Kernel Methods for Learning with Limited Labeled Data

by

Aniket Anand Deshmukh

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

Doctoral Committee:

Associate Professor Clayton Scott, Chair
Professor Alfred O. Hero, III
Assistant Professor Eric Schwartz
Associate Professor Ambuj Tewari

Aniket Anand Deshmukh

aniketde@umich.edu

ORCID iD: 0000-0002-7292-8436

To my parents

ACKNOWLEDGEMENTS

I would like to begin my thesis by thanking several mentors and friends who made this thesis possible. First, I would like to thank my advisor Prof. Scott with whom I worked for last five years. Prof. Scott has been an inspiration to me right from the start of my doctoral program. Prof. Scott was patient with me, and whenever I did not understand anything, he would start from very simple ideas and using them explain me the complex ones. During the last five years, we have had long sessions discussing proofs and brainstorming ideas. I am thankful to Prof. Scott for giving me a chance to work with him and investing many subsequent hours on me. Prof. Scott's openness to new ideas helped me think independently. Prof. Scott is very particular about thoroughness and depth of research in hand. He also helped me improve my writing and communication skills. I thank Prof. Scott for making me a good machine learning researcher, and I will always strive to do thorough research in the future.

I thank the committee members for timely feedback and giving me suggestions whenever I needed them. I thank Prof. Ambuj Tewari for helping me solve some of the queries I had about contextual bandits at the start of my research. His course - “Sequential Decision Making with mHealth Applications" boosted my research in the area of bandits. I also thank Prof. Eric Schwartz for helpful discussions on best arm identification. Prof. Eric helped me understand real-world applications of bandits. Prof. Alfred Hero's work in the area of sensor network has been helpful for me to understand the best sensor selection problem.

I would like to thank several members of the administrative staff at the University of

Michigan. To Becky Turanski, José-Antonio Rubio, Anne Rhoades, Karen Liska, Kristen Thornton, Amy Wicklund, and Judi Jones: thank you for helping me.

I would also like to thank my undergraduate professors at IIT Hyderabad, Prof. K Sri Rama Murty, Prof. Sumohana Channappayya, Prof. Soumya Jana, Prof. Balasubramaniam Jayaram and Prof. Mohammed Zafar Ali Khan for encouraging me to apply for PhD and instilling research ideology in me early in my undergraduate studies.

Next, I would like to thank Dr. Urun Dogan for the collaboration and his immense help in formulating problems during my PhD. Urun's enthusiasm for innovation in machine learning research and discussions with him on research papers have helped me understand fundamental challenges in machine learning.

I was introduced to online learning and bandits by Prof. Jacob Abernethy in his course "Theoretical Foundations of Machine Learning" in the second year of my PhD. I thank Jake for motivating me to explore the area of bandits. I thank Prof. Raj Rao Nadakuditi, Prof. Sandeep Pradhan and Prof. Laura Balzano for making my fundamentals stronger through various courses during PhD, which helped me conduct my research better.

I would like to thank my collaborator and friend Srinagesh for working with me during PhD. Discussions on mathematics, machine learning and interdisciplinary work with Srinagesh helped me broaden my horizons.

I thank David Hong and John Lipor for introducing me to new ideas and philosophies during my stay at Ann Arbor. I learned a lot through our exchanges on mathematics and philosophy. I also learned the importance of family and effective time management from John. I thank Matthew Kvalheim for collaborating with me and teaching me nonlinear dynamics and control systems. I also would like to thank Aditya Modi and Julian Katz-Samuels for discussing research areas in bandits and introducing me to new techniques of analyzing bandit algorithms. I thank Vikas Dhiman for studying "Regret Analysis of Stochastic and

Nonstochastic Multi-Armed Bandit Problems" book with me. Our study sessions helped me understand the proofs of various bandit algorithms. Our exchange of thoughts about the education system, activism has also helped me understand this world better.

I thank Abhishek Bafna, Sri Sharan Banagiri, Arnab Hazari, Niket Prakash, Naveen Murthy, Pragya Agrawal, Poornashree Rajendra and Kalyan Tej for being there for me during my ups and downs. Without all of you, my time at Ann Arbor would have never been so much fun. I had a good time with our dinner parties, political discussions and playing squash & racketball. I thank Abhishek Bafna, Naveen Murthy and Pragya Agrawal for collaborating with me on various course projects and teaching me various concepts during coursework.

I would like to thank Chaitanya Kulkarni, Sachin Rathod and Pravin Shedolkar for fun conversations and supporting me in my ambitions. I thank Omkar for being there for me and helping me during my time at Ann Arbor. I learned how to have fun in life and how to be a good cook from Omkar. Next, I would like to thank Piyush who has been a great support for me during the last 12 years and guided me on most major decisions in life. I would also like to thank Aditya who encouraged me to study speech processing and machine learning during my undergraduate. I have always had great technical discussions with Aditya, and he helped me keep up with the machine learning literature during my PhD.

I would like to thank my girlfriend Mansi for encouraging me during my bad times and supporting me to finish my PhD. This journey has been fun and easier with Mansi in my life. I would also like to thank my uncles Shankar and Dhananjay and my grandfather Dnyanoba for nurturing my interest in Science and Technology. Most importantly I would like to thank my brother Anup and my parents Ratnamala and Anand. Without their constant love, support and guidance, I would not have been able to pursue higher education and complete my PhD.

TABLE OF CONTENTS

DEDICATION ii

ACKNOWLEDGEMENTS iii

LIST OF TABLES x

LIST OF FIGURES xi

ABSTRACT xiii

CHAPTER

I Introduction 1

I.1 Background 2

I.1.1 Domain Generalization 2

I.1.2 Multi-Task Learning for Contextual Bandits 4

I.1.3 Simple Regret for Contextual Bandits 7

I.2 Contribution 8

II Domain Generalization 10

II.1 Introduction 11

II.2 Motivating Application: Automatic Gating of Flow Cytometry Data 12

II.3 Formal Setting 13

II.4 Learning Algorithm 14

II.4.1 Specifying the kernels 15
II.4.2 Relation to other kernel methods 17
II.5 Implementation 18
II.5.1 Representer theorem and hinge loss 18
II.5.2 Approximate Feature Mapping for Scalable Implementation 20
II.6 Experiments 27
II.6.1 Model Selection 28
II.6.2 Parkinson's disease telemonitoring dataset 28
II.6.3 Satellite Classification 29
II.6.4 Flow Cytometry Experiments 31
II.7 Multiclass Domain Generalization 36
II.7.1 Generalization Error Analysis 36
II.7.2 Experimental Results 38
II.8 Conclusion 40
II.9 Proofs 41
II.9.1 Proof of Theorem 2 41
II.9.2 Proof of Theorem 3 43
II.9.3 Proof of Theorem 4 44

III Multi-Task Learning for Contextual Bandits 51

III.1 Introduction 51
III.2 Related Work 54
III.3 KMTL-UCB 55
III.3.1 Upper Confidence Bound 56
III.3.2 Choice of Task Similarity Space and Kernel 58
III.4 Theoretical Analysis 60
III.4.1 Analysis of SupKMTL-UCB 60
III.4.2 Interpretation of Regret Bound 61

III.4.3 Comparison with CGP-UCB 63
III.5 Experiments 64
III.5.1 Synthetic News Article Data 64
III.5.2 Multi-class Datasets 65
III.6 Conclusions and future work 66
III.7 Proofs 67
III.7.1 KMTL Ridge Regression 67
III.7.2 Upper Confidence Bound 69
III.7.3 UCB Width 73
III.7.4 Regret Analysis 77
III.7.5 Proof of Theorem 6 83
III.7.6 Proof of Theorem 7 86
III.7.7 Proof of Theorem 8 86
III.7.8 Proof of Corollary 1 88

IV Simple Regret for Contextual Bandits **90**

IV.1 Introduction 91
IV.2 Motivation 93
IV.3 Formal Setting 96
IV.3.1 Problem Statement 97
IV.4 Algorithm 97
IV.4.1 Estimating Expected Rewards 98
IV.4.2 Contextual-Gap Algorithm 99
IV.4.3 Comparison of Contextual-Gap and Kernel-UCB 100
IV.5 Learning Theoretic Analysis 103
IV.6 Experimental Results and Discussion 107
IV.6.1 Synthetic Dataset 109
IV.6.2 Experimental Spacecraft Magnetic Field Dataset 110

IV.7 Conclusion 113

IV.8 Proofs 113

IV.8.1 Probabilistic Setting and Martingale Lemma 113

IV.8.2 Lower Bound on r^{th} Eigenvalue 115

IV.8.3 Monotonic Upper bound of $s_{a,t}(x)$ 123

IV.8.4 Simple Regret Analysis 127

V Conclusion and Future Work **140**

Bibliography **143**

LIST OF TABLES

TABLE

1.1 Contribution: Simple regret minimization for contextual bandits 7

2.1 RMSE of Marginal Transfer Learning on Parkinson's Disease Dataset 29

2.2 RMSE of Pooling on Parkinson's Disease Dataset 30

2.3 Average Classification Error of Marginal Transfer Learning on Satellite Dataset 32

2.4 Average Classification Error of Pooling on Satellite Dataset 32

2.5 Average Classification Error of Marginal Transfer Learning on Flow Cytometry Dataset 34

2.6 Average Classification Error of Pooling on Flow Cytometry Dataset 34

2.7 Summary of Multiclass Datasets for Domain Generalization 38

2.8 Percentage Classification Error on Multiclass Datasets 40

LIST OF FIGURES

FIGURE

1.1 Multi-armed Bandit Problem 4

1.2 Contextual Bandit Problem 6

2.1 Parkinson's disease telemonitoring dataset 31

2.2 Satellite dataset 33

2.3 Classification error rates for baseline and proposed method for different experimental settings, i.e., number of examples per task and number of tasks. . . 35

2.4 Synthetic Dataset: Three tasks $\theta = \{0, 90, 180\}$ 39

2.5 Synthetic Dataset: Thirteen tasks $\theta = \{0, 15, 30, ..., 180\}$ 39

2.6 MNIST Data with no rotation (first row) and 90 degree rotation (second row) 40

3.1 Synthetic Data 65

3.2 Results on Multiclass Datasets - Empirical Mean Regret 66

4.1 Scientific measurement: magnetic field lines of the Earth (Credit: NASA/Goddard Scientific Visualization Studio) 93

4.2 TBEx Small Satellite with Multiple Magnetometers [1, 2] 95

4.3 Contextual-Gap exploration policy: case 1 100

4.4 Contextual-Gap exploration policy: case 2 101

4.5 Average Simple Regret Evaluation on Synthetic Dataset 110

4.6 Average Simple Regret Evaluation on Spacecraft Magnetic Field Dataset . . 110

4.7 Histogram of Arm Selection during exploration 111

4.8 Worst Case Simple Regret Evaluation on Spacecraft Magnetic Field Dataset 112

ABSTRACT

Machine learning is a rapidly developing technology that enables a system to automatically learn and improve from experience. Modern machine learning algorithms have achieved state-of-the-art performances on a variety of tasks such as speech recognition [3], image classification [4], machine translation [5], playing games like Go [6], Dota 2 [7], etc. However, one of the biggest challenges in applying these machine learning algorithms in the real world is that they require huge amount of labeled data for the training. In the real world, the amount of labeled training data is often limited.

In this thesis, we address three challenges in learning with limited labeled data using kernel methods. In our first contribution, we provide an efficient way to solve an existing domain generalization algorithm and extend the theoretical analysis to multiclass classification. As a second contribution, we propose a multi-task learning framework for contextual bandit problems. We propose an upper confidence bound-based multi-task learning algorithm for contextual bandits, establish a corresponding regret bound, and interpret this bound to quantify the advantages of learning in the presence of high task (arm) similarity. Our third contribution is to provide a simple regret guarantee (best policy identification) in a contextual bandits setup. Our experiments examine a novel application to adaptive sensor selection for magnetic field estimation in interplanetary spacecraft and demonstrate considerable improvements of our algorithm over algorithms designed to minimize the cumulative regret.

CHAPTER I

Introduction

Machine learning is a rapidly developing technology and has the potential to solve many issues in speech processing, natural language processing, robotics, autonomous cars, and fields where data analysis is essential. Machine learning enables systems to automatically learn and improve from the experience without manually programming it for every scenario. Modern machine learning algorithms have achieved state-of-the-art performances on variety of tasks such as speech recognition [3], image classification[4], machine translation [5], playing games like Go [6], Dota 2 [7] ,etc. However, one of the biggest challenges in applying these machine learning algorithms in the real world is that they require huge amount of labeled data for the training. In real world, we are limited by the labeled data that's available during training, and which may also be distributionally different from the test data.

Consider an example of categorizing blood cells of a patient in two types, lymphocytes and non-lymphocytes. Doctors measure various physical and chemical properties of a cell using flow cytometry and then based on these properties, doctors have to manually label each cell into two types. Creating such a labeled dataset for learning algorithm is a very time consuming and expensive process. Also, physical and chemical properties of each blood cell may vary according to patient and that's why data used during training could be distributionally different than test data. One more example where collecting labeled data is very expensive is clinical trials. In clinical trials, doctors have number of options for drugs or type of treatments. Doctors try these options on various patients and check how effective the particular option is; which is equivalent to collecting labeled data or training data. Doctors have to minimize number of trials because trying wrong treatment could have adverse effects on patients. In this case, collecting labeled data is not only expensive but is also life threatening.

I.1 Background

In this section, I describe challenges addressed in the thesis and give necessary background. Specifically, I explore three main scenarios where I address issues that arise due to limited labeled data and solve these issues using kernel methods.

I.1.1 Domain Generalization

Transfer learning, domain adaptation, and weakly supervised learning all have the goal of generalizing without access to conventional labeled training data. One particular form of transfer learning that has garnered increasing attention in recent years is *domain generalization* (DG) [8, 9]. In this setting, the learner is given unlabeled data to classify, and must do so by leveraging labeled data sets from similar yet distinct classification problems. In other words, label training data drawn from the same distribution as the test data are not available, but are available from several related tasks (which may have slightly different distribution). We use the terms “task" and “domain" interchangeably throughout this chapter.

Applications of DG are numerous. For example, each task may be a prediction problem associated to a particular individual (e.g., handwritten digit recognition), and the variation between individuals accounts for the variation among the data sets. Domain generalization is needed when a new individual appears, and the only training data come from different subjects.

Consider the example of image classification, where one has images of different objects. Each image contains a single object and the goal is recognize or classify these images based on an object in it. We have multiple images of the same objects from different cameras, and we train our image classification model for these cameras (e.g., Apple's iPhone, Sony's DSLR, and Google's Pixel). During the test time, the goal is to classify images from Samsung's Note. Different cameras have different optical structure and so images from those cameras may look slightly different or may have different optical properties. What makes this more difficult is that there are no labeled images from Samsung's Note; and the classification model should be able to classify images from Samsung's Note without any labeled data.

As another application, we consider domain generalization for determining the orbits of microsatellites, which are increasingly deployed in space missions for a variety of scientific and technological purposes. Because of randomness in the launch process, the orbit of a microsatellite is random, and must be determined after the launch. Furthermore, ground antennae are not able to decode unique identifier signals transmitted by the microsatellites because of communication resource constraints and uncertainty in satellite position and dynamics. More concretely, suppose c microsatellites are launched together. Each launch is a random phenomenon and may be viewed as a task in our framework. One can simulate the launch of microsatellites using domain knowledge to generate highly realistic training data (feature vectors of ground antennae RF measurements, and labels of satellite ID). One can then transfer knowledge from the simulated training data to label (identify the satellite) the measurements from a real-world launch with high accuracy.

Domain generalization is the problem of assigning labels to an unlabeled data set, given several similar data sets for which labels have been provided [10]. More specifically, in domain generalization, the learning algorithm has N datasets during training with each dataset drawn from different probability distributions such that each point in each dataset has a label or class associated with it. The goal is to learn a classifier such that, given a new dataset (with no training data/labels) drawn from a different but similar probability distribution, it is possible to provide labels to its points.

I.1.2 Multi-Task Learning for Contextual Bandits

A multi-armed bandit (MAB) problem is a sequential decision-making problem where, at each time step, an agent chooses one of several “arms," and observes high reward for choosing the correct arm and smaller reward if it chooses some other arm. The name "multi-armed bandit" arises from an imaginary gambler who has access to number of slot machines. Slot machines here are also called "one-armed bandits". In order to achieve a goal of maximizing money in hand in certain number of trials, the gambler has to decide how many times to play each machine and in which order to play those machines [11].

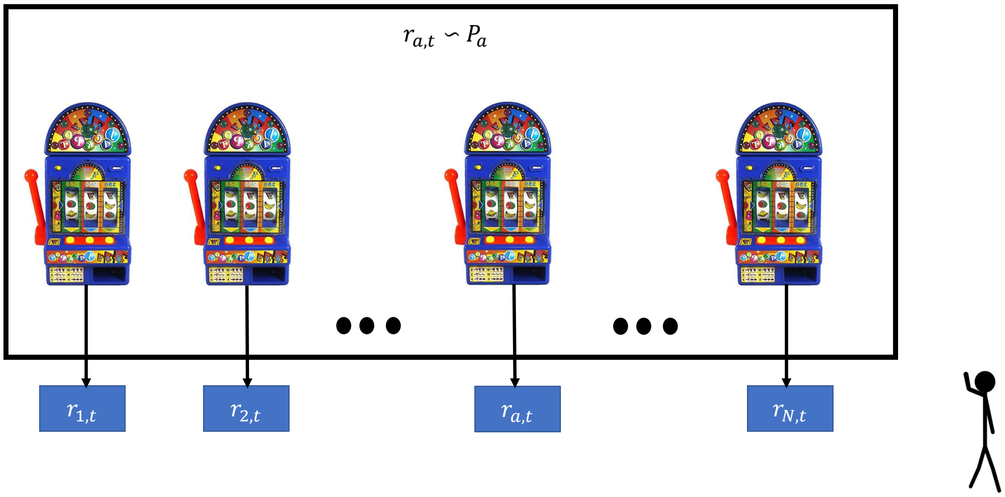


Figure 1.1: Multi-armed Bandit Problem

More formally, the gambler here is called the learner, each slot machine is an arm and the problem of making decision of choosing arms is called the multi-armed bandit (MAB) problem. The “regret" of the learner is the difference between the maximum possible reward and the reward resulting from the chosen action. The reward for each arm is random according to a fixed distribution, and the learner's goal is to either maximize its cumulative reward or minimize cumulative regret [12] through a combination of exploring different arms and exploiting those arms that have yielded high rewards in the past [13, 14]. For example, in Fig. 1.1, there are N arms to choose from, reward $r_{a,t}$ for each arm a at time t is sampled from a probability distribution P_a. In this case, the goal is to minimize cumulative regret $R_T = \max_{a \in [N]} \sum_{t=1}^{T} r_{a,t} - \sum_{t=1}^{T} r_{a_t,t}$, where $[N] = 1, \ldots, N$, a_t is the arm selected by a learner at time t and T are number of trials. If the learner explores too little, it may never find an optimal arm, which will in consequence increase its cumulative regret. If the learner explores too much, it may select sub-optimal arms too often which will also increase its cumulative regret.

The contextual bandit problem is an extension of the MAB problem where there is some side information, here called the context, associated with each arm [15]. The contextual bandit setting is also called associative reinforcement learning [16] and linear bandits [17, 18]. In Fig. 1.2, there is a context $x_{a,t}$ for an arm a at time t. The expected reward for each arm a given a context $x_{a,t}$ is some fixed but unknown function of $x_{a,t}$. More formally, $\mathbb{E}[r_{a_t,t} | x_{a,t}] = f_a(x_{a,t})$.

Contextual bandits have been used to model personalized news recommendations, ad placements, and other applications. Each context determines the distribution of rewards for the associated arm. The goal, therefore, in contextual bandits is still to maximize the cumulative reward or minimize cumulative regret, but now leveraging the contexts to predict the expected reward of each arm. i.e. $R_T = \sum_{t=1}^{T} r_{a_t^*,t} - \sum_{t=1}^{T} r_{a_t,t}$, where a_t^* is the arm with maximum reward at trial t. Note that arm with maximum reward in contextual bandits depends on the context unlike in MAB. Contextual bandits have been employed to model various applications like news article recommendations [19], computational advertisements

[20], website optimization [21] and clinical trials [22]. For example, in the case of a news article recommendation, the agent must select a news article to recommend to a particular user. The arms are articles, and contextual features are features derived from the article and the user. The reward is based on whether a user reads the recommended article.

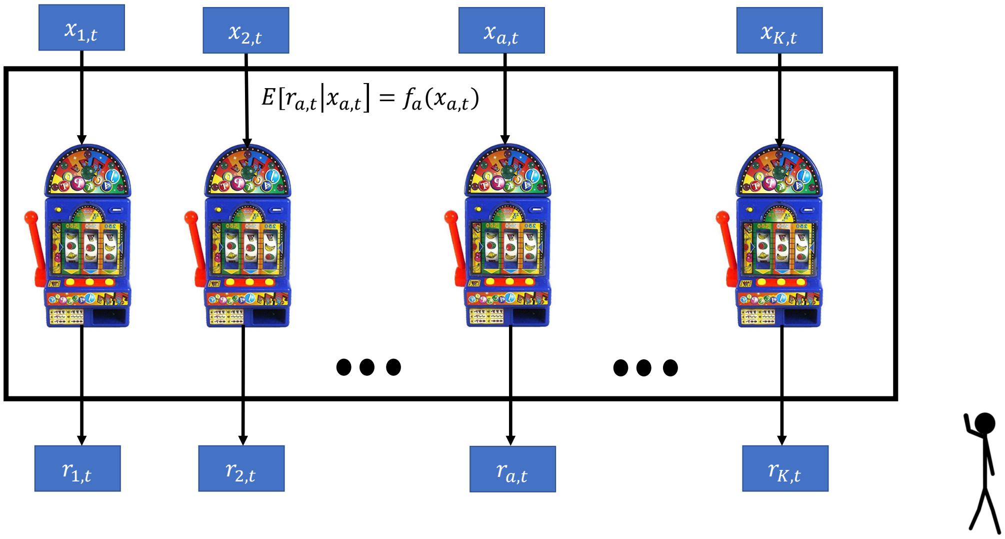


Figure 1.2: Contextual Bandit Problem

One common approach to contextual bandits is to fix the class of policy functions (i.e., functions from contexts to rewards) and try to learn the best function with time [23, 24, 25]. Most algorithms estimate rewards either separately for each arm or have one single estimator that is applied to all arms. But when rewards are estimated separately for each arm, we may be exploring more because arms could be similar to each other and if rewards are estimated together then we are assuming that there is a single estimator. Both these approaches are at one extreme and in reality arms could be similar to each other to a different extent. Therefore, I use an approach which adopts the perspective of multi-task learning (MTL) where separate estimators or one single estimator are special cases. The intuition is that some arms may be similar to each other, in which case it should be possible to pool the historical data for these arms to estimate the mapping from context to rewards more rapidly. For example, in the case of news article recommendations, there may be thousands of articles, and some of

	Cumulative Regret	Simple Regret
Multi-armed Bandits	Auer et al. 2002 [30]	Audibert et. al. 2012 [31]
Contextual Bandits	Chu et al. 2011 [19]	**This work**

Table 1.1: Contribution: Simple regret minimization for contextual bandits

those are bound to be similar to each other. In this case, when news article are similar to each other, we could benefit from estimating their rewards together and we may not need to explore too much to estimate their rewards.

I.1.3 Simple Regret for Contextual Bandits

The previous sub-section has discussed cumulative regret minimization in MAB and contextual bandit. In classical MABs, the goal of the learner is not always to minimize the cumulative regret. In some applications, there is a *pure exploration* phase during which the learning incurs no regret (i.e., no penalty for sub-optimal decisions), and performance is measured in terms of *simple regret*, which is the regret assessed at the end of the pure exploration phase. For example, in the best arm identification, the learner must guess the arm with a highest expected reward at the end of the exploration phase. Simple regret minimization clearly motivates different strategies, since there is no penalty for sub-optimal decisions during the exploration phase. Fixed budget and fixed confidence are the two main theoretical frameworks in which simple regret is generally analyzed [26, 27, 28, 29]. The number of trials for the exploration are fixed in the fixed budget setting and the goal is to maximize the probability of returning the best arm. In the fixed confidence setting, the goal is to achieve a fixed confidence about the quality of the returned arm in minimum possible number of trials. [26].

To date, work on contextual bandits has studied cumulative regret minimization i.e. $\sum_{t=1}^{T} r_{a_t^*,t} - \sum_{t=1}^{T} r_{a_t,t}$, which is motivated by applications in healthcare, web advertisement recommendations and news article recommendations [23]. In this thesis, I extend the idea of simple regret minimization to contextual bandits i.e. minimizing the regret at time t > T (after exploration phase) $r_{a_t^*,t} - r_{a_t,t}$. In this setting, there is a pure exploration phase during which no regret is incurred, followed by a *pure exploitation* phase during which regret is incurred, but there is no feedback so the learner cannot update its policy. To my knowledge, previous work has not addressed novel algorithms for this setting.

I.2 Contribution

The three major contributions of this thesis are summarized below.

1. Domain Generalization: In my first contribution (see chapter 2), I provide an efficient way to solve an existing kernel based domain generalization and extend the theoretical analysis to the multi-class classification. To be specific, I propose a kernel approximation technique which reduces the time complexity of the solver in the existing kernel based domain generalization approach to linear in terms of the number of samples. I give empirical evidence based on two medical datasets and one satellite dataset demonstrating the superiority of these algorithms over state-of-the-art ones. This work was done in collaboration with my advisor Prof. Clayton Scott, Prof. Gilles Blanchard and Dr. Urun Dogan at Microsoft Research.

2. Multi-Task Learning for Contextual Bandits: In chapter 3, I propose an upper confidence bound-based multi-task learning algorithm for contextual bandits, establish a corresponding regret bound, and interpret this bound to quantify the advantages of learning in the presence of high task (arm) similarity. I also describe an effective scheme for estimating task similarity from data and demonstrate my algorithm's performance using several data sets. This work was done in collaboration with Dr. Urun Dogan at Microsoft Research and my advisor Prof. Clayton Scott.

3. Simple Regret for Contextual Bandits: In chapter 4, I formulate a novel problem: that of simple regret minimization for contextual bandits and develop an algorithm, Contextual-Gap, for this setting. I present performance guarantees on the simple regret in the fixed budget framework and present experimental results for adaptive sensor selection in nano-satellites. This work was done in collaboration with Dr. Srinagesh Sharma, my advisor Prof. Clayton Scott, Prof. James Cutler and Prof. Mark Moldwin.

CHAPTER II

Domain Generalization

We consider the problem of assigning class labels to an unlabeled test data set, given several labeled training data sets drawn from similar distributions. This problem arises in several applications where data distributions fluctuate because of biological, technical, or other sources of variation. [32] has developed a distribution-free, kernel-based approach to the problem. This approach involves identifying an appropriate reproducing kernel Hilbert space and optimizing a regularized empirical risk over the space. But as dataset size increases, computational complexity of the SVM solver can be quadratic or cubic in terms of number of samples. We propose a kernel approximation technique which reduces the time complexity of the solver to linear in terms of number of samples. Kernel methods project input data points into high dimensional feature space (infinite-dimensional in case of Gaussian kernel) and find the optimal hyperplane in that feature space. Using kernel approximation techniques such as random Fourier features we map the input data to a randomized low-dimensional feature space and then apply existing fast linear SVM solvers. Experimental results are shown on three real world datasets. We also extend the generalization error analysis in [32] for the multi class setting and show supporting experimental results.

II.1 Introduction

Is it possible to leverage the solution of one classification problem to solve another? This is a question that has received increasing attention in recent years from the machine learning community, and has been studied in a variety of settings, including multi-task learning, covariate shift, and transfer learning. In this work, we study domain generalization, another setting in which this question arises, and one that incorporates elements of the three aforementioned settings, and is motivated by many practical applications.

To state the problem, let $\mathcal{X}$ be a feature space and $\mathcal{Y}$ a space of labels to predict. For a given distribution P_{XY}, we refer to the X marginal distribution P_X as simply the marginal distribution, and the conditional $P_{XY}(Y|X)$ as the posterior distribution. There are N similar but distinct distributions $P_{XY}^{(i)}$ on $\mathcal{X} \times \mathcal{Y}$, $i = 1, \ldots, N$. For each i, there is a training sample $S_i = (X_{ij}, Y_{ij})_{1 \leq j \leq n_i}$ of i.i.d. realizations of $P_{XY}^{(i)}$. There is also a test distribution P_{XY}^T that is similar to but again distinct from the "training distributions" $P_{XY}^{(i)}$. Finally, there is a test sample $(X_j^T, Y_j^T)_{1 \leq j \leq n_T}$ of i.i.d. realizations of P_{XY}^T, but in this case the labels Y_j are not observed. The goal of domain generalization is to correctly predict these unobserved labels. Essentially, given a random sample from the marginal test distribution P_X^T, we would like to predict the corresponding labels.

The goal is to predict these unobserved labels corresponding to samples drawn from the marginal test distribution. One of the methods to solve the transfer learning problem in the above setting is described in [32]. Their approach, marginal transfer learning, is a distribution-free, kernel-based and it involves identifying an appropriate reproducing kernel Hilbert space (RKHS) and optimizing a regularized empirical risk over the space. But as dataset size increases, computational complexity of this solver can be quadratic or cubic in terms of number of samples. We propose a kernel approximation to solve marginal transfer learning in linear time.

II.2 Motivating Application: Automatic Gating of Flow Cytometry Data

Flow cytometry is a high-throughput measurement platform that is an important clinical tool for the diagnosis of many blood-related pathologies. This technology allows for quantitative analysis of individual cells from a given population, derived for example from a blood sample from a patient. We may think of a flow cytometry data set as a set of d-dimensional attribute vectors $(X_j)_{1 \leq j \leq n}$, where n is the number of cells analyzed, and d is the number of attributes recorded per cell. These attributes pertain to various physical and chemical properties of the cell. Thus, a flow cytometry data set is a random sample from a patient-specific distribution.

Now suppose a pathologist needs to analyze a new (test) patient with data $(X_j^T)_{1 \leq j \leq n_T}$. Before proceeding, the pathologist first needs the data set to be “purified" so that only cells of a certain type are present. For example, lymphocytes are known to be relevant for the diagnosis of leukemia, whereas non-lymphocytes may potentially confound the analysis. In other words, it is necessary to determine the label $Y_j^T \in \{-1, 1\}$ associated to each cell, where $Y_j^T = 1$ indicates that the j-th cell is of the desired type.

In clinical practice this is accomplished through a manual process known as “gating." The data are visualized through a sequence of two-dimensional scatter plots, where at each stage a line segment or polygon is manually drawn to eliminate a portion of the unwanted cells. Because of the variability in flow cytometry data, this process is difficult to quantify in terms of a small subset of simple rules. Instead, it requires domain-specific knowledge and iterative refinement. Modern clinical laboratories routinely see dozens of cases per day, so it would be desirable to automate this process.

Since clinical laboratories maintain historical databases, we can assume access to a number (N) of historical patients that have already been expert-gated. Because of biological and technical variations in flow cytometry data, the distributions P^i_{XY} of the historical patients will vary. But every cell type of interest has a known tendency (e.g., high or low) for most measured attributes. Therefore, it is reasonable to assume that there is an underlying distribution (on distributions) governing flow cytometry data sets, that produces roughly similar distributions thereby making possible the automation of the gating process [32].

II.3 Formal Setting

As described in the last section let $\mathcal{X}$ be feature space and $\mathcal{Y}$ be the label space or output space. Further assume that we have samples from N distributions $S_i = (X_{ij}, Y_{ij})_{1 \le j \le n_i}$. For simplicity assume that $n_i = n$. Let $\mathcal{P}_{\mathcal{X}\times\mathcal{Y}}$ be the set of probability distributions on $\mathcal{X} \times \mathcal{Y}$, $\mathcal{P}_{\mathcal{X}}$ the set of probability distributions on $\mathcal{X}$, and $\mathcal{P}_{\mathcal{Y}|\mathcal{X}}$ the set of conditional probabilities of Y given X. Further, it is assumed that there exists a distribution μ on $\mathcal{P}_{\mathcal{X}\times\mathcal{Y}}$, where $P^1_{XY}, ..., P^N_{XY}$ are i.i.d. realizations from μ and as already described, samples S_i are i.i.d. realizations of (X, Y) following the distribution P^i_{XY}.

Suppose the user has training samples $S_i = ((X_{ij}, Y_{ij}))_{1 \le j \le n}$. Each data point X_{ij} along with its distribution P^i_X can be thought of as an extended data point $\tilde{X}_{ij} = (P^i_X, X_{ij})$ Now, consider a test sample $S^T = ((X^T_j, Y^T_j))_{1 \le j \le n_T}$, whose labels are not observed by a user. The goal here is to predict Y^T_j as accurately as possible. A decision function is a function $f : \mathcal{P}_{\mathcal{X}} \times \mathcal{X} \to \mathbb{R}$ (i.e. a classifier on extended feature space) that predicts $\hat{Y}_j = f(\hat{P_X}, X_j)$, where $\hat{P_X}$ is the associated empirical distribution. Let $\ell : \mathbb{R} \times \mathcal{Y} \to \mathbb{R}_+$ be the appropriate loss used, then the average loss incurred on the test sample is

$$L = \frac{1}{n_T} \sum_{j=1}^{n_T} \ell(\hat{Y}_j, Y_j). \tag{2.1}$$

Based on this the empirical error on test sample with sample size n_T is

$$\hat{\varepsilon}(f, n_T) = \frac{1}{T} \sum_{i=1}^{n_T} \ell(f(\hat{P_X^T}, X_i^T), Y_i^T), \tag{2.2}$$

and the generalization error of a decision function with respect to loss ℓ is

$$\varepsilon(f) = E_{P_{XY}^T \sim \mu} E_{(X^T, Y^T) \sim P_{XY}^T} \ell(f(P_X^T, X^T), Y^T). \tag{2.3}$$

Denoting $\tilde{X} = (P_X, X)$, the above can be written as

$$\varepsilon(f) = E_{P_{XY}^T \sim \mu} E_{(X^T, Y^T) \sim P_{XY}^T} \ell(f(\tilde{X}^T), Y^T). \tag{2.4}$$

Important points to note here are:

- At training time as well as at test time, the marginal distribution P_X for a sample is only known through the sample itself, that is, through the empirical marginal $\hat{P_X}$,
- Despite the similarity to standard binary classification in the infinite sample case, the learning task here is different, because the realizations $(\tilde{X}_{ij}, Y_{ij})$ are neither independent nor identically distributed

II.4 Learning Algorithm

We consider an approach based on kernels. The function $k : \Omega \times \Omega \to \mathbb{R}$ is called a *kernel* on Ω if the matrix $(k(x_i, x_j))_{1 \le i,j \le n}$ is symmetric and positive semi-definite for all positive integers n and all $x_1, \ldots, x_n \in \Omega$. It is well-known that if k is a kernel on Ω, then there exists a Hilbert space $\tilde{\mathcal{H}}$ and $\tilde{\Phi} : \Omega \to \tilde{\mathcal{H}}$ such that $k(x, x') = \langle \tilde{\Phi}(x), \tilde{\Phi}(x') \rangle_{\tilde{\mathcal{H}}}$. While $\tilde{\mathcal{H}}$ and $\tilde{\Phi}$ are not uniquely determined by k, the Hilbert space of functions (from Ω to $\mathbb{R}$) $\mathcal{H}_k = \{\langle v, \tilde{\Phi}(\cdot) \rangle_{\tilde{\mathcal{H}}} : v \in \tilde{\mathcal{H}}\}$ is uniquely determined by k, and is called the reproducing kernel Hilbert space (RKHS) of k.

One way to envision $\mathcal{H}_k$ is as follows. Define $\Phi(x) := k(\cdot, x)$, which is called the *canonical feature map* associated with k. Then the span of $\{\Phi(x) : x \in \Omega\}$, endowed with the inner product $\langle \Phi(x), \Phi(x') \rangle = k(x, x')$, is dense in $\mathcal{H}_k$. We also recall the *reproducing property*, which states that $\langle f, \Phi(x) \rangle = f(x)$ for all $f \in \mathcal{H}_k$.

Several well-known learning algorithms, such as support vector machines and kernel ridge regression, may be viewed as minimizers of a norm-regularized empirical risk over the RKHS of a kernel. A similar development has also been made for multi-task learning [33]. Inspired by this framework, we consider a general kernel algorithm as follows.

