{20n}{\delta}\right)}, 0\right)$$

$$|(\nabla^T f_0 \nabla f_0 + \lambda)^{-1}| \leq \frac{1}{\lambda + \max\left(\lambda_0 - \sqrt{\frac{32n^2}{m} \log\left(\frac{20n}{\delta}\right)}, 0\right)}$$

7. We will use McDiarmid's inequality now to show that $\sum_k \frac{2R}{\sqrt{m}} \mathbb{1}\{|w_k^T(0)x_i| \leq R\}$ is close to its expectation $\mathbb{E}\sum_k \frac{2R}{\sqrt{m}} \mathbb{1}\{|w_k^T(0)x_i| \leq R\}$ uniformly over $|x_i| \leq 1$. Also, for this concentration we are working in the law of large number $\frac{1}{m}$ regime instead of the $\frac{1}{\sqrt{m}}$ regime. Changing w_l to w_l' for some $l \in [m]$, the maximum change in function value is given by $\frac{4R}{\sqrt{m}}$. Hence, by McDiarmid's inequality,

$$\mathbb{P}\left(\sup_{|x_i|\leq 1} \sum_k \frac{2R}{\sqrt{m}} \mathbb{1}\{|w_k^T(0)x_i| \leq R\} \right.$$
$$\left. -\mathbb{E}\sup_{|x_i|\leq 1} \sum_k \frac{2R}{\sqrt{m}} \mathbb{1}\{|w_k^T(0)x_i| \leq R\} > t\right)$$
$$\leq \exp\left(\frac{-t^2}{32R^2}\right)$$

With probability more than $1 - \frac{\delta}{10}$,

$$\sup_{|x_i|\leq 1} \sum_k \frac{2R}{\sqrt{m}} \mathbb{1}\{|w_k^T(0)x_i| \leq R\}$$
$$\leq \mathbb{E}\sup_{|x_i|\leq 1} \sum_k \frac{2R}{\sqrt{m}} \mathbb{1}\{|w_k^T(0)x_i| \leq R\}$$
$$+ \sqrt{32R^2 \log\left(\frac{10}{\delta}\right)}$$

Also,

$$\begin{aligned}
&\mathbb{E} \sup_{|x_i|\leq 1} \sum_k \frac{2R}{\sqrt{m}} \mathbb{1}\{|w_k^T(0)x_i| \leq R\} \\
&\leq \mathbb{E} \sup_{|x_i|\leq 1} \sum_k \frac{2R}{\sqrt{m}} \big(\mathbb{1}\{|w_k^T(0)x_i| \leq R\} \\
&- \mathbb{P}(|w_k^T(0)x_i| \leq R)\big) \\
&+ \sup_{|x_i|\leq 1} \frac{2R}{\sqrt{m}} \sum_k \mathbb{P}(|w_k^T(0)x_i| \leq R)
\end{aligned}$$

We can use symmetrization for the first term and a trivial bound on the probability for the second term. With ϵ_k as i.i.d Rademacher random variable,

$$\begin{aligned}
&\mathbb{E} \sup_{|x_i|\leq 1} \sum_k \frac{2R}{\sqrt{m}} \mathbb{1}\{|w_k^T(0)x_i| \leq R\} \\
&\leq \mathbb{E} \sup_{|x_i|\leq 1} \sum_k \frac{4R}{\sqrt{m}} \epsilon_k (\mathbb{1}\{|w_k^T(0)x_i| \leq R\}) \\
&+ \frac{2R^2\sqrt{m}}{\kappa\sqrt{2\pi}}
\end{aligned}$$

Further, we can break the Rademacher average term into two,

$$\begin{aligned}
&\mathbb{E} \sup_{|x_i|\leq 1} \sum_k \frac{4R}{\sqrt{m}} \epsilon_k (\mathbb{1}\{|w_k^T(0)x_i| \leq R\}) \\
&= \mathbb{E} \sup_{|x_i|\leq 1} \sum_k \frac{4R}{\sqrt{m}} \epsilon_k (\mathbb{1}\{w_k^T(0)x_i \leq R\} \\
&\qquad\qquad - \mathbb{1}\{w_k^T(0)x_i \leq -R\}) \\
&\leq \mathbb{E} \sup_{|x_i|\leq 1} \sum_k \frac{4R}{\sqrt{m}} \epsilon_k (\mathbb{1}\{w_k^T(0)x_i \leq R\}) \\
&+ \mathbb{E} \sup_{|x_i|\leq 1} \sum_k \frac{4R}{\sqrt{m}} \epsilon_k (\mathbb{1}\{w_k^T(0)x_i \leq -R\})
\end{aligned}$$

We can bound each of the Rademacher average using VC theory. We can consider $\mathbb{1}\{w_k^T(0)x_i \leq R\}$ as a function which classifies $w_k(0)$ into zero or 1. Consider the class of functions $\mathcal{G}_x(w) = \{\mathbb{1}\{w^T x \leq R\}, |x| \leq 1\}$. $\mathcal{G}_x$ classifies a point w as one when $w^T x \leq R$. Since $\mathcal{G}_x$ is a linear classifier, its VC dimension is $d+1$. So, by VC theory,

$$\begin{aligned}&\mathbb{E} \sup_{|x_i|\leq 1} \sum_k \frac{4R}{\sqrt{m}} \epsilon_k \mathbb{1}\{|w_k^T(0)x_i| \leq R\} \\ &\leq 32R\sqrt{d\log(m+1)}\end{aligned}$$

Hence, w.p. more than $1 - \frac{\delta}{10}$ and $m > 10$,

$$\begin{aligned}&\sup_{|x_i|\leq 1} \sum_k \frac{2R}{\sqrt{m}} \mathbb{1}\{|w_k^T(0)x_i| \leq R\} \\ &\leq 32R\sqrt{d\log(m+1)} + \frac{2R^2\sqrt{m}}{\kappa\sqrt{2\pi}} + \sqrt{32R^2 \log\left(\frac{10}{\delta}\right)}\end{aligned}$$

8. Suppose the random variable

$$h(a, w(0)) := \sup_{|x|\leq 1} \sum_k \frac{a_k}{\sqrt{m}} \sigma(w_k^T(0)x)) = f_0(x)$$

Consider the set of random variable (a_k, w_k) for $k \in [m]$. We create a copy $(a_k^{(l)}, w_k^{(l)}) \forall k \in [m]$ as follows. For one index l, $(a_l^{(l)}, w_l^{(l)})$ is an i.i.d copy of (a_l, w_l). For all $k \neq l$, $(a_k^{(l)}, w_k^{(l)}) = (a_k, w_k)$. So, changing one coordinate l leads to a change in $h(.)$ by

$$|h(a, w) - h(a^{(l)}, w^{(l)})|^2 \leq \frac{2(|w_l|^2 + |w_l'|^2)}{m}$$

By Efron–Stein inequality,

$$\begin{aligned}
&Var(\sup_{|x|\leq 1} \sum_k \frac{a_k}{\sqrt{m}} \sigma(w_k^T(0)x)) \\
&\leq \mathbb{E} \sum_i |h(a,w) - h(a^{(i)}, w^{(i)})|^2 \\
&\leq \mathbb{E} \sum_i \frac{\mathbb{E}2(|w_i|^2 + |w_i'|^2)}{m} \leq 4d\kappa^2
\end{aligned}$$

Hence by Chebyshev's inequality, with probability more than $1 - \frac{\delta}{10}$

$$\begin{aligned}
&\sup_{|x|\leq 1} \sum_k \frac{a_k}{\sqrt{m}} \sigma(w_k^T(0)x) \\
&\leq \mathbb{E} \sup_{|x|\leq 1} \sum_k \frac{a_k}{\sqrt{m}} \sigma(w_k^T(0)x) + \sqrt{\frac{40d\kappa^2}{\delta}}
\end{aligned}$$

9. We have

$$|\nabla f_0(x_i)|^2 = \frac{1}{m} \sum_k \sigma'(w_k^T(0)x_i)|x_i|^2$$

We will use McDiarmid's inequality on $|\nabla f_0(x_i)|^2$. Upon changing one coordinate w_k to w_k' while keeping others the same, the maximum change in $|\nabla f_0(x_i)|^2$ is $\frac{2}{m}$. So, by McDiarmid's inequality,

$$\mathbb{P}\left(||\nabla f_0(x_i)|^2 - \mathbb{E}|\nabla f_0(x_i)|^2|\right) \leq 2\exp\left(-\frac{mt^2}{8}\right)$$

The mean $\mathbb{E}|\nabla f_0(x_i)|^2 = 0.5$. Hence, w.p. more than $1 - \frac{\delta}{10}$, $\forall i \in [n]$,

$$|\nabla f_0(x_i)|^2 \leq \frac{1}{2} + \sqrt{\frac{1}{m}\log\left(\frac{20n}{\delta}\right)}$$

10. We will use the generalization result from Theorem 11.3 from [57]. Suppose

$$\sup_{|y|\leq 1,|x|\leq 1} |y - f(x)| \leq M$$

Since the loss function $|y - f(x)|$ is $1-$Lipschitz, with probability more than $1 - \frac{\delta}{10}$ over the i.i.d. sampled data points $x_i, y_i i \in [n]$,

$$\mathbb{E}_{x,y}|y - f(x)| \leq \frac{1}{n}\sum_i |y_i - f(x_i)| + 2Rad(\mathcal{F}_w) + 3M\sqrt{\frac{\log\left(\frac{20}{\delta}\right)}{2n}}$$

For the next sections from B.2 to B.12, we will use the following bounds:

1. Expression $|\nabla_k f_0(\nabla^T f_0 \nabla f_0 + \lambda)^{-1} y|$ can be bound by $\frac{2|\nabla_k f_0|\sqrt{n}}{(\lambda+\lambda_0)}$ using condition (6). Further

$$|\nabla_k f_0|^2 = \sum_{i=1}^{n} \frac{1}{m}\sigma'(w_k^T(0)x_i)|x_i|^2 \leq \frac{n}{m}$$

So, $|\nabla_k f_0(\nabla^T f_0 \nabla f_0 + \lambda)^{-1} y| \leq \frac{2n}{\sqrt{m}(\lambda+\lambda_0)}$.

2. Now, $|w_k(\infty)| \leq |w_k(0)| + |\nabla_k f_0(\nabla^T f_0 \nabla f_0 + \lambda)^{-1} y| \leq 2\kappa\sqrt{d\log\left(\frac{20md}{\delta}\right)} + \frac{2n}{\sqrt{m}(\lambda+\lambda_0)}$ using condition (1) and (6).

B.2 Bounding err_k

In this section we will bound err_k by $O(\frac{1}{\sqrt{m}})$. We can expand err_k into the terms as below and apply triangle inequality to each.

$$\begin{aligned} err_k = &(\nabla_k f - \nabla_k f_0)y \\ &+ (\nabla_k f_0 \nabla_k^T f_0 - \nabla_k f \nabla_k^T f)(w_k - w_k(\infty)) \\ &+ (\nabla_k f_0 \nabla_k^T f_0 - \nabla_k f \nabla_k^T f) w_k(\infty) \\ &- \sum_{i \neq k} \nabla_k f \nabla_i^T f (w_i - w_i(\infty)) \\ &+ \sum_{i \neq k} (\nabla_k f_0 \nabla_i^T f_0 - \nabla_k f \nabla_i^T f) w_i(\infty) \\ &- \nabla_k f_0 f_0 \end{aligned} \tag{B.1}$$

We bound terms in err_k from (B.1) below.

- Expression $|(\nabla_k f - \nabla_k f_0)y| = |\sum_{i=1}^n \frac{a_k x_i y_i}{\sqrt{m}}(\sigma'(w_k^T x_i) - \sigma'(w_k^T(0)x_i))|$ is upper bounded by $\frac{2n}{\sqrt{m}}$.
- We have

$$\begin{aligned} &|(\nabla_k f_0 \nabla_k^T f_0 - \nabla_k f \nabla_k^T f)(w_k - w_k(\infty))| \\ &= |\sum_{i=1}^n \frac{1}{m}((\sigma'(w_k^T(0)x_i))^2 - (\sigma'(w_k^T x_i))^2) x_i x_i^T || w_k - w_k(\infty))| \\ &\leq \frac{2n}{m} |w_k - w_k(\infty)| \end{aligned}$$

- Using bound (2), $|(\nabla_k f_0 \nabla_k^T f_0 - \nabla_k f \nabla_k^T f) w_k(\infty)| \leq \frac{2n}{m} |w_k(\infty)| \leq \frac{4n}{m} \kappa \sqrt{d \log\left(\frac{20md}{\delta}\right)} + \frac{4n^2}{m\sqrt{m}(\lambda + \lambda_0)}$.