Consider the loss function $\ell : \mathbb{R} \times \mathcal{Y} \to \mathbb{R}_+$. Let $\overline{k}$ be a kernel on $\mathfrak{P}_X \times \mathcal{X}$, and let $\mathcal{H}_{\overline{k}}$ be the associated RKHS. For the sample S_i, let $\widehat{P}_X^{(i)} = \frac{1}{n_i} \sum_{j=1}^{n_i} \delta_{X_{ij}}$ denote the corresponding empirical X distribution. Also consider the extended input space $\mathfrak{P}_X \times \mathcal{X}$ and the extended data $\widetilde{X}_{ij} = (\widehat{P}_X^{(i)}, X_{ij})$. Note that $\widehat{P}_X^{(i)}$ plays a role analogous to the task index in multi-task learning. Now define

$$\widehat{f}_\lambda = \arg\min_{f \in \mathcal{H}_{\overline{k}}} \frac{1}{N} \sum_{i=1}^{N} \frac{1}{n_i} \sum_{j=1}^{n_i} \ell(f(\widetilde{X}_{ij}), Y_{ij}) + \lambda \|f\|^2 . \tag{2.5}$$

II.4.1 Specifying the kernels

In the rest of the chapter we will consider a kernel $\overline{k}$ on $\mathfrak{P}_{\mathcal{X}} \times \mathcal{X}$ of the product form

$$\overline{k}((P_1, x_1), (P_2, x_2)) = k_P(P_1, P_2) k_X(x_1, x_2), \tag{2.6}$$

where k_P is a kernel on $\mathfrak{P}_{\mathcal{X}}$ and k_X a kernel on $\mathcal{X}$.

Furthermore, we will consider kernels on $\mathfrak{P}_{\mathcal{X}}$ of a particular form. Let k'_X denote a kernel on $\mathcal{X}$ (which might be different from k_X) that is measurable and bounded. We define the *kernel mean embedding* $\Psi : \mathfrak{P}_\mathcal{X} \to \mathcal{H}_{k'_X}$:

$$P_X \mapsto \Psi(P_X) := \int_\mathcal{X} k'_X(x, \cdot) dP_X(x). \tag{2.7}$$

This mapping has been studied in the framework of "characteristic kernels" [34], and it has been proved that universality of k'_X implies injectivity of Ψ [35, 36].

Note that the mapping Ψ is linear. Therefore, if we consider the kernel $k_P(P_X, P'_X) = \left\langle \Psi(P_X), \Psi(P'_X) \right\rangle$, it is a linear kernel on $\mathfrak{P}_\mathcal{X}$ and cannot be a universal kernel. For this reason, we introduce yet another kernel $\mathfrak{K}$ on $\mathcal{H}_{k'_X}$ and consider the kernel on $\mathfrak{P}_\mathcal{X}$ given by

$$k_P(P_X, P'_X) = \mathfrak{K}\left(\Psi(P_X), \Psi(P'_X) \right) . \tag{2.8}$$

Note that particular kernels inspired by the finite dimensional case are of the form

$$\mathfrak{K}(v, v') = F(\|v - v'\|), \tag{2.9}$$

or

$$\mathfrak{K}(v, v') = G(\langle v, v' \rangle), \tag{2.10}$$

where F, G are real functions of a real variable such that they define a kernel. For example, $F(t) = \exp(-t^2/(2\sigma^2))$ yields a Gaussian-like kernel, while $G(t) = (1 + t)^d$ yields a polynomial-like kernel. Kernels of the above form on the space of probability distributions over a compact space $\mathcal{X}$ have been introduced and studied in [37]. Below we apply their results to deduce that $\overline{k}$ is a universal kernel for certain choices of k_X, k'_X, and $\mathfrak{K}$.

II.4.2 Relation to other kernel methods

By choosing $\overline{k}$ differently, one can recover other existing kernel methods. In particular, consider the class of kernels of the same product form as above, but where

$$k_P(P_X, P'_X) = \begin{cases} 1 & P_X = P'_X \\ \\ \tau & P_X \neq P'_X \end{cases}$$

If $\tau = 0$, the algorithm (2.5) corresponds to training N kernel machines $f(\widehat{P}_X^{(i)}, \cdot)$ using kernel k_X (e.g., support vector machines in the case of the hinge loss) on each training data set, independently of the others (note that this does not offer any generalization ability to a new dataset). If $\tau = 1$, we have a “pooling" strategy that, in the case of equal sample sizes n_i, is equivalent to pooling all training data sets together in a single data set, and running a conventional kernel learning supervised learning algorithm with kernel k_X (*i.e.*, this corresponds to trying to find a single “one-fits-all” prediction function which does not depend on the marginal). In the intermediate case $0 < \tau < 1$, the resulting kernel is a “multi-task kernel," and the algorithm recovers a multitask learning algorithm like that of [33]. We compare to the pooling strategy below in our experiments. We also examined the multi-task kernel with $\tau < 1$, but found that, as far as generalization to a new unlabeled task is concerned, it was always outperformed by pooling, and so those results are not reported. This fits the observation that the choice $\tau = 0$ does not provide any generalization to a new task, while $\tau = 1$ at least offers some form of generalization, if only by fitting the same decision function to all datasets.

In the special case where all labels Y_{ij} are the same value for a given task, and k_X is taken to be the constant kernel, the problem we consider reduces to “distributional" classification or regression, which is essentially standard supervised learning where a distribution (observed through samples) plays the role of the feature vector. Our analysis techniques could easily be specialized to analyze this problem.

II.5 Implementation

Implementation of the algorithm in (2.5) relies on techniques that are similar to those used for other kernel methods, but with some variations. The first subsection illustrates how, for the case of hinge loss, the optimization problem corresponds to a certain cost-sensitive support vector machine. Subsequent subsections focus on more scalable implementations based on approximate feature mappings.

II.5.1 Representer theorem and hinge loss

For a particular loss ℓ, existing algorithms for optimizing an empirical risk based on that loss can be adapted to the marginal transfer setting. We now illustrate this idea for the case of the hinge loss, $\ell(t, y) = \max(0, 1 - yt)$. To make the presentation more concise, we will employ the extended feature representation $\widetilde{X}_{ij} = (\widehat{P}_X^{(i)}, X_{ij})$, and we will also "vectorize" the indices (i, j) so as to employ a single index on these variables and on the labels. Thus the training data are $(\widetilde{X}_i, Y_i)_{1 \le i \le M}$, where $M = \sum_{i=1}^{N} n_i$, and we seek a solution to

$$\min_{f \in \mathcal{H}_{\overline{k}}} \sum_{i=1}^{M} c_i \max(0, 1 - Y_i f(\widetilde{X}_i)) + \frac{1}{2} \|f\|^2 .$$

Here $c_i = \dfrac{1}{\lambda N n_m}$, where m is the smallest positive integer such that $i \leq n_1 + \cdots + n_m$. By the representer theorem [38], the solution of (2.5) has the form

$$\widehat{f}_\lambda = \sum_{i=1}^{M} r_i \overline{k}(\widetilde{X}_i, \cdot)$$

for real numbers r_i. Plugging this expression into the objective function of (2.5), and introducing the auxiliary variables ξ_i, we have the quadratic program

$$\begin{aligned}
&\min_{r,\xi} \ \frac{1}{2} r^T \overline{K} r + \sum_{i=1}^{M} c_i \xi_i \\
&\text{s.t. } Y_i \sum_{j=1}^{M} r_j \overline{k}(\widetilde{X}_i, \widetilde{X}_j) \geq 1 - \xi_i, \ \forall i \\
&\qquad \xi_i \geq 0, \ \forall i,
\end{aligned}$$

where $\overline{K} := (\overline{k}(\widetilde{X}_i, \widetilde{X}_j))_{1 \leq i,j \leq M}$. Using Lagrange multiplier theory, the dual quadratic program is

$$\begin{aligned}
&\max_{\alpha} \ -\frac{1}{2} \sum_{i,j=1}^{M} \alpha_i \alpha_j Y_i Y_j \overline{k}(\widetilde{X}_i, \widetilde{X}_j) + \sum_{i=1}^{M} \alpha_i \\
&\text{s.t. } 0 \leq \alpha_i \leq c_i \ \forall i,
\end{aligned}$$

and the optimal function is

$$\widehat{f}_\lambda = \sum_{i=1}^{M} \alpha_i Y_i \overline{k}(\widetilde{X}_i, \cdot).$$

This is equivalent to the dual of a cost-sensitive support vector machine, without offset, where the costs are given by c_i. Therefore, we can learn the weights α_i using any existing software package for SVMs that accepts example-dependent costs and a user-specified kernel matrix, and allows for no offset. Returning to the original notation, the final predictor given a test

X-sample S^T has the form

$$\widehat{f}_\lambda(\widehat{P}_X^T, x) = \sum_{i=1}^{N} \sum_{j=1}^{n_i} \alpha_{ij} Y_{ij} \overline{k}((\widehat{P}_X^{(i)}, X_{ij}), (\widehat{P}_X^T, x))$$

where the α_{ij} are nonnegative. Like the SVM, the solution is often sparse, meaning most α_{ij} are zero.

Finally, we remark on the computation of $k_P(\widehat{P}_X, \widehat{P}_X')$. When $\mathfrak{K}$ has the form of (2.9) or (2.10), the calculation of k_P may be reduced to computations of the form $\left\langle \Psi(\widehat{P}_X), \Psi(\widehat{P}_X') \right\rangle$. If $\widehat{P}_X$ and $\widehat{P}_X'$ are empirical distributions based on the samples $X_1, \ldots, X_n$ and $X_1', \ldots, X_{n'}'$, then

$$\begin{aligned} \left\langle \Psi(\widehat{P}_X), \Psi(\widehat{P}_X') \right\rangle &= \left\langle \frac{1}{n} \sum_{i=1}^{n} k_X'(X_i, \cdot), \frac{1}{n'} \sum_{j=1}^{n'} k_X'(X_j', \cdot) \right\rangle \\ &= \frac{1}{nn'} \sum_{i=1}^{n} \sum_{j=1}^{n'} k_X'(X_i, X_j'). \end{aligned}$$

Note that when k_X' is a (normalized) Gaussian kernel, $\Psi(\widehat{P}_X)$ coincides (as a function) with a smoothing kernel density estimate for P_X.

II.5.2 Approximate Feature Mapping for Scalable Implementation

Assuming $n_i = n$, for all i, the computational complexity of a nonlinear SVM solver is between $O(N^2n^2)$ and $O(N^3n^3)$ [39, 40]. Thus, standard nonlinear SVM solvers may be insufficient when either or both N and n are very large.

One approach to scaling up kernel methods is to employ approximate feature mappings together with linear solvers. This is based on the idea that kernel methods are solving for a linear predictor after first nonlinearly transforming the data. Since this nonlinear transformation can have an extremely high-dimensional or even infinite-dimensional output, classical kernel methods avoid computing it explicitly. However, if the feature mapping can be approximated by a finite dimensional transformation with a relatively low-dimensional output, one can directly solve for the linear predictor, which can be accomplished in $O(Nn)$ time [41].

In particular, given a kernel $\overline{k}$, the goal is to find an approximate feature mapping $\overline{z}(\tilde{x})$ such that $\overline{k}(\tilde{x}, \tilde{x}') \approx \overline{z}(\tilde{x})^T \overline{z}(\tilde{x}')$. Given such a mapping $\overline{z}$, one then applies an efficient linear solver, such as Liblinear [42], to the training data $(\overline{z}(\tilde{X}_{ij}), Y_{ij})_{ij}$ to obtain a weight vector w. The final prediction on a test point $\tilde{x}$ is then $w^T \overline{z}(\tilde{x})$. As described in the previous subsection, the linear solver may need to be tweaked, as in the case of unequal sample sizes n_i, but this is usually straightforward.

Recently, such low-dimensional approximate future mappings $z(x)$ have been developed for several kernels. We examine two such techniques in the context of marginal transfer learning, the Nyström approximation [43, 44] and random Fourier features. The Nyström approximation applies to any kernel method, and therefore extends to the marginal transfer setting without additional work. On the other hand, we give a novel extension of random Fourier features to the marginal transfer learning setting (for the case of all Gaussian kernels), together with performance analysis.

II.5.2.1 Random Fourier Features

The approximation of Rahimi and Recht is based on Bochner's theorem, which characterizes shift invariant kernels [45].

Theorem 1. *A continuous kernel* $k(x, y) = k(x - y)$ *on* $\mathbb{R}^d$ *is positive definite iff* $k(x - y)$ *is the Fourier transform of a finite positive measure* $p(w)$*, i.e.,*

$$k(x-y) = \int_{\mathbb{R}^d} p(w) e^{jw^T(x-y)} dw \,. \tag{2.11}$$

If a shift invariant kernel $k(x-y)$ is properly scaled then Theorem 1 guarantees that $p(w)$ in (2.11) is a proper probability distribution.

Random Fourier features approximate the integral in (2.11) using samples drawn from $p(w)$. If $w_1, w_2, ..., w_L$ are i.i.d. draws from $p(w)$, then

$$\begin{aligned} k(x-y) &= \int_{\mathbb{R}^d} p(w) e^{jw^T(x-y)} dw \\ &= \int_{\mathbb{R}^d} p(w) \cos(w^T x - w^T y) dw \\ &\approx \frac{1}{L} \sum_{i=1}^{L} \cos(w_i^T x - w_i^T y) \\ &= \frac{1}{L} \sum_{i=1}^{L} \cos(w_i^T x)\cos(w_i^T y) + \sin(w_i^T x)\sin(w_i^T y) \\ &= \frac{1}{L} \sum_{i=1}^{L} [\cos(w_i^T x), \sin(w_i^T x)]^T [\cos(w_i^T y), \sin(w_i^T y)] \\ &= z_w(x)^T z_w(y) \,, \end{aligned} \tag{2.12}$$

where $z_w(x) = \frac{1}{\sqrt{L}}[\cos(w_1^T x), \sin(w_1^T x), ..., \cos(w_L^T x), \sin(w_L^T x)] \in \mathbb{R}^{2L}$ is an approximate nonlinear feature mapping of dimensionality $2L$. In the following, we extend the Random Fourier features methodology to the kernel $\bar{k}$ on the extended feature space $\mathfrak{P}_{\mathcal{X}} \times \mathcal{X}$. Let $X_1, \ldots, X_{n_1}$ and $X'_1, \ldots, X'_{n_2}$ be i.i.d. realizations of P_X and P'_X respectively, and let $\widehat{P}_X$ and $\widehat{P}'_X$ denote the corresponding empirical distributions. Given $x, x' \in \mathcal{X}$, denote $\tilde{x} = (\widehat{P}_X, x)$ and $\tilde{x}' = (\widehat{P}'_X, x')$. The goal is to find an approximate feature mapping $\bar{z}(\tilde{x})$ such that $\bar{k}(\tilde{x}, \tilde{x}') \approx \bar{z}(\tilde{x})^T \bar{z}(\tilde{x}')$. Recall that,

$$\bar{k}(\tilde{x}, \tilde{x}') = k_P(\widehat{P}_X, \widehat{P}'_X) k_X(x, x');$$

specifically, we consider k_X and k'_X to be Gaussian kernels and the kernel on distributions k_P

to have the Gaussian-like form

$$k_P(\widehat{P}_X, \widehat{P}'_X) = \exp\left\{\frac{1}{2\sigma_P^2}\|\Psi(\widehat{P}_X) - \Psi(\widehat{P}'_X)\|^2_{\mathcal{H}_{k'_X}}\right\}.$$

As noted earlier in this section, the calculation of $k_P(\widehat{P}_X, \widehat{P}'_X)$ reduces to the computation of

$$\langle\Psi(\widehat{P}_X), \Psi(\widehat{P}'_X)\rangle = \frac{1}{n_1 n_2}\sum_{i=1}^{n_1}\sum_{j=1}^{n_2} k'_X(X_i, X'_j). \tag{2.13}$$

We use Theorem 1 to approximate k'_X and thus $\langle\Psi(\widehat{P}_X), \Psi(\widehat{P}'_X)\rangle$. Let $w_1, w_2, ..., w_L$ be i.i.d. draws from $p'(w)$, the inverse Fourier transform of k'_X. Then we have:

$$\begin{aligned}
\langle\Psi(\widehat{P}_X), \Psi(\widehat{P}'_X)\rangle &= \frac{1}{n_1 n_2}\sum_{i=1}^{n_1}\sum_{j=1}^{n_2} k'_X(X_i, X'_j) \\
&\approx \frac{1}{Ln_1 n_2}\sum_{l=1}^{L}\sum_{i=1}^{n_1}\sum_{j=1}^{n_2} \cos(w_l^T X_i - w_l^T X'_j) \\
&= \frac{1}{Ln_1 n_2}\sum_{l=1}^{L}\sum_{i=1}^{n_1}\sum_{j=1}^{n_2} [\cos(w_l^T X_i)\cos(w_l^T X'_j) + \sin(w_l^T X_i)\sin(w_l^T X'_j)] \\
&= \frac{1}{Ln_1 n_2}\sum_{l=1}^{L}\{\sum_{i=1}^{n_1}[\cos(w_l^T X_i), \sin(w_l^T X_i)]^T \sum_{j=1}^{n_2}[\cos(w_l^T X'_j), \sin(w_l^T X'_j)]\} \\
&= Z_P(\widehat{P}_X)^T Z_P(\widehat{P}'_X),
\end{aligned}$$

where

$$Z_P(\widehat{P}_X) = \frac{1}{n_1\sqrt{L}}\sum_{i=1}^{n_1}\Big[\cos(w_1^T X_i), \sin(w_1^T X_i), ..., \cos(w_L^T X_i), \sin(w_L^T X_i)\Big], \tag{2.14}$$

and $Z_P(\widehat{P}'_X)$ is defined analogously with n_1 replaced by n_2. For the proof of Theorem 2, let z'_X denote the approximate feature map corresponding to k'_X, which satisfies $Z_P(\widehat{P}_X) = \frac{1}{n_1}\sum_{i=1}^{n_1} z'_X(X_i)$.

Note that the lengths of the vectors $Z_P(\widehat{P}_X)$ and $Z_P(\widehat{P}'_X)$ are $2L$. To approximate $\bar{k}$ we may write

$$\begin{aligned} \bar{k}(\tilde{x},\tilde{x}') &\approx \exp\frac{-\|Z_P(\widehat{P}_X)-Z_P(\widehat{P}'_X)\|^2_{\mathbb{R}^{2L}}}{2\sigma_P^2}\cdot\exp\frac{-\|x-x'\|^2_{\mathbb{R}^d}}{2\sigma_X^2} \\ &= \exp\frac{-(\sigma_X^2\|Z_P(\widehat{P}_X)-Z_P(\widehat{P}'_X)\|^2_{\mathbb{R}^{2L}}+\sigma_P^2\|x-x'\|^2_{\mathbb{R}^d})}{2\sigma_P^2\sigma_X^2} \\ &= \exp\frac{-(\|\sigma_X Z_P(\widehat{P}_X)-\sigma_X Z_P(\widehat{P}'_X)\|^2_{\mathbb{R}^{2L}}+\|\sigma_P x-\sigma_P x'\|^2_{\mathbb{R}^d})}{2\sigma_P^2\sigma_X^2} \\ &= \exp\frac{-\|(\sigma_X Z_P(\widehat{P}_X),\sigma_P x)-(\sigma_X Z_P(\widehat{P}'_X),\sigma_P x')\|^2_{\mathbb{R}^{2L+d}}}{2\sigma_P^2\sigma_X^2} \end{aligned} \tag{2.15}$$

This is also a Gaussian kernel, now on $\mathbb{R}^{2L+d}$. Again by applying Theorem 1, we have

$$\bar{k}(\widehat{P}_X,X),(\widehat{P}'_X,X')) \approx \int_{\mathbb{R}^{2L+d}} p(v)e^{jv^T((\sigma_X Z_P(P_X),\sigma_P X)-(\sigma_X Z_P(P'_X),\sigma_P X'))}dv.$$

Let $v_1, v_2, ..., v_q$ be drawn i.i.d. from $p(v)$, the inverse Fourier transform of the Gaussian kernel with bandwidth $\sigma_P\sigma_X$. Let $u = (\sigma_X Z_P(\widehat{P}_X), \sigma_P x)$ and $u' = (\sigma_X Z_P(\widehat{P}'_X), \sigma_P x')$. Then

$$\begin{aligned} \bar{k}(\tilde{x},\tilde{x}') &\approx \frac{1}{Q}\sum_{q=1}^{Q}\cos(v_q^T(u-u')) \\ &= \bar{z}(\tilde{x})^T\bar{z}(\tilde{x}'), \end{aligned}$$

where

$$\bar{z}(\tilde{x}) = \frac{1}{\sqrt{Q}}[\cos(v_1^T u),\sin(v_1^T u),...,\cos(v_Q^T u),\sin(v_Q^T u)] \in \mathbb{R}^{2Q} \tag{2.16}$$

and $\bar{z}(\tilde{x}')$ is defined similarly.

This completes the construction of the approximate feature map. The following result, which uses Hoeffding's inequality and generalizes a result of Rahimi and Recht [45], says that the approximation achieves any desired approximation error with very high probability as $L, Q \to \infty$.

Theorem 2. *Let L be the number of random features to approximate the kernel on distributions and Q be the number of features to approximate the final product kernel. For any $\epsilon_\ell > 0$, $\epsilon_q > 0$, $\tilde{x} = (\widehat{P}_X, x)$, $\tilde{x}' = (\widehat{P}'_X, x')$,*

$$P(|\bar{k}(\tilde{x}, \tilde{x}') - \bar{z}(\tilde{x})^T \bar{z}(\tilde{x}')| \geq \epsilon_l + \epsilon_q) \leq 2 \exp\Big(-\frac{Q\epsilon_q^2}{2}\Big) + 6n_1n_2 \exp\Big(-\frac{L\epsilon^2}{2}\Big), \tag{2.17}$$

where $\epsilon = \frac{\sigma_P^2}{2}\log(1+\epsilon_l)$, σ_P is the bandwidth parameter of the Gaussian-like kernel k_P, and n_1 and n_2 are the sizes of the empirical distributions $\widehat{P}_X$ and $\widehat{P}'_X$, respectively.

The above results holds for fixed $\tilde{x}$ and $\tilde{x}'$. Following again [45], one can use an ϵ-net argument to prove a stronger statement for every pair of points in the input space simultaneously.

Lemma 1. *Let $\mathcal{M}$ be a compact subset of $\mathbb{R}^d$ with diameter $r = \mathrm{diam}(\mathcal{M})$ and let D be the number of random Fourier features used. Then for the mapping defined in (2.12), we have*

$$P\Big(\sup_{x,y\in\mathcal{M}} |z_w(x)^T z_w(y) - k(x-y)| \geq \epsilon\Big) \leq 2^8\Big(\frac{\sigma r}{\epsilon}\Big)^2 \exp\Big(\frac{-D\epsilon^2}{2(d+2)}\Big)$$

where $\sigma = \mathbb{E}[w^T w]$ is the second moment of the Fourier transform of k [45].

Our RFF approximation of $\bar{k}$ is grounded on Gaussian RFF approximations on Euclidean spaces, and thus, the following results holds by invoking Lemma 1, and otherwise following the argument of Theorem 2.

Theorem 3. *Using the same notations as in Theorem 2 and Lemma 1,*

$$P\Big(\sup_{x,x'\in\mathcal{M}} |\bar{k}(\tilde{x}, \tilde{x}') - \bar{z}(\tilde{x})^T \bar{z}(\tilde{x}')| \geq \epsilon_l + \epsilon_q\Big) \tag{2.18}$$

$$\leq 2^8\Big(\frac{\sigma'_X r}{\epsilon_q}\Big)^2 \exp(\frac{-Q\epsilon_q^2}{2(d+2)}) + 2^9 3 n_1 n_2 \Big(\frac{\sigma_P \sigma_X r}{\epsilon_l}\Big)^2 \exp(\frac{-L\epsilon_l^2}{2(d+2)})$$

where σ'_X is the width of kernel k'_X in Eqn (2.13) and σ_P, σ_X are width of kernel k_P and k_X respectively.

II.5.2.2 Nyström Approximation

Like random Fourier features, the Nyström approximation is a technique to approximate kernel matrices. Unlike random Fourier features, for the Nyström approximation, the feature maps are data-dependent. Also, in the last subsection, all kernels were assumed to be shift invariant. With the Nyström approximation there is no such assumption.

For a general kernel k, the goal is to find a feature mapping $z : \mathbb{R}^d \to \mathbb{R}^L$, where $L > d$, such that $k(x, x') \approx z(x)^T z(x')$. Let r be the target rank of the final approximated kernel matrix, and m be the number of selected columns of the original kernel matrix. In general $r \leq m \ll n$.

Given data points $x_1, \ldots, x_n$, the Nyström method approximates the kernel matrix by first sampling m data points $x'_1, x'_2, ..., x'_m$ without replacement from the original sample, and then constructing a low rank matrix by $\widehat{K}_r = K_b \widehat{K}^{-1} K_b^T$, where $K_b = [k(x_i, x'_j)]_{n\times m}$, and $\widehat{K} = [k(x'_i, x'_j)]_{m\times m}$. Hence, the final approximated feature mapping is

$$z_n(x) = \widehat{D}^{-\frac{1}{2}} \widehat{V}^T [k(x, x'_1), ..., k(x, x'_m)], \tag{2.19}$$

where $\widehat{D}$ is the eigenvalue matrix of $\widehat{K}$ and $\widehat{V}$ is the corresponding eigenvector matrix.

The Nyström approximation holds for any positive definite kernel, but random Fourier features can be used only for shift invariant kernels. On the other hand, random Fourier features are very easy to implement and have a lower time complexity than the Nyström method (where one has to find the eigenvalue decomposition). Moreover, the Nyström method is useful only when the kernel matrix is low rank. In our experiments, we use random Fourier features when all kernels are Gaussian and the Nyström method otherwise.

II.6 Experiments

This section empirically compares our marginal transfer learning method with pooling. One implementation of the pooling algorithm was mentioned in Section II.4.2, where k_P is taken to be a constant kernel. Another implementation is to put all the training data sets together and train a single conventional kernel method. The only difference between the two implementations is that in the form, weights of $1/n_i$ are used for examples from training task i. In almost all of our experiments below, the various training tasks have the same sample sizes, in which case the two implementations coincide. The only exception is the fourth experiment when we use all training data, in which case we use the second of the two implementations mentioned above.

We consider three classification problems ($\mathcal{Y} = \{-1, 1\}$), for which the hinge loss is employed, and one regression problem ($\mathcal{Y} \subset \mathbb{R}$), where the ϵ-insensitive loss is employed. Thus, the algorithms implemented are natural extensions of support vector classification and regression to marginal transfer learning. The code is available online to reproduce all results [1].

[1] The code to reproduce our results is available at https://github.com/aniketde/DomainGeneralizationMarginal

II.6.1 Model Selection

The various experiments use different combinations of kernels. In all experiments, linear kernels $k(x_1, x_2) = {x_1}^T x_2$ and Gaussian kernels $k_\sigma(x_1, x_2) = \exp\Big(-\frac{||x_1 - x_2||^2}{2\sigma^2}\Big)$ were used.

The bandwidth σ of each Gaussian kernel and the regularization parameter λ of the machines were selected by grid search. For model selection, five-fold cross-validation has been used. In order to stabilize the cross-validation procedure, it was repeated 5 times over independent random splits into folds [46]. Thus, candidate parameter values were evaluated on the 5×5 validation sets and the configuration yielding the best average performance was selected. If any of the chosen hyper-parameters was at the grid boundary, the grid was extended accordingly, i.e., the same grid size has been used, however, the center of grid has been assigned to the previously selected point. The grid used for kernels was $\sigma \in \big(10^{-2}, 10^{4}\big)$ with logarithmic spacing, and the grid used for the regularization parameter was $\lambda \in \big(10^{-2}, 10^{1}\big)$ with logarithmic spacing.

II.6.2 Parkinson's disease telemonitoring dataset

We test our method in the regression setting using the Parkinson's disease telemonitoring dataset, which is composed of a range of biomedical voice measurements using a telemonitoring device from 42 people with early-stage Parkinson's. The recordings were automatically captured in the patients' homes. The aim is to predict the clinician's Parkinson's disease symptom score for each recording on the unified Parkinson's disease rating scale (UPDRS) [47]. Thus we are in a regression setting, and employ the ϵ-insensitive loss from support vector regression. All kernels are taken to be Gaussian, and the random Fourier features speedup is used.

There are around 200 recordings per patient. We randomly select 7 test users and then

Table 2.1: RMSE of Marginal Transfer Learning on Parkinson's Disease Dataset

Training examples	10	15	20	25	30	35
20	13.78	12.37	11.93	10.74	10.08	11.17
24	14.18	11.89	11.51	10.90	10.55	10.18
28	14.95	13.29	12.00	10.21	10.59	9.52
34	13.27	11.66	11.79	9.16	9.34	10.50
41	12.89	11.27	11.17	9.91	9.10	10.05
49	13.15	11.70	13.81	10.12	9.01	8.69
58	12.16	9.59	9.85	9.28	8.44	7.62
70	13.03	9.16	8.80	9.03	8.16	7.88
84	11.98	9.18	9.74	9.03	7.30	7.01
100	12.69	8.48	9.52	8.01	7.14	7.5

vary the number of training users $\boldsymbol{N}$ from 10 to 35 in steps of 5, and we also vary the number of training examples $\boldsymbol{n}$ per user from 20 to 100. We repeat this process several times to get the average errors which are shown in Fig 2.1 and Tables 2.1 and 2.2. The transfer learning method clearly outperforms pooling, especially as $\boldsymbol{N}$ and $\boldsymbol{n}$ increase.

II.6.3 Satellite Classification

Microsatellites are increasingly deployed in space missions for a variety of scientific and technological purposes. Because of randomness in the launch process, the orbit of a microsatellite is random, and must be determined after the launch. One recently proposed approach is to estimate the orbit of a satellite based on radiofrequency (RF) signals as measured in a ground sensor network. However, microsatellites are often launched in bunches, and for this approach to be successful, it is necessary to associate each RF measurement vector with a particular

Table 2.2: RMSE of Pooling on Parkinson's Disease Dataset

Training examples	10	15	20	25	30	35
20	13.64	11.93	11.95	11.06	11.91	12.08
24	13.80	11.83	11.70	11.98	11.68	11.48
28	13.78	11.70	11.72	11.18	11.58	11.73
34	13.71	12.20	12.04	11.17	11.67	11.92
41	13.69	11.73	12.08	11.28	11.55	12.59
49	13.75	11.85	11.79	11.17	11.34	11.82
58	13.70	11.89	12.06	11.06	11.82	11.65
70	13.54	11.86	12.14	11.21	11.40	11.96
84	13.55	11.98	12.03	11.25	11.54	12.22
100	13.53	11.85	11.92	11.12	11.96	11.84

satellite. Furthermore, the ground antennae are not able to decode unique identifier signals transmitted by the microsatellites, because (a) of constraints on the satellite/ground antennae links, including transmission power, atmospheric attenuation, scattering, and thermal noise, and (b) ground antennae must have low gain and low directional specificity owing to uncertainty in satellite position and dynamics. To address this problem, recent work has proposed to apply our marginal transfer learning methodology [48].

As a concrete instance of this problem, suppose two microsatellites are launched together. Each launch is a random phenomenon and may be viewed as a task in our framework. For each launch i, training data (X_{ij}, Y_{ij}), $j = 1, \ldots, n_i$, are generated using a highly realistic simulation model, where X_{ij} is a feature vector of RF measurements across a particular sensor network and at a particular time, and Y_{ij} is a binary label identifying which of the two microsatellites produced a given measurement. By applying our methodology, we can classify

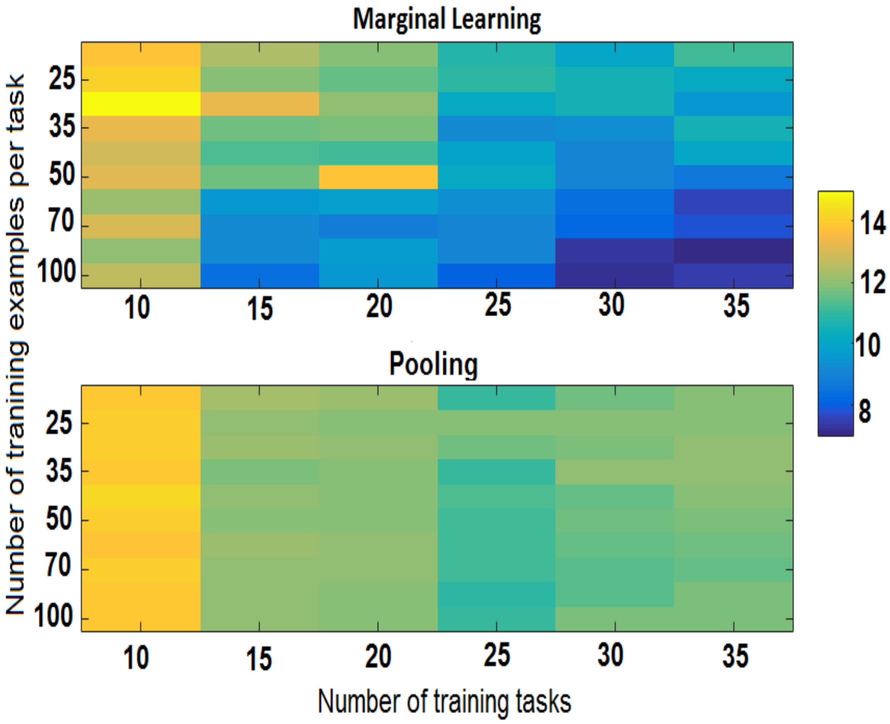


Figure 2.1: Parkinson's disease telemonitoring dataset

unlabeled measurements X_j^T from a new launch with high accuracy. Given these labels, orbits can subsequently be estimated using the observed RF measurements. We thank Srinagesh Sharma and James Cutler for providing us with their simulated data, and refer the reader to their paper for more details on the application [48].

To demonstrate this idea, we analyzed the data from [48] for $T = 50$ launches, viewing 40 as training data and 10 as testing. We use Gaussian kernels and the RFF kernel approximation technique to speed up the algorithm. Results are shown in Fig 2.2. As expected, the error for the proposed method is much lower than for pooling.

II.6.4 Flow Cytometry Experiments

We demonstrate the proposed methodology for the flow cytometry auto-gating problem, described in Sec. II.2. The pooling approach has been previously investigated in this context

Table 2.3: Average Classification Error of Marginal Transfer Learning on Satellite Dataset

Training examples	10	20	30	40
5	8.62	7.61	8.25	7.17
15	6.21	5.90	5.85	5.43
30	6.61	5.33	5.37	5.35
45	5.61	5.19	4.71	4.70
all training data	5.16	4.72	3.69	3.87

Table 2.4: Average Classification Error of Pooling on Satellite Dataset

Training examples	10	20	30	40
5	8.13	7.54	7.94	6.96
15	6.55	5.81	5.79	5.57
30	6.06	5.36	5.56	5.31
45	5.58	5.12	5.30	4.99
all training data	5.16	4.78	4.93	4.97

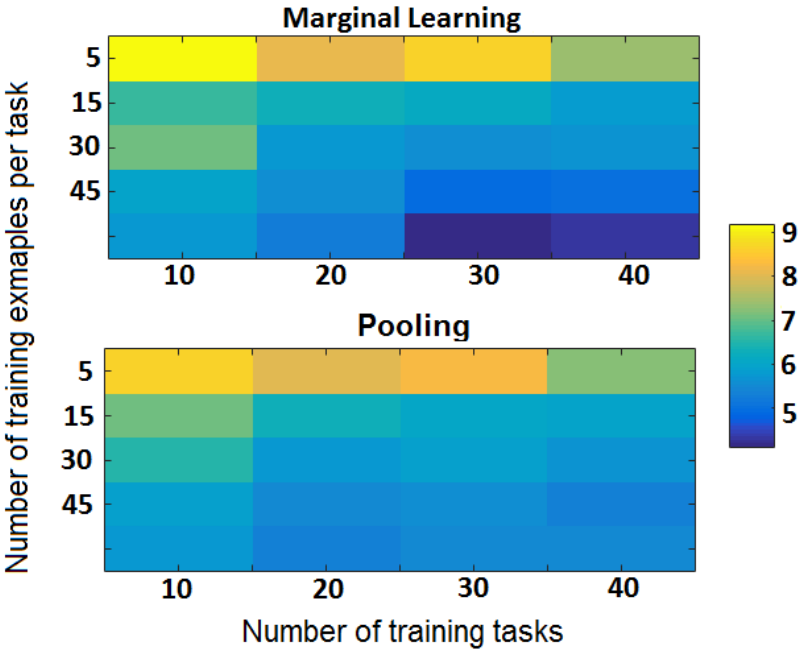


Figure 2.2: Satellite dataset

by [49]. We used a dataset that is a part of FlowCAP Challenges where the ground truth labels have been supplied by human experts [50]. We used the so-called “Normal Donors" dataset. The dataset contains 8 different classes and 30 subjects. Only two classes (0 and 2) have consistent class ratios, so we have restricted our attention to these two.

The corresponding flow cytometry data sets have sample sizes ranging from 18,641 to 59,411, and the proportion of class 0 in each data set ranges from 25.59 to 38.44%. We randomly selected 10 tasks as test tasks and used exactly the same tasks over all experiments. We varied the number of tasks in the training data from 5 to 20 with an additive step size of 5, and the number of training examples per task from 32 to 16384 with a multiplicative step size of 2. We repeated this process 10 times to get the average errors which are shown in Fig. 2.3 and Tables 2.5 and 2.6. The kernel k_P was Gaussian, and the other two were linear. The Nyström approximation was used to achieve an efficient implementation.

For nearly all settings the proposed method has a smaller error rate than the baseline. Furthermore, for the marginal transfer learning method, when one fixes the number of training

Table 2.5: Average Classification Error of Marginal Transfer Learning on Flow Cytometry Dataset

Training examples	5	10	15	20
1024	9	9.03	9.03	8.70
2048	9.12	9.56	9.07	8.62
4096	8.96	8.91	9.01	8.66
8192	9.18	9.20	9.04	8.74
16384	9.05	9.08	9.04	8.63

Table 2.6: Average Classification Error of Pooling on Flow Cytometry Dataset

Training examples	5	10	15	20
1024	9.41	9.48	9.32	9.52
2048	9.92	9.57	9.45	9.54
4096	9.72	9.56	9.36	9.40
8192	9.43	9.53	9.38	9.50
16384	9.42	9.56	9.40	9.33

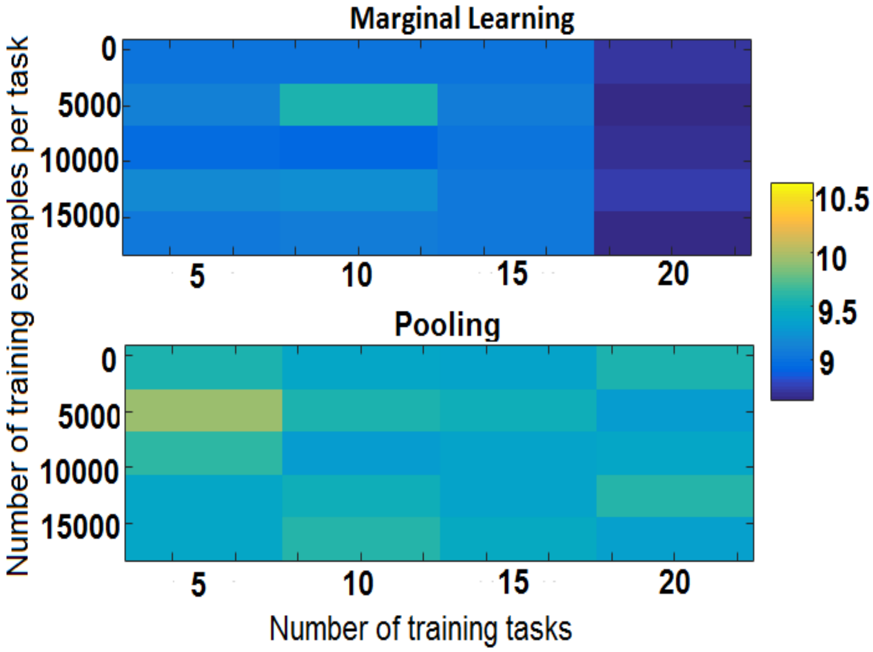


Figure 2.3: Classification error rates for baseline and proposed method for different experimental settings, i.e., number of examples per task and number of tasks.

examples and increases the number of tasks then the classification error rate drops.

II.7 Multiclass Domain Generalization

In this chapter, we have reviewed kernel based approach to address domain generalization [32] which addresses binary classification and regression. In this section, we extend the generalization error analysis to multiclass setting ($|\mathcal{Y}| = c$) and give supporting experimental results. While several aspects of the original analysis in [32] carry over to the multiclass case, others do not. In particular, we use an extension of the contraction lemma for Rademacher complexity of Lipschitz loss classes to prove the generalization error bound [51].

We modify our objective function for multiclass classification compared to Eqn. 2.5. We will find a decision function $f \in \mathcal{H}_{\bar{k}}^c := \mathcal{H}_{\bar{k}} \times \cdots \mathcal{H}_{\bar{k}}$ (c times) and has components $g_l \in \mathcal{H}_{\bar{k}}, l = 1, 2, ...c$, i.e., $f = \begin{bmatrix} g_1 & g_2 & \dots & g_c \end{bmatrix}$. Define

$$\hat{f}_\lambda = \arg\min_{f \in \mathcal{H}_{\bar{k}}^c} \frac{1}{N} \sum_{i=1}^{N} \frac{1}{n_i} \sum_{j=1}^{n_i} \ell(f(\tilde{X}_{ij}), Y_{ij}) + \lambda r(f), \tag{2.20}$$

as the empirical estimate of the optimal decision function. Define the regularizer $r(f)$ as $r(f) := \|f\|^2_{\mathcal{H}_{\bar{k}}^c} := \sum_{m=1}^{c} \|g_m\|^2_{\mathcal{H}_{\bar{k}}}$.

II.7.1 Generalization Error Analysis

We make the following assumptions to analyze the generalization error. For any kernel k, $\phi_k(x) := k(\cdot, x) \in \mathcal{H}_k$ denotes the canonical feature map, $\mathcal{B}_k(R)$ refers to the closed ball of radius R in $\mathcal{H}_k$ and $\mathcal{B}_k^c(R) := \prod_{m=1}^{c} \mathcal{B}_k(R)$ refers to the product space of c closed balls.

A I The loss function $\ell : \mathbb{R}^c \times \mathcal{Y} \to R$ is bounded by B_ℓ, and is L_ℓ-Lipschitz in the first variable: For all y, $|\ell(T_1, y) - \ell(T_2, y)| \leq L_\ell \|T_1 - T_2\|_2$ for $T_1, T_2 \in \mathbb{R}^c$.

A II Kernels k_x, k'_x, k_P are bounded by $B_k^2, B_{k'}^2, B_{k_P}^2$ respectively.

A III The canonical feature map $\phi_{k_P} : \mathcal{H}_{k'_x} \to \mathcal{H}_{k_P}$ is α-Hölder continuous, i.e., $\forall a, b \in \mathcal{B}_{k'_x}(B_{k'})$:

$$\|\phi_{k_P}(a) - \phi_{k_P}(b)\|_2 \leq L_{k_P}\|a - b\|_2^{\alpha}.$$

The above assumptions are similar to those presented in [52] translated to multiclass data. Condition **A III** holds with $\alpha = 1$ when k_P is the Gaussian-like kernel on $\mathcal{H}_{k'_x}$. Using the stated assumptions, we shall now develop generalization error bounds for multiclass DG. To generalize the analysis, an extension of Talagrand's lemma for bounding the Rademacher complexity is needed. Such an extension was provided by [53, 51] and [54].

Lemma 2. ***(Vector Valued Talagrand's Contraction Lemma)*** *[51] Let $\mathcal{F}$ be a class of functions from $\mathcal{X} \to \mathbb{R}^c$. Let $\{\mu_i\}_{i=1}^N$ and $\{\sigma_{ij}\}_{i=1,j=1}^{N,c}$ be two sets of independent Rademacher random variables. If $\varphi : \mathbb{R}^c \to \mathbb{R}$ is L-Lipschitz under $\|\cdot\|_p$ where $p \geq 2$, then*

$$\mathbb{E}_\mu[\sup_{f\in\mathcal{F}} \sum_{i=1}^N \mu_i \varphi(f(x_i))] \leq \sqrt{2}L\mathbb{E}_\sigma[\sup_{f\in\mathcal{F}} \sum_{i=1}^N \sum_{j=1}^c \sigma_{ij} g_j(x_i)].$$

For simplicity's sake, we assume that $n_i = n$ to state the generalization error bound.

Theorem 4. ***(Estimation error control)*** *Assuming that conditions **A I** - **A III** hold then for any $R > 0$, with probability at least $1 - \delta$:*

$$\sup_{f\in\mathcal{B}_k^c(R)} |\widehat{\varepsilon}(f) - \varepsilon(f)| \leq L_\ell L_{k_P} R B_k c (B_{k'})^\alpha \Bigg(\sqrt{\frac{2\log\frac{2N}{\delta}}{n}} + \sqrt{\frac{1}{n}} + \frac{4\log\frac{2N}{\delta}}{3n}\Bigg)^\alpha$$
$$+ \frac{8\sqrt{2}RL_\ell B_k B_{k_P} c}{\sqrt{N}} + B_\ell\sqrt{\frac{\log 8\delta^{-1}}{2N}}$$

Proof Sketch Let $\mathcal{E}(f) = |\widehat{\varepsilon}(f) - \varepsilon(f)|$.

$$\begin{aligned} \sup_{f \in \mathcal{B}^c_{\bar{k}}(R)} \mathcal{E}(f) &\leq \sup_{f \in \mathcal{B}^c_{\bar{k}}(R)} \left| \widehat{\varepsilon}(f) - \frac{1}{Nn} \sum_{i=1}^{N} \sum_{j=1}^{n} \ell(f(\tilde{X}_{ij}), Y_{ij}) \right| + \sup_{f \in \mathcal{B}^c_{\bar{k}}(R)} \left| \frac{1}{Nn} \sum_{i=1}^{N} \sum_{j=1}^{n} \ell(f(\tilde{X}_{ij}), Y_{ij}) - \varepsilon(f) \right| \\ &= (I) + (II) \end{aligned}$$

Term (I) is bounded by application of Lipschitz continuity of ℓ, union bounds for tasks and classes over f and through Hölder continuity in assumption **A III**. Bounding the term (II) is similar to bounding term (II) in Theorem 5 in [52] with modifications for multi-class loss. In addition, the modified Talagrand's lemma 2 is applied to bound the Rademacher complexity [51].

II.7.2 Experimental Results

We test the proposed algorithm on 4 multiclass datasets and compare it with pooling, where data from all the tasks are pooled together to learn one single classifier. Datasets description are given below and a summary is in Table 2.7.

Dataset	Training Tasks	Test Tasks	Examples Per Task	Classes
Synthetic	80	20	100	10
Satellite	400	100	77-165	3
HAR	20	10	300	6
MNIST-MOD	80	20	100	10

Table 2.7: Summary of Multiclass Datasets for Domain Generalization

Synthetic Dataset: Features for synthetic data are drawn from the unit square. Based on one of the dimensions, the data are labeled from 0 to 10, e.g., if the feature value is between 0 and 0.1, then it's labeled as 1, if it's in between 0.1 and 0.2, then it's labeled as 2, and so on. After that, the feature vectors are rotated clockwise by an angle randomly drawn from 0 to 180 degrees to get data for one task. The process is repeated 100 times to get data for 100 tasks out of which 80 are train tasks and 20 are test tasks. Fig. 2.4 shows 3 such tasks for $\theta = 0, 90$ and 180 where the supports don't overlap at all, and Fig. 2.5 shows 13 tasks where the supports overlap.

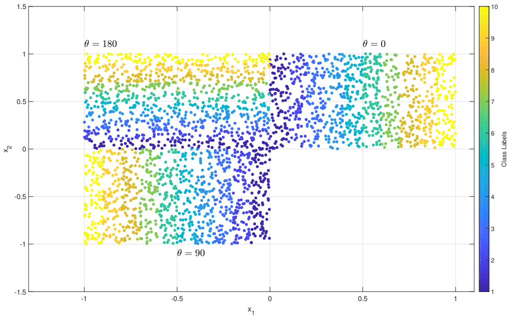


Figure 2.4: Synthetic Dataset: Three tasks $\theta = \{0, 90, 180\}$

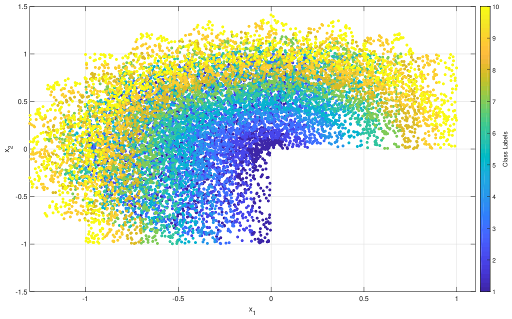


Figure 2.5: Synthetic Dataset: Thirteen tasks $\theta = \{0, 15, 30, ..., 180\}$

Satellite Dataset: The problem is described in the introduction, and we used the dataset presented by [48] modified for $c = 3$ spacecraft.

HAR Dataset: This is a human activity recognition using smart-phone dataset from UCI repository [55]. Each of 30 volunteers performed six activities (walking, walking upstairs, walking downstairs, sitting, standing, laying) wearing the smart-phone.

MNIST-MOD Dataset: We randomly draw 1000 images from MNIST's train dataset. Then we rotate each of this image by randomly drawn angle from 0 to 180 degrees and repeat this 100 times to get data for 100 tasks. Example for rotated MNIST dataset is shown in Fig. 2.6.

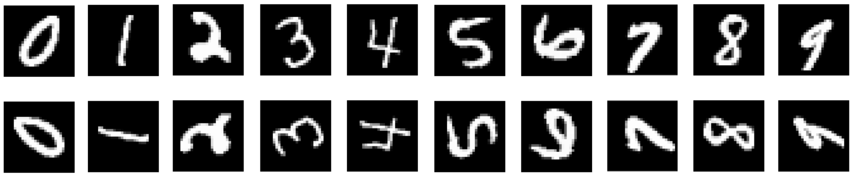


Figure 2.6: MNIST Data with no rotation (first row) and 90 degree rotation (second row)

We use all Gaussian kernels and a novel random Fourier Feature (RFF) approximation, which extends the usual RFF approximation on Euclidean space $\mathcal{X}$ [56] to the extended feature space $\mathcal{P}_\mathcal{X} \times \mathcal{X}$, to speed up the algorithm. We used Liblinear package for the implementation [42]. All hyperparameters were selected using five fold cross-validation and experiments were repeated 10 times. We show results in Table 2.8. The proposed method performs the best in three datasets and equally well in the one remaining dataset. The more our method outperforms pooling, the more knowledge can be shared between tasks.

Dataset	Pooling	Proposed Method
Synthetic	70.73 (±2.30)	**25.40** (±1.72)
Satellite	11.95 (±0.46)	**8.28** (±0.79)
HAR	1.69 (±0.56)	**1.68** (±0.58)
MNIST-MOD	22.79 (±1.38)	**21.39** (±1.24)

Table 2.8: Percentage Classification Error on Multiclass Datasets

II.8 Conclusion

In this work, we give scalable implementation of the kernel-based algorithm for domain generalization of [5] and extend the generalization error analysis to the multiclass setting.

We implemented the approach, demonstrating its improved performance with respect to a pooling strategy on multiple data sets.

II.9 Proofs

II.9.1 Proof of Theorem 2

Proof. Observe:

$$\bar{k}(\tilde{x}, \tilde{x}') = \exp\left\{\frac{-1}{2\sigma_P^2}\|\Psi(\widehat{P}_X) - \Psi(\widehat{P}'_X)\|^2\right\}\exp\left\{\frac{-1}{2\sigma_X^2}\|x - x'\|^2\right\},$$

and denote:

$$\tilde{k}(\tilde{x}, \tilde{x}') = \exp\left\{\frac{-1}{2\sigma_P^2}\|Z_P(\widehat{P}_X) - Z_P(\widehat{P}'_X)\|^2\right\}\exp\left\{\frac{-1}{2\sigma_X^2}\|x - x'\|^2\right\},$$

We omit the arguments of $\bar{k}, \tilde{k}$ for brevity. Let k_q be the final approximation ($k_q = \bar{z}(\tilde{x})^T\bar{z}(\tilde{x}')$) and then we have

$$|\bar{k} - k_q| = |\bar{k} - \tilde{k} + \tilde{k} - k_q| \leq |\bar{k} - \tilde{k}| + |\tilde{k} - k_q|. \tag{2.21}$$

From Eqn. (2.21) it follows that,

$$P(|\bar{k} - k_q| \geq \epsilon_l + \epsilon_q) \leq P(|\bar{k} - \tilde{k}| \geq \epsilon_l) + P(|\tilde{k} - k_q| \geq \epsilon_q). \tag{2.22}$$

By a direct application of Hoeffding's inequality,

$$P(|\tilde{k} - k_q| \geq \epsilon_q) \leq 2\exp(-\frac{Q\epsilon_q^2}{2}). \tag{2.23}$$

Recall that $\langle \Psi(\widehat{P}_X), \Psi(\widehat{P}'_X) \rangle = \frac{1}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} k'_X(X_i, X'_j)$. For a pair X_i, X'_j, we have again by Hoeffding

$$P(|z'_X(X_i)^T z'_X(X'_j) - k'_X(X_i, X'_j)| \geq \epsilon) \leq 2\exp(-\frac{L\epsilon^2}{2}).$$

Let Ω_{ij} be the event $|z'_X(X_i)^T z'_X(X'_j) - k'_X(X_i, X'_j)| \geq \epsilon$, for particular i, j. Using the union bound we have

$$P(\Omega_{11} \cup \Omega_{12} \cup \ldots \cup \Omega_{n_1 n_2}) \leq 2n_1 n_2 \exp(-\frac{L\epsilon^2}{2})$$

This implies

$$P(|Z_P(\widehat{P}_X)^T Z_P(\widehat{P}'_X) - \langle \Psi(\widehat{P}_X), \Psi(\widehat{P}'_X) \rangle| \geq \epsilon) \leq 2n_1 n_2 \exp(-\frac{L\epsilon^2}{2}). \tag{2.24}$$

Therefore,

$$\begin{aligned}
\left|\bar{k} - \tilde{k}\right| &= \left| \exp\left\{ \frac{-1}{2\sigma_X^2} \|x - x'\|^2 \right\} \left[\exp\left\{ \frac{-1}{2\sigma_P^2} \|\Psi(\widehat{P}_X) - \Psi(\widehat{P}'_X)\|^2 \right\} \right.\right. \\
&\quad \left.\left. - \exp\left\{ \frac{-1}{2\sigma_P^2} \|Z_P(\widehat{P}_X) - Z_P(\widehat{P}'_X)\|^2 \right\} \right] \right| \\
&\leq \left| \left[\exp\left\{ \frac{-1}{2\sigma_P^2} \|\Psi(\widehat{P}_X) - \Psi(\widehat{P}'_X)\|^2 \right\} - \exp\left\{ \frac{-1}{2\sigma_P^2} \|Z_P(\widehat{P}_X) - Z_P(\widehat{P}'_X)\|^2 \right\} \right] \right|
\end{aligned}$$

$$\begin{aligned}
&= \left| \exp\left\{ \frac{-1}{2\sigma_P^2} \|\Psi(\widehat{P}_X) - \Psi(\widehat{P}'_X)\|^2 \right\} \left[1 - \exp\left\{ \frac{-1}{2\sigma_P^2} \Big(\|Z_P(\widehat{P}_X) - Z_P(\widehat{P}'_X)\|^2 \right.\right.\right. \\
&\quad \left.\left.\left. - \|\Psi(\widehat{P}_X) - \Psi(\widehat{P}'_X)\|^2 \Big) \right\} \right] \right| \\
&\leq \left| \left[1 - \exp\left\{ \frac{-1}{2\sigma_P^2} \Big(\|Z_P(\widehat{P}_X) - Z_P(\widehat{P}'_X)\|^2 - \|\Psi(\widehat{P}_X) - \Psi(\widehat{P}'_X)\|^2 \Big) \right\} \right] \right| \\
&= \left| 1 - \exp\left\{ \frac{-1}{2\sigma_P^2} \Big(Z_P(\widehat{P}_X)^T Z_P(\widehat{P}_X) - \langle \Psi(\widehat{P}_X), \Psi(\widehat{P}_X) \rangle + Z_P(\widehat{P}'_X)^T Z_P(\widehat{P}'_X) \right.\right. \\
&\quad \left.\left. - \langle \Psi(\widehat{P}'_X), \Psi(\widehat{P}'_X) \rangle - 2\Big(Z_P(\widehat{P}_X)^T Z_P(\widehat{P}'_X) - \langle \Psi(\widehat{P}_X), \Psi(\widehat{P}'_X) \rangle \Big) \Big) \right\} \right|
\end{aligned}$$

$$\begin{aligned}
&\leq \Bigg|1 - \exp\Bigg\{\frac{1}{2\sigma_P^2}\Bigg(|Z_P(\widehat{P}_X)^T Z_P(\widehat{P}_X) - \langle\Psi(\widehat{P}_X), \Psi(\widehat{P}_X)\rangle| + |Z_P(\widehat{P}'_X)^T Z_P(\widehat{P}'_X) \\
&\quad - \langle\Psi(\widehat{P}'_X), \Psi(\widehat{P}'_X)\rangle| + 2|\Big(Z_P(\widehat{P}_X)^T Z_P(\widehat{P}'_X) - \langle\Psi(\widehat{P}_X), \Psi(\widehat{P}'_X)\rangle\Big)|\Bigg)\Bigg\}\Bigg|
\end{aligned}$$

The result now follows by applying the bound of Eqn. (2.24) to each of the three terms in the exponent of the preceding expression, together with the stated formula for ϵ in terms of ϵ_ℓ.

□

II.9.2 Proof of Theorem 3

Proof. The proof is very similar to the proof of Theorem 2. We use Lemma 1 to replace bound (2.23) with:

$$P(\sup_{x,x'\in\mathcal{M}} |\tilde{k} - k_q| \geq \epsilon_q) \leq 2^8\Big(\frac{\sigma'_X r}{\epsilon_q}\Big)^2 \exp(\frac{-Q\epsilon_q^2}{2(d+2)}). \tag{2.25}$$

Similarly, Eqn. (2.24) is replaced by

$$\begin{aligned}
&P(\sup_{x,x'\in\mathcal{M}} |Z_P(\widehat{P}_X)^T Z_P(\widehat{P}'_X) - \langle\Psi(\widehat{P}_X), \Psi(\widehat{P}'_X)\rangle| \geq \epsilon) \\
&\qquad \leq 2^9 n_1 n_2 \Big(\frac{\sigma_P \sigma_X r}{\epsilon_l}\Big)^2 \exp(\frac{-L\epsilon_l^2}{2(d+2)}).
\end{aligned} \tag{2.26}$$

The remainder of the proof now proceeds as in the previous proof.

□

Theorem 5. ***(Hoeffding's Inequality in Hilbert spaces [57])*** *Let* $(\Omega, \mathcal{A}, P)$ *be a prob-*

ability space, H be a separable Hilbert space, and $B > 0$. Furthermore, let $\eta_1, ..., \eta_n : \Omega \to H$ be independent H-valued random variables satisfying $\|\eta_i\|_\infty \leq B \ \forall i = 1, ..., n$. Then, for all $\delta \in (0, 1)$, we have

$$P\Big(\|\frac{1}{n}\sum_{i=1}^{n}(\eta_i - \mathbb{E}_P\eta_i)\|_H \geq B\sqrt{\frac{2\log\delta^{-1}}{n}} + B\sqrt{\frac{1}{n}} + \frac{4B\log\delta^{-1}}{3n}\Big) \leq \delta.$$

II.9.3 Proof of Theorem 4

Proof. The proof follows the general structure of the theorem presented in [52]. Without loss of generality, it is assumed that $n_i = n$. The function f can be split into c components $f = \begin{bmatrix} g_1 & g_2 & \cdots & g_c \end{bmatrix}$. We are interested in error bounds over $f \in \mathcal{B}^c_{\bar{k}}(R) := \prod_{m=1}^{c} \mathcal{B}_{\bar{k}}(R)$ and $g_m \in \mathcal{B}_{\bar{k}}(R), m \in \{1, 2, ..., c\}$.

$$\begin{aligned}
\sup_{f \in \mathcal{B}^c_{\bar{k}}(R)} |\widehat{\varepsilon}(f) - \varepsilon(f)| \quad &\leq \quad \sup_{f \in \mathcal{B}^c_{\bar{k}}(R)} \Big|\widehat{\varepsilon}(f) - \frac{1}{N}\sum_{i=1}^{N}\frac{1}{n_i}\sum_{j=1}^{n_i}\ell(f(P^i_X, X_{ij}), Y_{ij})\Big| \\
&+ \quad \sup_{f \in \mathcal{B}^c_{\bar{k}}(R)} \Big|\frac{1}{N}\sum_{i=1}^{N}\frac{1}{n_i}\sum_{j=1}^{n_i}\ell(f(P^i_X, X_{ij}), Y_{ij}) - \varepsilon(f)\Big|. \\
&=: \quad (I) + (II)
\end{aligned}$$

Let's bound the first term:

$$\begin{aligned}
\Big|\widehat{\varepsilon}(f) - \frac{1}{N}\sum_{i=1}^{N}\frac{1}{n_i}\sum_{j=1}^{n_i}\ell(f(P^i_X, X_{ij}), Y_{ij})\Big| \quad &\leq \quad L_\ell\frac{1}{N}\sum_{i=1}^{N}\frac{1}{n_i}\sum_{j=1}^{n_i}\Big\|f(\widehat{P}^i_X, X_{ij}) - f(P^i_X, X_{ij})\Big\|_2 \\
&\leq \quad L_\ell\frac{1}{N}\sum_{i=1}^{N}\frac{1}{n_i}\sum_{j=1}^{n_i}\Big\|f(\widehat{P}^i_X, X_{ij}) - f(P^i_X, X_{ij})\Big\|_1 \\
&= \quad L_\ell\frac{1}{N}\sum_{i=1}^{N}\frac{1}{n_i}\sum_{j=1}^{n_i}\sum_{l=1}^{c}\Big|g_l(\widehat{P}^i_X, X_{ij}) - g_l(P^i_X, X_{ij})\Big|.
\end{aligned}$$

$$\tag{2.27}$$

Now, let us look at the term $|g_l(\widehat{P}^i_X, X_{ij}) - g_l(P^i_X, X_{ij})|$ for some $l \in \{1, 2, ..., c\}$.

Using the reproducing property of the kernel

$$\begin{aligned}
|g_l(\widehat{P}_X, X) - g_l(P_X, X)| &= |\langle \bar{k}\Big((\widehat{P}_X, X), \cdot\Big) - \bar{k}\Big((P_X, X), \cdot\Big), g_l\rangle| \\
&\leq \|g_l\| \ \|\bar{k}\Big((\widehat{P}_X, X), \cdot\Big) - \bar{k}\Big((P_X, X), \cdot\Big)\| \\
&= \|g_l\| \ \Bigg(\bar{k}\Big((\widehat{P}_X, X), (\widehat{P}_X, X)\Big) + \bar{k}\Big((P_X, X), (P_X, X)\Big) \\
&\qquad - 2\bar{k}\Big((\widehat{P}_X, X), (P_X, X)\Big)\Bigg)^{\frac{1}{2}} \\
&= \|g_l\| \ k_x(X, X)^{\frac{1}{2}}\Bigg(k_P\Big(\Psi(\widehat{P}_X), \Psi(\widehat{P}_X)\Big) + k_P\Big(\Psi(P_X), \Psi(P_X)\Big) \\
&\qquad - 2k_P\Big(\Psi(\widehat{P}_X), \Psi(P_X)\Big)\Bigg)^{\frac{1}{2}} \\
&= \|g_l\| B_k \ \|\phi_{k_P}(\Psi(P_X)) - \phi_{k_P}(\Psi(\widehat{P}_X))\| \\
&\leq \|g_l\| B_k L_{k_P} \ \|\Psi(P_X) - \Psi(\widehat{P}_X)\|^{\alpha}
\end{aligned} \tag{2.28}$$

Where the last inequality is due to the Hölder continuity of the kernel k_P. We can bound $\|\Psi(P_X) - \Psi(\widehat{P}_X)\|$ using Hoeffding's inequality in the Hilbert space $\mathcal{H}_{k'_x}$.

Using Theorem 5: with probability at least $1 - \delta$ we have,

$$\|\Psi(P_X) - \Psi(\widehat{P}_X)\| \leq B_{k'}\sqrt{\frac{2\log\delta^{-1}}{n}} + B_{k'}\sqrt{\frac{1}{n}} + \frac{4B_{k'}\log\delta^{-1}}{3n}. \tag{2.29}$$

Combining equation 2.28 and 2.29, with at least probability $1-\delta$

$$\left|g_l(\widehat{P}^i_X, X_{ij}) - g_l(P^i_X, X_{ij})\right| \leq \|g_l\| B_k L_{k_P} \ \left(B_{k'}\sqrt{\frac{2\log\delta^{-1}}{n}} + B_{k'}\sqrt{\frac{1}{n}} + \frac{4B_{k'}\log\delta^{-1}}{3n}\right)^{\alpha}.$$

Using the union bound, with at least probability $1-\delta$

$$\frac{1}{N}\sum_{i=1}^{N}\left|g_l(\widehat{P}^i_X, X_{ij}) - g_l(P^i_X, X_{ij})\right| \leq \|g_l\| B_k L_{k_P} \ \left(B_{k'}\sqrt{\frac{2\log\frac{N}{\delta}}{n}} + B_{k'}\sqrt{\frac{1}{n}} + \frac{4B_{k'}\log\frac{N}{\delta}}{3n}\right)^{\alpha}. \qquad (2.30)$$

Combining equation 2.27 and 2.30, with at least probability $1-\delta$

$$\sup_{f\in\mathcal{B}^c_k(R)}\left|\widehat{\varepsilon}(f) - \frac{1}{N}\sum_{i=1}^{N}\frac{1}{n_i}\sum_{j=1}^{n_i}\ell(f(P^i_X, X_{ij}), Y_{ij})\right| \leq L_\ell L_{k_P} R B_k c(B_{k'})^{\alpha}\left(\sqrt{\frac{2\log\frac{N}{\delta}}{n}} + \sqrt{\frac{1}{n}} + \frac{4\log\frac{N}{\delta}}{3n}\right)^{\alpha}. \qquad (2.31)$$

Bounding the second term is similar to bounding term (II) in Theorem 5 in [52] with modifications for multi-class loss. We defined term (II) as

$$(II) := \sup_{f\in\mathcal{B}^c_k(R)}\left|\frac{1}{N}\sum_{i=1}^{N}\frac{1}{n_i}\sum_{j=1}^{n_i}\ell(f(P^i_X, X_{ij}), Y_{ij}) - \varepsilon(f)\right|$$

We define $(II)'$ as the one sided version of term (II) i.e.,

$$(II)' := \sup_{f\in\mathcal{B}^c_k(R)}\frac{1}{N}\sum_{i=1}^{N}\frac{1}{n_i}\sum_{j=1}^{n_i}\ell(f(P^i_X, X_{ij}), Y_{ij}) - \varepsilon(f)$$

We have

$$\begin{aligned}(II)' \leq & \sup_{f \in \mathcal{B}^c_{\bar{k}}(R)} \frac{1}{N}\sum_{i=1}^{N}\frac{1}{n_i}\sum_{j=1}^{n_i} \ell(f(P^i_X, X_{ij}), Y_{ij}) - \mathbb{E}[\ell(f(\tilde{X}, Y))\big| P^i_{XY}] \\ & + \sup_{f \in \mathcal{B}^c_{\bar{k}}(R)} \frac{1}{N}\sum_{i=1}^{N}\left(\mathbb{E}[\ell(f(\tilde{X}, Y))\big| P^i_{XY}] - \mathbb{E}[\ell(f(\tilde{X}, Y))]\right) \qquad (2.32) \\ =: & (IIa) + (IIb)\end{aligned}$$

Control of Term (IIa) Conditional to P^1_{XY},...,P^N_{XY} , the random variables $(X_{ij}, Y_{ij})_{ij}$ are independent (not identically distributed). We apply Azuma-McDiarmid's inequality to

$$\zeta((X_{ij}, Y_{ij})_{ij}) = \sup_{f \in \mathcal{B}^c_{\bar{k}}(R)} \frac{1}{N}\sum_{i=1}^{N}\frac{1}{n_i}\sum_{j=1}^{n_i} \ell(f(P^i_X, X_{ij}), Y_{ij}) - \mathbb{E}[\ell(f(\tilde{X}, Y))\big| P^i_{XY}]$$

We have with probability at least $1 - \delta$ that

$$\left|\zeta - \mathbb{E}[\zeta|(P^i_{XY})_{1\leq i\leq N}]\right| \leq \sqrt{C_\zeta \log(1/\delta)}$$

Where $C_\zeta = \frac{B^2_\ell}{Nn}$. Next we bound the expectation term using standard Rademacher complexity analysis and then applying the extension to Talagrand's convex concentration inequality (see [58] theorem 7 and lemma 22). Let $(\epsilon_{ij})_{1\leq i\leq N, 1\leq j\leq n_i}$ be i.i.d Rademacher random variables.

$$\begin{aligned}&\mathbb{E}[\zeta|(P^i_{XY})_{1\leq i\leq N}] \\ &= \mathbb{E}_{(X_{ij},Y_{ij})}\left[\sup_{f \in \mathcal{B}^c_{\bar{k}}(R)} \frac{1}{N}\sum_{i=1}^{N}\frac{1}{n_i}\sum_{j=1}^{n_i} \ell(f(P^i_X, X_{ij}), Y_{ij}) - \mathbb{E}[\ell(f(\tilde{X}, Y))\big| P^i_{XY}]\middle| (P^i_{XY})_{1\leq i\leq N}\right] \\ &\leq \frac{2}{N}\mathbb{E}_{(X_{ij},Y_{ij})}\mathbb{E}_{(\epsilon_{ij})}\left[\sup_{f \in \mathcal{B}^c_{\bar{k}}(R)} \sum_{i=1}^{N}\frac{1}{n_i}\sum_{j=1}^{n_i} \epsilon_{ij}\ell(f(P^i_X, X_{ij}), Y_{ij})\middle| (P^i_{XY})_{1\leq i\leq N}\right].\end{aligned}$$

We now apply the modified talagrand inequality as stated in Lemma 2. Let $(\sigma_{ijm})_{1\leq i\leq N, 1\leq j\leq n_i, 1\leq m\leq c}$

be i.i.d Rademacher random variables, then by assumption **A I** and Lemma 2,

$$\begin{aligned}&\mathbb{E}[\zeta|(P^i_{XY})_{1\le i\le N}]\\&\le\frac{2\sqrt{2}L_\ell}{N}\mathbb{E}_{(X_{ij},Y_{ij})}\mathbb{E}_{(\sigma_{ijm})}\left[\sup_{f\in\mathcal{B}^c_{\bar{k}}(R)}\sum_{i=1}^N\frac{1}{n_i}\sum_{j=1}^{n_i}\sum_{m=1}^{c}\sigma_{ijm}g_m(P^i_X,X_{ij}),Y_{ij})\bigg|(P^i_{XY})_{1\le i\le N}\right].\end{aligned}\tag{2.33}$$

Next, from the properties of supremum,

$$\begin{aligned}&\mathbb{E}[\zeta|(P^i_{XY})_{1\le i\le N}]\\&\le\frac{2\sqrt{2}L_\ell}{N}\mathbb{E}_{(X_{ij},Y_{ij})}\mathbb{E}_{(\sigma_{ijm})}\left[\sum_{m=1}^{c}\sup_{g_m\in\mathcal{B}_{\bar{k}}(R)}\sum_{i=1}^N\frac{1}{n_i}\sum_{j=1}^{n_i}\sigma_{ijm}g_m(P^i_X,X_{ij}),Y_{ij})\bigg|(P^i_{XY})_{1\le i\le N}\right]\\&=\frac{2\sqrt{2}L_\ell}{N}\sum_{m=1}^{c}\mathbb{E}_{(X_{ij},Y_{ij})}\mathbb{E}_{(\sigma_{ijm})}\left[\sup_{g_m\in\mathcal{B}_{\bar{k}}(R)}\sum_{i=1}^N\frac{1}{n_i}\sum_{j=1}^{n_i}\sigma_{ijm}g_m(P^i_X,X_{ij}),Y_{ij})\bigg|(P^i_{XY})_{1\le i\le N}\right].\end{aligned}$$

In the above equation, the terms in the summation with respect to m are over the same space $\mathcal{B}_{\bar{k}}(R)$. Now, for any $m\in\{1,2,...c\}$, we have

$$\mathbb{E}[\zeta|(P^i_{XY})_{1\le i\le N}]\le\frac{2\sqrt{2}L_\ell c}{N}\mathbb{E}_{(X_{ij},Y_{ij})}\mathbb{E}_{(\sigma_{ijm})}\left[\sup_{g_m\in\mathcal{B}_{\bar{k}}(R)}\sum_{i=1}^N\frac{1}{n_i}\sum_{j=1}^{n_i}\sigma_{ijm}g_m(P^i_X,X_{ij}),Y_{ij})\bigg|(P^i_{XY})_{1\le i\le N}\right].$$

Applying Lemma 22 from [58] and its related arguments,

$$\begin{aligned}&\mathbb{E}[\zeta|(P^i_{XY})_{1\le i\le N}]\\&\le\frac{4\sqrt{2}cRL_\ell B_k B_{k_P}}{N}\sqrt{\sum_{i=1}^N\sum_{m=1}^{c}\frac{1}{n_i}}.\end{aligned}$$

When $n_i=n$ we have

$$\mathbb{E}[\zeta|(P^i_{XY})_{1\le i\le N}]\le\frac{4\sqrt{2}RL_\ell B_k B_{k_P}c}{\sqrt{Nn}}.$$

Control of term (IIb) Let

$$\xi((P^i_{XY})_{1\leq i\leq N}) = \sup_{f\in\mathcal{B}^c_{\bar{k}}(R)} \frac{1}{N}\sum_{i=1}^{N}\Big(\mathbb{E}[\ell(f(\tilde{X},Y))\Big|P^i_{XY}] - \mathbb{E}[\ell(f(\tilde{X},Y))]\Big).$$

Since $(P^i_{XY})_{1\leq i\leq N}$ are i.i.d we can apply Azuma-McDiarmid inequality to ξ to obtain

$$|\xi - \mathbb{E}[\xi]| \leq B_\ell\sqrt{\frac{\log(1/\delta)}{2N}}.$$

We then bound $\mathbb{E}[\xi]$ using the standard Rademacher complexity analysis with the modified Talagrand's inequality similar to the bounding of (IIa). We have

$$\begin{aligned}
\mathbb{E}[\xi] =\ & \mathbb{E}_{(P^i_{XY})_{1\leq i\leq N}}\Bigg[\sup_{f\in\mathcal{B}^c_{\bar{k}}(R)} \frac{1}{N}\sum_{i=1}^{N}\Big(\mathbb{E}_{(X,Y)\sim P^i_{XY}}[\ell(f(\tilde{X},Y))] \\
&\qquad\qquad\qquad\qquad - \mathbb{E}_{P_{XY}\sim\mu,(X,Y)\sim P_{XY}}[\ell(f(\tilde{X},Y))]\Big)\Bigg] \\
\leq\ & \frac{2}{N}\mathbb{E}_{(P^i_{XY})_{1\leq i\leq N}}\mathbb{E}_{(\epsilon_i)_{1\leq i\leq N}}\Bigg[\sup_{f\in\mathcal{B}^c_{\bar{k}}(R)}\sum_{i=1}^{N}\epsilon_i\mathbb{E}_{(X_i,Y_i)\sim P^i_{XY}}[\ell(f(\tilde{X}_i,Y_i))]\Bigg] \\
\leq\ & \frac{2}{N}\mathbb{E}_{(P^i_{XY})_{1\leq i\leq N}}\mathbb{E}_{(X_i,Y_i)\sim P^i_{XY}}\mathbb{E}_{(\epsilon_i)_{1\leq i\leq N}}\Bigg[\sup_{f\in\mathcal{B}^c_{\bar{k}}(R)}\sum_{i=1}^{N}\epsilon_i[\ell(f(\tilde{X}_i,Y_i))]\Bigg] \\
\leq\ & \frac{4\sqrt{2}RL_\ell B_k B_{k_P}c}{\sqrt{N}}.
\end{aligned}$$

Where the third step is due to Jensen's inequality.