- Expanding the term $|\sum_{i\neq k}\nabla_k f\nabla_i^T f(w_i - w_i(\infty))|$,

$$
\begin{aligned}
&|\sum_{i\neq k}\nabla_k f\nabla_i^T f(w_i - w_i(\infty))| \\
&= |\sum_{i\neq k}\frac{a_i}{m}\sum_j \sigma'(w_k^T x_j)\sigma'(w_i^T x_j)x_j x_j^T(w_i - w_i(\infty))|
\end{aligned}
$$

Since $\max_i |\sum_j \sigma'(w_k^T x_j)\sigma'(w_i^T x_j)x_j x_j^T(w_i - w_i(\infty))| \leq n\max_i |w_i - w_i(\infty)| \leq nR$, we can use Lemma 5 to get a bound,

$$
\begin{aligned}
&|\sum_{i\neq k}\nabla_k f\nabla_i^T f(w_i - w_i(\infty)| \\
&\leq \frac{2n\max_i |w_i - w_i(\infty)|\sqrt{d\log\left(\frac{20md}{\delta}\right)}}{\sqrt{m}}
\end{aligned}
$$

- Expanding the term $|\sum_{i\neq k}(\nabla_k f_0\nabla_i^T f_0 - \nabla_k f\nabla_i^T f)w_i(\infty)|$,

$$
\begin{aligned}
&|\sum_{i\neq k}(\nabla_k f_0\nabla_i^T f_0 - \nabla_k f\nabla_i^T f)w_i(\infty)| \\
&= |\sum_{i\neq k}\frac{a_i}{m}\sum_j \big(\sigma'(w_k^T(0)x_j)\sigma'(w_i^T(0)x_j) \\
&-\sigma'(w_k^T x_j)\sigma'(w_i^T x_j)\big)\, x_j x_j^T w_i(\infty)|
\end{aligned}
$$

We have

$$
\begin{aligned}
&\max_i|\sum_j \big(\sigma'(w_k^T(0)x_j)\sigma'(w_i^T(0)x_j) \\
&\quad -\sigma'(w_k^T x_j)\sigma'(w_i^T x_j)\big)\, x_j x_j^T w_i(\infty)| \\
&\quad \leq 2n\max_i |w_i(\infty)| \qquad \text{(B.2)}
\end{aligned}
$$

We can use Lemma 5, bound (2) and (B.2) to get a bound,

$$
\begin{aligned}
&|\sum_{i\neq k}(\nabla_k f_0 \nabla_i^T f_0 - \nabla_k f \nabla_i^T f) w_i(\infty)| \\
&\leq \frac{4n \max_i |w_i(\infty)| \sqrt{d \log\left(\frac{20md}{\delta}\right)}}{\sqrt{m}} \\
&\leq \frac{8n\kappa d \log\left(\frac{20md}{\delta}\right)}{\sqrt{m}} + \frac{8n^2 \sqrt{d \log\left(\frac{20md}{\delta}\right)}}{m(\lambda+\lambda_0)}
\end{aligned}
$$

- Using condition (2), $|\nabla_k f_0 f_0| = |\sum_i \frac{a_k x_i f_0(x_i)}{\sqrt{m}} \sigma'(w_k^T(0) x_i)| \leq \frac{\max_i |f_0(x_i)|}{\sqrt{m}} \leq \frac{4\kappa}{\sqrt{m\delta}}$.

Combining the results above we can determine,

$$
\begin{aligned}
|err_k| = &\frac{2n}{\sqrt{m}} \\
&+ \frac{2n}{m}|w_k - w_k(\infty)| \\
&+ \frac{4n}{m}\kappa\sqrt{d\log\left(\frac{20md}{\delta}\right)} + \frac{4n^2}{m\sqrt{m}\,(\lambda+\lambda_0)} \\
&+ \frac{2n \max_i |w_i - w_i(\infty)| \sqrt{d \log\left(\frac{20md}{\delta}\right)}}{\sqrt{m}} \\
&+ \frac{8n\kappa d \log\left(\frac{20md}{\delta}\right)}{\sqrt{m}} + \frac{8n^2\sqrt{d\log\left(\frac{20md}{\delta}\right)}}{m(\lambda+\lambda_0)} \\
&+ \frac{4\kappa}{\sqrt{m\delta}} \qquad \text{(B.3)} \\
&\leq \frac{80nd \log\left(\frac{20md}{\delta}\right)(1 + \max_{i\neq k}|w_i - w_i(\infty)|)}{\sqrt{m\delta}} \\
&+ \frac{40n\sqrt{\log\left(\frac{20md}{\delta}\right)}(1 + |w_k - w_k(\infty)|)}{m} \\
&\leq \frac{120nd \log\left(\frac{20md}{\delta}\right)(1 + \max_{i\in[m]}|w_i - w_i(\infty)|)}{\sqrt{m\delta}} \qquad \text{(B.4)}
\end{aligned}
$$

B.3 Proof of Lemma 3.4.2

Suppose, at time $t = 0$ we have $|w_k(0) - w_k(\infty)| \leq |\nabla_k f_0(\nabla^T f_0 \nabla f_0 + \lambda)^{-1} y| + O(\frac{1}{\sqrt{m}}) \leq \frac{c_n}{\sqrt{m}}$ where c_n is $O(n)$. Let T be the first time that $w_k(t)$ exists the ball $2|w_k(0) - w_k(\infty)| + O(\frac{1}{\sqrt{m}}) \leq 2|\nabla_k f_0(\nabla^T f_0 \nabla f_0 + \lambda)^{-1} y| + O(\frac{1}{\sqrt{m}}) \leq \frac{2c_n}{\sqrt{m}}$. Consider the Lyapunov function $V(w_k) = |w_k(t) - w_k(\infty)|^2$. This satisfies,

$$\dot{V} \leq -\lambda V + |err_k| \sqrt{V}$$

Since $|err_k| \leq O(\frac{1}{\sqrt{m}})$ from Appendix B.2, we have,

$$\dot{\sqrt{V}} \leq -\lambda \sqrt{V} + O(\frac{1}{\sqrt{m}})$$

Solving this equation for all $t \leq T$,

$$|w_k(t) - w_k(\infty)| \leq |w_k(0) - w_k(\infty)| + O(\frac{1}{\lambda \sqrt{m}})$$

Hence, at time $t = T$, $|w_k(t) - w_k(\infty)|$ is strictly inside the ball $\frac{2c_n}{\sqrt{m}}$. So $T = \infty$. This shows that,

$$|w_k(t) - w_k(\infty)| \leq 2|w_k(0) - w_k(\infty)| + O(\frac{1}{\sqrt{m}}) \forall t > 0$$

By applying triangle inequality, we get a bound on $|w_k(t) - w_k(0)|$ as,

$$|w_k(t) - w_k(0)| \leq 3|w_k(0) - w_k(\infty)| + O(\frac{1}{\sqrt{m}}) \forall t > 0$$

Using bound (1), we arrive at the result.

B.4 Bound on err_f

- Using condition (4) and (6),

$$\begin{aligned}
&|\nabla^T f \nabla f - H| \\
&\leq |\nabla^T f \nabla f - \nabla^T f_0 \nabla f_0| + |\nabla^T f_0 \nabla f_0 - H| \\
&\leq \frac{8n^2 R}{\kappa\sqrt{2\pi}} + 4\sqrt{2}n^2 \sqrt{\frac{\log\left(\frac{20}{\delta}\right)}{m}} + \sqrt{\frac{32n^2}{m}\log\left(\frac{20n}{\delta}\right)} \\
&\leq \frac{8n^2 R}{\kappa\sqrt{2\pi}} + \sqrt{\frac{128n^4}{m}\log\left(\frac{20n}{\delta}\right)}
\end{aligned}$$

- By triangle inequality,

$$\begin{aligned}
|f - y| &\leq |f - f_\infty| + |\lambda(H + \lambda)^{-1} y| \\
&\leq |f - f_\infty| + \frac{\lambda\sqrt{n}}{\lambda + \lambda_0}
\end{aligned}$$

- Also by condition (2),

$$|\nabla^T f w(0)| \leq \sqrt{2n}\kappa \log\left(\frac{20n}{\delta}\right)$$

Now, we can easily bound the error term err_f by applying triangle inequality,

$$\begin{aligned}
&|err_f| \leq |\lambda \nabla^T f w(0)| + |(\nabla^T f \nabla f - H||f - y| \\
&\leq \sqrt{2n}\lambda\kappa \log\left(\frac{20n}{\delta}\right) + \frac{8n^2 R}{\kappa\sqrt{2\pi}}(|f - f_\infty| + \frac{\lambda\sqrt{n}}{\lambda + \lambda_0}) \\
&+ \sqrt{\frac{128n^4}{m}\log\left(\frac{20n}{\delta}\right)}(|f - f_\infty| + \frac{\lambda\sqrt{n}}{\lambda + \lambda_0})
\end{aligned}$$

B.5 Proof of Theorem 3.4.3

One can choose a Lyapunov function $V(f) = |f - f_\infty|^2$. From the dynamics of $\dot{f}$ in (3.13), we can compute the dynamics of V as,

$$\dot{V} \leq -(\lambda + \lambda_0)V + |err_f|\sqrt{V}$$
$$\dot{\sqrt{V}} \leq -(\lambda + \lambda_0)\sqrt{V} + O(\frac{1}{\sqrt{m}})\sqrt{V} + O(\frac{1}{\sqrt{n}})$$

when $m \geq poly(n, \frac{1}{\delta}, \frac{1}{\lambda})$. Solving this,

$$|f(t) - f_\infty| \leq \exp\left(-\left((\lambda + \lambda_0) - O(m^{-1/2})\right)t\right)|f_0 - f_\infty| + \frac{O(n^{-0.5})}{\lambda + \lambda_0 - O(m^{-0.5})}$$

So the training error is bounded as,

$$\begin{aligned}|f(t) - y| &\leq \exp\left(-((\lambda + \lambda_0) - O(m^{-0.5}))t\right)|f_0 - f_\infty| \\ &\quad + \frac{O(n^{-0.5})}{\lambda + \lambda_0 - O(m^{-0.5})} + |y - f_\infty| \\ &\leq \exp\left(-((\lambda + \lambda_0) - O(m^{-0.5}))t\right)|f_0 - f_\infty| \\ &\quad + \frac{O(n^{-0.5})}{\lambda + \lambda_0 - O(m^{-0.5})} + \lambda|(H + \lambda)^{-1}y|\end{aligned}$$

If $\lambda + \lambda_0 > O(m^{-0.5})$, then at steady state $t = \infty$ the training error is

$$\frac{1}{n}|f(\infty) - y|^2 \leq \frac{\lambda^2|(H + \lambda)^{-1}y|^2}{n} + O(n^{-0.5})$$

B.6 Bound on err

First, applying Matrix Woodbury identity on err simplifies it to

$$\begin{aligned} err = \nabla f \nabla^T f w(0) + (\nabla f \nabla^T f + \lambda)(\nabla f (\nabla^T f \nabla f + \lambda)^{-1} \\ - \nabla f_0 (\nabla^T f_0 \nabla f_0 + \lambda)^{-1}) y \end{aligned}$$

Again, breaking up the terms and applying Matrix Woodbury identity,

$$\begin{aligned} & err = \nabla f \nabla^T f w(0) + (\nabla f \nabla^T f + \lambda) \\ & \big[(\nabla f - \nabla f_0)(\nabla^T f \nabla f + \lambda)^{-1} \\ & + \nabla f_0 (\nabla^T f_0 \nabla f_0 + \lambda)^{-1} \\ & (\nabla^T f_0 \nabla f_0 - \nabla^T f \nabla f)(\nabla^T f \nabla f + \lambda)^{-1} \big] \, y \end{aligned}$$

We bound each term in RHS of err as below.

- $|\nabla f_0| \leq \sqrt{n}$.
- $|\nabla f \nabla^T f w(0)| \leq \sqrt{n} |\nabla^T f w(0)| \leq n\kappa \sqrt{2 \log\left(\frac{20n}{\delta}\right)}$ where we use condition (2).
- $|(\nabla f \nabla^T f + \lambda)| \leq n + \lambda$.
- $|(\nabla f \nabla^T f + \lambda)^{-1}| \leq \frac{1}{\lambda}$.
- From condition (3) $|\nabla f - \nabla f_0| \leq \sqrt{\frac{8nR}{\sqrt{2\pi}\kappa}} + \sqrt[1/4]{\frac{32n^2 \log\left(\frac{20}{\delta}\right)}{m}}$.
- From condition (4) $|\nabla^T f_0 \nabla f_0 - \nabla^T f \nabla f| \leq \frac{8n^2 R}{\kappa\sqrt{2\pi}} + 4\sqrt{2} n^2 \sqrt{\frac{\log\left(\frac{20}{\delta}\right)}{m}}$.