Combining terms (IIa) and (IIb), we can write

$$\begin{aligned}
(II)' &\leq \frac{4\sqrt{2}RL_\ell B_k B_{k_P}c}{\sqrt{Nn}} + \frac{4\sqrt{2}RL_\ell B_k B_{k_P}c}{\sqrt{N}} + B_\ell\sqrt{\frac{\log 2\delta^{-1}}{2N}} \\
&\leq \frac{8\sqrt{2}RL_\ell B_k B_{k_P}c}{\sqrt{N}} + B_\ell\sqrt{\frac{\log 2\delta^{-1}}{2N}}
\end{aligned} \tag{2.34}$$

since $n \geq 1$. The bound for term (II) can be obtained by replacing δ with $\delta/2$ as in standard Rademacher complexity analysis (see Theorem 2 of [59]). We obtain, with probability at least $1-\delta$

$$(II) \leq \frac{8\sqrt{2}RL_\ell B_k B_{k_P} c}{\sqrt{N}} + B_\ell \sqrt{\frac{\log 4\delta^{-1}}{2N}} \tag{2.35}$$

So for the final bound, combining 2.31, 2.35, with probability at least $1-\delta$

$$\begin{aligned} \sup_{f \in \mathcal{B}_{\bar{k}}^{c}(R)} |\widehat{\varepsilon}(f) - \varepsilon(f)| \leq & L_\ell L_{k_P} R B_k c (B_{k'})^\alpha \left(\sqrt{\frac{2\log\frac{2N}{\delta}}{n}} + \sqrt{\frac{1}{n}} + \frac{4\log\frac{2N}{\delta}}{3n} \right)^\alpha \\ & + \frac{8\sqrt{2}RL_\ell B_k B_{k_P} c}{\sqrt{N}} + B_\ell \sqrt{\frac{\log 8\delta^{-1}}{2N}} \end{aligned} \tag{2.36}$$

□

CHAPTER III

Multi-Task Learning for Contextual Bandits

Contextual bandits are a form of multi-armed bandit in which the agent has access to predictive side information (known as the context) for each arm at each time step, and have been used to model personalized news recommendation, ad placement, and other applications. In this work, we propose a multi-task learning framework for contextual bandit problems. Like multi-task learning in the batch setting, the goal is to leverage similarities in contexts for different arms so as to improve the agent's ability to predict rewards from contexts. We propose an upper confidence bound-based multi-task learning algorithm for contextual bandits, establish a corresponding regret bound, and interpret this bound to quantify the advantages of learning in the presence of high task (arm) similarity. We also describe an effective scheme for estimating task similarity from data, and demonstrate our algorithm's performance on several data sets.

III.1 Introduction

A multi-armed bandit (MAB) problem is a sequential decision making problem where, at each time step, an agent chooses one of several "arms," and observes some reward for the choice it made. The reward for each arm is random according to a fixed distribution, and the agent's goal is to maximize its cumulative reward [12] through a combination of exploring different arms and exploiting those arms that have yielded high rewards in the past [13, 14].

The contextual bandit problem is an extension of the MAB problem where there is some side information, called the context, associated to each arm [15]. Each context determines the distribution of rewards for the associated arm. The goal in contextual bandits is still to maximize the cumulative reward, but now leveraging the contexts to predict the expected reward of each arm. Contextual bandits have been employed to model various applications like news article recommendation [19], computational advertisement [20], website optimization [21] and clinical trials [22]. For example, in the case of news article recommendation, the agent must select a news article to recommend to a particular user. The arms are articles and contextual features are features derived from the article and the user. The reward is based on whether a user reads the recommended article.

One common approach to contextual bandits is to fix the class of policy functions (i.e., functions from contexts to arms) and try to learn the best function with time [23, 24, 25]. Most algorithms estimate rewards either separately for each arm, or have one single estimator that is applied to all arms. In contrast, our approach is to adopt the perspective of multi-task learning (MTL). The intuition is that some arms may be similar to each other, in which case it should be possible to pool the historical data for these arms to estimate the mapping from context to rewards more rapidly. For example, in the case of news article recommendation, there may be thousands of articles, and some of those are bound to be similar to each other.

The contextual bandit problem is formally stated in Problem 3.1. The total T trial reward is defined as $\sum_{t=1}^{T} r_{a_t,t}$ and the optimal T trial reward as $\sum_{t=1}^{T} r_{a_t^*,t}$, where $r_{a_t,t}$ is reward of the selected arm a_t at time t and a_t^* is the arm with maximum reward at trial t. The goal is to

Algorithm 3.1: Contextual Bandits

```
for $t = 1, ..., T$ do
    Observe context $x_{a,t} \in \mathbb{R}^d$ for all arms $a \in [N]$, where $[N] = \{1, ...N\}$
    Choose an arm $a_t \in [N]$
    Receive a reward $r_{a_t,t} \in \mathbb{R}$
    Improve arm selection strategy based on new observation $(x_{a_t,t}, a_t, r_{a_t,t})$
end
```

find an algorithm that minimizes the T trial regret

$$R(T) = \sum_{t=1}^{T} r_{a_t^*,t} - \sum_{t=1}^{T} r_{a_t,t}.$$

We focus on upper confidence bound (UCB) type algorithms for the remainder of the chapter. A UCB strategy is a simple way to represent the exploration and exploitation tradeoff. For each arm, there is an upper bound on reward, comprised of two terms. The first term is a point estimate of the reward, and the second term reflects the confidence in the reward estimate. The strategy is to select the arm with maximum UCB. The second term dominates when the agent is not confident about its reward estimates, which promotes exploration. On the other hand, when all the confidence terms are small, the algorithm exploits the best arm(s) [30].

In the popular UCB type contextual bandits algorithm called Lin-UCB, the expected reward of an arm is modeled as a linear function of the context, $\mathbb{E}[r_{a,t}|x_{a,t}] = x_{a,t}^T \theta_a^*$, where $r_{a,t}$ is the reward of arm a at time t and $x_{a,t}$ is the context of arm a at time t. To select the best arm, one estimate θ_a for each arm independently using the data for that particular arm [23]. In the language of multi-task learning, each arm is a task, and Lin-UCB learns each task independently.

In the theoretical analysis of the Lin-UCB [19] and its kernelized version Kernel-UCB [24] θ_a is replaced by θ, and the goal is to learn one single estimator using data from all the arms. In other words, the data from the different arms are pooled together and viewed as coming from a single task. These two approaches, independent and pooled learning, are two extremes, and reality often lies somewhere in between. In the MTL approach, we seek to pool some tasks together, while learning others independently.

We present an algorithm motivated by this idea and call it kernelized multi-task learning UCB (KMTL-UCB). Our main contributions are proposing a UCB type multi-task learning algorithm for contextual bandits, established a regret bound and interpreting the bound to reveal the impact of increased task similarity, introducing a technique for estimating task similarities on the fly, and demonstrating the effectiveness of our algorithm on several datasets.

This chapter is organized as follows. Section 2 describes related work and in Section 3 we propose a UCB algorithm using multi-task learning. Regret analysis is presented in Section 4, and our experimental findings are reported in Section 5. We conclude in Section 6.

III.2 Related Work

A UCB strategy is a common approach to quantify the exploration/exploitation tradeoff. At each time step t, and for each arm a, a UCB strategy estimates a reward $\hat{r}_{a,t}$ and a one-sided confidence interval above $\hat{r}_{a,t}$ with width $\hat{w}_{a,t}$. The term $ucb_{a,t} = \hat{r}_{a,t} + \hat{w}_{a,t}$ is called the UCB index or just UCB. Then at each time step t, the algorithm chooses the arm a with the highest UCB.

In contextual bandits, the idea is to view learning the mapping $x \mapsto r$ as a regression problem. Lin-UCB uses a linear regression model while Kernel-UCB uses a nonlinear regression model drawn from the reproducing kernel Hilbert space (RKHS) of a symmetric and positive definite (SPD) kernel. Either of these two regression models could be applied in either the *independent* setting or the *pooled* setting. In the independent setting, the regression function for each arm is estimated separately. This was the approach adopted by Li et al. [23] with a linear model. Regret analysis for both Lin-UCB and Kernel-UCB adopted the *pooled* setting [19, 24]. Kernel-UCB in the independent setting has not previously been considered to our knowledge, although the algorithm would just be a kernelized version of Li et al. [23]. We will propose a methodology that extends the above four combinations of setting (independent and pooled) and regression model (linear and nonlinear). Gaussian Process UCB (GP-UCB) uses a Gaussian prior on the regression function and is a Bayesian equivalent of Kernel-UCB [25].

There are some contextual bandit setups that incorporate multi-task learning. In Lin-UCB with Hybrid Linear Models the estimated reward consists of two linear terms, one that is arm-specific and another that is common to all arms [23]. Gang of bandits [60] uses a graph structure (e.g., a social network) to transfer the learning from one user to other for personalized recommendation. Collaborative filtering bandits [61] is a similar technique which clusters the users based on context. Contextual Gaussian Process UCB (CGP-UCB) builds on GP-UCB and has many elements in common with our framework [62]. We defer a more detailed comparison to CGP-UCB until later.

III.3 KMTL-UCB

We propose an alternate regression model that includes the independent and pooled settings as special cases. Our approach is inspired by work on transfer and multi-task learning in the batch setting [32, 63]. Intuitively, if two arms (tasks) are similar, we can pool the data for those arms to train better predictors for both.

Formally, we consider regression functions of the form

$$f : \tilde{X} \mapsto \mathcal{Y}$$

where $\tilde{X} = \mathcal{Z} \times \mathcal{X}$, and $\mathcal{Z}$ is what we call the *task similarity space*, $\mathcal{X}$ is the *context space* and $\mathcal{Y} \subseteq \mathbb{R}$ is the reward space. Every context $x_a \in \mathcal{X}$ is associated with an arm descriptor z_a, and we define $\tilde{x}_a = (z_a, x_a)$ to be the *augmented context.* Intuitively, z_a is a variable that can be used to determine the similarity between different arms. Examples of $\mathcal{Z}$ and z_a will be given below.

Let $\tilde{k}$ be a SPD kernel on $\tilde{X}$. In this work, we focus on kernels of the form

$$\tilde{k}\Big((z, x), (z', x')\Big) = k_{\mathcal{Z}}(z, z')k_{\mathcal{X}}(x, x'), \tag{3.1}$$

where $k_{\mathcal{X}}$ is a SPD kernel on $\mathcal{X}$, such as linear or Gaussian kernel if $\mathcal{X} = \mathcal{R}^d$, and $k_{\mathcal{Z}}$ is a kernel on $\mathcal{Z}$ (examples given below). Let $\mathcal{H}_{\tilde{k}}$ be the RKHS of functions $f : \tilde{X} \mapsto \mathbb{R}$ associated to $\tilde{k}$. Note that a product kernel is just one option for $\tilde{k}$, and other forms may be worth exploring.

III.3.1 Upper Confidence Bound

Instead of learning regression estimates for each arm separately, we effectively learn regression estimates for all arms at once by using all the available training data. Let N be the total number of distinct arms that the algorithm has to choose from. Define $[N] = \{1, ..., N\}$ and let the observed contexts at time t be $x_{a,t}, \forall a \in [N]$. Let $n_{a,t}$ be the number of times the algorithm has selected arm a up to and including time t so that $\sum_{a=1}^{N} n_{a,t} = t$. Define sets $t_a = \{\tau < t : a_\tau = a\}$, where a_τ is the arm selected at time τ. Notice that $|t_a| = n_{a,t-1}$ for all a.

We solve the following problem at time t:

$$\hat{f}_t = \arg\min_{f \in \mathcal{H}_{\tilde{k}}} \frac{1}{N} \sum_{a=1}^{N} \frac{1}{n_{a,t-1}} \sum_{\tau \in t_a} (f(\tilde{x}_{a,\tau}) - r_{a,\tau})^2 + \lambda \|f\|^2_{\mathcal{H}_{\tilde{k}}}, \tag{3.2}$$

where $\tilde{x}_{a,\tau}$ is the augmented context of arm a at time τ, and $r_{a,\tau}$ is the reward of an arm a selected at time τ. This problem (3.2) is a variant of kernel ridge regression. Applying the representer theorem [57] the optimal f can be expressed as $f = \sum_{a'=1}^{N} \sum_{\tau' \in t_{a'}} \alpha_{a',\tau'} \tilde{k}(\cdot, \tilde{x}_{a',\tau'})$, which yields the solution (detailed derivation is in the Section III.7)

$$\hat{f}_t(\tilde{x}) = \tilde{k}_{t-1}(\tilde{x})^T (\eta_{t-1} \tilde{K}_{t-1} + \lambda I)^{-1} \eta_{t-1} y_{t-1}, \tag{3.3}$$

where $\tilde{K}_{t-1}$ is the $(t-1) \times (t-1)$ kernel matrix on the augmented data $[\tilde{x}_{a_\tau,\tau}]_{\tau=1}^{t-1}$, $\tilde{k}_{t-1}(\tilde{x}) = [\tilde{k}(\tilde{x}, \tilde{x}_{a_\tau,\tau})]_{\tau=1}^{t-1}$ is a vector of kernel evaluations between $\tilde{x}$ and the past data, $y_{t-1} = [r_{a_\tau,\tau}]_{\tau=1}^{t-1}$ are all observed rewards, and η_{t-1} is the $(t-1) \times (t-1)$ diagonal matrix $\eta_{t-1} = \text{diag}[\frac{1}{n_{a_\tau,t-1}}]_{\tau=1}^{t-1}$.

When $\tilde{x} = \tilde{x}_{a,t}$, we write $\tilde{k}_{a,t} = \tilde{k}_{t-1}(\tilde{x}_{a,t})$. With only minor modifications to the argument in Valko et al [24], we have the following:

Lemma 3. *Suppose the rewards $[r_{a_\tau,\tau}]_{\tau=1}^{T}$ are independent random variables with means $\mathbb{E}[r_{a_\tau,\tau} | x_{a_\tau,\tau}] = f^*(\tilde{x}_{a_\tau,\tau})$, where $f^* \in \mathcal{H}_{\tilde{k}}$ and $\|f^*\|_{\mathcal{H}_{\tilde{k}}} \le c$. Let $\alpha = \sqrt{\frac{\log(2TN/\delta)}{2}}$ and $\delta > 0$. With probability at least $1 - \frac{\delta}{T}$, we have that $\forall a \in [N]$*

$$|\hat{f}_t(\tilde{x}_{a,t}) - f^*(\tilde{x}_{a,t})| \le w_{a,t} := (\alpha + c\sqrt{\lambda}) s_{a,t} \tag{3.4}$$

where $s_{a,t} = \lambda^{-1/2} \sqrt{\tilde{k}(\tilde{x}_{a,t}, \tilde{x}_{a,t}) - \tilde{k}_{a,t}^T (\eta_{t-1} \tilde{K}_{t-1} + \lambda I)^{-1} \eta_{t-1} \tilde{k}_{a,t}}$.

The result in Lemma 3 motivates the UCB

$$ucb_{a,t} = \hat{f}_t(x_{a,t}) + w_{a,t}$$

and inspires Algorithm 3.2.

```
Algorithm 3.2: KMTL-UCB
Input: Input: $\beta \in R_+$,
for $t = 1, ..., T$ do
    Update the (product) kernel matrix $\tilde{K}_{t-1}$ and $\eta_{t-1}$
    Observe context features at time $t$: $x_{a,t}$ for each $a \in [N]$.
    Determine arm descriptor $z_a$ for each $a \in [N]$ to get augmented context $\tilde{x}_{a,t}$.
    for all $a$ at time $t$ do
        $p_{a,t} \leftarrow \hat{f}_t(x_{a,t}) + \beta s_{a,t}$
    end

    Choose arm $a_t = \arg\max p_{a,t}$, observe a real valued payoff $r_{a_t,t}$ and update $y_t$ .
    Output: Output: $a_t$

end
```

Before an arm has been selected at least once, $\hat{f}_t(x_{a,t})$ and the second term in $s_{a,t}$, i.e., $\tilde{k}_{a,t}^T(\eta_{t-1}\tilde{K}_{t-1} + \lambda I)^{-1}\eta_{t-1}\tilde{k}_{a,t}$, are taken to be 0. In that case, the algorithm only uses the first term of $s_{a,t}$, i.e., $\sqrt{\tilde{k}(\tilde{x}_{a,t}, \tilde{x}_{a,t})}$, to form the UCB.

III.3.2 Choice of Task Similarity Space and Kernel

To illustrate the flexibility of our framework, we present the following three options for $\mathcal{Z}$ and $k_{\mathcal{Z}}$:

1. Independent: $\mathcal{Z} = \{1, ..., N\}$, $k_{\mathcal{Z}}(a, a') = \mathbb{1}_{a=a'}$. The augmented context for a context x_a from arm a is just (a, x_a).

2. Pooled: $\mathcal{Z} = \{1\}$, $k_{\mathcal{Z}} \equiv 1$. The augmented context for a context x_a for arm a is just $(1, x_a)$.

3. Multi-Task: $\mathcal{Z} = \{1, ..., N\}$ and $k_{\mathcal{Z}}$ is a PSD matrix reflecting arm/task similarities. If this matrix is unknown, it can be estimated as discussed below.

Algorithm 3.2 with the first two choices specializes to the independent and pooled settings mentioned previously. In either setting, choosing a linear kernel for $k_{\mathcal{X}}$ leads to Lin-UCB, while a more general kernel essentially gives rise to Kernel-UCB. We will argue that the multi-task setting facilitates learning when there is high task similarity.

We also introduce a fourth option for $\mathcal{Z}$ and $k_{\mathcal{Z}}$ that allows task similarity to be estimated when it is unknown. In particular, we are inspired by the kernel transfer learning framework of Blanchard et al. [32]. Thus, we define the arm similarity space to be $\mathcal{Z} = \mathcal{P}_{\mathcal{X}\mathcal{Y}}$, the set of all probability distributions on $\mathcal{X} \times \mathcal{Y}$. In this case $\mathcal{X}$ is a space of contexts and $\mathcal{Y}$ is a space of rewards. We further assume that contexts for arm a are drawn from probability measure P_a. Given a context x_a for arm a, we define its augmented context to be (P_a, x_a).

To define a kernel on $\mathcal{Z} = \mathcal{P}_{\mathcal{X}\mathcal{Y}}$, we use the same construction described in [32], originally introduced by Steinwart and Christmann [64]. In particular, in our experiments we use a Gaussian-like kernel

$$k_{\mathcal{Z}}(P_a, P_{a'}) = \exp(-\|\Psi(P_a) - \Psi(P_{a'})\|^2 / 2\sigma_{\mathcal{Z}}^2), \tag{3.5}$$

where $\Psi(P) = \displaystyle\int k'_{\mathcal{X}}(\cdot, x) y dP_{xy}$ is the kernel mean embedding of a distribution P. This embedding is defined by yet another SPD kernel $k'_{\mathcal{X}}$ on $\mathcal{X}$, which could be different from the $k_{\mathcal{X}}$ used to define $\tilde{k}$. We may estimate $\Psi(P_a)$ via $\Psi(\widehat{P}_a) = \dfrac{1}{n_{a,t-1}} \displaystyle\sum_{\tau \in t_a} r_{a,\tau} k'_{\mathcal{X}}(\cdot, x_{a_\tau,\tau})$, which leads to an estimate of $k_{\mathcal{Z}}$.

III.4 Theoretical Analysis

To simplify the analysis we consider a modified version of the original problem 3.2:

$$\hat{f}_t = \arg\min_{f \in \mathcal{H}_{\tilde{k}}} \frac{1}{N} \sum_{a=1}^{N} \sum_{\tau \in t_a} (f(\tilde{x}_{a,\tau}) - r_{a,\tau})^2 + \lambda \|f\|^2_{\mathcal{H}_{\tilde{k}}}. \tag{3.6}$$

In particular, this modified problem omits the terms $\dfrac{1}{n_{a,t-1}}$ as they obscure the analysis. In practice, these terms should be incorporated.

In this case $s_{a,t} = \lambda^{-1/2}\sqrt{\tilde{k}(\tilde{x}_{a,t}, \tilde{x}_{a,t}) - \tilde{k}_{a,t}^T(\tilde{K}_{t-1} + \lambda I)^{-1}\tilde{k}_{a,t}}$. Under this assumption Kernel-UCB is exactly KMTL-UCB with $k_{\mathcal{Z}} \equiv 1$. On the other hand, KMTL-UCB can be viewed as a special case of Kernel-UCB on the augmented context space $\tilde{\mathcal{X}}$. Thus, the regret analysis of Kernel-UCB applies to KMTL-UCB, but it does not reveal the potential gains of multi-task learning. We present an interpretable regret bound that reveals the benefits of MTL. We also establish a lower bound on the UCB width that decreases as task similarity increases (presented in the Section III.7).

III.4.1 Analysis of SupKMTL-UCB

It is not trivial to analyze algorithm 3.2 because the reward at time t is dependent on the past rewards. We follow the same strategy originally proposed in [16] and used in [19, 24] which uses SupKMTL-UCB as a master algorithm, and BaseKMTL-UCB (which is called by SupKMTL-UCB) to get estimates of reward and width. SupKMTL-UCB builds mutually exclusive subsets of $[T]$ such that rewards in any subset are independent. This guarantees that the independence assumption of Lemma 3 is satisfied. We describe these algorithms in the Section III.7.

Theorem 6. *Assume that $r_{a,t} \in [0,1], \forall a \in [N]$, $T \geq 1$, $\|f^*\|_{\mathcal{H}_{\tilde{k}}} \leq c$, $\tilde{k}(\tilde{x},\tilde{x}) \leq c_{\tilde{k}}, \forall \tilde{x} \in \tilde{X}$ and the task similarity matrix K_Z is known. With probability at least $1-\delta$, SupKMTL-UCB satisfies*

$$\begin{aligned} R(T) &\leq 2\sqrt{T} + 10\left(\sqrt{\frac{\log\left(2TN(\log(T)+1)/\delta\right)}{2}} + c\sqrt{\lambda}\right)\sqrt{2m\log g([T])}\sqrt{T\lceil\log(T)\rceil} \\ &= O\left(\sqrt{T\log(g([T]))}\right) \end{aligned}$$

where $g([T]) = \dfrac{\det(\tilde{K}_{T+1}+\lambda I)}{\lambda^{T+1}}$ *and* $m = \max(1, \dfrac{c_{\tilde{k}}}{\lambda})$.

Note that this theorem assumes that task similarity is known. In the experiments for real datasets using the approach discussed in subsection III.3.2 we estimate the task similarity from the available data.

III.4.2 Interpretation of Regret Bound

The following theorems help us interpret the regret bound by looking at

$$g([T]) = \frac{\det(\tilde{K}_{T+1}+\lambda I)}{\lambda^{T+1}} = \prod_{t=1}^{T+1}\frac{(\lambda_t+\lambda)}{\lambda},$$

where, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{T+1}$ are the eigenvalues of the kernel matrix $\tilde{K}_{T+1}$.

As mentioned above, the regret bound of Kernel-UCB applies to our method, and we are able to recover this bound as a corollary of Theorem 6. In the case of Kernel-UCB $\tilde{K}_t = K_{X_t}, \forall t \in [T]$ as all arm estimators are assumed to be the same. We define the effective rank of $\tilde{K}_{T+1}$ in the same way as [24] defines the effective dimension of the kernel feature space.

Definition 1. *The effective rank of $\tilde{K}_{T+1}$ is defined to be* $r := \min\{j : j\lambda \log T \geq \sum_{i=j+1}^{T+1} \lambda_i\}$.

In the following result, the notation $\tilde{O}$ hides logarithmic terms.

Corollary 1. $\log(g([T])) \leq r \log\left(2T \frac{2(T+1)c_{\tilde{k}} + r\lambda - r\lambda \log T}{r\lambda}\right)$*, and therefore* $R(T) = \tilde{O}(\sqrt{rT})$

However, beyond recovering a known bound, Theorem 6 can also be interpreted to reveal the potential gains of multi-task learning. To interpret the regret bound in Theorem 6, we make a further assumption that after time t, $n_{a,t} = \frac{t}{N}$ for all $a \in [N]$. For simplicity define $n_t = n_{a,t}$. Let ($\odot$) denote the Hadamard product, ($\otimes$) denote the Kronecker product and $\mathbb{1}_n \in R^n$ be the vector of ones. Let $K_{X_t} = [k_{\mathcal{X}}(x_{a_\tau,\tau}, x_{a_{\tau'},\tau'})]_{\tau,\tau'=1}^t$ be the $t \times t$ kernel matrix on contexts, $K_{Z_t} = [k_{\mathcal{Z}}(z_{a_\tau}, z_{a_{\tau'}})]_{\tau,\tau'=1}^t$ be the associated $t \times t$ kernel matrix based on arm similarity, and $K_Z = [k_{\mathcal{Z}}(z_a, z_a)]_{a=1}^N$ be the $N \times N$ arm/task similarity matrix between N arms, where $x_{a_\tau,\tau}$ is the observed context and z_{a_τ} is the associated arm descriptor. Using eqn. (3.1), we can write $\tilde{K}_t = K_{Z_t} \odot K_{X_t}$. We rearrange the sequence of $x_{a_\tau,\tau}$ to get $[x_{a,\tau}]_{a=1,\tau=(t+1)_a}^N$ such that elements $(a-1)n_t$ to an_t belong to arm a. Define $\tilde{K}_t^r, K_{X_t}^r$ and $K_{Z_t}^r$ to be the rearranged kernel matrices based on the re-ordered set $[x_{a,\tau}]_{a=1,\tau=(t+1)_a}^N$. Notice that we can write $\tilde{K}_t^r = (K_Z \otimes \mathbb{1}_{n_t}\mathbb{1}_{n_t}^T) \odot K_{X_t}^r$ and the eigenvalues $\lambda(\tilde{K}_t)$ and $\lambda(\tilde{K}_t^r)$ are equal. To summarize, we have

$$\begin{aligned} \tilde{K}_t &= K_{Z_t} \odot K_{X_t} \\ \lambda(\tilde{K}_t) &= \lambda\Big((K_Z \otimes \mathbb{1}_{n_t}\mathbb{1}_{n_t}^T) \odot K_{X_t}^r\Big). \end{aligned} \tag{3.7}$$

Theorem 7. *Let the rank of matrix $K_{X_{T+1}}$ be r_x and the rank of matrix K_Z be r_z. Then* $\log(g([T])) \leq \; r_z r_x \log\left(\frac{(T+1)c_{\tilde{k}} + \lambda}{\lambda}\right)$

This means that when the rank of the task similarity matrix is low, which reflects a high degree of inter-task similarity, the regret bound is tighter. For comparison, note that when all tasks are independent, $r_z = N$ and when all tasks are the same (pooled), then $r_z = 1$. In the case of Lin-UCB [19] where all arm estimators are assumed to be the same and $k_{\mathcal{X}}$ is a linear kernel, the regret bound in Theorem 6 evaluates to $\tilde{O}(\sqrt{dT})$, where d is the dimension of the context space. In the original Lin-UCB algorithm [23] where all arm estimators are different, the regret bound would be $\tilde{O}(\sqrt{NdT})$.

We can further comment on $g([T])$ when all distinct tasks (arms) are similar to each other with task similarity equal to μ. Thus, define $K_Z(\mu) := (1-\mu)I_N + \mu \mathbb{1}_N \mathbb{1}_N^T$ and $\tilde{K}_t^r(\mu) = (K_Z(\mu) \otimes \mathbb{1}_{n_t}\mathbb{1}_{n_t}^T) \odot K_{X_t}^r$.

Theorem 8. *Let* $g_\mu([T]) = \dfrac{\det(\tilde{K}_{T+1}^r(\mu) + \lambda I)}{\lambda^{T+1}}$. *If* $\mu_1 \leq \mu_2$ *then* $g_{\mu_1}([T]) \geq g_{\mu_2}([T])$.

This shows that when there is more task similarity, the regret bound is tighter.

III.4.3 Comparison with CGP-UCB

CGP-UCB transfers the learning from one task to another by leveraging additional known task-specific context variables [62], similar in spirit to KTML-UCB. Indeed, with slight modifications, KMTL-UCB can be viewed as a frequentist analogue of CGP-UCB, and similarly CGP-UCB could be modified to address our setting. Furthermore, the term $g([T])$ appearing in our regret bound is equivalent to an information gain term used to analyze CGP-UCB. In the agnostic case of CGP-UCB where there is no assumption of a Gaussian prior on decision functions, their regret bound is $O(\log(g([T]))\sqrt{T})$, while their regret bound matches ours when they adopt a GP prior on f^*. Thus, our primary contributions with respect to CGP-UCB are to quantify the gains of multi-task learning in the form of Theorems 2 and 3, and a technique for estimating task similarity which is critical for real-world applications. In contrast to our examples given below, the experiments in [62] assume a known task similarity matrix.

III.5 Experiments

We test our algorithm on synthetic data and some multi-class classification datasets. In the case of multi-class datasets, the number of arms N is the number of classes and the reward is 1 if we predict the correct class, otherwise it is 0. We separate the data into two parts - validation set and test set. We use all Gaussian kernels and we estimate the task similarity between arms only when both arms were selected correctly at least 5 times. We preselect 200 hyperparameter configurations and run the algorithm on validation set (with different sequences of streaming data) and select the best hyperparameter configuration based on minimum mean regret. Then we run the algorithm on the test set 10 times (with different sequences of streaming data) and report the mean regret. For the synthetic data, we compare Kernel-UCB in the independent setting (Kernel-UCB-Ind) and pooled setting (Kernel-UCB-Pool), KMTL-UCB with known task similarity, and KMTL-UCB-Est which estimates task similarity on the fly. For the real datasets in the multi-class classification setting, we compare Kernel-UCB-Ind and KMTL-UCB-Est. In this case, the pooled setting is not valid because $x_{a,t}$ is the same for all arms (only z_a differs) and KMTL-UCB is not valid because the task similarity matrix is unknown. The code is available online to reproduce all results [2].

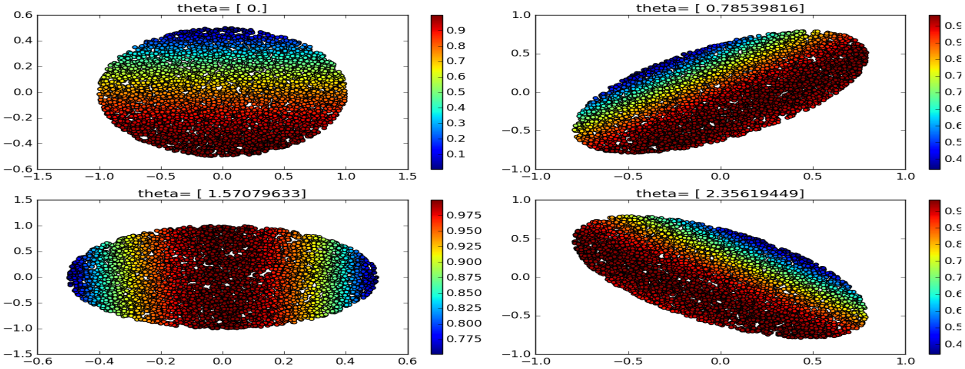
III.5.1 Synthetic News Article Data

Suppose an agent has access to a pool of articles and their context features. The agent then sees a user along with his/her features for which it needs to recommend an article. Based on user features and article features the algorithm gets a combined context $x_{a,t}$. The user context $x_{u,t} \in \mathbb{R}^2, \forall t$ is randomly drawn from an ellipse centered at $(0, 0)$ with major axis

[2] The code to reproduce our results is available at https://github.com/aniketde/MultiTaskLearningContextualBandits

length 1 and minor axis length 0.5. Let $x_{u,t}[:,1]$ be the minor axis and $x_{u,t}[:,2]$ be the major axis. Article context $x_{art,t}$ is any angle $\theta \in [0, \frac{\pi}{2}]$. To get the overall summary $x_{a,t}$ of user and article the user context $x_{u,t}$ is rotated with $x_{art,t}$. Rewards for each article are defined based on the minor axis $r_{a,t} = \left(1.0 - (x_{u,t}[:,1] - \frac{a}{N} + 0.5)^2\right)$. Figure 3.1 shows one such example for 4 different arms. The color code describes the reward, the two axes show the information about user context, and theta is the article context. We take $N = 5$. For KMTL-UCB, we use a Gaussian kernel on $x_{art,t}$ to get the task similarity.

Figure 3.1: Synthetic Data



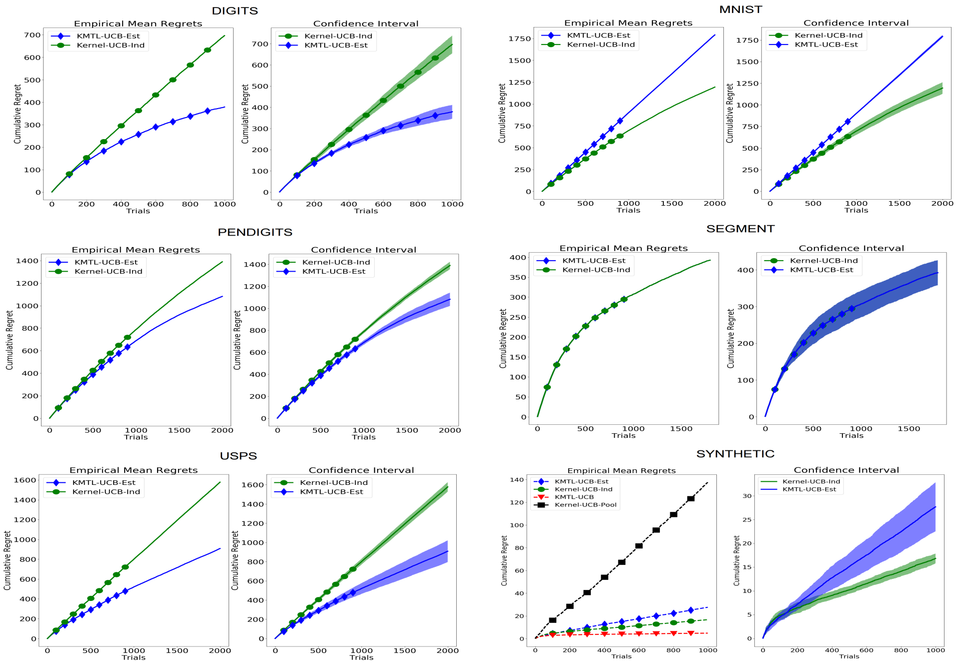
The results of this experiment are shown in last block of Figure 3.2. As one can see, Kernel-UCB-Pool performs the worst. That means for this setting combining all the data and learning a single estimator is not efficient. KMTL-UCB beats the other methods in all 10 runs.

III.5.2 Multi-class Datasets

In the case of multi-class classification, each class is an arm and the features of an example for which the algorithm needs to recommend a class are the contexts. We consider the following datasets: Digits $(N = 10, d = 64)$, MNIST $(N = 10,\ d = 780\ )$, Pendigits $(N = 10, d = 16)$, Segment $(N = 7, d = 19)$ and USPS $(N = 10, d = 256)$. Empirical mean regrets are shown in

Figure 3.2. KMTL-UCB-Est performs the best in three of the datasets and performs equally well in the one of the datasets.

Figure 3.2: Results on Multiclass Datasets - Empirical Mean Regret



III.6 Conclusions and future work

We present a multi-task learning framework in the contextual bandit setting and describe a way to estimate task similarity when it is not given. We give theoretical analysis, interpret the regret bound, and support the theoretical analysis with extensive experiments. We also establish a lower bound on the UCB width, and argue that it decreases as task similarity increases.

III.7 Proofs

III.7.1 KMTL Ridge Regression

Let $n_{a,t}$ be the number of times the algorithm has selected arm a up and including time t so that $\sum_{a=1}^{N} n_{a,t} = t$. Define sets $t_a = \{\tau < t : a_\tau = a\}$, where a_τ is the arm selected at time τ. Notice that $|t_a| = n_{a,t-1}$ for all a. We solve the following problem at time t:

$$\hat{f}_t = \arg\min_{f \in \mathcal{H}_{\tilde{k}}} \frac{1}{N} \sum_{a=1}^{N} \frac{1}{n_{a,t-1}} \sum_{\tau \in t_a} (f(\tilde{x}_{a,\tau}) - r_{a,\tau})^2 + \lambda \|f\|^2_{\mathcal{H}_{\tilde{k}}}, \tag{3.8}$$

where $\tilde{x}_{a,\tau}$ is augmented context and $r_{a,\tau}$ is the reward of arm a selected at time τ. We can minimize (3.8) by solving a variant of kernel ridge regression. Applying the representer theorem [57] the optimal f can be expressed as $f = \sum_{a'=1}^{N} \sum_{\tau' \in t_a} \alpha_{a'\tau'} \tilde{k}(\cdot, \tilde{x}_{a',\tau'})$. Plugging this in, we have the objective function

$$\begin{aligned} J(f) &= \frac{1}{N} \sum_{a=1}^{N} \frac{1}{n_{a,t-1}} \sum_{\tau \in t_a} (\sum_{a'=1}^{N} \sum_{\tau' \in t_a} \alpha_{a'\tau'} \tilde{k}(\tilde{x}_{a,\tau}, \tilde{x}_{a',\tau'}) - r_{a,\tau})^2 + \lambda \|f\|^2_{\mathcal{H}_{\tilde{k}}} \\ &= (y_{t-1} - \tilde{K}_{t-1}\alpha)^T \eta_{t-1} (y_{t-1} - \tilde{K}_{t-1}\alpha) + \lambda \alpha^T \tilde{K}_{t-1} \alpha \\ &= y_{t-1}^T \eta_{t-1} y_{t-1} - y_{t-1}^T \eta_{t-1} \tilde{K}_{t-1} \alpha - \alpha^T \tilde{K}_{t-1} \eta_{t-1} y_{t-1} \\ &\quad + \alpha^T \tilde{K}_{t-1} \eta_{t-1} \tilde{K}_{t-1} \alpha + \lambda \alpha^T \tilde{K}_{t-1} \alpha. \end{aligned}$$

Taking the gradient, we have

$$\frac{\partial J}{\partial \alpha} \quad = \quad -2\tilde{K}_{t-1}\eta_{t-1}y_{t-1} + 2\tilde{K}_{t-1}\eta_{t-1}\tilde{K}_{t-1}\alpha + 2\lambda\tilde{K}_{t-1}\alpha = 0.$$

Solving for α yields

$$\alpha \quad = \quad (\eta_{t-1}\tilde{K}_{t-1} + \lambda I)^{-1}\eta_{t-1}y_{t-1},$$

which implies

$$\hat{f}_t(\tilde{x}) \quad = \quad \tilde{k}_{t-1}(\tilde{x})^T(\eta_{t-1}\tilde{K}_{t-1} + \lambda I)^{-1}\eta_{t-1}y_{t-1}. \tag{3.9}$$

Here $\tilde{K}_{t-1}$ is the $(t-1)\times(t-1)$ kernel matrix on the augmented data $[\tilde{x}_{a_\tau,\tau}]_{\tau=1}^{t-1}$, $\tilde{k}_{t-1}(\tilde{x}) = [\tilde{k}(\tilde{x}, \tilde{x}_{a_\tau,\tau})]_{\tau=1}^{t-1}$ is a vector of kernel evaluations between $\tilde{x}$ and the past data, $y_{t-1} = [r_{a_\tau,\tau}]_{\tau=1}^{t-1}$ are all observed labels or rewards and η_{t-1} is the $(t-1)\times(t-1)$ diagonal matrix $\eta_{t-1} = \text{diag}[\frac{1}{n_{a_\tau}}]_{\tau=1}^{t-1}$.

We can also derive the solution without using the representer theorem. Let ϕ be a feature map associated with kernel $\tilde{k}$. Let

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{N}\sum_{a=1}^{N}\frac{1}{n_{a,t-1}}\sum_{\tau\in t_a}(\phi(\tilde{x}_{a,\tau})^T\theta - r_{a,\tau})^2 + \lambda\|\theta\|^2. \tag{3.10}$$

Minimizing eqn. (3.10) over θ gives,

$$\hat{\theta}_t = D_{t-1}^{-1}\Phi_{t-1}^T\eta_{t-1}y_{t-1}, \tag{3.11}$$

where $D_{t-1} = (\Phi_{t-1}^T \eta_{t-1} \Phi_{t-1} + \lambda I)$, $\Phi_t = [\phi(\tilde{x}_{a_\tau,\tau})^T]_{\tau=1}^t \in \mathbb{R}^{t \times \tilde{d}}$ and $\tilde{d}$ is the dimension of feature space $\phi(x)$. The equivalence between eqn. (3.9) and (3.11) follows from the matrix inversion lemma.

III.7.2 Upper Confidence Bound

Lemma 4. *Suppose the rewards $[r_{a_\tau,\tau}]_{\tau=1}^T$ are independent random variables with means $\mathbb{E}[r_{a_\tau,\tau}|x_{a_\tau,\tau}] = \phi(\tilde{x}_{a_\tau,\tau})^T\theta^*$, where $\|\theta^*\| \leq c$. Let $\alpha = \sqrt{\frac{\log(2TN/\delta)}{2}}$ and $\delta > 0$. With probability at least $1 - \frac{\delta}{T}$, we have that $\forall a \in [N]$*

$$|\phi(\tilde{x}_{a,t})^T\hat{\theta}_t - \phi(\tilde{x}_{a,t})^T\theta^*| \leq (\alpha + c\sqrt{\lambda})s_{a,t},$$

where $s_{a,t} = \sqrt{\phi(\tilde{x}_{a,t})^T D_t^{-1} \phi(\tilde{x}_{a,t})}$.

Proof. Proof of this theorem is similar to proof of Lemma 1 in [19]. For simplicity we write $D_{t-1} = D, \Phi_{t-1} = \Phi$, $y_{t-1} = y$ and $\eta_{t-1} = \eta$. Now

$$\begin{aligned}
\phi(\tilde{x}_{a,t})^T\hat{\theta}_t - \phi(\tilde{x}_{a,t})^T\theta^* &= \phi(\tilde{x}_{a,t})^T D^{-1}\Phi^T\eta y - \phi(\tilde{x}_{a,t})^T D^{-1} D\theta^* \\
&= \phi(\tilde{x}_{a,t})^T D^{-1}\Phi^T\eta y - \phi(\tilde{x}_{a,t})^T D^{-1}(\Phi^T\eta\Phi + \lambda I)\theta^* \\
&= \phi(\tilde{x}_{a,t})^T D^{-1}\Phi^T\eta y - \phi(\tilde{x}_{a,t})^T D^{-1}(\Phi^T\eta\Phi\theta^* + \lambda\theta^*) \\
&= \phi(\tilde{x}_{a,t})^T D^{-1}\Phi^T\eta(y - \Phi\theta^*) - \phi(\tilde{x}_{a,t})^T D^{-1}\lambda\theta^*.
\end{aligned}$$

Therefore

$$\begin{aligned}
|\phi(\tilde{x}_{a,t})^T\hat{\theta}_t - \phi(\tilde{x}_{a,t})^T\theta^*| &\leq |\phi(\tilde{x}_{a,t})^T D^{-1}\Phi^T\eta(y - \Phi\theta^*)| + \|\theta^*\|\|\phi(\tilde{x}_{a,t})^T D^{-1}\lambda\| \\
&\leq |\phi(\tilde{x}_{a,t})^T D^{-1}\Phi^T\eta(y - \Phi\theta^*)| + c\lambda||\phi(\tilde{x}_{a,t})^T D^{-1}||
\end{aligned}$$

where the first inequality is due to Cauchy-Schwarz.

Now we know that $\mathbb{E}y = \mathbb{E}[r_{a_\tau,\tau}]_{\tau=1,\dots,t-1} = \Phi\theta^* \implies \mathbb{E}[y - \Phi\theta^*] = 0$. Let $f(y^1, ..., y^{t-1}) = |\phi(\tilde{x}_{a,t})^T D^{-1} \Phi^T \eta (y - \Phi\theta^*)|$ and vector $V = \phi(\tilde{x}_{a,t})^T D^{-1} \Phi^T \eta$. Then

$$|f(y^1, ...y^i, ..., y^{t-1}) - f(y^1, ...\hat{y}^i, ..., y^{t-1})| = |V_i(y^i - \hat{y}^i)| \le |V_i|.$$

That means any component y_i can change $f(y^1, ..., y^{t-1})$ by at most $|V_i|$.

Using statistical independence of all random variables $r_{a_\tau,\tau}$ in a vector y and using McDiarmid's Inequality:

$$\begin{aligned} P(|\phi(\tilde{x}_{a,t})^T D^{-1} \Phi^T \eta (y - \Phi\theta^*)| \ge \alpha s_{a,t}) &\le 2\exp(-\frac{2\alpha^2 s_{a,t}^2}{\|V\|^2}) \\ &\le 2\exp(-2\alpha^2) \\ &= \frac{\delta}{TN} \end{aligned}$$

where the second inequality is due to

$$\begin{aligned} s_{a,t}^2 &= \phi(\tilde{x}_{a,t})^T D^{-1} \phi(\tilde{x}_{a,t}) \\ &= \phi(\tilde{x}_{a,t})^T D^{-1} (\Phi^T \eta \Phi + \lambda I) D^{-1} \phi(\tilde{x}_{a,t}) \\ &\ge \phi(\tilde{x}_{a,t})^T D^{-1} \Phi^T \eta \Phi D^{-1} \phi(\tilde{x}_{a,t}) \\ &\ge \phi(\tilde{x}_{a,t})^T D^{-1} \Phi^T \eta^2 \Phi D^{-1} \phi(\tilde{x}_{a,t}) \\ &= \|\eta \Phi D^{-1} \phi(\tilde{x}_{a,t})\|^2 \end{aligned}$$

$$= \quad \|V\|^2.$$

Now applying the union bound we can see that, with probability at least $1 - \frac{\delta}{T}$, $\forall a \in [N]$

$$|\phi(\tilde{x}_{a,t})^T D^{-1} \Phi^T \eta (y - \Phi\theta_a^*)| \leq \alpha s_{a,t}.$$

Bounding the second term:

$$\begin{aligned} c\lambda||\phi(\tilde{x}_{a,t})^T A_a^{-1}|| &= c\lambda\sqrt{\phi(\tilde{x}_{a,t})^T D^{-1} I D^{-1} \phi(\tilde{x}_{a,t})} \\ &\leq c\sqrt{\lambda}\sqrt{\phi(\tilde{x}_{a,t})^T D^{-1}(\lambda I + \Phi^T\Phi) D^{-1}\phi(\tilde{x}_{a,t})} \\ &= c\sqrt{\lambda}\sqrt{\phi(\tilde{x}_{a,t})^T D^{-1}\phi(\tilde{x}_{a,t})} \\ &= c\sqrt{\lambda} s_{a,t}. \end{aligned}$$

□

We kernelize $s_{a,t}$ in the following result.

III.7.2.1 Proof of Lemma 3

Proof. We use Lemma 4 to get the width and then kernelize it using techniques in [24]. Note that $\Phi\phi(\tilde{x}) = \tilde{k}_{t-1}(\tilde{x})$. When $\tilde{x} = \tilde{x}_{a,t}$, we write $\tilde{k}_{a,t} = \tilde{k}_{t-1}(\tilde{x}_{a,t})$. For simplicity we write $\eta_{t-1} = \eta$ and $\Phi_{t-1} = \Phi$. Since the matrices $(\Phi^T\eta\Phi + \lambda I)$, $(\eta\Phi\Phi^T + \lambda I)$ are regularized, they are strictly positive definite and hence their inverses are defined. Observe that

$$(\Phi^T\eta\Phi + \lambda I)\Phi^T \quad = \quad \Phi^T(\eta\Phi\Phi^T + \lambda I) \tag{3.12}$$

by associative property of matrix multiplication and

$$\Phi^T(\eta\Phi\Phi^T + \lambda I)^{-1} \quad = \quad (\Phi^T\eta\Phi + \lambda I)^{-1}\Phi^T \tag{3.13}$$

by multiplication of $(\Phi^T\eta\Phi + \lambda I)^{-1}$ and $(\eta\Phi\Phi^T + \lambda I)^{-1}$ on both sides. Also observe that

$$(\Phi^T\eta\Phi + \lambda I)\phi(\tilde{x}_{a,t}) \quad = \quad (\Phi^T\eta\tilde{k}_{a,t} + \lambda\phi(\tilde{x}_{a,t}))$$

by associative property of matrix multiplication and using $\Phi\phi(\tilde{x}_{a,t}) = \tilde{k}_{a,t}$. Multiplying on the left by $(\Phi^T\eta\Phi + \lambda I)^{-1}$,

$$\begin{aligned}\phi(\tilde{x}_{a,t}) \quad &= \quad (\Phi^T\eta\Phi + \lambda I)^{-1}(\Phi^T\eta\tilde{k}_{a,t} + \lambda\phi(\tilde{x}_{a,t}))\\ &= \quad (\Phi^T\eta\Phi + \lambda I)^{-1}\Phi^T\eta\tilde{k}_{a,t} + \lambda(\Phi^T\eta\Phi + \lambda I)^{-1}\phi(\tilde{x}_{a,t})\\ &= \quad \Phi^T(\eta\Phi\Phi^T + \lambda I)^{-1}\eta\tilde{k}_{a,t} + \lambda(\Phi^T\eta\Phi + \lambda I)^{-1}\phi(\tilde{x}_{a,t}) \qquad (3.14)\end{aligned}$$

where the last step is due to eqn. (3.13).

Multiplying both sides of eqn. (3.14) by $\phi(\tilde{x}_{a,t})^T$ we get,

$$\phi(\tilde{x}_{a,t})^T\phi(\tilde{x}_{a,t}) \quad = \quad \tilde{k}_{a,t}^T(\eta\Phi\Phi^T + \lambda I)^{-1}\eta\tilde{k}_{a,t} + \lambda\phi(\tilde{x}_{a,t})^T(\Phi^T\eta\Phi + \lambda I)^{-1}\phi(\tilde{x}_{a,t})$$

or, equivalently,

$$\tilde{k}(\tilde{x}_{a,t}, \tilde{x}_{a,t}) \quad = \quad \tilde{k}_{a,t}^T(\eta\tilde{K}_{t-1} + \lambda I)^{-1}\eta\tilde{k}_{a,t}^T + \lambda s_{a,t}^2.$$

By rearranging terms, we get

$$s_{a,t} = \lambda^{-1/2}\sqrt{\tilde{k}(\tilde{x}_{a,t}, \tilde{x}_{a,t}) - \tilde{k}_{a,t}^T(\eta_{t-1}\tilde{K}_{t-1} + \lambda I)^{-1}\eta_{t-1}\tilde{k}_{a,t}}. \tag{3.15}$$

□

III.7.3 UCB Width

In this subsection we establish a lower bound on the UCB width. To simplify the analysis we consider a problem:

$$\hat{f}_t = \arg\min_{f\in\mathcal{H}_{\tilde{k}}} \frac{1}{N}\sum_{a=1}^{N}\sum_{\tau\in t_a}(f(\tilde{x}_{a,\tau}) - r_{a,\tau})^2 + \lambda\|f\|_{\mathcal{H}_{\tilde{k}}}^2, \tag{3.16}$$

as $\frac{1}{n_{a,t-1}}$ obscures the analysis. In this case $s_{a,t} = \lambda^{-1/2}\sqrt{\tilde{k}(\tilde{x}_{a,t}, \tilde{x}_{a,t}) - \tilde{k}_{a,t}^T(\tilde{K}_{t-1} + \lambda I)^{-1}\tilde{k}_{a,t}}$. Let ($\odot$) denote the Hadamard product and ($\otimes$) denote the Kronecker product.

Lemma 5. [65] *Let A be a positive definite matrix partitioned according to*

$$A = \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right].$$

Then

$$A_{22} \geq A_{22} - A_{12}^T A_{11}^{-1} A_{12} \geq \frac{4\lambda_{\max}\lambda_{\min}}{\left(\lambda_{\max} + \lambda_{\min}\right)^2} A_{22}$$

where $\lambda_{\max}$ and $\lambda_{\min}$ are the maximum and minimum eigenvalues of A and $A \geq B$ means $A - B$ is a positive semidefinite matrix.

Lemma 6. [66] *Let D, C be positive semidefinite matrices. Any eigenvalue $\lambda(D \odot C)$ of $D \odot C$*

satisfies

$$\lambda(D \odot C) \leq \lambda_{\max}(D \odot C) \leq |\max_i d_{ii}|\lambda_{max}(C)$$

and

$$|\min_i d_{ii}|\lambda_{min}(C) \leq \lambda_{\min}(D \odot C) \leq \lambda(D \odot C).$$

Lemma 7. [67] *Let $D \in \mathbb{R}^{n\times n}$ and $C \in \mathbb{R}^{m\times m}$. Any eigenvalue $\lambda(D \otimes C)$ of $D \otimes C \in \mathbb{R}^{nm\times nm}$ is equal to the product of an eigenvalue of D and an eigenvalue of C.*

We assume that $n_{a,t} = \frac{t}{N}$ after time t to get interpretibility (this is not needed for the general regret bound that we prove in Theorem 6). For simplicity define $n_t = n_{a,t}$. Let $(\odot)$ denote the Hadamard product, $(\otimes)$ denote the Kronecker product and $\mathbb{1}_n \in R^n$ be the vector of ones. Let $K_{X_t} = [k_{\mathcal{X}}(x_{a_\tau,\tau}, x_{a_{\tau'},\tau'})]_{\tau,\tau'=1}^t$ be the $t \times t$ kernel matrix on contexts, $K_{Z_t} = [k_{\mathcal{Z}}(z_{a_\tau}, z_{a_{\tau'}})]_{\tau,\tau'=1}^t$ be the associated $t \times t$ kernel matrix based on arm similarity, and $K_Z = [k_{\mathcal{Z}}(z_a, z_a)]_{a=1}^N$ be the $N \times N$ arm similarity matrix between N arms, where $x_{a_\tau,\tau}$ is observed context and z_{a_τ} is an associated arm descriptor. Using the definition of tildek, $\tilde{k}\Big((z,x),(z',x')\Big) = k_{\mathcal{Z}}(z,z')k_{\mathcal{X}}(x,x')$, we can write $\tilde{K}_t = K_{Z_t} \odot K_{X_t}$. We rearrange a sequence of $x_{a_\tau,\tau}$ to get $[x_{a,\tau}]_{a=1,\tau=(t+1)_a}^N$ such that elements $(a-1)n_t$ to an_t belong to arm a. Define $\tilde{K}_t^r, K_{X_t}^r$ and $K_{Z_t}^r$ be rearranged kernel matrices based on the re-ordered set $[x_{a,\tau}]_{a=1,\tau=(t+1)_a}^N$. Notice that we can write $\tilde{K}_t^r = (K_Z \otimes \mathbb{1}_{n_t}\mathbb{1}_{n_t}^T) \odot K_{X_t}^r$ and the eigenvalues $\lambda(\tilde{K}_t)$ and $\lambda(\tilde{K}_t^r)$ are equal. To summarize, we have

$$\tilde{K}_t \quad = \quad K_{Z_t} \odot K_{X_t}$$

and

$$\lambda(\tilde{K}_t) \quad = \quad \lambda\Big((K_Z \otimes \mathbb{1}_{n_t}\mathbb{1}_{n_t}^T) \odot K_{X_t}^r\Big). \tag{3.17}$$

Lemma 8. *Assume $\tilde{k}(\tilde{x},\tilde{x}) \leq c_{\tilde{k}}, \forall \tilde{x} \in \tilde{X}$, and let $\tilde{K}_t$ be the final product kernel matrix and*

K_Z be the task similarity matrix. Also write

$$\tilde{K}_t + \lambda I_t = \left[\begin{array}{c|c} \tilde{K}_{t-1} + \lambda I_{t-1} & \tilde{k}_{a,t} \\ \hline \tilde{k}_{a,t}^T & \tilde{k}(\tilde{x}_{a,t}, \tilde{x}_{a,t}) + \lambda \end{array}\right].$$

Then

$$L_{s_{a,t}} = \frac{4nc_{\tilde{k}}\lambda_{\max}(K_Z) + \lambda}{\left(nc_{\tilde{k}}\lambda_{\max}(K_Z) + 2\lambda\right)^2}\left(\tilde{k}(\tilde{x}_{a,t}, \tilde{x}_{a,t}) + \lambda\right) - 1 \le s_{a,t}^2 \le \frac{c_{\tilde{k}}}{\lambda}. \tag{3.18}$$

Proof. Using Lemma 5,

$$\tilde{k}(\tilde{x}_{a,t}, \tilde{x}_{a,t}) + \lambda - \tilde{k}_{a,t}^T(\tilde{K}_{t-1} + \lambda I_{t-1})^{-1}\tilde{k}_{a,t} \quad \le \quad \tilde{k}(\tilde{x}_{a,t}, \tilde{x}_{a,t}) + \lambda.$$

Subtracting λ from both sides,

$$\lambda s_{a,t}^2 \quad \le \quad \tilde{k}(\tilde{x}_{a,t}, \tilde{x}_{a,t})$$

and therefore

$$s_{a,t}^2 \quad \le \quad \frac{c_{\tilde{k}}}{\lambda}.$$

This proves the upper bound. Again by using Lemma 5,

$$\tilde{k}(\tilde{x}_{a,t}, \tilde{x}_{a,t}) + \lambda - \tilde{k}_{a,t}^T(\tilde{K}_{t-1} + \lambda I_{t-1})^{-1}\tilde{k}_{a,t} \ge \frac{4\lambda_{\max}(\tilde{K}_t + \lambda I_t)\lambda_{\min}(\tilde{K}_t + \lambda I_t)}{\left(\lambda_{\max}(\tilde{K}_t + \lambda I_t) + \lambda_{\min}(\tilde{K}_t + \lambda I_t)\right)^2}\left(\tilde{k}(\tilde{x}_{a,t}, \tilde{x}_{a,t}) + \lambda\right)$$

Notice that the right hand side of the above equation is a monotonically decreasing function of $\frac{\lambda_{\max}}{\lambda_{\min}}$. Then

$$\begin{aligned}
\lambda s_{a,t}^2 + \lambda &\geq \frac{4\lambda_{\max}(\tilde{K}_t + \lambda I_t)\lambda_{\min}(\tilde{K}_t + \lambda I_t)}{\left(\lambda_{\max}(\tilde{K}_t + \lambda I_t) + \lambda_{\min}(\tilde{K}_t + \lambda I_t)\right)^2}\left(\tilde{k}(\tilde{x}_{a,t}, \tilde{x}_{a,t}) + \lambda\right) \\
&= \frac{\frac{4\lambda_{\max}(\tilde{K}_t)+\lambda}{\lambda_{\min}(\tilde{K}_t)+\lambda}}{\left(\frac{\lambda_{\max}(\tilde{K}_t)+\lambda}{\lambda_{\min}(\tilde{K}_t)+\lambda} + 1\right)^2}\left(\tilde{k}(\tilde{x}_{a,t}, \tilde{x}_{a,t}) + \lambda\right) \\
&= \frac{\frac{4\lambda_{\max}(\tilde{K}_t^r)+\lambda}{\lambda_{\min}(\tilde{K}_t^r)+\lambda}}{\left(\frac{\lambda_{\max}(\tilde{K}_t^r)+\lambda}{\lambda_{\min}(\tilde{K}_t^r)+\lambda} + 1\right)^2}\left(\tilde{k}(\tilde{x}_{a,t}, \tilde{x}_{a,t}) + \lambda\right) \\
&\geq \frac{\frac{4c_{\tilde{k}}\lambda_{\max}(K_{Z_t}^r)+\lambda}{\min_i K_{X_t}^r(ii)\lambda_{\min}(K_{Z_t}^r)+\lambda}}{\left(\frac{c_{\tilde{k}}\lambda_{\max}(K_{Z_t}^r)+\lambda}{\min_i K_{X_t}^r(ii)\lambda_{\min}(K_{Z_t}^r)+\lambda} + 1\right)^2}\left(\tilde{k}(\tilde{x}_{a,t}, \tilde{x}_{a,t}) + \lambda\right)
\end{aligned}$$

where $K_{X_t}^r(ii)$ are the diagonal elements of $K_{X_t}^r$ and the last inequality is due to Lemma 6. The smallest eigenvalue of $\mathbb{1}_{n_t}\mathbb{1}_{n_t}^T$ is zero and therefore according to Lemma 7, the smallest eigenvalue of $K_{Z_t}^r$ is zero. This implies

$$\begin{aligned}
\lambda s_{a,t}^2 + \lambda &\geq \frac{\frac{4nc_{\tilde{k}}\lambda_{\max}(K_{Z_t}^r)+\lambda}{\lambda}}{\left(\frac{nc_{\tilde{k}}\lambda_{\max}(K_{Z_t}^r)+\lambda}{\lambda} + 1\right)^2}\left(\tilde{k}(\tilde{x}_{a,t}, \tilde{x}_{a,t}) + \lambda\right) \\
&= \frac{4nc_{\tilde{k}}\lambda_{\max}(K_Z) + \lambda}{\left(nc_{\tilde{k}}\lambda_{\max}(K_Z) + 2\lambda\right)^2}\left(\tilde{k}(\tilde{x}_{a,t}, \tilde{x}_{a,t}) + \lambda\right)\lambda
\end{aligned}$$

where the last equality is again due to Lemma 7. Dividing both sides by λ and then subtracting one gives

$$s_{a,t}^2 \geq \frac{4nc_{\tilde{k}}\lambda_{\max}(K_Z) + \lambda}{\left(nc_{\tilde{k}}\lambda_{\max}(K_Z) + 2\lambda\right)^2}\left(\tilde{k}(\tilde{x}_{a,t}, \tilde{x}_{a,t}) + \lambda\right) - 1$$

□

Theorem 9 below says that the lower bound on width decreases as task similarity increases. In particular, assume that all distinct tasks are similar to each other with task similarity equal to μ and there are N tasks (arms). Thus $K_Z(\mu) := (1-\mu)I_N + \mu \mathbb{1}_N \mathbb{1}_N^T$.

Define

$$L_{s_{a,t}}(\mu) := \frac{4nc_{\tilde{k}}\lambda_{\max}(K_Z(\mu)) + \lambda}{\left(nc_{\tilde{k}}\lambda_{\max}(K_Z(\mu)) + 2\lambda\right)^2}\left(\tilde{k}(\tilde{x}_{a,t}, \tilde{x}_{a,t}) + \lambda\right) - 1.$$

Theorem 9. *Let $L_{s_{a,t}}$ be the lower bound on width as defined in Lemma 8. If $\mu_1 \leq \mu_2$ then*

$$L_{s_{a,t}}(\mu_1) \geq L_{s_{a,t}}(\mu_2). \tag{3.19}$$

Proof. The eigenvalues of $K_Z(\mu) = (1-\mu)I_N + \mu \mathbb{1}_N \mathbb{1}_N^T$ are $1 + \mu(N-1)$ with multiplicity 1 and $1-\mu$ with multiplicity $N-1$.

That means $\lambda_{\max}(K_Z(\mu))$ is highest when tasks are more similar and it decreases as task similarity μ goes to zero. The theorem follows as $L_{s_{a,t}(\mu)}$ is a monotonically decreasing function of $\lambda_{\max}(K_Z(\mu))$ □

This is important because if the lower bound on $s_{a,t}$ is small then we may be more confident about the reward estimates and this may lead to a tighter regret bound. In the next subsection we discuss the upper bound on regret.

III.7.4 Regret Analysis

We use the Lemma 24to prove the Lemma 10

Lemma 9 (Lemma 1.1 in [68]). *Let $A \in \mathbb{R}^{n \times n}$ be a positive definite matrix partitioned*

```
Algorithm 3.3: BaseKMTL-UCB at step $t$
Input: Input: $\alpha \in R_+, c, \lambda, \Psi \subseteq \{1, 2, ..., t-1\}$
Get $\tilde{K}_\Psi = \Phi_\Psi \Phi_\Psi^T$, where $\Phi_\Psi = [\phi(\tilde{x}_{a_\tau,\tau})^T]_{\tau \in \Psi}$
Get $y_\Psi = \begin{bmatrix} r_{a_\tau,\tau} \end{bmatrix}_{\tau \in \Psi}$
Observe context features at time $t$: $x_{a,t}$ for each $a \in N$
Calculate $\tilde{k}_{a,\Psi} = \Phi_\Psi^T \phi(\tilde{x}_{a,t})$ and $\tilde{k}(\tilde{x}_{a,t}, \tilde{x}_{a,t})$ for each $a \in N$.
for all $a$ at time $t$ do
    $s_{a,t} = \lambda^{-1/2} \sqrt{\tilde{k}(\tilde{x}_{a,t}, \tilde{x}_{a,t}) - \tilde{k}_{a,\Psi}^T (\tilde{K}_\Psi + \lambda I)^{-1} \tilde{k}_{a,\Psi}}$
    $ucb_{a,t} \leftarrow \tilde{k}_{a,\Psi}^T (\tilde{K}_\Psi + \lambda I)^{-1} y_\Psi + (\alpha + c\sqrt{\lambda}) s_{a,t}$
end
```

according to

$$A = \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{12}^T & A_{22} \end{array}\right].$$

where $A_{11} \in \mathbb{R}^{(n-1)\times(n-1)}, A_{12} \in \mathbb{R}^{(n-1)}$ *and* $A_{22} \in \mathbb{R}^1$. *Then* $\det(A) = \det(A_{11})(A_{22} - A_{12}^T A_{11}^{-1} A_{12})$.

Using the notations of BaseKMTL-UCB, we write $\tilde{K}_\Psi = \Phi_\Psi \Phi_\Psi^T$ and $\tilde{k}_{a,\Psi} = \Phi_\Psi^T \phi(\tilde{x}_{a,t})$ where $\Phi_\Psi = [\phi(\tilde{x})_{a_\tau,\tau}^T]_{\tau \in \Psi}$ and $\Psi \subseteq \{1, ..., t-1\}$. Define

$$\tilde{K}_{\Psi+1} + \lambda I = \left[\begin{array}{c|c} \tilde{K}_\Psi + \lambda I_{|\Psi|} & \tilde{k}_{a,\Psi} \\ \hline \tilde{k}_{a,\Psi}^T & \tilde{k}(\tilde{x}_{a,t}, \tilde{x}_{a,t}) + \lambda \end{array}\right]$$

Also, define $\tilde{k}_1 = \tilde{k}(\tilde{x}_{a_\sigma,\sigma}, \tilde{x}_{a_\sigma,\sigma})$, where σ is the smallest element of Ψ.

Algorithm 3.4: SupKMTL-UCB

```
Using same notation as in [19]:
Input: Input: $\alpha \in R_+, T \in \mathbb{N}$
$Q \leftarrow \lceil \log T \rceil$, $\Psi_1^q \leftarrow \emptyset$ and $\forall q \in [Q]$.
for $t = 1, \ldots, T$ do
    $q \leftarrow 1$ and $\hat{A}_1 \leftarrow [N]$
    repeat
        $s_{a,t}, ucb_{a,t} \leftarrow$ BaseKMTL-UCB with $\Psi_t^q$ and $\alpha$, for all $a \in \hat{A}_q$
        $w_{a,t} = (\alpha + c\sqrt{\lambda}) s_{a,t}$
        if $w_{a,t} \le \dfrac{1}{\sqrt{T}}$ for all $a \in \hat{A}_q$ then
            Choose $a_t = \arg\max_{a \in \hat{A}_q} ucb_{a,t}$ and $\Psi_{t+1}^{q'} \leftarrow \Psi_t^{q'}$ for all $q' \in [Q]$
        end

        else if $w_{a,t} \le 2^{-q}$ for all $a \in \hat{A}_q$ then
            $\hat{A}_{q+1} \leftarrow \{a \in \hat{A}_q | ucb_{a,t} \ge \max_{a' \in \hat{A}_q} ucb_{a',t} - 2^{1-q}\}$ and $q \leftarrow q + 1$
        end

        else
            Choose $a_t \in \hat{A}_q$ such that $w_{a_t,t} > 2^{-q}$
            Update $\Psi_{t+1}^q \leftarrow \Psi_t^q \cup \{t\}$ and $\forall q' \ne q$, $\Psi_{t+1}^{q'} \leftarrow \Psi_t^{q'}$
        end

    until $a_t$ is found

    Observe reward $r_{a_t,t}$
end
```

Lemma 10. *Using notations in BaseKMTL-UCB and suppose* $|\Psi| \geq 2$. *Then*

$$\sum_{\tau \in \Psi} s^2_{a_\tau,\tau} \leq 2m \log g(\Psi),$$

where $m = \max(1, \frac{c_{\tilde{k}}}{\lambda})$ *and*

$$g(\Psi) = \frac{\det(\tilde{K}_{\Psi+1} + \lambda I)}{\lambda^{|\Psi|+1}}.$$

Proof. Using the Lemma 24,

$$\begin{aligned}
\det(\tilde{K}_{\Psi+1} + \lambda I) &= (\tilde{k}_1 + \lambda) \prod_{\tau \in \Psi \setminus \{\sigma\}} \lambda(1 + s^2_{a_\tau,\tau}) \\
&= \lambda(\frac{\tilde{k}_1}{\lambda} + 1) \prod_{\tau \in \Psi \setminus \{\sigma\}} \lambda(1 + s^2_{a_\tau,\tau}) \\
&= \lambda \prod_{\tau \in \Psi} \lambda(1 + s^2_{a_\tau,\tau}),
\end{aligned}$$

where the last step is because $s^2_{a_\sigma,\sigma} = \frac{k_1}{\lambda}$.