By triangle inequality we can bound $|err|$ as,

$$\begin{aligned}
|err| &\leq n\kappa\sqrt{2\log\left(\frac{20n}{\delta}\right)} \\
&+ \frac{\sqrt{n}(n+\lambda)}{\lambda}\left(\sqrt{\frac{8nR}{\sqrt{2\pi}\kappa}} + \sqrt[1/4]{\frac{32n^2\log\left(\frac{20}{\delta}\right)}{m}}\right) \\
&+ \frac{2n(n+\lambda)}{\lambda(\lambda+\lambda_0)}\left(\frac{8n^2R}{\kappa\sqrt{2\pi}} + 4\sqrt{2}n^2\sqrt{\frac{\log\left(\frac{20}{\delta}\right)}{m}}\right) \\
&\leq o(1) \text{ when } \kappa = \frac{1}{n^{1+\epsilon}}, \forall\epsilon > 0
\end{aligned}$$

B.7 Proof of Lemma 3.4.4

Now we can analyze (3.14). Choose $V(w) = |w - w(\infty)|^2$. So,

$$\begin{aligned}
\dot{V} &\leq -\lambda V + |err|\sqrt{V} \\
\dot{\sqrt{V}} &\leq -\lambda\sqrt{V} + o(1)
\end{aligned}$$

Therefore, one can solve the above equation to get a bound,

$$|w(t) - w(\infty)| \leq e^{-\lambda t}|w(0) - w(\infty)| + \frac{o(1)}{\lambda}$$

which gives us a bound on $|w(t) - w(0)|, \forall t > 0$

$$|w(t) - w(0)| \leq 2|w(0) - w(\infty)| + o(1)$$

Also,

$$\begin{aligned}
|w(0) - w(\infty)|^2 &= y^T(\nabla^T f_0 \nabla f_0 + \lambda)^{-1} y \\
&- \lambda y^T(\nabla^T f_0 \nabla f_0 + \lambda)^{-2} y \\
&= y^T(H+\lambda)^{-1} y - \lambda y^T(H+\lambda)^{-2} y + err_w
\end{aligned}$$

where we define

$$\begin{aligned} err_w := y^T((\nabla^T f_0 \nabla f_0 + \lambda)^{-1} - (H + \lambda)^{-1})y \\ - \lambda y^T((\nabla^T f_0 \nabla f_0 + \lambda)^{-2} - (H + \lambda)^{-2})y \end{aligned}$$

We bound err_w in Appendix B.8. Hence, we can finally conclude that,

$$\begin{aligned} |w(t) - w(0)| &\leq 2\sqrt{y^T(H + \lambda)^{-1}H(H + \lambda)^{-1}y} + o(1) \\ |w(\infty) - w(0)| &\leq \sqrt{y^T(H + \lambda)^{-1}H(H + \lambda)^{-1}y} + o(1) \end{aligned}$$

B.8 Bound on err_w

- By Matrix Woodbury identity,

$$\begin{aligned} &|(\nabla^T f_0 \nabla f_0 + \lambda)^{-1} - (H + \lambda)^{-1}| \\ &= |(H + \lambda)^{-1}(\nabla^T f_0 \nabla f_0 - H)(\nabla^T f_0 \nabla f_0 + \lambda)^{-1}| \end{aligned}$$

  We can use triangle inequality in conjunction with condition (6) and (6) to have,

$$\begin{aligned} &|(\nabla^T f_0 \nabla f_0 + \lambda)^{-1} - H^{-1}| \\ &\leq \frac{2}{(\lambda + \lambda_0)^2}\sqrt{\frac{32n^2}{m}\log\left(\frac{20n}{\delta}\right)} \end{aligned}$$

- Similarly, we can apply Matrix Woodbury identity along with triangle inequality to have

$$\begin{aligned} &|(\nabla^T f_0 \nabla f_0 + \lambda)^{-2} - H^{-2}| \\ &\leq |(H + \lambda)^{-2}||\nabla^T f_0 \nabla f_0 - H|^2|(\nabla^T f_0 \nabla f_0 + \lambda)^{-2}| \\ &+ 2|(H + \lambda)^{-2}||\nabla^T f_0 \nabla f_0 - H||(\nabla^T f_0 \nabla f_0 + \lambda)^{-1}| \end{aligned}$$

Further, condition (6) gives us,

$$|(\nabla^T f_0 \nabla f_0 + \lambda)^{-2} - H^{-2}| \leq \frac{4|\nabla^T f_0 \nabla f_0 - H|^2}{(\lambda + \lambda_0)^4} + \frac{4|\nabla^T f_0 \nabla f_0 - H|}{(\lambda + \lambda_0)^3}$$

Finally applying condition (6) we get,

$$|(\nabla^T f_0 \nabla f_0 + \lambda)^{-2} - H^{-2}| \leq \frac{128n^2}{m(\lambda + \lambda_0)^4} \log\left(\frac{20n}{\delta}\right) + \frac{4}{(\lambda + \lambda_0)^3}\sqrt{\frac{32n^2}{m}\log\left(\frac{20n}{\delta}\right)}$$

Combining the results from above, we can bound $|err_w|$ as,

$$\begin{aligned} |err_w| &\leq n|(\nabla^T f_0 \nabla f_0 + \lambda)^{-1} - (H + \lambda)^{-1}| \\ &\quad + n\lambda|(\nabla^T f_0 \nabla f_0 + \lambda)^{-2} - H^{-2}| \\ &\leq \frac{2n}{(\lambda + \lambda_0)^2}\sqrt{\frac{32n^2}{m}\log\left(\frac{20n}{\delta}\right)} \\ &\quad + \frac{128n^3\lambda}{m(\lambda + \lambda_0)^4}\log\left(\frac{20n}{\delta}\right) \\ &\quad + \frac{4n\lambda}{(\lambda + \lambda_0)^3}\sqrt{\frac{32n^2}{m}\log\left(\frac{20n}{\delta}\right)} \\ &= o(1) \end{aligned}$$

when $m \geq poly\left(n, \log\left(\frac{1}{\delta}\right), \lambda, \frac{1}{\lambda+\lambda_0}\right)$.

B.9 Proof of Lemma 3.4.5

We can write $f(x_i) = \nabla^T f(x_i)(w - w(0)) + \nabla^T f(x_i) w(0)$. So,

$$Rad(\mathcal{F}_w) = \frac{1}{n}\mathbb{E}_\epsilon \sup_{w \in \mathcal{F}_w} \left(\left(\sum_i \epsilon_i \nabla f_0(x_i) \right)^T (w - w(0)) \right.$$
$$+ \left(\sum_i \epsilon_i (\nabla f(x_i) - \nabla f_0(x_i)) \right)^T (w - w(0))$$
$$\left. + \sum_i \epsilon_i \nabla^T f(x_i) w(0) \right)$$

Using $|w - w(0)| \leq B$, the Cauchy-Schwartz inequality and the triangle inequality,

$$Rad(\mathcal{F}_w) \leq \frac{1}{n}\mathbb{E}_\epsilon \left(B |\sum_i \epsilon_i \nabla f_0(x_i)| \right)$$
$$+ \frac{B}{\sqrt{n}} \sup_{w \in \mathcal{F}_w} \sqrt{\sum_i |\nabla f(x_i) - \nabla f_0(x_i)|^2}$$
$$+ \sup_{w \in \mathcal{F}_w} \max_i |\nabla^T f(x_i) w(0)|$$

Further using condition (2), condition (3) and Jensen's inequality,

$$Rad(\mathcal{F}_w) \leq \frac{B}{n} \sqrt{\mathbb{E}_\epsilon |\sum_i \epsilon_i \nabla f_0(x_i)|^2}$$
$$+ B\sqrt{\frac{8R}{\sqrt{2\pi}\kappa}} + B \left(\frac{32 \log\left(\frac{20}{\delta}\right)}{m} \right)^{1/4} + o(1)$$

Also, using condition (9), $\mathbb{E}_\epsilon|\sum_i \epsilon_i \nabla f(x_i)|^2 = \sum_i |\nabla f_0(x_i)|^2 \leq \frac{n}{2} + \sqrt{\frac{n^2}{m}\log\left(\frac{20n}{\delta}\right)}$. Hence,

$$Rad(\mathcal{F}_w) \leq \frac{B}{\sqrt{2n}} + B\sqrt{\frac{1}{m}\log\left(\frac{20n}{\delta}\right)} + B\sqrt{\frac{8R}{\sqrt{2\pi}\kappa}} + B\left(\frac{32\log\left(\frac{20}{\delta}\right)}{m}\right)^{1/4} + o(1)$$

The result follows as $\kappa = \frac{1}{n^{1+\epsilon}}$ for any $\epsilon > 0$.

B.10 Proof of Theorem 3.4.6

In order to relate the test error with the Rademacher complexity result from Lemma 3.4.5, we can use condition (10). But, we will need to compute an upper bound M to the loss function $\sup_{|y|\leq 1,|x|\leq 1}|y - f(x)|$. We prove an upper bound on M in Appendix B.11,

$$M := \sup_{|y_i|\leq 1,|x_i|\leq 1} |y_i - f(x_i)| \leq o(n^{1/2-\epsilon/2}) + o(1)$$

Using this value of M the theorem statement follows.

B.11 Bounding M

Suppose w is such that $|w - w(0)| \leq B$ where B is given Lemma 3.4.5. The loss function $|y_i - f(x_i)|$ is bounded over the range of $|y_i| \leq 1, |x_i| \leq 1$,

$$\begin{aligned}
&|y_i - f(x_i)| = |y_i - \nabla^T f(x_i) w| \\
&\leq |y_i| + |\nabla^T f(x_i)(w - w(0))| \\
&+ |(\nabla f(x_i) - \nabla f_0(x_i))^T w(0)| + |\nabla^T f_0(x_i) w(0)|
\end{aligned}$$

We will bound each term separately,

- $|y_i| \leq 1$.
- By Cauchy-Schwartz inequality $|\nabla^T f(x_i)(w - w(0))| \leq |w - w(0)| \leq B \leq o(n^{1/2-\epsilon/2})$.
- Let us expand the term $|(\nabla f(x_i) - \nabla f_0(x_i))^T w(0)|$ as below,

$$\begin{aligned}
&|(\nabla f(x_i) - \nabla f_0(x_i))^T w(0)| \\
&\leq |\sum_k \frac{a_k(\sigma'(w_k^T x_i) - \sigma'(w_k^T(0)x_i))w_k^T(0)x_i}{\sqrt{m}}|
\end{aligned}$$

Further $\sigma'(w_k^T x_i) - \sigma'(w_k^T(0)x_i)$ is nonzero only when $\mathbb{1}\{|w_k^T(0)x_i| \leq R\}$. Following a decomposition similar to proof of Lemma 5.4 in [41],

$$\begin{aligned}
&|\sum_k \frac{a_k(\sigma'(w_k^T x_i) - \sigma'(w_k^T(0)x_i))w_k^T(0)x_i}{\sqrt{m}}| \\
&= |\sum_k \frac{a_k}{\sqrt{m}} \mathbb{1}\{|w_k^T(0)x_i| \leq R\} \\
&(\sigma'(w_k^T x_i) - \sigma'(w_k^T(0)x_i))w_k^T(0)x_i| \\
&= |\sum_k \frac{a_k}{\sqrt{m}} \mathbb{1}\{|w_k^T(0)x_i| \leq R\} \\
&(\sigma'(w_k^T x_i)(w_k(0) - w_k)^T x_i + \sigma(w_k^T x_i) - \sigma(w_k^T(0)x_i))|
\end{aligned}$$

Using the 1-Lipschitz property of $\sigma(.)$ and triangle inequality we can further upper bound,

$$\begin{aligned}
&|\sum_k \frac{a_k(\sigma'(w_k^T x_i) - \sigma'(w_k^T(0)x_i))w_k^T(0)x_i}{\sqrt{m}}| \\
&\leq \sum_k \frac{2R}{\sqrt{m}} \mathbb{1}\{|w_k^T(0)x_i| \leq R\}
\end{aligned}$$

Using condition (7) we can upper bound,

$$\begin{aligned}
&|(\nabla f(x_i) - \nabla f_0(x_i))^T w(0)| \\
&\leq 32R\sqrt{d\log(m+1)} + \frac{2R^2\sqrt{m}}{\kappa\sqrt{2\pi}} + \sqrt{32R^2\log\left(\frac{10}{\delta}\right)} \\
&= o(1)
\end{aligned}$$

as $R = O(\frac{1}{\sqrt{m}})$.

- Now, consider the term $f_0(x_i)$ over the range of $|x_i| \leq 1$.

$$f_0(x_i) = \sum_k \frac{a_k}{\sqrt{m}}\sigma(w_k^T(0)x_i)$$

From condition (8), we know that $\sup_{|x_i|\leq 1} f_0(x_i)$ is close to its expectation.

$$|f_0(x_i)| \leq \mathbb{E}\sup_{|x_i|\leq 1}\sum_k \frac{a_k}{\sqrt{m}}\sigma(w_k^T(0)x_i) + \sqrt{\frac{40d\kappa^2}{\delta}}$$

Since $\sigma(.)$ is $1-$Lipschitz, the expected value can be bounded by using the contraction principle with respect to distribution of a_k's,

$$\mathbb{E}\sup_{|x_i|\leq 1}\sum_k \frac{a_k}{\sqrt{m}}\sigma(w_k^T(0)x_i) \leq \mathbb{E}\sup_{|x_i|\leq 1}\sum_k \frac{a_k}{\sqrt{m}}w_k^T(0)x_i$$

Now using Cauchy-Schwartz followed by Jensen's inequality,

$$\begin{aligned}
&\mathbb{E}\sup_{|x_i|\leq 1}\sum_k \frac{a_k}{\sqrt{m}}\sigma(w_k^T(0)x_i) \leq \sqrt{\mathbb{E}\left|\sum_k \frac{a_k}{\sqrt{m}}w_k(0)\right|^2} \\
&= \sqrt{\mathbb{E}\sum_k \frac{1}{m}|w_k(0)|^2} = \kappa\sqrt{d}
\end{aligned}$$

Finally, we can bound $|f_0(x_i)|$ over $|x_i| \leq 1$.

$$|f_0(x_i)| \leq \kappa\sqrt{d} + \sqrt{\frac{40d\kappa^2}{\delta}} = o(1)$$

whenever $\kappa = O(\frac{\delta}{nd})$.

Combining the results from above,

$$M := \sup_{|y_i|\leq 1, |x_i|\leq 1} |y_i - f(x_i)| \leq o(n^{1/2-\epsilon/2}) + o(1)$$

B.12 Extension to $\sigma > 0$

For added noise $\sigma > 0$, we can assume a similar approach to bounding $\mathbb{E}_{x,y}|y - f(x)|$. Suppose we observe $\tilde{y} = y + \epsilon$ where ϵ is uncorrelated to y and sub-Gaussian with variance proxy σ^2. Now, following a similar procedure as done in this Chapter 3 and Appendix B, $\mathbb{E}_{x,y}|y - f(x)|$ is broken down into the training loss and the Rademacher complexity. The Rademacher complexity term simply changes to,

$$\sqrt{\frac{\tilde{y}^T(H+\lambda)^{-1}H(H+\lambda)^{-1}\tilde{y}}{2n}}$$

which can then be upper bounded by

$$\sqrt{\frac{y^T(H+\lambda)^{-1}H(H+\lambda)^{-1}y}{2n}} + O(\frac{\sigma}{\sqrt{\lambda}})$$

The training loss computed on y is upper bounded by the training loss computed on $\tilde{y}$ as the added noise ϵ is uncorrelated to y.

$$\frac{1}{n}\sum_i |y_i - f(x_i)|^2 \leq \frac{1}{n}\sum_i |\tilde{y}_i - f(x_i)|^2 w.h.p.$$

Now, the training loss on $\tilde{y}$ is exactly as computed in Chapter 3 and Appendix B.

APPENDIX C

PROOFS FROM CHAPTER 4

C.1 Proof of Theorem 3

The proof of this theorem is divided into Section C.1.2, which provides a bound on $V_\perp$, Section C.1.3, which bounds $V_\parallel$ and Section C.1.4 which provides a bound on $\|w - w_L^*\|$ and further proves that $w(t)$ converges by proving that it forms a Cauchy sequence. Section C.1.1 provides the probability conditions that are assumed to be true in Sections C.1.2-C.1.4.

C.1.1 Probability Conditions

Lemma 12 is proved in [79], it requires use of various concentration inequalities.

Lemma 12. *With probability greater than* $1 - \delta$ *over initialization* $a, w(0)$ *the following conditions are true. Lemma 5 is true and*

1. *Suppose* $S_w := \{w : \max_k |w_k - w_k(0)| \leq R \ \ \forall k \in [m]\}$. *Then,*

$$\sup_{w \in S_w} \|\nabla f - \nabla f_0\|_F^2 \leq \frac{8dnR}{\sqrt{2\pi}\kappa} + 4\sqrt{2}dn\sqrt{\frac{\log\left(\frac{20}{\delta}\right)}{m}}$$

$$\sup_{w \in S_w} \|\nabla^T f \nabla f - \nabla^T f_0 \nabla f_0\| \leq \frac{8dn^2R}{\kappa\sqrt{2\pi}} + 4\sqrt{2}dn^2\sqrt{\frac{\log\left(\frac{20}{\delta}\right)}{m}}$$

2. *At initialization* $\|f_0\|$ *is bounded,* $\|f_0\| \leq \frac{10\kappa\sqrt{dn}}{\delta}$.

3. *The perturbation between $\nabla^T f_0 \nabla f_0$ and $\mathbb{E}_{w(0)}[\nabla^T f_0 \nabla f_0]$ is bounded from above by*

$$\|\nabla^T f_0 \nabla f_0 - H\| \leq \sqrt{\frac{32n^2 d}{m} \log\left(\frac{20n}{\delta}\right)}$$

C.1.2 Bound on $V_\perp$

$$\dot{V}_\perp = -\langle P_0^\perp (w - w_L^*), P_0^\perp \nabla f (f - Y)\rangle$$

Since $P_0^\perp \nabla f_0 = 0$

$$\dot{V}_\perp = -\langle P_0^\perp (w - w_L^*), P_0^\perp (\nabla f - \nabla f_0)(f - Y)\rangle$$

Applying the Cauchy-Schwarz inequality,

$$\dot{V}_\perp \leq \sqrt{V_\perp} \|\nabla f - \nabla f_0\| \|f - Y\|$$

From Lemma 12 $\|\nabla f - \nabla f_0\| = O\left(\frac{dn\sqrt{C_y}}{\sqrt{c\delta} m^{0.25}}\right)$ and from Lemma 5 $\|f - Y\|^2 \leq \exp(-ct/2) \|f_0 - Y\|^2$. So

$$\begin{aligned}\sqrt{V_\perp} &= \int_0^t \dot{\sqrt{V_\perp}} ds \leq O\left(\frac{dn\sqrt{C_y}}{\sqrt{c\delta} m^{0.25}}\right) \int_0^\infty \exp(-cs/4) \|f_0 - Y\| ds \\ &= O\left(\frac{(dnC_y)^{1.5}}{c^{1.5}\delta^{1.5} m^{0.25}}\right)\end{aligned}$$

C.1.3 Bound on $V_\parallel$

We can decompose $\dot{w}$ as

$$\dot{w} = -\nabla f_0 \nabla^T f_0 (w - w_L^*) + (\nabla f_0 - \nabla f)(f - Y) + \nabla f_0 (\nabla^T f_0 w - f)$$

Define

$$e_w := (\nabla f_0 - \nabla f)(f - Y) + \nabla f_0(\nabla^T f_0 w - f)$$

Plugging this into $\dot{V}_\parallel$, we get

$$\dot{V}_\parallel = -\langle P_0(w - w_L^*), \nabla f_0 \nabla^T f_0 (w - w_L^*)\rangle + \langle P_0(w - w_L^*), P_0 e_w\rangle \tag{C.1}$$

First, we are going to bound each term in $\|e_w\|$ below.

- $\|(\nabla f_0 - \nabla f)(f - Y)\| \leq O\left(\frac{dn\sqrt{C_y}}{\sqrt{c\delta}m^{0.25}}\right)\exp(-ct/4)\|f_0 - Y\| = O\left(\frac{(dnC_y)^{1.5}}{\sqrt{c}\delta^{1.5}m^{0.25}}\right)$.
- Using $\nabla^T f_0 w - f = \int_0^t (\nabla^T f_0 \dot{w} - \dot{f})ds$ and $\dot{f} = \nabla^T f \dot{w}$

$$\begin{aligned}
\|\nabla^T f_0 w - f\| &\leq \int_0^\infty \|(\nabla^T f_0 - \nabla^T f)\dot{w}\| dt \\
&\leq \int_0^\infty \|(\nabla^T f_0 - \nabla^T f)\|\|\nabla f\|\|f - Y\| dt \\
&\leq \int_0^\infty \|(\nabla f_0 - \nabla f)\|\|\nabla f\| \exp(-ct/4)\|f_0 - Y\| dt \\
&= O\left(\frac{(dn)^2 C_y^{1.5}}{c^{1.5}\delta^{1.5}m^{0.25}}\right)
\end{aligned}$$

Hence $\|e_w\| = O\left(\frac{(dn)^{2.5}C_y^{1.5}}{c^{1.5}\delta^{1.5}m^{0.25}}\right)$. Using this bound on $\|e_w\|$ together with Cauchy-Schwarz inequality we can simplify (C.1) as

$$\dot{V}_\parallel \leq -\langle P_0(w - w_L^*), \nabla f_0 \nabla^T f_0 (w - w_L^*)\rangle + \sqrt{V_\parallel} O\left(\frac{(dn)^{2.5}C_y^{1.5}}{c^{1.5}\delta^{1.5}m^{0.25}}\right)$$

From Lemma 12 $\nabla^T f_0 \nabla f_0 \succeq c/2$ when $m = O(n^2 d \log(n/\delta)/c^2)$. Hence $\nabla f_0 \nabla^T f_0 \succeq c/2 P_0$. So

$$\dot{V}_\parallel \leq -c/2 V_\parallel + \sqrt{V_\parallel} O\left(\frac{(dn)^{2.5}C_y^{1.5}}{c^{1.5}\delta^{1.5}m^{0.25}}\right)$$

Solving the above differential equation implies

$$\sqrt{V_\parallel} \leq \exp(-c/2t)\sqrt{V_\parallel(0)} + O\left(\frac{(dn)^{2.5}C_y^{1.5}}{c^{2.5}\delta^{1.5}m^{0.25}}\right)$$

C.1.4 Convergence of w

In this subsection, we first show that the results in the previous two subsections imply that $\|w - w_L^*\|$ is bounded. Combining results on $V_\perp$ and $V_\parallel$,

$$\|w - w_L^*\| \leq \sqrt{V_\perp} + \sqrt{V_\parallel} \leq \exp(-c/2t)\sqrt{V_\parallel(0)} + O\left(\frac{(dn)^{2.5}C_y^{1.5}}{c^{2.5}\delta^{1.5}m^{0.25}}\right)$$

We know that

$$w(t) = w(0) + \int_0^t \nabla f(f - Y)dt$$

For any $t > s > T$, we show that $\|w(t) - w(s)\| = O(\exp(-T))$. Then it will follow that $\{w(t_n)\}_{n\geq 1}$ is a Cauchy sequence for any sequence $t_1 \leq t_2 \ldots$ and that it converges.

$$\begin{aligned}\|w(t) - w(s)\| &\leq \int_s^t \|\nabla f(f - Y)\|dt \leq \int_s^t \sqrt{nd}\exp\left(-\frac{ct}{4}\right)\|(f_0 - Y)\|dt \\ &= O\left(\frac{dnC_y}{c\delta}\right)\exp(-T)\end{aligned}$$

This shows that $w(t) \to w^*$ as $t \to \infty$ for some w^*.

C.2 Proof of Lemma 4

We can write the $H_{ij} = \frac{\tilde{x}_i^\top \tilde{x}_j}{4} + \sum_{p\geq 1} c_{2p}(\frac{\tilde{x}_i^\top \tilde{x}_j}{d+1})^{2p}$, $c_{2p} = \frac{(2p-3)!!(d+1)}{2\pi(2p-2)!!(2p-1)}$. Consider the sequence of $n \times n$ matrices $H^{(p)}, p \geq 1$ where we define the i, j element of $H^{(p)}$ as $H_{ij}^{(p)} = (\frac{\tilde{x}_i^\top \tilde{x}_j}{d+1})^{2p}$. We can write $H = 0.25\tilde{X}^T\tilde{X} + \sum_{p\geq 1} c_{2p}H^{(p)}$ where $\tilde{X} = [\tilde{x}_1 \ldots \tilde{x}_n]$. Note that $H^{(p)}, p \geq 1$ are positive semidefinite as we can write it

as $H^{(p)} = \frac{1}{(d+1)^{2p}} \tilde{X}^{(p)T} \tilde{X}^{(p)}, \tilde{X}^{(p)} = [\tilde{x}_1^{\otimes 2p} \dots \tilde{x}_n^{\otimes 2p}]$. So H is a positive semidefinite matrix. We will lower bound the smallest eigenvalue of H by computing the smallest eigenvalue for each of $H^{(p)}$ using Gershgorin circle theorem.

$$\lambda_{\min}(H) = \min_{u:\|u\|=1} u^T H u \geq \sum_{p \geq 1} c_{2p} \min_{u:\|u\|=1} u^T H^{(p)} u = \sum_{p \geq 1} c_{2p} \lambda_{\min}(H^{(p)})$$

The diagonal elements of $H^{(p)}$ are 1. Hence by Gershgorin circle theorem the eigenvalues of $H^{(p)}$ can be lower bounded by $1 - \max_i \sum_{j:j \neq i} H_{ij}^{(p)}$. This bound is positive when $\max_i \sum_{j:j \neq i} H_{ij}^{(p)}$ is less than 1. Denote $\cos\theta_{\min} := \max_{i \neq j} \frac{\tilde{x}_j^T \tilde{x}_j}{d+1}$. So $\max_i \sum_{j:j \neq i} H_{ij}^{(p)} \leq (n-1)(\cos\theta_{\min})^{2p}$. For all $p \geq k = \frac{\log(2n)}{\log(1/\cos\theta_{\min})}$, $(n-1)(\cos\theta_{\min})^{2p} \leq 0.5$. Hence

$$\lambda_{\min}(H) \geq 0.5 \sum_{p \geq k} c_{2p} \geq \sum_{p \geq k} \frac{d+1}{8\pi\sqrt{p-1}(2p-1)} \geq \int_{k+1}^{\infty} \frac{d+1}{8\pi\sqrt{x-1}(2x-1)} dx$$
$$\geq \frac{d+1}{8\pi\sqrt{k+1}} = \frac{d+1}{8\pi} \left(\sqrt{\frac{\log(1/\cos\theta_{\min})}{\log(2n/\cos\theta_{\min})}} \right)$$

where we use Wallis' inequality to lower bound the ratio of the double factorial [80].