From Lemma 8, $\max s^2_{a_\tau,\tau} = \frac{c_{\tilde{k}}}{\lambda}$. When $\frac{c_{\tilde{k}}}{\lambda} \leq 1$, using $x \leq 2\log(1+x), \forall x \in [0,1]$, $s^2_{a_\tau,\tau} \leq 2\log(1 + s^2_{a_\tau,\tau})$. In this case,

$$\begin{aligned}
\sum_{\tau \in \Psi} s^2_{a_\tau,\tau} &\leq 2 \sum_{\tau \in \Psi} \log(1 + s^2_{a_\tau,\tau}) \\
&= 2 \log \prod_{\tau \in \Psi} (1 + s^2_{a_\tau,\tau}) \\
&= 2 \log \frac{\det(\tilde{K}_{\Psi+1} + \lambda I)}{\lambda^{|\Psi|+1}}.
\end{aligned}$$

When $\frac{c_{\tilde{k}}}{\lambda} > 1$,

$$\begin{aligned}\sum_{\tau\in\Psi} \frac{c_{\tilde{k}}}{\lambda}\frac{\lambda}{c_{\tilde{k}}} s_{a_\tau,\tau}^2 &\le \frac{2c_{\tilde{k}}}{\lambda}\sum_{\tau\in\Psi}\log(1+\frac{\lambda}{c_{\tilde{k}}}s_{a_\tau,\tau}^2)\\ &\le \frac{2c_{\tilde{k}}}{\lambda}\sum_{\tau\in\Psi}\log(1+s_{a_\tau,\tau}^2)\\ &= \frac{2c_{\tilde{k}}}{\lambda}\log\prod_{\tau\in\Psi}(1+s_{a_\tau,\tau}^2)\\ &= \frac{2c_{\tilde{k}}}{\lambda}\log\frac{\det(\tilde{K}_{\Psi+1}+\lambda I)}{\lambda^{|\Psi|+1}}.\end{aligned}$$

Combining both cases,

$$\begin{aligned}\sum_{\tau\in\Psi} s_{a_\tau,\tau}^2 &\le 2\max(1,\frac{c_{\tilde{k}}}{\lambda})\log\frac{\det(\tilde{K}_{\Psi+1}+\lambda I)}{\lambda^{|\Psi|+1}}\\ &= 2m\log g(\Psi).\end{aligned}$$

□

Lemma 11. *Using the same notations as in Lemma 10,*

$$\sum_{\tau\in\Psi} s_{a_\tau,\tau} \le \sqrt{2m|\Psi|\log g(\Psi)}$$

Proof.

$$\begin{aligned}\sum_{t\in\Psi} s_{a_\tau,\tau} &\le \sqrt{|\Psi|\sum_{\tau\in\Psi} s_{a_\tau,\tau}^2}\\ &\le \sqrt{2|\Psi|m\log\frac{\det(\tilde{K}_{\Psi+1}+\lambda I)}{\lambda^{|\Psi|+1}}}\end{aligned}$$

where the first inequality is due to Cauchy-Schwarz and the last inequality is due to Lemma

10. □

Lemma 12. [16] *Using notations in SupKMTL-UCB, for each $t \in [T]$, $q \in [Q]$, and any fixed sequence of feature vectors $x_{a_t,t}$ with $t \in \Psi_t^q$, the corresponding rewards $r_{a_t,t}$ are independent random variables such that $\mathbb{E}[r_{a_t,t}] = \phi(\tilde{x}_{a_t,t})^T\theta^*$.*

Lemma 13. [16] *Using notations in SupKMTL-UCB, let $\|\theta^*\| \leq c$ and a_t^* be the best arm at time t. With probability $1 - \delta Q$ and $\forall t \in [T], q \in [Q]$, the following hold*

- $|\phi(\tilde{x}_{a,t})^T\hat{\theta}_t - \mathbb{E}[r_{a,t}|x_{a,t}]| \leq \left(\sqrt{\frac{\log 2TN/\delta}{2}} + \sqrt{\lambda}c\right)s_{a,t}$
- $a_t^* \in \hat{A}_q$
- $\mathbb{E}[r_{a_t^*,t}] - \mathbb{E}[r_{a,t}] \leq 2^{3-q}$.

Lemma 14. *Using notations in SupKMTL-UCB, $\forall q \in [Q]$,*

$$|\Psi_{T+1}^q| \leq 2^q\left(\sqrt{\frac{\log 2TN/\delta}{2}} + c\sqrt{\lambda}\right)\sqrt{2m\Big(\log g([T])\Big)|\Psi_{T+1}^q|}$$

where $[T] = \{1, ..., T\}$.

Proof.

$$\begin{aligned}\sum_{t\in\Psi_{T+1}^q} w_{a_t,t} &= \sum_{t\in\Psi_{T+1}^q}\left(\sqrt{\frac{\log 2TN/\delta}{2}} + c\sqrt{\lambda}\right)s_{a_t,t} \\ &\leq \left(\sqrt{\frac{\log 2TN/\delta}{2}} + c\sqrt{\lambda}\right)\sqrt{2m|\Psi_{T+1}^q|\log g(\Psi_{T+1}^q)} \\ &\leq \left(\sqrt{\frac{\log 2TN/\delta}{2}} + c\sqrt{\lambda}\right)\sqrt{2m\Big(\log g([T])\Big)|\Psi_{T+1}^q|}\end{aligned}$$

where the first inequality is due to Lemma 11 and the last inequality holds because $1 + s_{a_t,t}^2 \geq 1$ for all t.

From the third step (line 16) in SupKMTL-UCB algorithm 3.4, we choose and alternative $a_t \in \hat{A}_q$ such that $w_{a_t,t} \geq 2^{-q}$ and include that t in Ψ^q_{t+1} for the next round of estimates. Therefore,

$$\sum_{t \in \Psi^q_{T+1}} w_{a_t,t} \geq 2^{-q} |\Psi^q_{T+1}|$$

.

Combining the above two equations completes the proof. □

Lemma 15. [Azuma's inequality [69]] *Let $r_1, ..., r_T$ be random variables with $|r_\tau| \leq a_\tau$, for some $a_1, ..., a_T \geq= 0$. Then*

$$P\left(\left| \sum_{\tau=1}^{T} r_\tau - \sum_{\tau=1}^{T} \mathbb{E}[r_\tau | r_1, ..., r_{\tau-1}] \right| \geq B \right) \leq 2 \exp\left(- \frac{B^2}{2 \sum_{\tau=1}^{T} a_\tau^2} \right) \tag{3.20}$$

III.7.5 Proof of Theorem 6

We use same proof technique proposed by Auer et al. [16].

Proof. Let Ψ_0 be the set of trials for which an alternative ($w_{a,t} \leq \frac{1}{\sqrt{T}}$) at line 9 of SupKMTL-UCB algorithm 3.4 is chosen . Since $2^{-Q} \leq \frac{1}{\sqrt{T}}$, we have $\{1, ..., T\} = \Psi_0 \cup \bigcup_q \Psi^q_{T+1}$.

With probability $1 - \delta Q$,

$$\begin{aligned}
\mathbb{E}[R(T)] &= \sum_{t=1}^{T} \mathbb{E}[r_{a^*_t,t}] - \mathbb{E}[r_{a_t,t}] \\
&= \sum_{t \in \Psi_0} \mathbb{E}[r_{a^*_t,t}] - \mathbb{E}[r_{a_t,t}] + \sum_{q=1}^{Q} \sum_{t \in \Psi^q_{T+1}} \mathbb{E}[r_{a^*_t,t}] - \mathbb{E}[r_{a_t,t}] \\
&\leq \frac{2}{\sqrt{T}} \Psi_0 + \sum_{q=1}^{Q} \sum_{t \in \Psi^q_{T+1}} \mathbb{E}[r_{a^*_t,t}] - \mathbb{E}[r_{a_t,t}]
\end{aligned}$$

$$
\begin{aligned}
&\leq \frac{2}{\sqrt{T}}T + \sum_{q=1}^{Q} \sum_{t\in\Psi_{T+1}^{q}} 2^{3-q} \\
&\leq 2\sqrt{T} + \sum_{q=1}^{Q} 2^{3-q}|\Psi_{T+1}^{q}| \\
&\leq 2\sqrt{T} + \sum_{q=1}^{Q} 2^{3-q}2^{q}\Big(\sqrt{\frac{\log 2TN/\delta}{2}} + c\sqrt{\lambda}\Big)\sqrt{2m\Big(\log g([T])\Big)|\Psi_{T+1}^{q}|} \\
&\leq 2\sqrt{T} + 8\Big(\sqrt{\frac{\log 2TN/\delta}{2}} + c\sqrt{\lambda}\Big)\sqrt{2m\Big(\log g([T])\Big)}\sum_{q=1}^{Q}\sqrt{|\Psi_{T+1}^{q}|} \\
&\leq 2\sqrt{T} + 8\Big(\sqrt{\frac{\log 2TN/\delta}{2}} + c\sqrt{\lambda}\Big)\sqrt{2m\Big(\log g([T])\Big)}\sqrt{Q\sum_{q=1}^{Q}|\Psi_{T+1}^{q}|} \\
&\leq 2\sqrt{T} + 8\Big(\sqrt{\frac{\log 2TN/\delta}{2}} + c\sqrt{\lambda}\Big)\sqrt{2m\Big(\log g([T])\Big)}\sqrt{QT}
\end{aligned}
$$

where the first inequality is because of line 9 of SupKMTL-UCB algorithm 3.4, the second inequality is due to Lemma 13 and the fourth inequality is due to Lemma 14.

Using $B = \sqrt{2T\log(2/\delta)}$ and $a_\tau = 1$ in Azuma's inequality (Lemma 15), with probability at least $1-\delta(Q+1)$,

$$
\begin{aligned}
R(T) &\leq \mathbb{E}[R(T)] + \sqrt{2T\log(2/\delta)} \\
&\leq 2\sqrt{T} + 8\Big(\sqrt{\frac{\log 2TN/\delta}{2}} + c\sqrt{\lambda}\Big)\sqrt{2m\Big(\log g([T])\Big)}\sqrt{QT} + \sqrt{2T\log(2/\delta)} \\
&\leq 2\sqrt{T} + 10\Big(\sqrt{\frac{\log 2TN/\delta}{2}} + c\sqrt{\lambda}\Big)\sqrt{2m\Big(\log g([T])\Big)}\sqrt{QT}.
\end{aligned}
$$

Replacing δ with $\dfrac{\delta}{Q+1}$, we get that with probability at least $1-\delta$,

$$
R(T) \leq 2\sqrt{T} + 10\Big(\sqrt{\frac{\log 2TN(Q+1)/\delta}{2}} + c\sqrt{\lambda}\Big)\sqrt{2m\Big(\log g([T])\Big)}\sqrt{QT} \tag{3.21}
$$

$$\leq \quad 2\sqrt{T} + 10\left(\sqrt{\frac{\log 2TN(\log(T)+1)/\delta}{2}} + c\sqrt{\lambda}\right)\sqrt{2m \log g([T]}\sqrt{T\lceil\log(T)\rceil}. \quad (3.22)$$

□

We use following definitions and lemmas to interpret the regret bound and to establish a regret bound in terms of the effective rank of the kernel matrix.

Definition 2. *Let* $x, y \in \mathbb{R}^n$ *and* $x_1 \geq x_2 \geq .... \geq x_n$, $y_1 \geq y_2 \geq .... \geq y_n$. *We say* x *is majorized by* y, *i.e.* $x \prec y$, *if* $\sum_{i=1}^{k} x_i \leq \sum_{i=1}^{k} y_i$, *for* $k = 1, ..., n-1$ *and* $\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i$.

Definition 3. *A real valued function on* g *defined on set* $\mathcal{S} \subset \mathbb{R}^n$ *is said to be Schur concave on* $\mathcal{S}$ *if* $x \prec y \implies g(x) \geq g(y)$.

Lemma 16. [70] *If* $x, y \in \mathbb{R}_+^n$ *and* $x \prec y$, *then* $\prod_{i=1}^{n} x_i \geq \prod_{i=1}^{n} y_i$. *This means* $\prod x_i$ *is a Schur concave function.*

Lemma 17. [71] *Let* A, B *be positive semidefinite matrices of the same size and let all elements on diagonal of* B *are 1. Then* $\lambda(A \odot B) \prec \lambda(A)$.

Lemma 18. [67] *Let* A, B *be matrices of size* $\mathbb{R}^{n\times m}$ *then* $\operatorname{rank}(A \odot B) \leq \operatorname{rank}(A)\operatorname{rank}(B)$.

Lemma 19. [Arithmetic Mean-Geometric Mean Inequality [72]] *For every sequence of nonnegative real numbers* $a_1, a_2, ...a_n$ *one has*

$$(\prod_{i=1}^{n} a_i)^{1/n} \leq \frac{\sum_{i=1} a_i}{n}$$

with equality if and only if $a_1 = a_2 = ... = a_n$.

III.7.6 Proof of Theorem 7

Suppose the rank of $\tilde{K}_{T+1}$ is r. Hence only the first r eigenvalues are non zero. In that case $g([T])$ attains its maximum when each of these r eigenvalues is equal to $\dfrac{\text{trace}(\tilde{K}_{T+1})}{r}$ (using Lemma 26). Thus,

$$\begin{aligned} g([T]) &= \frac{\prod_{i=1}^{T+1}(\lambda_i + \lambda)}{\lambda^{T+1}} \\ &\leq \frac{\prod_{i=1}^{r}(\text{trace}(\tilde{K}_{T+1})/r + \lambda)}{\lambda^{r}} \\ &= \left(\frac{\text{trace}(\tilde{K}_{T+1})/r + \lambda}{\lambda}\right)^{r}. \end{aligned}$$

It follows that,

$$\begin{aligned} \log(g([T])) &\leq r \log\left(\frac{\text{trace}(\tilde{K}_{T+1})/r + \lambda}{\lambda}\right) \\ &\leq r \log\left(\frac{\text{trace}(\tilde{K}_{T+1}) + \lambda}{\lambda}\right) \\ &= r_z r_x \log\left(\frac{\text{trace}(\tilde{K}_{T+1}) + \lambda}{\lambda}\right) \\ &\leq r_z r_x \log\left(\frac{(T+1)c_{\tilde{k}} + \lambda}{\lambda}\right), \end{aligned}$$

where the second inequality is due to Lemma 18.

III.7.7 Proof of Theorem 8

Proof. Suppose the $\tilde{K}_{T+1}(\mu_1)$ and $\tilde{K}_{T+1}(\mu_2)$ are final kernel matrices after time T, $K_{Z^r_{T+1}}(\mu_1)$ and $K_{Z^r_{T+1}}(\mu_2)$ are corresponding matrices using the definition 3.17. Also suppose that $K_Z(\mu_1)$ and $K_Z(\mu_2)$ are task similarity matrices. The eigenvalues of $K_Z(\mu) = (1-\mu)I_N + \mu \mathbb{1}_N \mathbb{1}_N^T$ are

$1 + \mu(N-1)$ with multiplicity 1 and $1-\mu$ with multiplicity $N-1$.

Let n be positive integer with $n \leq N-1$ and define df to be the difference between sum of largest $n+1$ eigenvalues of $K_Z(\mu_1)$ and $K_Z(\mu_2)$. Thus,

$$\begin{aligned}
df &= 1+\mu_1(N-1)+n(1-\mu_1)-\Big(1+\mu_2(N-1)+n(1-\mu_2)\Big) \\
&= (N-1)(\mu_1-\mu_2)+n(1-\mu_1-1+\mu_2) \\
&= (N-1)(\mu_1-\mu_2)+n(\mu_2-\mu_1) \\
&= (\mu_1-\mu_2)(N-1-n) \\
&\leq 0
\end{aligned}$$

where the last inequality holds because $\mu_1 \leq \mu_2$. This implies

$$\lambda(K_Z(\mu_1)) \prec \lambda(K_Z(\mu_2))$$

and the Lemma 7 implies

$$\lambda(K^r_{Z_{T+1}}(\mu_1)) \prec \lambda(K^r_{Z_{T+1}}(\mu_2)).$$

Using the Lemma 17 and the definition 3.17, we have

$$\lambda(\tilde{K}_{T+1}(\mu_1)) \prec \lambda(\tilde{K}_{T+1}(\mu_2))$$

This implies

$$\lambda(\tilde{K}_{T+1}(\mu_1)) + \lambda \prec \lambda(\tilde{K}_{T+1}(\mu_2)) + \lambda.$$

Using the Lemma 16, we conclude that

$$\prod_{t=1}^{T+1}(\lambda_t(\tilde{K}_{T+1}(\mu_1)) + \lambda) \geq \prod_{t=1}^{T+1}(\lambda_t(\tilde{K}_{T+1}(\mu_2)) + \lambda).$$

This completes the proof. □

III.7.8 Proof of Corollary 1

Proof. Let's find the upper bound of maximum of $g([T])$. We know that $r\lambda \log T \geq \sum_{i=r+1}^{T+1} \lambda_i$. Let ϵ be a constant such that $r\lambda \log T = \sum_{i=r+1}^{T+1} \lambda_i + \epsilon$. Notice that $\epsilon \leq (T+1)c_{\tilde{k}}$. Consider

$$\begin{aligned} \max \quad & \prod_{i=1}^{T+1} \quad (\lambda_i + \lambda) \\ s.t. \quad & \sum_{i=1}^{r} \quad \lambda_i + \lambda = (T+1)c_{\tilde{k}} + r\lambda - r\lambda \log T + \epsilon \\ and \quad & \sum_{i=r+1}^{T+1} \quad \lambda_i + \lambda = r\lambda \log T - \epsilon + (T+1-r)\lambda \end{aligned}$$

Using Lemma 26, the maximum of above constrained optimization problem occurs at

$$\lambda_i + \lambda = \begin{cases} \dfrac{(T+1)c_{\tilde{k}} + r\lambda - r\lambda \log T + \epsilon}{r}, & \text{if } \lambda_i \leq r, \\ \dfrac{r\lambda \log T + (T+1-r)\lambda}{(T+1-r)} - \dfrac{\epsilon}{T+1-r} & \text{otherwise.} \end{cases} \tag{3.23}$$

Therefore,

$$
\begin{aligned}
g([T]) &= \prod_{t=1}^{T+1} \frac{(\lambda_t + \lambda)}{\lambda} \\
&\leq \left(\frac{(T+1)c_{\tilde{k}} + r\lambda - r\lambda \log T + \epsilon}{r\lambda}\right)^r \left(\frac{r\lambda \log T + (T+1-r)\lambda}{(T+1-r)\lambda}\right)^{T+1-r} \\
&= \left(\frac{(T+1)c_{\tilde{k}} + r\lambda - r\lambda \log T + \epsilon}{r\lambda}\right)^r \left(\frac{r \log T + (T+1-r)}{(T+1-r)}\right)^{T+1-r} \\
&= \left(\frac{(T+1)c_{\tilde{k}} + r\lambda - r\lambda \log T + \epsilon}{r\lambda}\right)^r \left(\frac{r \log T}{T+1-r} + 1\right)^{T+1-r} \\
&= \left(\frac{(T+1)c_{\tilde{k}} + r\lambda - r\lambda \log T + \epsilon}{r\lambda}\right)^r \left(\frac{r \log T}{T+1-r} + 1\right)^{T+1-r} \\
&\leq \left(\frac{(T+1)c_{\tilde{k}} + r\lambda - r\lambda \log T + \epsilon}{r\lambda}\right)^r \left(\frac{r \log(T+r-1)}{T} + 1\right)^{T} \\
&\leq \left(\frac{(T+1)c_{\tilde{k}} + r\lambda - r\lambda \log T + \epsilon}{r\lambda}\right)^r \exp\Big(r \log(T+r-1)\Big)
\end{aligned}
$$

where the first inequality is due to eqn. (3.23), the second inequality holds because $(1+\frac{\log(x)}{x})^x$ is monotonically increasing function $\forall x \geq 1$ and the last inequality holds because $\log(1+x) \leq x, \forall x > -1$.

Taking log on both sides

$$
\begin{aligned}
\log(g([T])) &\leq r \log\left(\frac{(T+1)c_{\tilde{k}} + r\lambda - r\lambda \log T + \epsilon}{r\lambda}\right) + r \log(T+r-1) \\
&\leq r \log\left(\frac{(T+1)c_{\tilde{k}} + r\lambda - r\lambda \log T + \epsilon}{r\lambda}\right) + r \log(2T) \\
\log(g([T])) &\leq r \log\left(2T \frac{2(T+1)c_{\tilde{k}} + r\lambda - r\lambda \log T}{r\lambda}\right).
\end{aligned}
$$

□

CHAPTER IV

Simple Regret for Contextual Bandits

There are two variants of the classical multi-armed bandit (MAB) problem that have received considerable attention from machine learning researchers in recent years: contextual bandits and simple regret minimization. Contextual bandits are a sub-class of MABs where, at every time step, the learner has access to side information that is predictive of the best arm. Simple regret minimization assumes that the learner only incurs regret after a pure exploration phase. In this work, we study simple regret minimization for contextual bandits. Motivated by applications where the learner has separate training and autonomous modes, we assume that, the learner experiences a *pure exploration* phase, where feedback is received after every action but no regret is incurred, followed by a *pure exploitation* phase in which regret is incurred but there is no feedback. We present the Contextual-Gap algorithm and establish performance guarantees on the simple regret, i.e., the regret during the pure exploitation phase. Our experiments examine a novel application to adaptive sensor selection for magnetic field estimation in interplanetary spacecraft, and demonstrate considerable improvement over algorithms designed to minimize the cumulative regret.

IV.1 Introduction

The multi-armed bandit (MAB) is a framework for sequential decision making where, at every time step, the learner selects (or "pulls") one of several possible actions (or "arms"), and receives a reward based on the selected action. The regret of the learner is the difference between the maximum possible reward and the reward resulting from the chosen action. In the classical MAB setting, the goal is to minimize the sum of all regrets, or *cumulative regret*, which naturally leads to an exploration/exploitation trade-off problem [30]. If the learner explores too little, it may never find an optimal arm which will increase its cumulative regret. If the learner explores too much, it may select sub-optimal arms too often which will also increase its cumulative regret. There are a variety of algorithms that solve this exploration/exploitation trade-off problem [30, 16, 73, 17, 12].

The contextual bandit problem extends the classical MAB setting, with the addition of time-varying side information, or *context*, made available at every time step. The best arm at every time step depends on the context, and intuitively the learner seeks to determine the best arm as a function of context. To date, work on contextual bandits has studied cumulative regret minimization, which is motivated by applications in health care, web advertisement recommendations and news article recommendations [23]. The contextual bandit setting is also called associative reinforcement learning [16] and linear bandits [17, 18].

In classical (non-contextual) MABs, the goal of the learner isn't always to minimize the cumulative regret. In some applications, there is a *pure exploration* phase during which the learning incurs no regret (i.e., no penalty for sub-optimal decisions), and performance is measured in terms of *simple regret*, which is the regret assessed at the end of the pure exploration phase. For example, in top-arm identification, the learner must guess the arm with highest expected reward at the end of the exploration phase. Simple regret minimization clearly motivates different strategies, since there is no penalty for sub-optimal decisions during the exploration phase. Fixed budget and fixed confidence are the two main theoretical frameworks in which simple regret is generally analyzed [26, 27, 28, 29].

In this chapter, we extend the idea of simple regret minimization to contextual bandits. In this setting, there is a pure exploration phase during which no regret is incurred, following by a *pure exploitation* phase during which regret is incurred, but there is no feedback so the learner cannot update its policy. To our knowledge, previous work has not addressed novel algorithms for this setting. [74] provide simple regret guarantees for the policy of uniform sampling of arms in the i.i.d setting. The contextual bandit algorithm of [75] also has distinct exploration and exploitation phases, but unlike our setting, the agent has control over which phase it is in, i.e., when it wants to receive feedback. In the work of [76, 77, 78, 79] there is a single best arm even when contexts are observed (directly or indirectly). Our algorithm, Contextual-Gap, generalizes the idea of [26] and [76] to the contextual bandits setting.

We make the following contributions: 1. We formulate a novel problem: that of simple regret minimization for contextual bandits. 2. We develop an algorithm, Contextual-Gap, for this setting. 3. We present performance guarantees on the simple regret in the fixed budget framework. 4. We present experimental results for adaptive sensor selection in nano-satellites.

The chapter is organized as follows. In section 2, we motivate the new problem based on the real-life application of magnetometer selection in spacecraft. In section 3, we state the problem formally, and to solve this new problem, we present the Contextual-Gap algorithm in section 4. In section 5, we present the learning theoretic analysis and in section 6, we present and discuss experimental results. Section 7 concludes.

IV.2 Motivation

Our work is motivated by autonomous systems that go through an initial training phase (the pure exploration phase) where they learn how to accomplish a task without being penalized for sub-optimal decisions, and then are deployed in an environment where they no longer receive feedback, but regret is incurred (the pure exploitation phase).

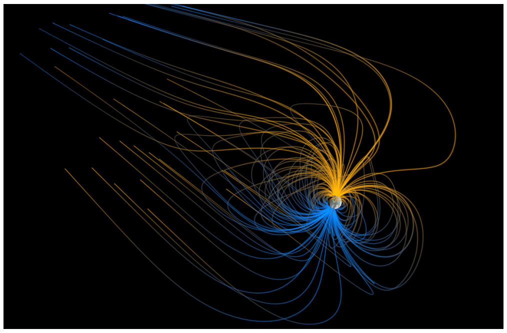


Figure 4.1: Scientific measurement: magnetic field lines of the Earth (Credit: NASA/Goddard Scientific Visualization Studio)

An example scenario arises in the problem of estimating weak interplanetary magnetic fields (Figure 4.1) in the presence of noise using resource-constrained spacecraft known as nano-satellites or CubeSats. Spacecraft systems generate their own spatially localized magnetic field noise due to large numbers of time-varying current paths in the spacecraft. Historically, with large spacecraft, such noise was minimized by physically separating the sensor from the spacecraft using a rigid boom. In highly resource-constrained satellites such as nano-satellites, however, structural constraints limit the use of long rigid booms, requiring sensors to be close to or inside the CubeSat (Figure 4.2). Thus, recent work has focused on nano-satellites equipped with multiple magnetic field sensors (magnetometers) [80].

A natural problem that arises in nano-satellites with multiple sensors is that of determining the sensor with the reading closest to the true magnetic field. At each time step, whenever sensor is selected, one has to calibrate the sensor readings because sensor behaviours change due to rapid changes in temperature and movement (rotations or maneuvers) of the satellite which introduce a further errors in magnetic field readings. This calibration process is expensive in terms of computation and memory [81, 82], particularly when dealing with many magnetic field sensors. To get accurate readings from different sensors one has to repeat this calibration process for every sensor and it's not feasible because of the limited power resources on the spacecraft. These constraints motivate the selection of a single sensor at each time step.

Furthermore, the best sensor changes with time. This stems from the time-varying localization of noise in the spacecraft, which in turn results from different operational events such as data transmission, spacecraft maneuvers, and power generation. This dynamic sensor selection problem is readily cast as a contextual bandit problem. The context is given by the spacecraft's telemetry system which provides real-time measurements related to spacecraft operation, including solar panel currents, temperatures, momentum wheel information, and real-time current consumption [83].

In this application, however, conventional contextual bandit algorithms are not applicable because feedback is not always available. Feedback requires knowledge of sensor noise, which in turn requires knowledge of the true magnetic field. Yet the true magnetic field is known only during certain portions of a spacecraft's orbit (e.g., when the satellite is near other spacecraft, or when the earth shields the satellite from sun-induced magnetic fields). Moreover, when the true magnetic field is known, there is no need to estimate the magnetic field in the first place! This suggests a learning scenario where the agent (the sensor scheduler) operates in two phases, one where it has feedback but incurs no regrets (because the field being estimated is known), and another where it does not receive feedback, but nonetheless needs to produce estimates. This is precisely the problem we study.

In the magnetometer problem defined above, the exploration and exploitation times occur in phases, as the satellite moves into and out of regions where the true magnetic field is known.

For simplicity, we will the address the problem in which the first T time steps belong to the exploration phase, and all subsequent time steps to the exploitation phase. Nonetheless, the algorithm we introduce can switch between phases indefinitely, and does not need to know in advance when a new phase is beginning.

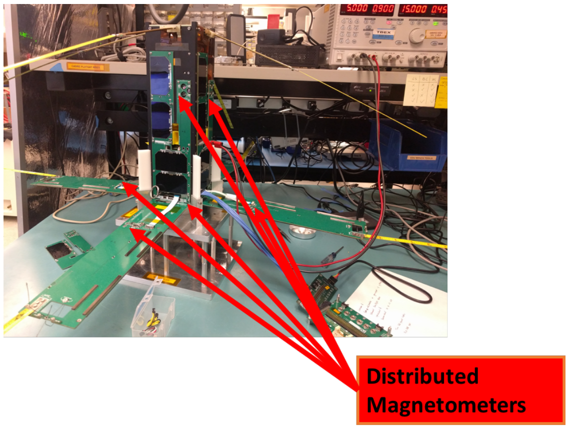


Figure 4.2: TBEx Small Satellite with Multiple Magnetometers [1, 2]

Sensor management, adaptive sensing, and sequential resource allocation have historically been viewed in the decision process framework where the learner takes actions on selecting the sensor based on previously collected data. There have been many proposed solutions based on Markov decision processes (MDPs) and partially observable MDPs, with optimality bounds for cumulative regret [84, 85, 86, 87, 88]. In fact, sensor management and sequential resource allocation was one of the original motivating settings for the classical MAB problem [89, 12, 84], again with the goal of cumulative regret minimization. We are interested in an adaptive sensing setting where the optimal decisions and rewards also depend on the context, but where the actions can be separated into a pure exploration and pure exploitation phases, with no regret during exploration, and with no feedback during pure exploitation.

IV.3 Formal Setting

We denote the context space as $\mathcal{X} = \mathbb{R}^d$. Let $\{x_t\}_{t=1}^{\infty}$ denote the sequence of observed contexts. Let the total number of arms be A. For each x_t, the learner is required to choose an arm $a \in [A]$, where $[A] := \{1, 2, ..., A\}$.

For arm $a \in [A]$, let $f_a : \mathcal{X} \to \mathbb{R}$ be a function that maps context to expected reward when arm a is selected. Let a_t denote the arm selected at time t, and assume the reward at time t obeys $r_t := f_{a_t}(x_t) + \zeta_t$, where ζ_t is noise (described in more detail below). We assume that for each a, f_a belongs to a reproducing kernel Hilbert space (RKHS) defined on $\mathcal{X}$. The first T time steps belong to the *exploration phase* where the learner observes context x_t, chooses arm a_t and obtains reward r_t. The time steps after T belong to an *exploitation phase* where the learner observes context x_t, chooses arm a_t and earns an implicit reward r_t that is not returned to the learner.

For the theoretical results below, the following general probabilistic framework is adopted, following [18] and [90]. We assume that ζ_t is a zero mean, ρ-conditionally sub-Gaussian random variable, i.e., ζ_t is such that for some $\rho > 0$ and $\forall \gamma \in \mathbb{R}$,

$$\mathbb{E}[e^{\gamma \zeta_t} | \mathcal{H}_{t-1}] \leq \exp\left(\frac{\gamma^2 \rho^2}{2}\right). \tag{4.1}$$

Here $\mathcal{H}_{t-1} = \{x_1, \ldots, x_{t-1}, \zeta_1, \ldots, \zeta_{t-1}\}$ is the history at time t (see Section IV.8 for additional details).

We also define the following terms. Let $D_{a,t}$ be the set of all time indices when arm a was selected up to time $t-1$ and set $N_{a,t} = |D_{a,t}|$. Let $X_{a,t}$ be the data matrix whose columns are $\{x_\tau\}_{\tau \in D_{a,t}}$ and similarly let $Y_{a,t}$ denote the column vector of rewards $\{r_\tau\}_{\tau \in D_{a,t}}$. Thus, $X_{a,t} \in \mathbb{R}^{d \times N_{a,t}}$ and $Y_{a,t} \in \mathbb{R}^{N_{a,t}}$.

IV.3.1 Problem Statement

At every time step t, the learner observes context x_t. During the exploration phase $t \leq T$, the learner chooses a series of actions to explore and learn the mappings f_a from context to reward. During the exploitation phase $t > T$, the goal is to select the best arm as a function of context. We define the *simple regret* associated with choosing arm $a \in [A]$, given context x, as:

$$R_a(x) := f^*(x) - f_a(x), \tag{4.2}$$

where $f^*(x) := \max_{i \in [A]} f_i(x)$ is the expected reward for the best arm for context x. The learner aims to minimize the simple regret for $t > T$. To be more precise, let Ω be the fixed policy mapping context to arm during the exploitation phase. The goal is to determine policies for the exploration and exploitation phases such that for all $\epsilon > 0$ and $t > T$

$$\mathbb{P}(R_{\Omega(x_t)}(x_t) \geq \epsilon | x_t) \leq b_\epsilon(T),$$

where $b_\epsilon(T)$ is an expression that decreases to 0 as $T \to \infty$.

The following section presents an algorithm to solve this problem.

IV.4 Algorithm

We propose an algorithm that extends the Bayes Gap algorithm [76] to the contextual setting. Note that Bayes Gap itself is originally motivated from UGapEb [26].

IV.4.1 Estimating Expected Rewards

A key ingredient of our extension is an estimate of f_a, for each a, based on the current history. We use kernel methods to estimate f_a. Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a symmetric positive definite kernel function on $\mathcal{X}$, $\mathcal{H}$ be the corresponding RKHS and $\phi(x) = k(\cdot, x)$ be the associated canonical feature map. Let $\phi(X_{a,t}) := [\phi(x_j)]_{j \in D_{a,t}}$. We define the kernel matrix associated with $X_{a,t}$ as $K_{a,t} := \phi(X_{a,t})^T \phi(X_{a,t}) \in \mathbb{R}^{N_{a,t} \times N_{a,t}}$ and the kernel vector of context x as $k_{a,t}(x) := \phi(X_{a,t})^T \phi(x)$. Let $I_{a,t}$ be the identity matrix of size $N_{a,t}$. We estimate f_a at time t, via kernel ridge regression, i.e.,

$$\hat{f}_{a,t}(x) = \arg\min_{f_a \in \mathcal{H}} \sum_{j \in D_{a,t}} (f_a(x_j) - r_j)^2 + \lambda \|f_a\|^2.$$

The solution to this optimization problem is $\hat{f}_{a,t}(x) = k_{a,t}(x)^T (K_{a,t} + \lambda I_{a,t})^{-1} Y_{a,t}$. Furthermore, [90] establish a confidence interval for $f_a(x)$ in terms of $\hat{f}_{a,t}(x)$ and the "variance" $\hat{\sigma}^2_{a,t}(x) := k(x,x) - k_{a,t}(x)^T (K_{a,t} + \lambda I_{a,t})^{-1} k_{a,t}(x)$.