We can also get a simple upper bound on the minimum eigenvalue of $\lambda_{\min}(H)$. Suppose $a, b = \arg\max_{ij} x_i^T x_j$. Choose vector $u \in \mathbb{R}^n$ with $u_a = \frac{1}{\sqrt{2}}, u_b = \frac{1}{\sqrt{2}}$.

$$\begin{aligned} \lambda_{\min}(H) = \min_{u:\|u\|=1} u^T H u &\leq \frac{1}{2}(H_{aa} + H_{bb} - 2H_{ab}) \\ &= 0.5(d+1)\left(1 - \left(1 - \frac{\theta_{\min}}{\pi}\right)\cos\theta_{\min}\right) \end{aligned}$$

Using the identity $1 - \theta^2/2 \leq \cos\theta \leq 1 - \theta^2/4, \theta < 1$ we arrive at the approximations.

C.3 Proof of Theorem 4

It is proved in [56] that gradient descent achieves zero training error in overparameterized networks.

Lemma 13. *([56]) The discrete-time gradient descent algorithm achieves zero loss if* $m = \Omega\left(\frac{d^4 C_y^2 n^6}{c^4 \delta^3}\right), \kappa = 1$ *and* $\eta = O\left(\frac{c}{d^2 n^2}\right)$. *With probability more than* $1-\delta$ *over initialization,* $\|f_k - Y\|^2 \leq \left(1 - \frac{c\eta}{2}\right)^k \|f_0 - Y\|^2$. *Also, the weights stay close to initialization,* $\|w_k(k) - w_k(0)\| = O\left(\frac{dnC_y}{\delta c\sqrt{m}}\right)$ *for* $k = \{0, 1, \ldots\}$.

C.3.1 Bound on $V_\perp$

Since, $P_0^\perp(w(0) - w_L^*) = 0$, we can use a telescoping expansion of $P_0^\perp(w(k+1) - w_L^*)$ followed by the use of triangle inequality to get

$$\begin{aligned}
\sqrt{V_\perp(k+1)} &\leq \sum_{l=0}^{k} \|P_0^\perp(w(l+1) - w(l))\| \\
&= \eta \sum_{l=0}^{k} \|P_0^\perp(\nabla f_l - \nabla f_0)(f_l - Y)\| \\
&\leq \eta \sum_{l=0}^{k} O\left(\frac{dn\sqrt{C_y}}{\sqrt{c\delta} m^{0.25}}\right) \|f_l - Y\| \\
&\leq \eta O\left(\frac{dn\sqrt{C_y}}{\sqrt{c\delta} m^{0.25}}\right) \sum_{l=0}^{k} (1 - c\eta)^{l/2} \|f_0 - Y\| \\
&\leq \eta O\left(\frac{dn\sqrt{C_y}}{\sqrt{c\delta} m^{0.25}}\right) \frac{1}{1 - \sqrt{1 - c\eta}} \|f_0 - Y\| \\
&\leq O\left(\frac{dn\sqrt{C_y}}{c\sqrt{c\delta} m^{0.25}}\right) \|f_0 - Y\|
\end{aligned}$$

C.3.2 Bound on $V_{\|}$

We have

$$V_{\|}(k+1) = \|P_0(w_{k+1} - w_L^*)\|^2 = \|P_0(w_k - w_L^*)\|^2 + \|P_0(w_{k+1} - w_k)\|^2 + 2\langle P_0(w_k - w_L^*), P_0(w_{k+1} - w_k)\rangle \tag{C.2}$$

We can expand $P_0(w_{k+1} - w_k)$ as

$$\begin{aligned} P_0(w_{k+1} - w_k) &= -\eta P_0 \nabla f_k (f_k - Y) \\ &= -\eta (P_0(\nabla f_k - \nabla f_0)(f_k - Y) + \nabla f_0 (f_k - \nabla^T f_0 w_k) \\ &+ \nabla f_0 \nabla^T f_0 (w_k - w_L^*)) \\ &= e_w - \eta \nabla f_0 \nabla^T f_0 (w_k - w_L^*)) \end{aligned} \tag{C.3}$$

where we define $e_w := -\eta(P_0(\nabla f_k - \nabla f_0)(f_k - Y) + \nabla f_0(f_k - \nabla^T f_0 w_k))$. We can analyze the terms in (C.2) separately as below using the decomposition in (C.3).

- $\|P_0(w_{k+1} - w_k)\|^2$: Using the bound on $\|\nabla f - \nabla f_0\|$ from Lemma 12,

$$\begin{aligned} \|P_0(w_{k+1} - w_k)\| \leq \eta O\left(\frac{dn\sqrt{C_y}}{\sqrt{c\delta}m^{0.25}}\right)\|f_k - Y\| \\ + \eta\sqrt{dn}\|f_k - \nabla^T f_0 w_k\| + \eta dn \|P_0(w_k - w_L^*)\| \end{aligned}$$

$$\begin{aligned} f_k - \nabla^T f_0 w_k &= \sum_{l=1}^{k} f_l - \nabla^T f_0 w_l - f_{l-1} - \nabla^T f_0 w_{l-1} \\ &= \sum_{l=1}^{k} (\nabla f - \nabla f_0)^T (w_l - w_{l-1}) \\ &= \eta \sum_{l=1}^{k} (\nabla f - \nabla f_0)^T \nabla f_{l-1}(f_{l-1} - Y) \end{aligned}$$

By using the triangle inequality,

$$\begin{aligned}
\|f_k - \nabla^T f_0 w_k\| &\leq \eta \sum_{l=1}^{k} O(\frac{(dn)^{1.5}\sqrt{C_y}}{\sqrt{c\delta}m^{0.25}})\|f_{l-1} - Y\| \\
&\leq \eta \sum_{l=1}^{k} O(\frac{(dn)^{1.5}\sqrt{C_y}}{\sqrt{c\delta}m^{0.25}})(1-c\eta)^{l/2}\|f_0 - Y\| \\
&\leq O\left(\frac{(dn)^{1.5}\sqrt{C_y}}{c\sqrt{c\delta}m^{0.25}}\right)\|f_0 - Y\| \\
&= O\left(\frac{(dn)^2 C_y^{1.5}}{c^{1.5}\delta^{1.5}m^{0.25}}\right)
\end{aligned}$$

Using this bound we can upper bound $\|P_0(w_{k+1} - w_k)\|$ as

$$\begin{aligned}
\|P_0(w_{k+1} - w_k)\| &\leq \eta O\left(\frac{dn\sqrt{C_y}}{\sqrt{c\delta}m^{0.25}}\right)(1-c\eta)^{k/2}\|f_0 - Y\| \\
&\quad + \eta O\left(\frac{(dn)^{2.5}C_y^{1.5}}{c^{1.5}\delta^{1.5}m^{0.25}}\right) + \eta dn\sqrt{V_\|} \\
&\leq \eta O\left(\frac{(dn)^{2.5}C_y^{1.5}}{c^{1.5}\delta^{1.5}m^{0.25}}\right) + \eta dn\sqrt{V_\|} \qquad \text{(C.4)}
\end{aligned}$$

So $\|P_0(w_{k+1} - w_k)\|^2$ can be bounded as

$$\begin{aligned}
\|P_0(w_{k+1} - w_k)\|^2 &\leq O\left(\frac{\eta^2(dn)^5 C_y^3}{c^3\delta^3 m^{0.5}}\right) + \eta^2(dn)^2 V_\| \\
&\quad + O\left(\frac{\eta^2(dn)^{3.5}C_y^{1.5}}{c^{1.5}\delta^{1.5}m^{0.25}}\right)\sqrt{V_\|}
\end{aligned}$$

Using the bound $\sqrt{V_\parallel} \leq 1 + V_\parallel$

$$\begin{aligned}\|P_0(w_{k+1} - w_k)\|^2 &\leq O\left(\frac{\eta^2(dn)^5 C_y^3}{c^3\delta^3 m^{0.5}}\right) + \eta^2(dn)^2 V_\parallel + O\left(\frac{\eta^2(dn)^{3.5}C_y^{1.5}}{c^{1.5}\delta^{1.5}m^{0.25}}\right) \\ &\quad + O\left(\frac{\eta^2(dn)^{3.5}C_y^{1.5}}{c^{1.5}\delta^{1.5}m^{0.25}}\right) V_\parallel \qquad \text{(C.5)} \\ &\leq O\left(\frac{\eta^2(dn)^5 C_y^3}{c^3\delta^3 m^{0.25}}\right) \\ &\quad + \left(\eta^2(dn)^2 + O\left(\eta^2 \frac{(dn)^{3.5}C_y^{1.5}}{c^{1.5}\delta^{1.5}m^{0.25}}\right)\right) V_\parallel \qquad \text{(C.6)}\end{aligned}$$

- $\langle P_0(w_k - w_L^*), P_0(w_{k+1} - w_k)\rangle$: We can expand $P_0(w_{k+1} - w_k)$ as in (C.3) and use Cauchy-Schwarz inequality to get

$$\begin{aligned}&\langle P_0(w_k - w_L^*), P_0(w_{k+1} - w_k)\rangle \\ &\leq \|e_w\|\sqrt{V_\parallel} - \eta\langle P_0(w_k - w_L^*), \nabla f_0 \nabla^T f_0(w_k - w_L^*))\rangle\end{aligned}$$

Using the bound $\nabla f_0 \nabla^T f_0 \succeq cP_0$ when $m \geq n^2 d\log(n/\delta)/c^2$ from Lemma 12

$$\langle P_0(w_k - w_L^*), P_0(w_{k+1} - w_k)\rangle \leq \|e_w\|\sqrt{V_\parallel} - c\eta V_\parallel \qquad \text{(C.7)}$$

Now we can use the bound $\|e_w\| = \eta O\left(\frac{(dn)^{2.5}C_y^{1.5}}{c^{1.5}\delta^{1.5}m^{0.25}}\right)$ and $\sqrt{V_\parallel} \leq 1 + V_\parallel$ to have

$$\begin{aligned}\langle P_0(w_k - w_L^*), P_0(w_{k+1} - w_k)\rangle \leq &-\left(c\eta - \eta O\left(\frac{(dn)^{2.5}C_y^{1.5}}{c^{1.5}\delta^{1.5}m^{0.25}}\right)\right) V_\parallel \\ &+ \eta O\left(\frac{(dn)^{2.5}C_y^{1.5}}{c^{1.5}\delta^{1.5}m^{0.25}}\right) \qquad \text{(C.8)}\end{aligned}$$

We are now ready to combine the bounds from (C.6) and (C.8) into the recursion in

(C.2). If we have $\eta \leq \frac{c}{2(dn)^2}$, and $m = \Omega\left(\frac{(dn)^{10}C_y^6}{c^{10}\delta^6}\right)$

$$\begin{aligned}
V_{\|}(k+1) &\leq V_{\|}(k) + O\left(\frac{\eta^2(dn)^5 C_y^3}{c^3\delta^3 m^{0.25}}\right) \\
&\quad + \left(\eta^2(dn)^2 + O\left(\eta^2\frac{(dn)^{3.5}C_y^{1.5}}{c^{1.5}\delta^{1.5}m^{0.25}}\right)\right)V_{\|}(k) \\
&\quad - \left(c\eta - \eta O\left(\frac{(dn)^{2.5}C_y^{1.5}}{c^{1.5}\delta^{1.5}m^{0.25}}\right)\right)V_{\|} + \eta O\left(\frac{(dn)^{2.5}C_y^{1.5}}{c^{1.5}\delta^{1.5}m^{0.25}}\right) \qquad \text{(C.9)} \\
&\leq \left(1 - \frac{c\eta}{2}\right)V_{\|}(k) + O\left(\frac{\eta(dnC_y)^3}{c^2\delta^3 m^{0.25}}\right) \qquad \text{(C.10)} \\
&\leq \left(1 - \frac{c\eta}{2}\right)^k V_{\|}(0) + O\left(\frac{(dnC_y)^3}{c^3\delta^3 m^{0.25}}\right) \qquad \text{(C.11)}
\end{aligned}$$

where the final step follows from induction.

C.3.3 Putting It All Together

We combine the bounds on $V_{\|}$ and $V_{\perp}$ to get

$$\begin{aligned}
\|w_k - w_L^*\| \leq \sqrt{V_{\|}} + \sqrt{V_{\perp}} &\leq O\left(\frac{dn\sqrt{C_y}}{c\sqrt{c\delta}m^{0.25}}\right)\|f_0 - Y\| \\
&\quad + \sqrt{\left(1 - \frac{c\eta}{2}\right)^k V_{\|}(0) + O\left(\frac{(dnC_y)^3}{c^3\delta^3 m^{0.25}}\right)} \\
&\leq \left(1 - \frac{c\eta}{2}\right)^{k/2}\|w_0 - w_L^*\| + O\left(\frac{(dnC_y)^{1.5}}{c^{1.5}\delta^{1.5}m^{0.125}}\right)
\end{aligned} \qquad \text{(C.12)}$$

C.3.4 Convergence of Iterates for GD

Similar to the continuous-time dynamics of gradient flow, we show that the iterates w_k form a Cauchy sequence. For $n > m > N$,

$$\begin{aligned}\|w_n - w_m\| &\leq \eta \sum_{k=m}^{n} \|\nabla f_k (f_k - Y)\| \leq \eta\sqrt{dn} \sum_{k=m}^{n} (1 - c\eta)^{k/2} \|(f_0 - Y)\| \\ &\leq \eta\sqrt{dn}(1 - c\eta)^{m/2} \frac{1}{1 - \sqrt{1 - c\eta}} \|(f_0 - Y)\| \\ &\leq \frac{2\sqrt{dn}}{c}(1 - c\eta)^{N/2} \|(f_0 - Y)\|\end{aligned}$$

Hence, $\{w_k\}_{k=1}^{\infty}$ converges.