Theorem 10 (Restatement of Theorem 2.1 in [90])**.** *Consider the contextual bandit scenario described in section IV.3. For any $\beta > 0$, with probability at least $1 - e^{-\beta^2}$, it holds simultaneously over all $x \in \mathcal{X}$ and all $t \leq T$,*

$$|f_a(x) - \hat{f}_{a,t}(x)| \leq (C_1 \beta + C_2) \frac{\hat{\sigma}_{a,t}(x)}{\sqrt{\lambda}}, \tag{4.3}$$

where $C_1 = \rho\sqrt{2}$ and $C_2 = \rho \sqrt{\sum_{\tau=2}^{T} \ln(1 + \frac{1}{\lambda} \hat{\sigma}_{a,\tau-1}(x_\tau))} + \sqrt{\lambda} \|f_a\|_{\mathcal{H}}$.

In the Section IV.8, we show that $C_2 = O(\rho\sqrt{\ln T})$. For convenience, we denote the width of the confidence interval $s_{a,t}(x) := 2(C_1\beta + C_2)\frac{\hat{\sigma}_{a,t}(x)}{\sqrt{\lambda}}$. Thus, the upper and lower confidence bounds of $f_a(x)$ are $U_{a,t}(x) := \hat{f}_{a,t}(x) + \frac{s_{a,t}(x)}{2}$ and $L_{a,t}(x) := \hat{f}_{a,t}(x) - \frac{s_{a,t}(x)}{2}$. The upper confidence bound is the most optimistic estimate of the reward and the lower confidence bound is the most pessimistic estimate of the reward.

IV.4.2 Contextual-Gap Algorithm

During the exploration phase, the Contextual-Gap algorithm proceeds as follows. First, the algorithm has a burn-in period where it cycles through the arms (ignoring context) and pulls each one N_λ times. Following this burn-in phase, when the algorithm is presented with context x at time $t \leq T$, the algorithm identifies two candidate arms, $J_t(x)$ and $j_t(x)$, as follows. For each arm a the *contextual gap* is defined as $B_{a,t}(x) := \max_{i \neq a} U_{i,t}(x) - L_{a,t}(x)$. $J_t(x)$ is the arm that *minimizes* $B_{a,t}(x)$ and $j_t(x)$ is the arm (excluding $J_t(x)$) whose upper confidence bound is maximized. Among these two candidates, the one with the widest confidence interval is selected. Note that quantity $B_{a,t}(x)$ upper bounds the the simple regret for corresponding arm a and is the basis of definition of the best arm $J_t(x)$. We use $j_t(x) = \arg\max_{a \neq J_t(x)} U_{a,t}(x)$ as the second candidate because optimistically $j_t(x)$ has a chance to be the best arm and it may give more information about how bad the choice of $J_t(x)$ could be.

In the exploitation phase, for a given context x, the contextual gap for all time steps in the exploration phase are evaluated. The arm with the smallest gap over the entire exploration phase for the given context x is chosen as the best arm associated with context x. Because there is no feedback during the exploitation phase, the algorithm moves to the next exploitation step without modification to the learning history. The exact description is presented in Algorithm 4.1.

During the exploitation phase, looking back at all history may be computationally prohibitive. Thus, in practice, we just select the best arm as $J_T(x_t), \forall t > T$. As described in the experimental section, this works well in practice. Theoretically, N_λ has to be bigger than a certain number defined in Lemma 21, but for experimental results we keep $N_\lambda = 1$.

IV.4.3 Comparison of Contextual-Gap and Kernel-UCB

In this section, we illustrate the difference between the policies of Kernel-UCB (which minimizes cumulative regret) and exploration phase of Contextual-Gap (which aims to minimize simple regret). At each time step, Contextual-Gap selects one of two arms: $J_t(x)$, the arm with highest pessimistic reward estimate, or $j_t(x)$, the arm excluding $J_t(x)$ with highest optimistic reward estimate. Kernel-UCB, in contrast, selects the arm with the highest optimistic reward estimate (i.e., with the maximum upper confidence bound).

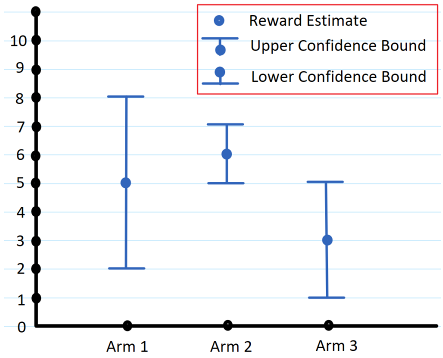


Figure 4.3: Contextual-Gap exploration policy: case 1

Consider a three arm scenario at some time τ with context x_τ. Suppose that the estimated rewards and confidence intervals are as in Figures 4.3 and 4.4, reflecting two different cases.

- Case 1 (Figure 4.3): In this case, Kernel-UCB would pick arm 1, because it has the maximum upper confidence bound. Kernel-UCB's policy is designed to be optimistic in the case of uncertainty. In the Contextual-Gap, we first calculate $J_\tau(x_\tau)$ which minimizes $B_{a,\tau}(x_\tau)$. Note that $B_{1,\tau}(x_\tau) = U_{2,\tau}(x_\tau) - L_{1,\tau}(x_\tau) = 7 - 2 = 5$, $B_{2,\tau}(x_\tau) = 3$ and $B_{3,\tau}(x_\tau) = 7$. In this case, $J_\tau(x_\tau) = 2$ and hence $j_\tau(x_\tau) = 1$. Finally, Contextual-Gap would choose among arm 1 and arm 2, and would finally choose arm 1 because it has the largest confidence interval. Hence, in case 1, Contextual-Gap chooses the same arm as that of Kernel-UCB.

- Case 2 (Figure 4.4): In this case, Kernel-UCB would pick arm 1. Note that $B_{1,\tau}(x_\tau) =$

$U_{2,\tau}(x_\tau) - L_{1,\tau}(x_\tau) = 7 - 4 = 3$, $B_{2,\tau}(x_\tau) = 7$ and $B_{3,\tau}(x_\tau) = 4$. Then $J_\tau(x_\tau) = 1$ and hence $j_\tau(x_\tau) = 2$. Finally, Contextual-Gap chooses arm 2, because it has the widest confidence interval. Hence, in case 2, Contextual-Gap chooses a different arm compared to that of Kernel-UCB.

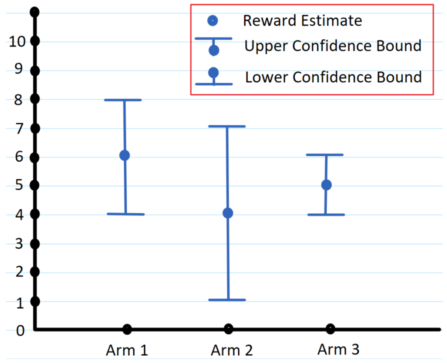


Figure 4.4: Contextual-Gap exploration policy: case 2

Clearly, the use of the lower confidence bound along with upper confidence bound allows Contextual-Gap to explore more than kernel-UCB. However, Contextual-Gap doesn't explore just any arm, but rather it explores only among arms with some likelihood of being optimal. The following section details high probability bounds on the simple regret of the Contextual-Gap algorithm.

Algorithm 4.1: Contextual-Gap

```
Input: Number of arms $A$, Time Steps $T$, parameter $\beta$, regularization parameter $\lambda$,
       burn-in phase constant $N_\lambda$.
// Exploration Phase I: Burn-in Period //
for $t = 1, ..., AN_\lambda$ do
    Observe context $x_t$ and choose $a_t = t \mod A$
    Receive reward $r_t \in \mathbb{R}$
end
//Exploration Phase II: Contextual-Gap Policy //
for $t = AN_\lambda + 1, \ldots, T$ do
    Observe context $x_t$
    Learn reward estimators $\hat{f}_{a,t}(x_t)$ and confidence interval $s_{a,t}(x_t)$ based on history
    $U_{a,t}(x_t) = \hat{f}_{a,t}(x_t) + \frac{s_{a,t}(x_t)}{2}$, $L_{a,t}(x_t) = \hat{f}_{a,t}(x_t) - \frac{s_{a,t}(x_t)}{2}$
    $B_{a,t}(x_t) = \max_{i \neq a} U_{i,t}(x_t) - L_{a,t}(x_t)$ , $J_t(x_t) = \arg\min_a B_{a,t}(x_t)$, $j_t(x_t) = \arg\max_{a \neq J_t(x_t)} U_{a,t}(x_t)$
    Choose $a_t = \arg\max_{a \in \{j_t(x_t), J_t(x_t)\}} s_{a,t}(x_t)$
    Receive reward $r_t \in \mathbb{R}$
end
// Exploitation Phase //
for $t > T$ do
    Observe context $x_t$.
    for $\tau = AN_\lambda + 1, \ldots, T$ do
        Evaluate and collect $J_\tau(x_t), B_{J_\tau(x_t)}(x_t)$
    end
    $\iota = \arg\min_{AN_\lambda + 1 \leq \tau \leq T} B_{J_\tau(x_t),t}(x_t)$
    Choose $\Omega(x_t) = J_\iota(x_t)$.
end
```

IV.5 Learning Theoretic Analysis

We now analyze high probability simple regret bounds which depend on the gap quantity $\Delta_a(x) := |\max_{i \neq a} f_i(x) - f_a(x)|$. The bounds are presented in the non-i.i.d setting described in Section IV.3. For the confidence interval to be useful, it needs to shrink to zero with high probability over the feature space as each arm is pulled more and more. This requires the smallest non-zero eigenvalue of the sample covariance matrix of the data for each arm to be lower bounded by a certain value. We make an assumption that allows for such a lower bound, and use it to prove that the confidence intervals shrink with high probability under certain assumptions. Finally, we bound the simple regret using the result of shrinking confidence interval, the gap quantity, and the special exploration strategy described in Algorithm 4.1. We now make additional assumptions to the problem setting.

A I $\{\mathcal{X}_t\}_{t \geq 1} \subset \mathbb{R}^d$, is a random process on compact space endowed with a finite positive Borel measure.

A II Kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is bounded by a constant L, the canonical feature map $\phi : \mathcal{X} \to \mathcal{H}$ of k is a continuous function, and $\mathcal{H}$ is separable.

We denote $\mathbb{E}_{t-1}[\cdot] := \mathbb{E}[\cdot | x_1, x_2, \ldots, x_{t-1}]$ and by $\lambda_r(A)$ the r^{th} largest eigenvalue of a compact self adjoint operator A. For a context x, the operator $\phi(x)\phi(x)^T : \mathcal{H} \to \mathcal{H}$ is a compact self-adjoint operator. Based on this notation, we make the following assumption:

A III There exists a subspace of dimension d^* with projection P, and a constant $\lambda_x > 0$, such that $\forall t$, $\lambda_r(P^T \mathbb{E}_{t-1}[\phi(x_t)\phi(x_t)^T]P) > \lambda_x$ for $r \leq d^*$ and $\lambda_r((I-P)^T \mathbb{E}_{t-1}[\phi(x_t)\phi(x_t)^T](I-P)) = 0, \forall r > d^*$.

Assumption **A III** facilitates the generalization of Bayes gap [76] to the kernel setting with non-i.i.d, time varying contexts. It allows us to lower bound, with high probability, the

r^{th} eigenvalue of the cumulative second moment operator $S_t := \sum_{s=1}^{t} \phi(x_s)\phi(x_s)^T$ so that it is possible to learn the reward behavior in the low energy directions of the context at the same rate as the high energy ones with high probability.

We now provide a lower bound on the r^{th} eigenvalue of a compact self-adjoint operator. There are similar results in the setting where reward is a linear function of context, including Lemma 2 in [91] and Lemma 7 in [92] which provides lowest eigenvalue bounds with the assumption of linear reward and full rank covariance, and Theorem 2.2 in [93] which assumes more structure to the contexts generated. We extend these results to the setting of a compact self-adjoint operator scenario with data occupying a finite dimensional subspace. Let $W_t := \sum_{s=1}^{t} \mathbb{E}_{s-1}[(\phi(x_s)\phi(x_s)^T)^2] - (\mathbb{E}_{s-1}[\phi(x_s)\phi(x_s)^T])^2$. By construction and Assumption **A III** we can show that W_t has d^* non-zero eigenvalues (See Section IV.8).

Lemma 20 (Lower bound on r^{th} Eigen-value of compact self-adjoint operators)**.** *Let $x_t \in \mathcal{X}$, $t \geq 1$ be generated sequentially from a random process. Assume that conditions **A I-A III** hold. Let $p(t) = \min(-t, 1)$ and $\forall b \geq 0, a > \frac{1}{6}(L^2 + \sqrt{L^4 + 36b})$ let $\tilde{d} := 50 \sum_{r=1}^{d^*} p(-\frac{a\lambda_r(\mathbb{E}W_t)}{L^2 b}) \leq 50d^*$. Let*

$$A(t, \delta) = \log \frac{(tL^4 + 1)(tL^4 + 3)\tilde{d}}{\delta},$$

and

$$h(t, \delta) = \Big(t\lambda_x - \frac{L^2}{3}\sqrt{18tA(t, \delta) + A(t, \delta)^2}) - \frac{L^2}{3}A(t, \delta)\Big).$$

Then for any $\delta > 0$,

$$\lambda_r(S_t) \geq h(t, \delta)_+$$

holds for all $t > 0$ with probability at least $1 - \delta$. Furthermore, if L =1, $r \leq d^*$ and $0 < \delta \leq \frac{1}{8}$, then the event

$$\lambda_r(S_t) \geq \frac{t\lambda_x}{2}, \forall t \geq \frac{256}{\lambda_x^2} \log(\frac{128\tilde{d}}{\lambda_x^2 \delta}),$$

holds with probability at least $1 - \delta$.

Lemma 20 provides high probability lower bounds on the minimum nonzero eigenvalue of the cumulative second moment operator S_t. Using the preceding lemma and the confidence interval defined in Theorem 10, it is possible to provide high probability monotonic bounds on the confidence interval widths $s_{a,t}(x)$.

Lemma 21 (Monotonic upper bound of $s_{a,t}(x_t)$)**.** *Consider a contextual bandit simple regret minimization problem with assumptions **A I-A III** and fix T. Assume $\|\phi(x)\| \leq 1$, $\lambda > 0$ and $\forall a \in [A]$, $N_{a,t} > N_\lambda := \max\Big(\frac{2(1-\lambda)}{\lambda_x}, d^*, \frac{256}{\lambda_x^2}\log(\frac{128\tilde{d}}{\lambda_x^2\delta})\Big)$. Then, for any $0 < \delta \leq \frac{1}{8}$,*

$$s_{a,t}(x_t)^2 \leq g_{a,t}(N_{a,t})$$

with probability at least $1-\delta$, for the monotonically decreasing function $g_{a,t}$ defined as $g_{a,t}(N_{a,t}) := 8(C_1\beta + C_2)^2\Big(\frac{1}{\lambda + N_{a,t}\lambda_x/2}\Big)$.

The condition $N_{a,t} > N_\lambda$ results in a minimum number of tries that arm a has to be selected before any bound will hold. In $N_\lambda := \max\Big(\frac{2(1-\lambda)}{\lambda_x}, d^*, \frac{256}{\lambda_x^2}\log(\frac{128\tilde{d}}{\lambda_x^2\delta})\Big)$, the first and third term in the max are needed so that we can give concentration bounds on eigenvalues and prove that the confidence width shrinks. The second term is needed because one has to get at least d^* contexts for every arm so that at least some energy is added to the lowest eigenvalues.

These high probability monotonic upper bounds on the confidence estimate can be used to upper bound the simple regret. The upper bound depends on a context-based hardness quantity defined for each arm a (similar to [76]) as

$$H_{a,\epsilon}(x) = \max(\frac{1}{2}(\Delta_a(x) + \epsilon), \epsilon). \tag{4.4}$$

Denote its lowest value as $H_{a,\epsilon} := \inf_{x\in\mathcal{X}} H_{a,\epsilon}(x)$. Let total hardness be defined as $H_\epsilon := \sum_{a\in[A]} H_{a,\epsilon}^{-2}$

(Note that $H_\epsilon \le \frac{A}{\epsilon^2}$). The recommended arm after time $t \ge T$ is defined as

$$\Omega(x) = J_{\arg\min_{AN_\lambda+1 \le \tau \le T} B_{J_\tau(x_t),t}(x_t)}(x_t)$$

from Algorithm 4.1. We now upper bound the simple regret as follows:

Theorem 11. *Consider a contextual bandit problem as defined in Section IV.3 with assumptions* ***A I-A III****. For* $0 < \delta \le \frac{1}{8}$, $\epsilon > 0$ *and* $N_\lambda := \max\left(\frac{2(1-\lambda)}{\lambda_x}, d^*, \frac{256}{\lambda_x^2}\log(\frac{128\tilde{d}}{\lambda_x^2\delta})\right)$, *let*

$$\beta = \sqrt{\frac{\lambda_x(T - N_\lambda(A-1)) + 2A\lambda}{16C_1^2 H_\epsilon}} - \frac{C_2}{C_1}. \tag{4.5}$$

For all $t > T$ *and* $\epsilon > 0$,

$$\mathbb{P}(R_{\Omega(x_t)}(x_t) < \epsilon | x_t) \ge 1 - A(T - AN_\lambda)e^{-\beta^2} - A\delta. \tag{4.6}$$

Note that the term C_2 in (4.5) grows logarithmically in T (see Section IV.8). For β to be positive, T should be greater than $\frac{16H_\epsilon C_2^2 - 2A\lambda}{\lambda_x} + N_\lambda(A-1)$. We compare the term $e^{-\beta^2}$ in our bound with the uniform sampling technique in [74] which leads to a bound that decay like $Ce^{-cT^{\frac{2}{(d_1+d)}}} \ge Ce^{-cT^{\frac{2}{(2+d)}}}$, where $d_1 \ge 2$, d is the context dimension, and C and c are constants. In our case, the decay rate has the form $C'Te^{-c'T}$ for constants C', c'. Clearly, our bound is superior for $\forall d \ge 1$. We can also compare Theorem 11 with Bayes Gap [76] and UGapEb [26] which provide simple regret guarantees in the multi-armed bandit setting. Bayes Gap and UGapEb have regret bounds of order $O(ATe^{-\frac{T-A}{H_\epsilon}})$ and we provide bounds of order $O(A(T - AN_\lambda)e^{-\frac{T-AN_\lambda}{H_\epsilon}})$. Ignoring other constants, our method has the additional term N_λ which is required because algorithm needs to see enough number of contexts to get information about context space and to become confident in the reward estimates in that context space. The simple regret bound is also dependent on the gap between the arms. A larger gap quantity Δ_a implies a larger $H_{a,\epsilon}$ which implies that quantity $e^{-\frac{1}{H_\epsilon}}$ is small. This means that a larger gap quantity leads to a faster rate.

Note that there are two choices for simple regret analysis: 1) bounding the simple regret uniformly (Theorem 11) and 2) average simple regret $\left(\sum_{t>T} R_{\Omega(x_t)}(x_t)\right)$. We bound the simple regret uniformly and it may require stronger distributional assumptions (e.g. Assumption **A III**) compared to average simple regret. Furthermore, we provide uniform bounds and not average simple regret bounds since our problem setting of simple regret minimization and the motivating application require performance guarantees for every time step during exploitation, as opposed to average simple regret guarantees.

IV.6 Experimental Results and Discussion

We present results from two different experimental setups, first is synthetic data, and second from a lab generated non-i.i.d. spacecraft magnetic field as described in the motivation Section. Cross validation was performed to minimize *average simple regret* for the exploitation phase while training with the exploration phase, both from the cross validation dataset. The value of T selected in both the cross validation and evaluation datasets were of similar magnitude. Evaluation of the algorithm for average simple regret behavior is performed with the evaluation dataset.

We present average simple regret comparisons of the Contextual-Gap algorithm against four baselines:

1. Uniform Sampling: We equally divide the exploration budget T among arms and learn a reward estimating function $f_a : \mathcal{X} \rightarrow \mathbb{R}$ for each of the arm during the exploration phase. During the exploitation phase, we select the best arm based on estimated reward function f_a.

2. Epsilon Greedy: At every step, we select the best arm (according to estimated f_a) with probability $1 - \epsilon_t$ and other arms with probability ϵ_t. We use $\epsilon_t = 0.99^t$, where t is the time step.

3. Kernel-UCB: We implement kernel-UCB from [24] for both exploration and exploitation.

4. Kernel-UCB-Mod: We implement kernel-UCB from [24] for exploration but use best arm based on estimated reward function f_a for exploitation.

5. Kernel-TS: We use kernelized version of Thompson Sampling from [94].

For all the algorithms, we use the Gaussian kernel. The algorithm was implemented with the best arm chosen with a history of one i.e., $\Omega(x_t) = J_T(x_t)$. For speed and scalability in implementation, the kernel inverse for arm a, $(K_{a,t} + \lambda I_{a,t})^{-1}$ and the kernel vector $k_{a,t}(x)$ updates were implemented as rank one updates. To tune kernel bandwidth and regularization parameters, we use following procedure: The dataset was split into two parts for hold-out (HO) and evaluation (EV). Each part was further split into two phases: exploration and exploitation. The value of T selected in both the hold-out and evaluation datasets were of similar magnitude. On the hold-out dataset, a grid search was used to set the tuning parameters by optimizing the average simple regret of the exploitation phase. The tuned parameters were used with the evaluation datasets to generate the plots. The code is available online to reproduce all results [3]. As our implementation performs rank one updates of the kernel matrix and its inverse, our algorithm has $O(T^2)$ as both computational and memory complexity in the worst case scenario, where T is the length of the exploration phase.

The exploration parameter $\alpha := C_1\beta + C_2$ is set to 1 for the results in this section and we show results for different values of α in Sharma et. al. (2018) [95] and [96].

The algorithm was implemented with the best arm chosen with a history of one i.e.,

[3] The code to reproduce our results is available at https://github.com/aniketde/ContextualGap

$\Omega(x_t) = J_T(x_t)$. For speed and scalability in implementation, the kernel inverse for arm a, $(K_{a,t} + \lambda I_{N_{a,t}})^{-1}$ and the kernel vector $k_{a,t}(x)$ updates were implemented as rank one updates. To tune kernel bandwidth and regularization parameters, we use following procedure: The dataset was split into two parts for hold-out (HO) and evaluation (EV). Each part was further split into two phases: exploration and exploitation. The value of T selected in both the hold-out and evaluation datasets were of similar magnitude. On the hold-out dataset, a grid search was used to set the tuning parameters by optimizing the average simple regret of the exploitation phase. The tuned parameters were used with the evaluation datasets to generate the plots. The code is available online to reproduce all results . As our implementation performs rank one updates of the kernel matrix and its inverse, our algorithm has $O(T^2)$ as both computational and memory complexity in the worst case scenario, where T is the length of the exploration phase.

IV.6.1 Synthetic Dataset

We present results of contextual simple regret minimization for synthetic dataset. At every time step, we observe a one dimensional feature vector $x_t \sim \mathcal{U}[0, 2\pi]$, where $\mathcal{U}$ is a uniform distribution. There are 20 arms and reward for each arm a is $r_{a,t} = \sin(a * x_t)$, where $a = [1, 2..., 20]$. The arm with the highest reward at time t is the best arm. At every time step, we only observe the reward for the arm that the algorithm selects.

Since the dataset is i.i.d. in nature, multiple simple regret evaluations are performed by shuffling the evaluation dataset, and the average curves are reported. Note that the algorithms have been cross validated for simple regret minimization. The plots are generated by varying the length of the exploration phase and keeping the exploitation dataset constant for evaluation of simple regret. It can be seen that the simple regret of the Contextual-Gap converges faster than the simple regret of other baselines.

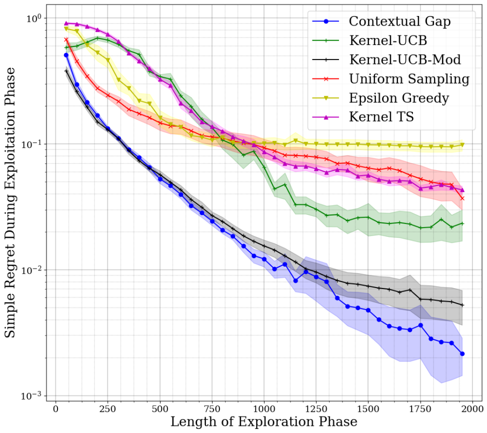


Figure 4.5: Average Simple Regret Evaluation on Synthetic Dataset

IV.6.2 Experimental Spacecraft Magnetic Field Dataset

We present the experimental setup and results associated with a lab generated, realistic spacecraft magnetic field dataset with *non-i.i.d. contexts*. In spacecraft magnetic field data, we are interested in identifying the least noisy sensor for every time step (see Section 2).

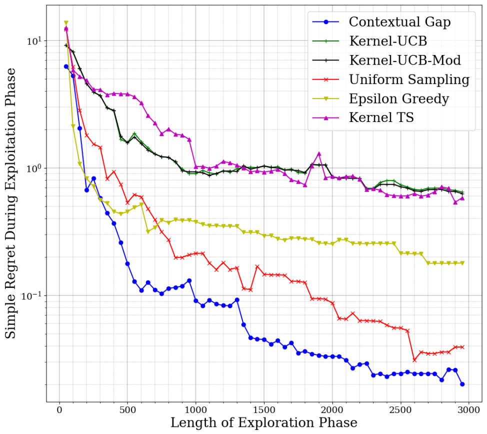


Figure 4.6: Average Simple Regret Evaluation on Spacecraft Magnetic Field Dataset

The dataset was generated with contexts x_t consisting of measured variables associated with the electrical behavior of the GRIFEX spacecraft [97, 98], and reward is the negative of the magnitude of the sensor noise measured at every time step.

Data were collected using 3 sensors (arms), and sensor readings were downloaded for all three sensors at all times steps, although the algorithm does not know these in advance and must select one sensor at each time step. The context information was used in conjunction with a realistic simulator to generate spacecraft magnetic field, and hence a realistic model of sensor noise, as a function of context. The true magnetic field was computed using models of the earth's magnetic field.

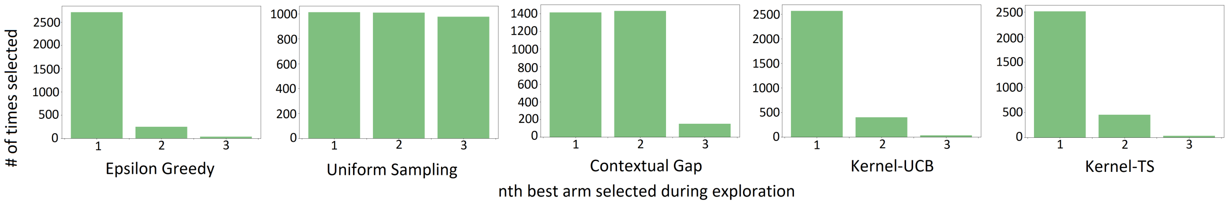


Figure 4.7: Histogram of Arm Selection during exploration

Histogram of number times the best, second best and third best arms are selected during exploration is shown in Figure 4.7. As expected, algorithms designed to minimize cumulative regret focus on the best arm more and Contextual-Gap explores best and second best arms.

The contextual gap algorithm presented is a solution to simple regret minimization, and not average simple regret minimization. Hence,we present the worst case simple regret among all the data present in the exploitation phase as additional empirical evidence of simple regret minimization (Figure 4.8).

Figure 4.6 shows the average simple regret minimization curves for the spacecraft data-set and even in this case Contextual-Gap converges faster compared to other algorithm. From the results above and additional experimental results in Sharma et. al. (2018) [95], one can conclude that Contextual-Gap has advtantage only if top best arms are closer to each other. In the case when best and second best are very far from each other, exploring second best does not give any additional advantage.

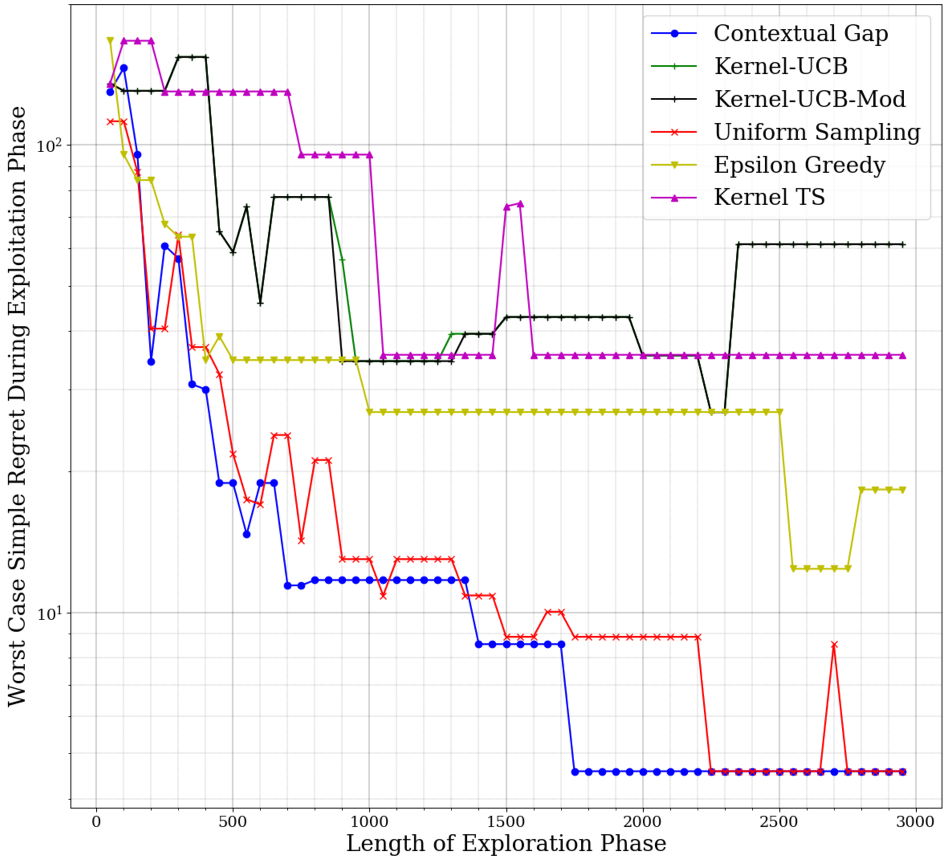


Figure 4.8: Worst Case Simple Regret Evaluation on Spacecraft Magnetic Field Dataset

IV.7 Conclusion

In this work, we present a novel problem: that of simple regret minimization in the contextual bandit setting. We propose the Contextual-Gap algorithm, give a regret bound for the simple regret, and show empirical results on lab-based spacecraft magnetometer dataset. It can be seen that in this scenario persistent and efficient exploration of the best and second best arms with the Contextual-Gap algorithm provides improved results compared against algorithms designed to optimize cumulative regret.

IV.8 Proofs

IV.8.1 Probabilistic Setting and Martingale Lemma

For the theoretical results, the following general probabilistic framework is adopted, following [18] and [90]. We formalize the notion of history H_t defined in the Section 3 of the chapter using filtration. A filtration is a sequence of σ-algebras $\{\mathcal{F}_t\}_{t=1}^{\infty}$ such that $\mathcal{F}_1 \subseteq \mathcal{F}_2 \subseteq \cdots \subseteq \mathcal{F}_n \subseteq \cdots$. Let $\{\mathcal{F}_t\}_{t=1}^{\infty}$ be a filtration such that x_t is $\mathcal{F}_{t-1}$ measurable, and ζ_t is $\mathcal{F}_t$ measurable. For example, one may take $\mathcal{F}_t := \sigma(x_1, x_2, \cdots, x_{t+1}, \zeta_1, \zeta_2, \cdots, \zeta_t)$, i.e., $\mathcal{F}_t$ is the $\sigma-$algebra generated by $x_1, x_2, \cdots, x_{t+1}, \zeta_1, \zeta_2, \cdots, \zeta_t$.

We assume that ζ_t is a zero mean, ρ-conditionally sub-Gaussian random variable, i.e., ζ_t is such that for some $\rho > 0$ and $\forall \gamma \in \mathbb{R}$,

$$\mathbb{E}[e^{\gamma \zeta_t} | \mathcal{F}_{t-1}] \leq \exp\left(\frac{\gamma^2 \rho^2}{2}\right). \tag{4.7}$$

Definition IV.1 (Definition 4.11 in [99]). Let $(\Sigma, \mathcal{F}, Pr)$ be a probability space with filtration

$\mathcal{F}_0, \mathcal{F}_1, \ldots$. Suppose that $Z_0, Z_1, \ldots$ are random variables such that for all $i > 0$, Z_i is $\mathcal{F}_i$ measurable. The sequence $Z_0, Z_1, \ldots$ is a martingale provided for all $i \geq 0$,

$$\mathbb{E}[Z_{i+1}|\mathcal{F}_i] = Z_i.$$

Lemma 22 (Theorem 4.12 in [99])**.** *Any subsequence of a martingale is also a martingale (relative to the corresponding subsequence of the underlying filter).*

The above Lemma is important because we construct confidence intervals for each arm separately. Note that we define a subset of time indices ($D_{a,t}$ of each arm a), when the arm a was selected. Based on these indices we can form sub-sequences of the main context $\{x_t\}_{t=1}^{\infty}$ and noise sequence $\{\zeta_t\}_{t=1}^{\infty}$ such that the assumptions on the main sequence hold for subsequences.

IV.8.1.1 Proof of Theorem 10

Theorem 4.1 is a slight modification of Theorem 2.1 in [90]. In the contextual bandit setting in [90], for any $\delta \in (0, 1]$, Theorem 2.1 in [90] establishes that with probability at least $1 - \delta$, it holds simultaneously over all $x \in \mathcal{X}$ and $t \geq 0$,

$$|f_a(x) - \hat{f}_{a,t}(x)| \leq \frac{\hat{\sigma}_{a,t}(x)}{\sqrt{\lambda}}\left[\sqrt{\lambda}\|f_a\|_{\mathcal{H}} + \rho\sqrt{2\ln(1/\delta) + 2\gamma_t(\lambda)}\right],$$

where $\gamma_t(\lambda) = \frac{1}{2}\sum_{\tau=1}^{t} \ln(1 + \frac{1}{\lambda}\hat{\sigma}_{a,\tau-1}(x_\tau))$

For $T \geq t$, one can replace t in the log terms with T. Then $\forall x, \forall t \geq 1$, we have

$$1 - \delta \leq \mathbb{P}\left(|f_a(x) - \hat{f}_{a,t}(x)| \leq \frac{\hat{\sigma}_{a,t}(x)}{\sqrt{\lambda}}\left[\sqrt{\lambda}\|f_a\|_{\mathcal{H}} + \rho\sqrt{2\ln(1/\delta) + 2\gamma_T(\lambda)}\right]\right).$$

Let $\delta = e^{-\beta^2}$. In that case,

$$1 - e^{-\beta^2} \leq \mathbb{P}\left(|f_a(x) - \hat{f}_{a,t}(x)| \leq \frac{\hat{\sigma}_{a,t}(x)}{\sqrt{\lambda}} \left[\sqrt{\lambda} \|f_a\|_{\mathcal{H}} + \rho \sqrt{2\beta^2 + 2\gamma_T(\lambda)} \right] \right).$$

Using triangle inequality $\sqrt{p+q} \leq \sqrt{p} + \sqrt{q}$ for any $p, q \geq 0$,

$$1 - e^{-\beta^2} \leq \mathbb{P}\left(|f_a(x) - \hat{f}_{a,t}(x)| \leq \frac{\hat{\sigma}_{a,t}(x)}{\sqrt{\lambda}} \left[\sqrt{\lambda} \|f_a\|_{\mathcal{H}} + \rho \sqrt{2\beta^2} + \rho \sqrt{2\gamma_T(\lambda)} \right] \right).$$

Let $C_1 = \rho\sqrt{2}$ and $C_2 = \sqrt{\lambda}\|f_a\|_{\mathcal{H}} + \rho\sqrt{2\gamma_T(\lambda)}$. Hence, we have

$$1 - e^{-\beta^2} \leq \mathbb{P}\left(|f_a(x) - \hat{f}_{a,t}(x)| \leq \frac{\hat{\sigma}_{a,t}(x)}{\sqrt{\lambda}} [C_1\beta + C_2] \right).$$

IV.8.2 Lower Bound on r^{th} Eigenvalue

First we state the Lemmas that we use to prove Lemma 20.