C.4 Proof of Lemma 6

We can decompose $f(x, w(k)) - f_{KR}(x)$ as

$$\begin{aligned}&(f(x, w(k)) - f_L(x, w(k))) + (f_L(x, w(k)) - f_L(x, w_L^*)) \\ &+ (\nabla^\top f(x, w(0)) P_0^\perp w(0)) + (\nabla^\top f(x, w(0)) \nabla f_0 (\nabla^\top f_0 \nabla f_0)^{-1} Y - f_{KR}(x))\end{aligned}$$

We bound each of the above terms below. In the derivation below we use the bounds resulting from Theorem 4, Lemma 13 and Lemma 12 deterministically. Section C.4.1 shows the probabilistic events that are assumed to be true w.p. greater than $1 - \delta$ in the proof below.

- We begin by showing that $\mathbb{E}_x(f(x, w(k)) - f_L(x, w(k)))^2$ is small. We can expand $f(x, w(k)) - f_L(x, w(k))$ as

  $$\frac{1}{\sqrt{m}} \sum_j a_j (\sigma(w_j^\top(k)\tilde{x}) - \sigma'(w_j^\top(0)\tilde{x}) w_j^\top(k)\tilde{x})$$

  Define $S(x, w(0)) := \{j \in [m] : \sigma'(w_j^\top(0)\tilde{x}) \neq \sigma'(w_j^\top(k)\tilde{x})\}$, the set of indices from $1 \ldots m$ for which $w_j^\top(0)\tilde{x}$ has a different sign than $w_j^\top(k)\tilde{x}$.

This implies that $f(x, w(k)) - f_L(x, w(k))$ can be equivalently written as

$$\frac{1}{\sqrt{m}} \sum_{j \in S(x,w(0))} a_j(\sigma(w_j^\top(k)\tilde{x}) - \sigma(w_j^\top(0)\tilde{x}) + \sigma(w_j^\top(0)\tilde{x}) - \sigma'(w_j^\top(0)\tilde{x}) w_j^\top(k)\tilde{x})$$

We can use triangle inequality followed by 1-Lipschitz property of $\sigma(.)$ to bound $|f(x, w(k)) - f_L(x, w(k))|$ by

$$\frac{2}{\sqrt{m}} \sum_{j \in S(x,w(0))} |w_j^\top(k)\tilde{x} - w_j^\top(0)\tilde{x}|$$

Denote the bound on $\|w_j(k) - w_j(0)\|$ from Lemma 13 as $R := O\left(\frac{dnC_y}{\delta c\sqrt{m}}\right)$. So

$$|f(x, w(k)) - f_L(x, w(k))|^2 \leq \frac{4dR^2|S(x, w(0))|^2}{m} \tag{C.13}$$

where $|S(x, w(0))|$ denotes the cardinality of set $S(x, w(0))$. From Section C.4.1, we have that

$$\mathbb{E}_x|S(x, w(0))|^2 \leq \frac{\mathbb{E}_{x,w(0)}|S(x, w(0))|^2}{\delta}$$

When $|w_j^\top(0)\tilde{x}| \leq |w_j^\top(0)\tilde{x} - w_j^\top(k)\tilde{x}|$ there is a difference in sign between $w_j^\top(0)\tilde{x}$ and $w_j^\top(k)\tilde{x}$. Hence

$$\mathbb{E}_{x,w(0)}|S(x, w(0))|^2 \leq \mathbb{E}_{x,w(0)} \left(\sum_j \mathbb{1}\{|w_j^\top(0)\tilde{x}| \leq |w_j^\top(0)\tilde{x} - w_j^\top(k)\tilde{x}|\} \right)^2$$

By triangle inequality followed by Cauchy-Schwartz

$$\mathbb{E}_{x,w(0)}|S(x, w(0))|^2 \leq \sum_{j,l} \sqrt{\mathbb{E}_{x,w_j(0)}\mathbb{1}\{|w_j^\top(0)\tilde{x}| \leq R\}\mathbb{E}_{x,w_l(0)}\mathbb{1}\{|w_l^\top(0)\tilde{x}| \leq R\}}$$

Since $w_j(0)$ is distributed $\mathcal{N}(0, \kappa^2 I_d)$ and x is a distributed as $\mathcal{X}$ which is supported on the ball $\|x\| = \sqrt{d}$, $w_j^\top(0)\tilde{x}$ is distributed as $\mathcal{N}(0, \kappa^2(d+1))$. Hence $\mathbb{P}(|w_j^\top(0)\tilde{x}| \leq R)$ is bounded by $\frac{R}{\sqrt{2\pi d}\kappa}$. This shows

$$\mathbb{E}_{x,w(0)}|S(x, w(0))|^2 = O\left(\frac{m^2 R}{\sqrt{d}\kappa}\right)$$

Plugging this bound back into (C.13) we have

$$E_x|f(x, w(k)) - f_L(x, w(k))|^2 = O\left(\frac{\sqrt{d}mR^3}{\delta\kappa}\right) = O\left(\frac{d^{3.5}n^3C_y^3}{\kappa\delta^4 c^3 m^{0.5}}\right)$$

- We can upper bound the term $E_x|f_L(x, w(k)) - f_L(x, w_L^*)|^2 \leq \|w(k) - w_L^*\|^2$. Using the bound on $\|w(k) - w_L^*\|$ from Theorem 4

$$\begin{aligned} E_x|f_L(x, w(k)) - f_L(x, w_L^*)|^2 \leq \left(1 - \frac{c\eta}{2}\right)^{k/2} \|w(0) - w_L^*\| \\ + O\left(\frac{(dnC_y)^{1.5}}{c^{1.5}\delta^{1.5}m^{0.125}}\right) \end{aligned}$$

From the definition of w_L^* we can write $w(0) - w_L^* = P_0 w(0) - \nabla f_0(\nabla^T f_0 \nabla f_0)^{-1} Y$. Hence

$$\begin{aligned} \|w(0) - w_L^*\|^2 &\leq 2\|P_0 w(0)\|^2 + 2\|\nabla f_0(\nabla^T f_0 \nabla f_0)^{-1} Y\|^2 \\ &= 2f_0^T(\nabla f_0^\top \nabla f_0)^{-1} f_0 + 2Y^T(\nabla f_0^\top \nabla f_0)^{-1} Y \end{aligned}$$

Using the bound on $\|f_0\|^2$ from Section C.4.1 $f_0^T(\nabla f_0^\top \nabla f_0)^{-1} f_0 = O\left(\frac{1}{\sqrt{dn}}\right)$ when $\kappa^2 = O\left(\frac{c\delta}{(dn)^{1.5}}\right)$. Now consider the term $Y^T(\nabla^T f_0 \nabla f_0)^{-1} Y$.

$$Y^T(\nabla^T f_0 \nabla f_0)^{-1} Y = Y^T H^{-1} Y + Y^T((\nabla^T f_0 \nabla f_0)^{-1} - H^{-1})Y$$

Using the identity $A^{-1} - B^{-1} = A^{-1}(B - A)B^{-1}$ and the concentration

results from Lemma 12

$$Y^T((\nabla^T f_0 \nabla f_0)^{-1} - H^{-1})Y \leq \frac{n^2 C_y^2 \sqrt{d}}{c^2\sqrt{m}} \sqrt{\log\left(\frac{n}{\delta}\right)} = O\left(1\right)$$

when $m = \Omega(dn^4 C_y^4 \log(n/\delta)/c^4)$. Thus $\|w(0) - w_L^*\|^2 \leq Y^T H^{-1} Y + O\left(1\right)$.

- Consider $\nabla^\top f(x, w(0)) P_0^\perp w(0) = f(x, w(0)) - \nabla^\top f(x, w(0)) \nabla f_0 (\nabla f_0^\top \nabla f_0)^{-1} f_0$. From Section C.4.1

$$\mathbb{E}_x f^2(x, w(0)) \leq \frac{\kappa^2 d}{2\delta} \tag{C.14}$$

For the second term, note that $E_x(\nabla^\top f(x, w(0)) \nabla f_0 (\nabla f_0^\top \nabla f_0)^{-1} f_0)^2 \leq d f_0^T (\nabla f_0^\top \nabla f_0)^{-1} f_0$ since $\mathbb{E}_x \nabla f(x, w(0)) \nabla^\top f(x, w(0)) \preccurlyeq dI$ and so

$$E_x(\nabla^\top f(x, w(0)) \nabla f_0 (\nabla f_0^\top \nabla f_0)^{-1} f_0)^2 \leq \frac{d\|f_0\|^2}{c}$$

when $m = \Omega(n^2 d \log(n/\delta)/c^2)$. Using the bound on $\|f_0\|^2$ from Section C.4.1

$$\frac{\|f_0\|^2}{c} = O\left(\frac{\kappa^2 dn}{c\delta}\right)$$

Combining this with (C.14)

$$\mathbb{E}_x(\nabla^\top f(x, w(0)) P_0^\perp w(0))^2 \leq \frac{\kappa^2 d^2 n}{c\delta} = O\left(\frac{1}{\sqrt{n}}\right)$$

when $\kappa^2 = O\left(\frac{c\delta}{d^2 n^{1.5}}\right)$.

- Finally, consider the term $\nabla^\top f(x, w(0)) \nabla f_0 (\nabla^\top f_0 \nabla f_0)^{-1} Y - f_{KR}(x)$. We can expand it as

$$\left(\nabla^\top f(x, w(0)) \nabla f_0 - h^\top\right) (\nabla^\top f_0 \nabla f_0)^{-1} Y + h^\top \left((\nabla^\top f_0 \nabla f_0)^{-1} - H^{-1}\right) Y$$

Applying triangle inequality we can upper bound

$\mathbb{E}_x \left(\nabla^\top f(x, w(0)) \nabla f_0 (\nabla^\top f_0 \nabla f_0)^{-1} Y - f_{KR}(x)\right)^2$ as

$$E_x \left(\frac{nC_y}{c^2} \|\nabla^\top f_0 \nabla f(x, w(0)) - h\|^2 \right) + \frac{(dnC_y)^2}{16} \|(\nabla^\top f_0 \nabla f_0)^{-1} - H^{-1}\|^2 \tag{C.15}$$

First consider the left term in (C.15). From Section C.4.1 the left term in (C.15) can be bounded by

$$\frac{nC_y}{c^2 \delta} \sum_i \mathbb{E}_{x, w(0), x_i} \left(\frac{1}{m} \sum_j \sigma'(w_j^\top(0)\tilde{x}) \sigma'(w_j^\top(0)\tilde{x}_i) \tilde{x}^\top \tilde{x}_i - K(x, x_i) \right)^2$$

Since $w_j^\top(0)\tilde{x}_i$ and $w_j^\top(0)\tilde{x}$ are statistically independently for all j given x, x_i, we have

$$E_x \left(\frac{nC_y}{c^2} \|\nabla^\top f(x, w(0)) \nabla f_0 - h^\top\|^2 \right) \leq \frac{2nC_y}{mc^2\delta} \tag{C.16}$$

Now consider the second term in (C.15). Using the identity $A^{-1} - B^{-1} = A^{-1}(B - A)B^{-1}$ and the concentration results from Lemma 12

$$\frac{(dnC_y)^2}{16} \|(\nabla^\top f_0 \nabla f_0)^{-1} - H^{-1}\|^2 \leq \frac{d^2 n^4 C_y^2}{c^4 m} \log\left(\frac{20n}{\delta}\right) \tag{C.17}$$

when $m = O(n^2 d \log(n/\delta)/c^2)$. Combining the above bound with (C.16)

$$\mathbb{E}_x \left(\nabla^\top f(x, w(0)) \nabla f_0 (\nabla^\top f_0 \nabla f_0)^{-1} Y - f_{KR}(x)\right)^2 \leq \frac{d^2 n^4 C_y^2 \log n}{c^4 m \delta}$$

C.4.1 Probability Conditions for Lemma 6

We can use a union bound to show the below events occur simultaneously with probability at least $1 - \delta$:

- Theorem 4, Lemma 13 and Lemma 12 are true.

- By Markov inequality, we have that

$$\mathbb{E}_x|S(x,w(0))|^2 \leq \frac{10\mathbb{E}_{x,w(0)}|S(x,w(0))|^2}{\delta} \text{ w.p. greater than } 1-\delta/10$$

- Using Markov inequality

$$\mathbb{E}_x f^2(x,w(0)) \leq \frac{10d\kappa^2}{\delta} \text{ w.p. greater than} 1-\delta/10$$

- By Markov inequality

$$\begin{aligned}&\mathbb{E}_x\left(\|\nabla^\top f(x,w(0))\nabla f_0 - h^\top\|^2\right)\\ &\leq \mathbb{E}_{x,w(0),x_i}\left(\frac{10}{\delta}\|\nabla^\top f(x,w(0))\nabla f_0 - h^\top\|^2\right)\end{aligned}$$

  with probability greater than $1-\delta/10$.

- Again, we apply Markov inequality.

$$\frac{\|f_0\|^2}{c} \leq \frac{\mathbb{E}\|f_0\|^2}{c\delta} \leq \frac{10\kappa^2 dn}{c\delta} \text{ w.p. greater than } 1-\delta/10$$

C.5 Proof of Lemma 7

Note that $K(x,y) = \tilde{x}^T\tilde{y}\frac{\pi-\arccos\left(\frac{\tilde{x}^T\tilde{y}}{d+1}\right)}{2\pi} = \frac{\tilde{x}^\top\tilde{y}}{4} + \sum_{p\geq 1} c_{2p}(\frac{\tilde{x}^\top\tilde{y}}{d+1})^{2p}, c_{2p} = \frac{(2p-3)!!(d+1)}{2\pi(2p-2)!!(2p-1)}$. Using $\tilde{x} = \{x, 1\}$

$$\begin{aligned}
K(x,y) &= \frac{1}{4} + \frac{x^\top y}{4} + \sum_{p\geq 1} c_{2p}\left(\frac{x^\top y + 1}{d+1}\right)^{2p} \\
&= \frac{1}{4} + \frac{x^\top y}{4} + \sum_{p\geq 1} \frac{c_{2p}}{(d+1)^{2p}} \sum_{k=0}^{2p} \binom{2p}{k} (x^\top y)^k \\
&= \left(\frac{1}{4} + \sum_{p\geq 1} \frac{c_{2p}}{(d+1)^{2p}}\right) + \left(\frac{1}{4} + \sum_{p\geq 1} \frac{2pc_{2p}}{(d+1)^{2p}}\right) x^\top y \\
&+ \sum_{k\geq 2} \sum_{p\geq \lceil k/2 \rceil} \frac{c_{2p}}{(d+1)^{2p}} \binom{2p}{k} (x^\top y)^k
\end{aligned}$$

Denoting the coefficient of $(x^\top y)^k$ by d_k

$$K(x,y) = \sum_{k\geq 0} d_k (x^\top y)^k = \sum_{k\geq 0} d_k (x^{\otimes k})^\top y^{\otimes k} = \phi^\top(x)\phi(y) \tag{C.18}$$

where $\phi(x) = [\sqrt{d_0}, \sqrt{d_1}x, \sqrt{d_2}x^{\otimes 2}, \sqrt{d_3}x^{\otimes 3}, \ldots]$.

C.6 Proof of Theorem 5

Denote matrix $\Phi = [\phi(x_1), \phi(x_2), \ldots \phi(x_n)]$ with n columns. Note that

$$f_{KR}(x) = \phi^\top(x)\Phi(\Phi^\top\Phi)^{-1}\Phi^\top\bar{w} = \phi^\top(x)P_\Phi\bar{w}$$

where P_Φ is the projection matrix onto the columns space of Φ. First we center the random variable $\phi(x)$ around its expectation and denote $\tilde{\phi}(x) = \phi(x) - \mathbb{E}_x\phi(x)$. Since the columns space of Φ remains unchanged due to this transformation, $P_{\tilde{\Phi}} =$

P_Φ. So

$$\begin{aligned} E_x(y - f_{KR}(x))^2 &= E_x(\tilde{\phi}^\top(x)(I - P_{\tilde{\Phi}})\bar{w} + \mathbb{E}_x\phi^\top(x)(I - P_{\tilde{\Phi}})\bar{w})^2 \\ &\leq 2\bar{w}^\top(I - P_{\tilde{\Phi}})\mathbb{E}_x\tilde{\phi}(x)\tilde{\phi}^\top(x)(I - P_{\tilde{\Phi}})\bar{w} \\ &+ 2\bar{w}^\top(I - P_\Phi)\mathbb{E}_x\phi(x)\mathbb{E}_x\phi^\top(x)(I - P_\Phi)\bar{w} \end{aligned}$$

Using $\|I - P_{\tilde{\Phi}}\| \leq 1$ and $P_\Phi^\perp\phi(x_i) = 0\ \forall i \in [n]$, we can further upper bound

$$\begin{aligned} E_x(y - f_{KR}(x))^2 \leq 2\|\bar{w}\|^2\|\mathbb{E}_x\tilde{\phi}(x)\tilde{\phi}^\top(x) - n^{-1}\sum_i \tilde{\phi}(x_i)\tilde{\phi}^\top(x_i)\| \\ + 2\|\bar{w}\|^2\|n^{-1}\sum_i \tilde{\phi}(x_i)\|^2 \end{aligned} \tag{C.19}$$

Now we will apply McDiarmid's inequality to the first term in (C.19). Note that typically one would need to use more involved concentration inequality for sample covariance matrix (like [81]). But our data points $\tilde{\phi}(x)$ are bounded and hence we can simply use McDiarmid's inequality. If we change one of the $\tilde{\phi}(x_i)$ with its i.i.d. copy then we can apply triangle inequality to show that $\|\mathbb{E}_x\tilde{\phi}(x)\tilde{\phi}^\top(x) - n^{-1}\sum_i \tilde{\phi}(x_i)\tilde{\phi}^\top(x_i)\|$ changes by a maximum of $2(d+1)/n$. Hence by McDiarmid's inequality

$$\begin{aligned} &\|\mathbb{E}_x\tilde{\phi}(x)\tilde{\phi}^\top(x) - n^{-1}\sum_i \tilde{\phi}(x_i)\tilde{\phi}^\top(x_i)\| \\ &\leq \mathbb{E}_{x_i}\|\mathbb{E}_x\tilde{\phi}(x)\tilde{\phi}^\top(x) - n^{-1}\sum_i \tilde{\phi}(x_i)\tilde{\phi}^\top(x_i)\| + \sqrt{\frac{(d+1)\log(1/\delta)}{n}} \text{ w.p. } 1 - \delta \end{aligned}$$

We can now upper bound $\mathbb{E}_{x_i}\|\mathbb{E}_x\tilde{\phi}(x)\tilde{\phi}^\top(x) - n^{-1}\sum_i \tilde{\phi}(x_i)\tilde{\phi}^\top(x_i)\|$ as

$$\begin{aligned}
&\mathbb{E}_{x_i}\|\mathbb{E}_x\tilde{\phi}(x)\tilde{\phi}^\top(x) - n^{-1}\sum_i \tilde{\phi}(x_i)\tilde{\phi}^\top(x_i)\| \\
&\leq \sqrt{\mathbb{E}_{x_i}\|\mathbb{E}_x\tilde{\phi}(x)\tilde{\phi}^\top(x) - n^{-1}\sum_i \tilde{\phi}(x_i)\tilde{\phi}^\top(x_i)\|_F^2} \\
&= \sqrt{\mathbb{E}_{x_i} n^{-2}\sum_i \|\mathbb{E}_x\tilde{\phi}(x)\tilde{\phi}^\top(x) - \tilde{\phi}(x_i)\tilde{\phi}^\top(x_i)\|_F^2} \\
&\leq \sqrt{\frac{2(d+1)}{n}}
\end{aligned}$$

as $\|\tilde{\phi}(x)\| \leq d+1$. Hence

$$\begin{aligned}
&\|\mathbb{E}_x\tilde{\phi}(x)\tilde{\phi}^\top(x) - n^{-1}\sum_i \tilde{\phi}(x_i)\tilde{\phi}^\top(x_i)\| \\
&= O\left(\sqrt{\frac{(d+1)\log(1/\delta)}{n}}\right) \text{ w.p. mote than } 1-\delta
\end{aligned} \tag{C.20}$$

We will apply McDiarmid's inequality to the second term in (C.19) as well. Since $\|n^{-1}\sum_i \tilde{\phi}(x_i)\|$ changes by a maximum of $2(d+1)/n$ by changing one of the $\tilde{\phi}(x_i)$ with an i.i.d copy

$$\begin{aligned}
&\|n^{-1}\sum_i \tilde{\phi}(x_i)\| \\
&\leq \mathbb{E}_{x_i}\|n^{-1}\sum_i \tilde{\phi}(x_i)\| + \sqrt{\frac{(d+1)\log(1/\delta)}{n}} \text{ w.p. mote than } 1-\delta
\end{aligned} \tag{C.21}$$

The mean $\mathbb{E}_{x_i}\|n^{-1}\sum_i \tilde{\phi}(x_i)\| \leq \sqrt{\mathbb{E}_{x_i}\|n^{-1}\sum_i \tilde{\phi}(x_i)\|^2} \leq \sqrt{\frac{d+1}{n}}$. Putting this together with (C.21)

$$\|n^{-1}\sum_i \tilde{\phi}(x_i)\| \leq \sqrt{\frac{d+1}{n}} + \sqrt{\frac{(d+1)\log(1/\delta)}{n}} \text{ w.p. mote than } 1-\delta \tag{C.22}$$

Combining the result in (C.22) and (C.20) with (C.19), we arrive at the bound on $E_x(y - f_{KR}(x))^2$. Finally, using the bound on $E_x(y - f(x, w(k)))^2$ from Lemma 6 we arrive at the result.

C.7 Proof of Corollary 1

If $y = \left(x^\top \beta\right)^p, \|\beta\| \leq 1$ then y can be equivalently written as

$$y = (\sqrt{d_p} x^{\otimes p})^\top \frac{1}{d_p} \beta^{\otimes p} = \phi^\top(x)\bar{w}, \bar{w} = [0, \ldots, 0, \frac{1}{\sqrt{d_p}} \beta^{\otimes p}, 0, \ldots]$$

In order to apply Theorem 5 we need to compute an upper bound on $\|\bar{w}\|^2 = \frac{\|\beta\|^{2p}}{d_p}$. It boils down to computing a lower bound on d_p. $d_p \geq 0.25$ for $p = 0, 1$. For $p \geq 2$ $d_p = \sum_{p' \geq \lceil p/2 \rceil} \frac{c_{2p'}}{(d+1)^{2p'}} \binom{2p'}{p}$. Using the results in [80] to lower bound the ratio of double factorial,

$$c_{2p'} = \frac{(2p' - 3)!!(d+1)}{2\pi(2p' - 2)!!(2p' - 1)} \geq \frac{d+1}{10(2p')^{1.5}}$$

Combining this with the bound $\binom{2p'}{k} \geq \left(\frac{2p'}{k}\right)^k$

$$\sum_{p' \geq \lceil p/2 \rceil} \frac{c_{2p'}}{(d+1)^{2p'}} \binom{2p'}{p} \geq \sum_{p' \geq \lceil p/2 \rceil} \frac{1}{10(2p')^{1.5}(d+1)^{2p'-1}} \left(\frac{2p'}{p}\right)^p$$
$$\geq \frac{1}{10(p+1)^{1.5}(d+1)^p}$$

Hence $\|\bar{w}\|^2 \leq 10(p+1)^{1.5}(d+1)^p$ and we arrive at the result.

REFERENCES

[1] H. K. Khalil and J. W. Grizzle, *Nonlinear Systems*. Prentice Hall Upper Saddle River, NJ, 2002, vol. 3.

[2] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 2018.

[3] C. Callegari, S. Giordano, M. Pagano, and T. Pepe, "Detecting anomalies in backbone network traffic: A performance comparison among several change detection methods," *International Journal of Sensor Networks*, vol. 11, no. 4, pp. 205–214, 2012.

[4] T. Li, L. Shwartz, and G. Y. Grabarnik, "System event mining: Algorithms and applications," *KDD 2017 Tutorial*, 2017. [Online]. Available: https://users.cs.fiu.edu/~taoli/event-mining/

[5] T. Li, C. Zeng, Y. Jiang, W. Zhou, L. Tang, Z. Liu, and Y. Huang, "Data-driven techniques in computing system management," *ACM Comput. Surv.*, vol. 50, no. 3, pp. 45:1–45:43, July 2015. [Online]. Available: http://doi.acm.org/10.1145/3092697

[6] S. Satpathi, S. Deb, R. Srikant, and H. Yan, "Learning latent events from network message logs," *IEEE/ACM Transactions on Networking*, vol. 27, no. 4, pp. 1728–1741, 2019.

[7] D. S. Matteson and N. A. James, "A nonparametric approach for multiple change point analysis of multivariate data," *Journal of the American Statistical Association*, vol. 109, no. 505, pp. 334–345, 2014. [Online]. Available: https://doi.org/10.1080/01621459.2013.849605

[8] A. A. Makanju, A. N. Zincir-Heywood, and E. E. Milios, "Clustering event logs using iterative partitioning," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '09. New York, NY, USA: ACM, 2009. [Online]. Available: http://doi.acm.org/10.1145/1557019.1557154 pp. 1255–1264.

[9] T. Li, F. Liang, S. Ma, and W. Peng, "An integrated framework on mining logs files for computing system management," in *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, ser. KDD '05. New York, NY, USA: ACM, 2005. [Online]. Available: http://doi.acm.org/10.1145/1081870.1081972 pp. 776–781.

[10] L. Tang and T. Li, "LogTree: A framework for generating system events from raw textual logs," in *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, 2010, pp. 491–500.

[11] T. Qiu, Z. Ge, D. Pei, J. Wang, and J. Xu, "What happened in my network: Mining network events from router syslogs," in *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '10. New York, NY, USA: ACM, 2010. [Online]. Available: http://doi.acm.org/10.1145/1879141.1879202 pp. 472–484.

[12] F. Wu, P. Anchuri, and Z. Li, "Structural event detection from log messages," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '17. New York, NY, USA: ACM, 2017. [Online]. Available: http://doi.acm.org/10.1145/3097983.3098124 pp. 1175–1184.

[13] R. Agrawal and R. Srikant, "Mining sequential patterns," in *Proceedings of the Eleventh International Conference on Data Engineering*, ser. ICDE '95. Washington, DC, USA: IEEE Computer Society, 1995. [Online]. Available: http://dl.acm.org/citation.cfm?id=645480.655281 pp. 3–14.

[14] D. Cheng, M. T. Bahadori, and Y. Liu, "FBLG: A simple and effective approach for temporal dependence discovery from time series data," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '14. New York, NY, USA: ACM, 2014. [Online]. Available: http://doi.acm.org/10.1145/2623330.2623709 pp. 382–391.

[15] C. Zeng, Q. Wang, W. Wang, T. Li, and L. Shwartz, "Online inference for time-varying temporal dependency discovery from time series," in *Big Data (Big Data), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1281–1290.

[16] C. H. Mooney and J. F. Roddick, "Sequential pattern mining – Approaches and algorithms," *ACM Comput. Surv.*, vol. 45, no. 2, pp. 19:1–19:39, Mar. 2013. [Online]. Available: http://doi.acm.org/10.1145/2431211.2431218

[17] J. A. Silva, E. R. Faria, R. C. Barros, E. R. Hruschka, A. C. P. L. F. d. Carvalho, and J. a. Gama, "Data stream clustering: A survey," *ACM Comput. Surv.*, vol. 46, no. 1, pp. 13:1–13:31, July 2013. [Online]. Available: http://doi.acm.org/10.1145/2522968.2522981

[18] Y. Jiang, C.-S. Perng, and T. Li, "Natural event summarization," in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, ser. CIKM '11. New York, NY, USA: ACM, 2011. [Online]. Available: http://doi.acm.org/10.1145/2063576.2063688 pp. 765–774.

[19] P. Wang, H. Wang, M. Liu, and W. Wang, "An algorithmic approach to event summarization," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '10. New York, NY, USA: ACM, 2010. [Online]. Available: http://doi.acm.org/10.1145/1807167.1807189 pp. 183–194.

[20] W. Peng, C. Perng, T. Li, and H. Wang, "Event summarization for system management," in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '07. New York, NY, USA: ACM, 2007. [Online]. Available: http://doi.acm.org/10.1145/1281192.1281305 pp. 1028–1032.

[21] N. Tatti and J. Vreeken, "The long and the short of it: Summarising event sequences with serial episodes," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '12. New York, NY, USA: ACM, 2012. [Online]. Available: http://doi.acm.org/10.1145/2339530.2339606 pp. 462–470.

[22] M. Du, F. Li, G. Zheng, and V. Srikumar, "DeepLog: Anomaly detection and diagnosis from system logs through deep learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '17. New York, NY, USA: ACM, 2017. [Online]. Available: http://doi.acm.org/10.1145/3133956.3134015 pp. 1285–1298.

[23] Y. Kawahara and M. Sugiyama, "Sequential change-point detection based on direct density-ratio estimation," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 5, no. 2, pp. 114–127, 2012.

[24] A. Lung-Yut-Fong, C. Lévy-Leduc, and O. Cappé, "Homogeneity and change-point detection tests for multivariate data using rank statistics," *arXiv preprint arXiv:1107.1971*, 2011.

[25] H. Chen and N. Zhang, "Graph-based change-point detection," *Ann. Statist.*, vol. 43, no. 1, pp. 139–176, 02 2015. [Online]. Available: https://doi.org/10.1214/14-AOS1269

[26] X. Wang and A. McCallum, "Topics over time: A non-Markov continuous-time model of topical trends," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '06. New York, NY, USA: ACM, 2006. [Online]. Available: http://doi.acm.org/10.1145/1150402.1150450 pp. 424–433.

[27] C. Wang, D. Blei, and D. Heckerman, "Continuous time dynamic topic models," *arXiv preprint arXiv:1206.3298*, 2012.

[28] T. L. Griffiths and M. Steyvers, "Finding scientific topics," *Proceedings of the National Academy of Sciences*, vol. 101, no. suppl 1, pp. 5228–5235, 2004.

[29] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=944919.944937

[30] M. Hoffman, F. R. Bach, and D. M. Blei, "Online learning for Latent Dirichlet Allocation," in *Advances in Neural Information Processing Systems*, 2010, pp. 856–864.

[31] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, "Stochastic variational inference," *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 1303–1347, 2013.

[32] A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, and M. Telgarsky, "Tensor decompositions for learning latent variable models," *Journal of Machine Learning Research*, vol. 15, pp. 2773–2832, 2014. [Online]. Available: http://jmlr.org/papers/v15/anandkumar14b.html

[33] T. Bansal, C. Bhattacharyya, and R. Kannan, "A provable SVD-based algorithm for learning topics in dominant admixture corpus," in *Advances in Neural Information Processing Systems 27*, 2014, pp. 1997–2005.

[34] C. Wang, J. Paisley, and D. Blei, "Online variational inference for the hierarchical dirichlet process," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 2011, pp. 752–760.

[35] E. Sy, S. A. Jacobs, A. Dagnino, and Y. Ding, "Graph-based clustering for detecting frequent patterns in event log data," in *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, Aug 2016, pp. 972–977.

[36] D. Li, T. Ding, and R. Sun, "On the benefit of width for neural networks: Disappearance of bad basins," *arXiv preprint arXiv:1812.11039*, 2018.

[37] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, "Visualizing the loss landscape of neural nets," in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 6389–6399. [Online]. Available: http://papers.nips.cc/paper/7875-visualizing-the-loss-landscape-of-neural-nets.pdf

[38] Y. Cooper, "The loss landscape of overparameterized neural networks," *arXiv preprint arXiv:1804.10200*, 2018.

[39] S. Liang, R. Sun, Y. Li, and R. Srikant, "Understanding the loss surface of neural networks for binary classification," in *International Conference on Machine Learning*, 2018, pp. 2835–2843.

[40] S. Liang, R. Sun, and R. Srikant, "Revisiting landscape analysis in deep neural networks: Eliminating decreasing paths to infinity," *arXiv preprint arXiv:1912.13472*, 2019.

[41] S. Arora, S. S. Du, W. Hu, Z. Li, and R. Wang, "Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks," *arXiv preprint arXiv:1901.08584*, 2019.

[42] G. Zhang, J. Martens, and R. B. Grosse, "Fast convergence of natural gradient descent for overparameterized neural networks," in *Advances in Neural Information Processing Systems*, 2019, pp. 8080–8091.

[43] W. Hu, Z. Li, and D. Yu, "Simple and effective regularization methods for training on noisily labeled data with generalization guarantee," in *International Conference on Learning Representations*, 2020.

[44] Z. Ji and M. Telgarsky, "Polylogarithmic width suffices for gradient descent to achieve arbitrarily small test error with shallow ReLU networks," *arXiv preprint arXiv:1909.12292*, 2019.

[45] Z. Chen, Y. Cao, D. Zou, and Q. Gu, "How much overparameterization is sufficient to learn deep ReLU networks?" *arXiv preprint arXiv:1911.12360*, 2019.

[46] Y. Cao and Q. Gu, "Generalization error bounds of gradient descent for learning overparameterized deep ReLU networks," *arXiv preprint arXiv:1902.01384*, 2019.

[47] D. Zou and Q. Gu, "An improved analysis of training overparameterized deep neural networks," in *Advances in Neural Information Processing Systems*, 2019, pp. 2053–2062.

[48] Z. Allen-Zhu, Y. Li, and Z. Song, "A convergence theory for deep learning via overparameterization," in *International Conference on Machine Learning*, 2019, pp. 242–252.

[49] D. Zou, Y. Cao, D. Zhou, and Q. Gu, "Stochastic gradient descent optimizes overparameterized deep ReLU networks," *arXiv preprint arXiv:1811.08888*, 2018.

[50] Z. Allen-Zhu, Y. Li, and Y. Liang, "Learning and generalization in overparameterized neural networks, going beyond two layers," in *Advances in Neural Information Processing Systems*, 2019, pp. 6155–6166.

[51] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.

[52] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034.

[53] A. Krogh and J. A. Hertz, "A simple weight decay can improve generalization," in *Advances in Neural Information Processing Systems*, 1992, pp. 950–957.

[54] L. Chizat, E. Oyallon, and F. Bach, "On lazy training in differentiable programming," in *Advances in Neural Information Processing Systems*, 2019, pp. 2933–2943.

[55] A. Jacot, F. Gabriel, and C. Hongler, "Neural Tangent Kernel: Convergence and generalization in neural networks," in *Advances in Neural Information Processing Systems*, 2018, pp. 8571–8580.

[56] S. S. Du, X. Zhai, B. Poczos, and A. Singh, "Gradient descent provably optimizes overparameterized neural networks," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=S1eK3i09YQ

[57] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. MIT Press, 2018.

[58] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[59] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[60] K. Hornik, "Some new results on neural network approximation," *Neural Networks*, vol. 6, no. 8, pp. 1069–1072, 1993.

[61] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals and Systems*, vol. 2, no. 4, pp. 303–314, 1989.

[62] J. Lee, L. Xiao, S. Schoenholz, Y. Bahri, R. Novak, J. Sohl-Dickstein, and J. Pennington, "Wide neural networks of any depth evolve as linear models under gradient descent," in *Advances in Neural Information Processing Systems*, 2019, pp. 8572–8583.

[63] S. Du, J. Lee, H. Li, L. Wang, and X. Zhai, "Gradient descent finds global minima of deep neural networks," in *International Conference on Machine Learning*, 2019, pp. 1675–1685.

[64] Z. Allen-Zhu, Y. Li, and Y. Liang, "Learning and generalization in overparameterized neural networks, going beyond two layers," *arXiv preprint arXiv:1811.04918*, 2018.

[65] S. Oymak and M. Soltanolkotabi, "Towards moderate overparameterization: Global convergence guarantees for training shallow neural networks," *IEEE Journal on Selected Areas in Information Theory*, 2020.

[66] Z. Ji and M. Telgarsky, "Polylogarithmic width suffices for gradient descent to achieve arbitrarily small test error with shallow ReLU networks," *arXiv preprint arXiv:1909.12292*, 2019.

[67] Z. Chen, Y. Cao, D. Zou, and Q. Gu, "How much overparameterization is sufficient to learn deep ReLU networks?" *arXiv preprint arXiv:1911.12360*, 2019.

[68] B. Neyshabur, "Implicit regularization in deep learning," *arXiv preprint arXiv:1709.01953*, 2017.

[69] S. Oymak and M. Soltanolkotabi, "Overparameterized nonlinear learning: Gradient descent takes the shortest path?" in *International Conference on Machine Learning*, 2019, pp. 4951–4960.

[70] Z. Ji and M. Telgarsky, "Risk and parameter convergence of logistic regression," *arXiv preprint arXiv:1803.07300*, 2018.

[71] D. Soudry, E. Hoffer, M. S. Nacson, S. Gunasekar, and N. Srebro, "The implicit bias of gradient descent on separable data," *The Journal of Machine Learning Research*, vol. 19, no. 1, pp. 2822–2878, 2018.

[72] L. Chizat and F. Bach, "Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss," *arXiv preprint arXiv:2002.04486*, 2020.

[73] B. Ghorbani, S. Mei, T. Misiakiewicz, and A. Montanari, "Linearized two-layers neural networks in high dimension," *arXiv preprint arXiv:1904.12191*, 2019.

[74] P. L. Bartlett, P. M. Long, G. Lugosi, and A. Tsigler, "Benign overfitting in linear regression," *Proceedings of the National Academy of Sciences*, 2020.

[75] Z. Ji, M. Telgarsky, and R. Xian, "Neural Tangent Kernels, transportation mappings, and universal approximation," *arXiv preprint arXiv:1910.06956*, 2019.

[76] A. Krogh and J. A. Hertz, "A simple weight decay can improve generalization," in *Advances in Neural Information Processing Systems*, 1992, pp. 950–957.

[77] S. Liang, R. Sun, and R. Srikant, "Achieving small test error in mildly overparameterized neural networks," *arXiv preprint arXiv:2104.11895*, 2021.

[78] J. A. Tropp, "User-friendly tail bounds for sums of random matrices," *Foundations of Computational Mathematics*, vol. 12, no. 4, pp. 389–434, 2012.

[79] S. Satpathi, H. Gupta, S. Liang, and R. Srikant, "The role of regularization in overparameterized neural networks," 2020. [Online]. Available: https://drive.google.com/drive/folders/1McH_Cb7b7ct89bQFBNt38Q1-7C5cNvjW

[80] C.-P. Chen and F. Qi, "The best bounds in Wallis' inequality," *Proceedings of the American Mathematical Society*, pp. 397–401, 2005.

[81] V. Koltchinskii and K. Lounici, "Concentration inequalities and moment bounds for sample covariance operators," *arXiv preprint arXiv:1405.2468*, 2014.