Lemma 23 (Lemma 9 in [92]). *If $a > 0, b > 0, ab \geq e$, then for all $t \geq 2a\log(ab)$,*

$$t \geq a\log(bt). \tag{4.8}$$

Lemma 24 (Lemma 1.1 in [68]). *Let $A \in \mathbb{R}^{n\times n}$ be a symmetric positive definite matrix partitioned according to*

$$A = \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{12}^T & A_{22} \end{array}\right],$$

where $A_{11} \in \mathbb{R}^{(n-1)\times(n-1)}, A_{12} \in \mathbb{R}^{(n-1)}$ and $A_{22} \in \mathbb{R}^1$. Then $\det(A) = \det(A_{11})(A_{22} - A_{12}^T A_{11}^{-1} A_{12})$.

Lemma 25 (Special case of extended Horn's inequality (Theorem 4.5 of [100]))**.** *Let A, B be compact self-adjoint operators. Then for any $p \geq 1$,*

$$\lambda_p(A + B) \leq \lambda_1(A) + \lambda_p(B). \tag{4.9}$$

Theorem 12 (Freedman's inequality for self adjoint operators, Thm 3.2 & section 3.2 in [101])**.** *Let $\{\Phi_t\}_{t=1,...}$ be a sequence of self-adjoint Hilbert Schmidt operators $\Phi_t : \mathcal{H} \to \mathcal{H}$ acting on a seperable Hilbert space ($\mathbb{E}\Phi$ is a operator such that $\langle(\mathbb{E}\Phi)z_1, z_2\rangle_{\mathcal{H}} = \mathbb{E}\langle\Phi z_1, z_2\rangle_{\mathcal{H}}$ for any $z_1, z_2 \in \mathcal{H}$). Additionally, assume that $\{\Phi_t\}_{t=1,...}$ is a martingale difference sequence of self adjoint operators such that $\|\Phi_t\| \leq L^2$ almost surely for all $1 \leq t \leq T$ and some positive $L \in \mathbb{R}$. Denote by $W_t = \sum_{s=1}^{t} \mathbb{E}_{s-1}[\Phi_s^2]$ and $p(t) = \min(-t, 1)$. Then for any $a \geq \frac{1}{6}(L^2 + \sqrt{L^4 + 36b}), b \geq 0$,*

$$\mathbb{P}\left(\|\sum_{j=1}^{t} \Phi_j\| > a \text{ and } \lambda_1(W_t) \leq b\right) \leq \tilde{d} \cdot \exp\left(-\frac{a^2/2}{b + aL^2/3}\right),$$

where $\|\cdot\|$ is the operator norm and $\tilde{d} := 50 \sum_{r=1}^{\infty}(p(-\frac{a\lambda_r(\mathbb{E}W_t)}{L^2 b}))$.

Note that $\tilde{d}$ is a function of t but it's upper bounded by d^* which is the rank of $\mathbb{E}_{s-1}[\phi(x)\phi(x)^T]$.

IV.8.2.1 Proof of Lemma 20

Lemma 7 in [92] gives the lower bound on minimum eigenvalue (finite dimensional case) when reward depends linearly on context. We extend it to r^{th} largest eigenvalue (infinite dimensional case) and the case when reward depends non-linearly on context.

Proof. $\mathcal{X} \subset \mathbb{R}^d$ is a compact space endowed with a finite positive Borel measure. For a continuous kernel k the canonical feature map ϕ is a continuous function $\phi : \mathcal{X} \to \mathcal{H}$, where

$\mathcal{H}$ is a separable Hilbert space (See section 2 of [102] for a construction such that $\mathcal{H}$ is separable). In such a setting, $\phi(\mathcal{X})$ is also compact space with a finite positive Borel measure [102]. We now define a few terms on $\phi(\mathcal{X})$.

Define the random variable $\Phi_t := \mathbb{E}_{t-1}[\phi(x_t)\phi(x_t)^T] - \phi(x_t)\phi(x_t)^T$. Let $Z_t := \sum_{s=1}^{t} \Phi_s = \sum_{s=1}^{t} \mathbb{E}_{s-1}[\phi(x_t)\phi(x_t)^T] - S_t = V_t - S_t$.

By construction, $\{Z_t\}_{t=1,2,\ldots}$ is a martingale and $\{\Phi_s\}_{s=1,2,\ldots}$ is the martingale difference sequence. Notice that $\lambda_1(\Phi_t) \leq L^2$. To use the Freedman's inequality, we lower bound the operator norm of Z_t, $\|Z_t\|$ and upper bound the largest eigenvalue of W_t, $\lambda_1(W_t)$. Let $\nu(A) = \max_i |\lambda_i(A)|$ be the spectral radius of operator A. We work with the spectral radius because it is not necessary that Z_t is a positive definite operator. It is well known that

$$\nu(A) \leq \|A\|. \tag{4.10}$$

By assumption **A III**, $\mathbb{E}_{s-1}[\phi(x)\phi(x)^T]$ lies in a fixed d^* dimensional subspace with its eigenvalues $\lambda_r(\mathbb{E}_{s-1}[\phi(x)\phi(x)^T]) > \lambda_x$ for $r \leq d^*$. Thus, for $V_t = \sum_{s=1}^{t} \mathbb{E}_{s-1}[\phi(x)\phi(x)^T]$, $\lambda_r(V_t) \geq t\lambda_x$.

Bound on $\|Z_t\|$: By definition, $V_t = Z_t + S_t$. Hence, $\lambda_r(V_t) \leq \lambda_1(Z_t) + \lambda_r(S_t)$ by using Horn's inequality (Lemma 25).

$$\begin{aligned}
\lambda_1(Z_t) &\geq \lambda_r(V_t) - \lambda_r(S_t) \\
\lambda_1(Z_t) &\geq t\lambda_x - \lambda_r(S_t) \\
\nu(Z_t) &\geq t\lambda_x - \lambda_r(S_t),
\end{aligned}$$

where the second step is due to **A III** and the third step is by definition of spectral radius.

By Eqn. (4.10), we have

$$\|Z_t\| \quad \geq \quad t\lambda_x - \lambda_r(S_t). \tag{4.11}$$

Bound on $\lambda_1(W_t)$: To bound the term $\lambda_1(W_t)$, write

$$\begin{aligned} W_t &= \sum_{s=1}^{t} \mathbb{E}_{s-1}[\Phi_s^2] \\ &= \sum_{s=1}^{t} \mathbb{E}_{s-1}[(\mathbb{E}_{s-1}[\phi(x_s)\phi(x_s)^T] - \phi(x_s)\phi(x_s)^T)^2]. \end{aligned}$$

By using square expansion,

$$\begin{aligned} W_t &= \sum_{s=1}^{t} \mathbb{E}_{s-1}[(\mathbb{E}_{s-1}[\phi(x_s)\phi(x_s)^T]^2 + (\phi(x_s)\phi(x_s)^T)^2 \\ &\quad -\mathbb{E}_{s-1}[\phi(x_s)\phi(x_s)^T](\phi(x_s)\phi(x_s)^T) \\ &\quad -(\phi(x_s)\phi(x_s)^T)\mathbb{E}_{s-1}[\phi(x_s)\phi(x_s)^T]] \\ &= \sum_{s=1}^{t} \mathbb{E}_{s-1}[(\phi(x_s)\phi(x_s)^T)^2] - \mathbb{E}_{s-1}[\phi(x_s)\phi(x_s)^T]^2. \end{aligned}$$

Taking norm on both sides,

$$\|W_t\| = \|\sum_{s=1}^{t} \mathbb{E}_{s-1}[(\phi(x_s)\phi(x_s)^T)^2] - \mathbb{E}_{s-1}[\phi(x_s)\phi(x_s)^T]^2\|.$$

As both terms on the right hand side are positive semi-definite matrices,

$$\|W_t\| \leq \|\sum_{s=1}^{t} \mathbb{E}_{s-1}[(\phi(x_s)\phi(x_s)^T)^2]\|.$$

Next, we use convexity properties of norms to get the upper bound.

$$\begin{aligned}
\|W_t\| &\leq \sum_{s=1}^{t} \|\mathbb{E}_{s-1}[(\phi(x_s)\phi(x_s)^T)^2]\| \\
&= \sum_{s=1}^{t} \|\mathbb{E}_{s-1}[(\phi(x_s)\left(\phi(x_s)^T\phi(x_s)\right)\phi(x_s)^T)]\| \\
&\leq L^2 \sum_{s=1}^{t} \|\mathbb{E}_{s-1}[(\phi(x_s)\phi(x_s)^T)]\| \\
&\leq L^2 \sum_{s=1}^{t} \mathbb{E}_{s-1}[\|(\phi(x_s)\phi(x_s)^T)\|]
\end{aligned}$$

where the first step is due to the triangle inequality and the third step is due to the upper bound $\|\phi(x)\| \leq L$, the fourth step is due to the convexity of the operator norm and Jensen's inequality. Using the properties of Hilbert Schmidt operators, we can write

$$\begin{aligned}
\mathbb{E}_{s-1}[\|(\phi(x_s)\phi(x_s)^T)\|] &\leq \mathbb{E}_{s-1}[\|(\phi(x_s)\phi(x_s)^T)\|_{HS}] \\
&= \mathbb{E}_{s-1}[\|\phi(x_s)\|^2] \leq L^2
\end{aligned}$$

Therefore, we can bound the norm $\|W_t\|$ as

$$\begin{aligned}
\|W_t\| &\leq L^2 \sum_{s=1}^{t} L^2 \\
&= tL^4,
\end{aligned}$$

Again, by using Eqn. (4.10), we have

$$\lambda_1(W_t) \leq tL^4. \tag{4.12}$$

Now, we shall construct a parameter A such that

$$\frac{a^2/2}{b + aL^2/3} \geq A. \tag{4.13}$$

For this inequality to hold, one can see, by its quadratic solution, $a \geq f(A, b) := \frac{1}{3}AL^2 + \sqrt{\frac{1}{9}A^2L^4 + 2Ab}$. Note that for $A > 1$, the condition of $a \geq f(A, b)$ also satisfies the conditions of Friedman's inequality in Theorem 12.

Let $A(m, \delta) = \log \frac{(m+1)(m+3)}{\delta}$ and P be the probability of event $\Big[\exists t : \lambda_r(\boldsymbol{S}_t) \leq t\lambda_x - f(A(tL^4, \delta), tL^4)\Big]$.

$$\begin{aligned}
& P \\
= \;& \mathbb{P}\Big[\exists t : \lambda_r(\boldsymbol{S}_t) \leq t\lambda_x - f(A(tL^4, \delta), tL^4)\Big] && (4.14)\\
\leq \;& \mathbb{P}\Big[\exists t : \lambda_r(\boldsymbol{S}_t) \leq t\lambda_x \\
& -f(A(\lambda_1(W_t), \delta), \lambda_1(W_t))\Big] && (4.15)\\
\leq \;& \sum_{m=0}^{\infty} \mathbb{P}\Big[\exists t : \lambda_r(\boldsymbol{S}_t) \leq t\lambda_x \\
& -f(A(m, \delta), m), \lambda_1(W_t) \leq m\Big] && (4.16)\\
\leq \;& \sum_{m=0}^{\infty} \mathbb{P}\Big[\exists t : \|Z_t\| \geq f(A(m, \delta), m), \\
& \lambda_1(W_t) \leq m\Big] && (4.17)\\
\leq \;& \tilde{d} \sum_{m=0}^{\infty} \exp\left(-A(m, \delta)\right) && (4.18)\\
= \;& \tilde{d} \sum_{m=0}^{\infty} \frac{\delta}{(m+1)(m+3)} \\
\leq \;& \tilde{d} \cdot \delta, && (4.19)
\end{aligned}$$

where (4.15) is because A is increasing in m, f is increasing in A, b, and Eqn. (4.12). Eqn. (4.16) is by application of the union bound over all the events for which $\lambda_1(W_t) \leq m$. Also, Eqn. (4.17) is due to Eqn. (4.11) and Eqn. (4.18) is due to Theorem 12.

The result is obtained by replacing δ by $\frac{\delta}{\tilde{d}}$.

For the second part. Let $\tilde{\lambda}_x := \frac{\lambda_x}{L}$. By definition of L, $\tilde{\lambda}_x \leq 1$. Let $t \geq \frac{256}{\tilde{\lambda}_x^2} \log \frac{128\tilde{d}}{\tilde{\lambda}_x^2 \delta}$. Then by using the Lemma 23,

$$t \geq \frac{128}{\tilde{\lambda}_x^2} \log \frac{t\tilde{d}}{\delta}. \tag{4.20}$$

Rearranging the terms, we get

$$\frac{t\tilde{\lambda}_x^2}{4} \geq 32 \log \frac{t\tilde{d}}{\delta}$$

Taking square root and then multiplying by $\sqrt{t}$ on both sides

$$\begin{aligned}
\frac{t\tilde{\lambda}_x}{2} &\geq \sqrt{32t \log \frac{t\tilde{d}}{\delta}} \\
&= \frac{2}{3}\sqrt{72t \log \frac{t\tilde{d}}{\delta}} \\
&= \frac{2}{3}\sqrt{36t \log \frac{t\tilde{d}}{\delta} + 36t \log \frac{t\tilde{d}}{\delta}}.
\end{aligned}$$

Using equation (4.20),

$$\begin{aligned}
\frac{t\tilde{\lambda}_x}{2} &\geq \frac{2}{3}\sqrt{36t \log \frac{t\tilde{d}}{\delta} + \frac{36 \cdot 128}{\tilde{\lambda}_x^2}\left(\log \frac{t\tilde{d}}{\delta}\right)^2} \\
&= \frac{2}{3}\sqrt{36t \log \frac{t\tilde{d}}{\delta} + \frac{36 \cdot 32}{\tilde{\lambda}_x^2} 4\left(\log \frac{t\tilde{d}}{\delta}\right)^2}.
\end{aligned}$$

Since $\tilde{\lambda}_x^2 \leq 1$ we have

$$\begin{aligned}
\frac{t\tilde{\lambda}_x}{2} &\geq \frac{2}{3}\sqrt{36t \log \frac{t\tilde{d}}{\delta} + (36 \cdot 32)\left(2 \log \frac{t\tilde{d}}{\delta}\right)^2} \\
&> \frac{2}{3}\sqrt{18t \cdot 2 \log \frac{t\tilde{d}}{\delta} + \left(2 \log \frac{t\tilde{d}}{\delta}\right)^2}. \qquad (4.21)
\end{aligned}$$

Now we use the condition on δ as stated in the Theorem statement: $0 \leq \delta \leq \frac{1}{8}$. We can see that

$$\frac{1}{8} \leq \frac{t^2 \tilde{d}^2}{(t+1)(t+3)}, \tag{4.22}$$

because $\frac{t^2 \tilde{d}^2}{(t+1)(t+3)}$ is a monotonically increasing function for both $t, \tilde{d}$ for $t, \tilde{d} \geq 1$. Simplifying Eqn. (4.22), we get

$$\frac{t^2 \tilde{d}^2}{\delta^2} \geq \frac{(t+1)(t+3)}{\delta}. \tag{4.23}$$

Taking log of both sides,

$$2 \log \frac{t\tilde{d}}{\delta} \geq \log(\frac{(t+1)(t+3)}{\delta}) = A(t, \delta).$$

Without loss of generality, we will assume that $L = 1$. From Eqn. (4.23) and Eqn. (4.21), we have

$$\begin{aligned}
\frac{t\lambda_x}{2} &\geq \frac{2}{3}\sqrt{18t \cdot A(t,\delta) + A(t,\delta)^2} \\
&= \frac{1}{3}\sqrt{18t \cdot A(t,\delta) + A(t,\delta)^2} \\
&\quad + \frac{1}{3}\sqrt{18t \cdot A(t,\delta) + A(t,\delta)^2} \\
&\geq \frac{1}{3}\sqrt{18t \cdot A(t,\delta) + A(t,\delta)^2} + \frac{1}{3}A(t,\delta)
\end{aligned}$$

Therefore,

$$\frac{t\lambda_x}{2} \geq f(A(t,\delta), t). \tag{4.24}$$

Equations (4.14) and (4.24) complete the proof. □

IV.8.3 Monotonic Upper bound of $s_{a,t}(x)$

Lemma 26. *[Arithmetic Mean-Geometric Mean Inequality [72]] For every sequence of nonnegative real numbers $a_1, a_2, \ldots a_n$ one has*

$$(\prod_{i=1}^{n} a_i)^{1/n} \leq \frac{\sum_{i=1} a_i}{n}$$

with equality if and only if $a_1 = a_2 = \ldots = a_n$.

Lemma 27. *If $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d > 0$, and $\mu_1 \geq 0, \mu_2 \geq 0 \cdots \mu_d \geq 0$ such that $\sum_j \mu_j = L$ and $\lambda_d \geq L$ then*

$$\prod_{i=1}^{d} \left(1 + \frac{\mu_i}{\lambda_i}\right) - 1 \leq \frac{2L}{\lambda_d}.$$

Proof. By replacing each λ_i with the smallest element λ_d we get,

$$\begin{aligned}
\prod_{i=1}^{d} \left(1 + \frac{\mu_i}{\lambda_i}\right) - 1 &\leq \prod_{i=1}^{d} \left(1 + \frac{\mu_i}{\lambda_d}\right) - 1 \\
&= \prod_{i=1}^{d} \left(\frac{\lambda_d + \mu_i}{\lambda_d}\right) - 1 \\
&= \left(\frac{\prod_{i=1}^{d} (\lambda_d + \mu_i)}{\lambda_d^d}\right) - 1 \\
&\leq \left(\frac{\sum_{i=1}^{d} (\lambda_d + \mu_i)}{d\lambda_d}\right)^d - 1 \\
&= \left(\frac{d\lambda_d + L}{d\lambda_d}\right)^d - 1 \\
&= \left(1 + \frac{L}{d\lambda_d}\right)^d - 1 \\
&\leq e^{L/\lambda_d} - 1,
\end{aligned}$$

where the fourth inequality is by Lemma 26 and last inequality holds because $(1 + \frac{a}{x})^x$ approaches e^a as $x \to \infty$ and $(1 + \frac{a}{x})^x$ is a monotonically increasing function of x.

By $e^x \leq 1 + 2x$ for $x \in [0,1]$ and the assumption that $\lambda_d \geq L$,

$$\begin{aligned}\prod_{i=1}^{d}\left(1+\frac{\mu_i}{\lambda_i}\right)-1 &\leq e^{L/\lambda_d}-1 \\ &\leq 1+\frac{2L}{\lambda_d}-1 \\ &= \frac{2L}{\lambda_d}.\end{aligned}$$

□

IV.8.3.1 Proof of Lemma 21

Proof. We will assume that $L = 1$. We write

$$K_{a,t+1} + \lambda_{I_{a,t+1}} = \left[\begin{array}{c|c} K_{a,t} + \lambda_{I_{a,t}} & k_{a,t}(x) \\ \hline k_{a,t}(x)^T & k(x,x) + \lambda \end{array}\right].$$

Let $\mu_i = \lambda_i(K_{a,t+1} + \lambda_{I_{a,t+1}}) - \lambda_i(K_{a,t} + \lambda_{I_{a,t}})$.

Using Lemma 24,

$$\begin{aligned} & \det(K_{a,t+1} + \lambda_{I_{a,t+1}}) \\ = \quad & \det(K_{a,t} + \lambda_{I_{a,t}})\Big(k(x,x) + \lambda \\ & -k_{a,t}(x)^T(K_{a,t} + \lambda_{I_{a,t}})^{-1}k_{a,t}(x)\Big). \end{aligned}$$

Rearranging,

$$k(x,x) - k_{a,t}(x)^T(K_{a,t} + \lambda_{I_{a,t}})^{-1}k_{a,t}(x)$$

$$= \frac{\det(K_{a,t+1} + \lambda_{I_{a,t+1}})}{\det(K_{a,t} + \lambda_{I_{a,t}})} - \lambda.$$

Dividing both sides by λ,

$$\begin{aligned}
&\frac{k(x,x) - k_{a,t}(x)^T (K_{a,t} + \lambda_{I_{a,t}})^{-1} k_{a,t}(x)}{\lambda} \\
= &\frac{\det(K_{a,t+1} + \lambda_{I_{a,t+1}})}{\lambda \det(K_{a,t} + \lambda_{I_{a,t}})} - 1. \qquad (4.25)
\end{aligned}$$

Notice that the left hand side is equal to $\frac{\hat{\sigma}_{a,t}(x)}{\lambda}$. Using the definitions of $s_{a,t}(x)$ and $\hat{\sigma}_{a,t}(x)$, we can write,

$$\begin{aligned}
s_{a,t}(x)^2 &= 4(C_1\beta + C_2)^2 \frac{\hat{\sigma}_{a,t}(x)^2}{\lambda} \\
&= 4(C_1\beta + C_2)^2 \left(\frac{\det(K_{a,t+1} + \lambda I_{a,t+1})}{\lambda \det(K_{a,t} + \lambda I_{a,t})} - 1 \right) \\
&= 4(C_1\beta + C_2)^2 \left(\frac{\prod_{i=1}^{N_{a,t}+1} \lambda_{i,a,t+1}}{\lambda \prod_{i=1}^{N_{a,t}} \lambda_{i,a,t}} - 1 \right)
\end{aligned}$$

By assumption in the statement of the Lemma, $N_{a,t} \geq d^*$. Hence, all eigenvalues above d^* are λ.

By replacing all eigenvalues $\lambda_{i,a,\tau}$ by λ for $\tau = \{t, t+1\}$ and $i > d^*$, we get

$$s_{a,t}(x)^2 = 4(C_1\beta + C_2)^2 \left(\prod_{i=1}^{d^*} \frac{\lambda_{i,a,t+1}}{\lambda_{i,a,t}} - 1 \right).$$

Note that $\lambda_{i,a,t+1} = \lambda_{i,a,t} + \mu_i$. By replacing $\lambda_{i,a,t+1}$, we get

$$\begin{aligned}
s_{a,t}(x)^2 &= 4(C_1\beta + C_2)^2 \left(\prod_{i=1}^{d^*} \frac{\lambda_{i,a,t} + \mu_i}{\lambda_{i,a,t}} - 1 \right) \\
&= 4(C_1\beta + C_2)^2 \left(\prod_{i=1}^{d^*} \left(1 + \frac{\mu_i}{\lambda_{i,a,t}} \right) - 1 \right) \\
&\leq 4(C_1\beta + C_2)^2 \left(1 + \frac{2L}{\lambda_{d^*,a,t}} - 1 \right),
\end{aligned}$$

where the third inequality is due to Lemma 27.

For $L = 1$,

$$\begin{aligned} s_{a,t}(x)^2 &\leq 4(C_1\beta + C_2)^2\left(1 + \frac{2}{\lambda_{d^*,a,t}} - 1\right) \\ &= 4(C_1\beta + C_2)^2\left(\frac{2}{\lambda_{d^*,a,t}}\right). \end{aligned}$$

Note that $\lambda_{d^*,a,t} = \lambda_{d^*}(K_{a,t+1} + \lambda I_{a,t+1}) = \lambda_{d^*}(K_{a,t+1}) + \lambda$. By Lemma 20, $\lambda_{d^*}(K_{a,t+1}) \geq N_{a,t}\lambda_x$. We can apply Lemma 27 only when

$$\frac{1}{\lambda + N_{a,t}\lambda_x/2} < 1$$

or

$$N_{a,t} > \frac{2(1-\lambda)}{\lambda_x}.$$

The assumption in the statement of the lemma satisfies the above equation. Hence, we have

$$\begin{aligned} s_{a,t}(x)^2 &\leq 4(C_1\beta + C_2)^2\left(\frac{2}{\lambda + N_{a,t}\lambda_x/2}\right) \\ &= 8(C_1\beta + C_2)^2\left(\frac{1}{\lambda + N_{a,t}\lambda_x/2}\right) \\ &= g_{a,t}(N_{a,t}). \end{aligned}$$

This concludes the proof. □

IV.8.3.2 Closed form of $g_{a,t}^{-1}(s)$

Now we calculate a closed form expression of $N_{a,t}$. Setting the upper bound on confidence in the Theorem 10 to s, we calculate the inverse in terms of $N_{a,t}$,

$$8(C_1\beta + C_2)^2\left(\frac{1}{\lambda + N_{a,t}\lambda_x/2}\right) \quad = \quad s^2.$$

Rearranging all the terms, we get

$$\begin{aligned} 8(C_1\beta + C_2)^2 &= s^2(\lambda + N_{a,t}\lambda_x/2) \\ (\lambda + N_{a,t}\lambda_x/2) &= \frac{8(C_1\beta + C_2)^2}{s^2} \\ N_{a,t} &= \frac{16(C_1\beta + C_2)^2}{s^2\lambda_x} - \frac{2\lambda}{\lambda_x}. \end{aligned}$$

Define

$$g_{a,t}^{-1}(s) = \frac{16(C_1\beta + C_2)^2}{s^2\lambda_x} - \frac{2\lambda}{\lambda_x}. \tag{4.26}$$

IV.8.4 Simple Regret Analysis

Lemma 28 (Value of β). *Assume the conditions in Theorem 10 and Lemma 21. If* $\sum_{a\in[A]} g_{a,t}^{-1}(H_{a\epsilon}) = T - N_\lambda(A-1)$, *then*

$$\beta = \sqrt{\frac{\lambda_x(T - N_\lambda(A-1)) + 2A\lambda}{16C_1^2 H_\epsilon}} - \frac{C_2}{C_1}. \tag{4.27}$$

Proof. We have

$$\sum_{a\in[A]} g_{a,t}^{-1}(H_{a\epsilon}) \quad = \quad T - N_\lambda(A-1).$$

By using Eqn. (4.26),

$$\begin{aligned}
\sum_{a\in[A]} \frac{16(C_1\beta + C_2)^2}{H_{a\epsilon}^2 \lambda_x} - \frac{2\lambda}{\lambda_x} &= T - N_\lambda(A-1) \\
\frac{16(C_1\beta + C_2)^2}{\lambda_x} \sum_{a\in[A]} \frac{1}{H_{a\epsilon}^2} - \frac{2A\lambda}{\lambda_x} &= T - N_\lambda(A-1).
\end{aligned}$$

By using definition of H_ϵ,

$$\frac{16(C_1\beta + C_2)^2 H_\epsilon}{\lambda_x} - \frac{2A\lambda}{\lambda_x} = T - N_\lambda(A-1)$$

Rearranging the terms,

$$\begin{aligned}
16(C_1\beta + C_2)^2 H_\epsilon &= \lambda_x(T - N_\lambda(A-1)) + 2A\lambda \\
(C_1\beta + C_2)^2 &= \frac{\lambda_x(T - N_\lambda(A-1)) + 2A\lambda}{16H_\epsilon} \\
\beta &= \sqrt{\frac{\lambda_x(T - N_\lambda(A-1)) + 2A\lambda}{16C_1^2 H_\epsilon}} - \frac{C_2}{C_1}.
\end{aligned}$$

□

IV.8.4.1 Proof of Theorem 11

Let $[A] = \{1, ..., A\}$. We define a feasible set $A'(x) \subseteq [A]$ such that elements of $A'(x)$ contain possible set of arms that may be pulled if context x was observed at all times $AN_\lambda < t \leq T$. The set $A'(x)$ is used to discount the arms that will never be pulled with context x.

Proof. The proof broadly follows the same structure presented in Theorem 2 of [76]. We will provide the simple regret bound at the recommendation of time $T+1$, since the algorithm operates in a pure exploitation setting, the recommended arm Ω_{T+2} will follow the same properties.

Fix $x \in \mathcal{X}$ such that x can be generated from the filtration. We define the event $\mathcal{E}_{a,t}(x)$ to be the event in which for arm $a \leq A$, $f_a(x)$ lies between the upper and lower confidence bounds given $x_1, x_2, ..., x_{t-1}$ More precisely,

$$\mathcal{E}_{a,t}(x) = \{L_{a,t}(x_t) \leq f_a(x) \leq U_{a,t}(x) | x_1, x_2, \cdots, x_{t-1}\}.$$

For events $\mathcal{E}_{a,t}$, from Theorem 10,

$$\mathbb{P}(\mathcal{E}_{a,t}(x)) \geq 1 - e^{-\beta^2}.$$

Let $N_{a,T}$ denote the number of times each arm has been tried upto time T. Clearly $\sum_{a=1}^{A} N_{a,T} = T$. Also, note that we try each arm at least N_λ number of times before we run our algorithm. We define event $\mathcal{E}$ as $\mathcal{E} := \bigcup_{a \leq A, AN_\lambda < t \leq T} \mathcal{E}_{a,t}(x)$. By the union bound we can show that

$$\mathbb{P}(\mathcal{E}) \geq 1 - A(T - AN_\lambda)e^{-\beta^2}.$$

The next part of the proof works by contradiction.

Let $\epsilon > 0$. The recommended arm at the end of time T for context x is defined as follows: let $t^* := \arg\min_{AN_\lambda < t \leq T} B_{J_t(x),t}(x)$ then the recommended arm is $\Omega := \Omega_{T+1} := J_{t^*}(x)$.

Conditioned on event $\mathcal{E}$, we will assume that the event $R_\Omega(x) > \epsilon$ is true and arrive at a contradiction with high probability. Note that if $R_\Omega(x) > \epsilon$, the recommended arm Ω is necessarily sub-optimal (regret is zero for the optimal arm).

Define $M_{a,T}(x)$ as number of times arm $a \in [A]$ would be selected in $AN_\lambda < t \leq T$, if we had seen context x at all those times. Hence, $\sum\limits_{a\in A'(x)} M_{a,T}(x) = T - AN_\lambda$. Also, note that $N_{a,T}(x) = M_{a,T}(x) + N_\lambda$ for $a \in A'(x)$ and $N_{a,T}(x) = N_\lambda$ otherwise. Let $t_a = t_a(x)$ be the last time instant for which arm $a \in A'(x)$ may have been selected using the Contextual-Gap algorithm if context x was observed throughout.

The following holds for the recommended arm Ω with context x:

$$\begin{aligned} \min(0, s_{a,t_a}(x) - \Delta_a(x)) + s_{a,t_a}(x) &\geq B_{J_{t_a}(x),t_a}(x) \\ &\geq B_{\Omega,T+1}(x) \\ &\geq R_\Omega(x) \\ &> \epsilon. \end{aligned}$$

Where the first inequality holds due to Lemma 31, the second inequality holds by definition of $B_{\Omega,T+1}$, the third inequality holds due to Lemma 29 and the last inequality holds due to the event $R_\Omega > \epsilon$. The preceding inequality can also be written as

$$\begin{aligned} s_{a,t_a}(x) > 2s_{a,t_a}(x) - \Delta_a(x) > \epsilon, &\qquad \text{if } \Delta_a(x) > s_{a,t_a}(x). \\ 2s_{a,t_a}(x) - \Delta_a(x) > s_{a,t_a}(x) > \epsilon, &\qquad \text{if } \Delta_a(x) < s_{a,t_a}(x). \end{aligned}$$

This leads to the following bound on the confidence diameter of $a \in [A]$,

$$s_{a,t_a}(x) > \max(\frac{1}{2}(\Delta_a(x) + \epsilon), \epsilon) =: H_{a\epsilon}(x).$$

For any arm a, we consider the final number of arm pulls $M_{a,T}(x) + N_\lambda$. From Lemma 21 we can write, using the strict monotonicity and there by invertibility of $g_{a,T}$, with probability at least $1 - \delta$ as

$$\begin{aligned} M_{a,T}(x) + N_\lambda &\leq g_{a,T}^{-1}(s_{a,t_a}(x)) \\ &< g_{a,T}^{-1}(H_{a\epsilon}(x)) \\ &\leq g_{a,T}^{-1}(H_{a\epsilon}), \end{aligned}$$

where $H_{a\epsilon} = \inf_x H_{a\epsilon}(x)$. Last two equations hold as $g_{a,T}$ is a monotonically decreasing function. By summing both sides with respect to $a \in A'(x)$ we can write

$$T - AN_\lambda + |A'(x)|N_\lambda \quad < \quad \sum_{a \in A'(x)} g_{a,T}^{-1}(H_{a\epsilon}),$$

We can make RHS even bigger by adding terms $a \in [A] \backslash A'(x)$. Hence, we get

$$T - (A - |A'(x)|)N_\lambda \quad < \quad \sum_{a \in [A]} g_{a,T}^{-1}(H_{a\epsilon}).$$

We can make LHS even smaller by noting that minimum value of $|A'(x)|$ is one.

$$T - AN_\lambda + N_\lambda \quad < \quad \sum_{a \in [A]} g_{a,T}^{-1}(H_{a\epsilon}).$$

Rearranging the terms, we get

$$\begin{aligned} T - AN_\lambda + N_\lambda \quad &< \quad \sum_{a \in [A]} g_{a,T}^{-1}(H_{a\epsilon}) \\ T - N_\lambda(A-1) \quad &< \quad \sum_{a \in [A]} g_{a,T}^{-1}(H_{a\epsilon}). \end{aligned}$$

which contradicts our definition of $g_{a,T}$ in the theorem statement. Therefore $R_{\Omega_T}(x) \leq \epsilon$.

From the preceding argument we have that if $\sum_{a \in [A]} g_{a,T}^{-1}(H_{a\epsilon}) \leq T - N_\lambda(A-1)$, then for any $x \in \mathcal{X}$ generated from the filtration,

$$\mathbb{P}(R_{\Omega_T} < \epsilon | x) \geq 1 - A(T - AN_\lambda)e^{-\beta^2} - A\delta.$$

In the above equation, $1 - A(T - AN_\lambda)e^{-\beta^2}$ is from the event $\mathcal{E}$ and $1 - A\delta$ is due to the fact that the monotonic upper bounds holds only with probability $1 - \delta$ for each of the arms. Setting β such that $\sum_{a \in [A]} g_{a,T}^{-1}(H_{a\epsilon}) = T - N_\lambda(A - 1)$ (See Lemma 28), we have for

$$\beta = \sqrt{\frac{\lambda_x(T - N_\lambda(A - 1)) + 2A\lambda}{16C_1^2 H_\epsilon}} - \frac{C_2}{C_1},$$

that

$$\mathbb{P}(R_{\Omega_T} < \epsilon | x) \geq 1 - A(T - AN_\lambda)e^{-\beta^2} - A\delta,$$

for $C_1 = \rho\sqrt{2}$ and $C_2 = \rho\sqrt{\sum_{\tau=2}^{T} \ln(1 + \frac{1}{\lambda}\hat{\sigma}_{a,\tau-1}(x_\tau))} + \sqrt{\lambda}\|f_a\|_{\mathcal{H}}$.

Since C_2 depends on T, to complete the proof and validity of the bound, we will show that C_2 grows logarithmically in T. When assumption **A III** holds and $\|\phi(x)\| \leq 1$, similar to the analysis in [18, 90], we have

$$\begin{aligned}
C_2 &= \rho\sqrt{\sum_{\tau=2}^{T} \ln(1 + \frac{1}{\lambda}\hat{\sigma}_{a,\tau-1}(x_\tau))} + \sqrt{\lambda}\|f_a\|_{\mathcal{H}} \\
&= \rho\sqrt{\sum_{\tau=2}^{T} \ln(1 + \frac{1}{\lambda}\phi(x_\tau)^T(I + \frac{1}{\lambda}K_{a,\tau-1})^{-1}\phi(x_\tau))} \\
&+ \sqrt{\lambda}\|f_a\|_{\mathcal{H}} \\
&= \rho\sqrt{\ln(\det(I + \frac{1}{\lambda}K_{a,T}))} + \sqrt{\lambda}\|f_a\|_{\mathcal{H}} \\
&\leq \rho\sqrt{d^* \ln\left(\frac{1}{d^*}\left(1 + \frac{T}{\lambda}\right)\right)} + \sqrt{\lambda}\|f_a\|_{\mathcal{H}}.
\end{aligned}$$

Since C_2 depends on $\sqrt{\ln(T)}$, we fix $C_2 = O(\rho\sqrt{\ln(T)})$. As $T \to \infty$ the RHS of the probability bound goes to unity and we have the resulting theorem.

□

IV.8.4.2 Lemmas over event $\mathcal{E}$

For arm a at time t, we define event $\mathcal{E}_{a,t}$ as

$$\mathcal{E}_{a,t}(x) = \{L_{a,t}(x_t) \leq f_a(x) \leq U_{a,t}(x) | x_1, x_2, \cdots, x_{t-1}\}.$$

We define event $\mathcal{E}$ as $\mathcal{E} := \bigcup_{a \leq A, AN_\lambda < t \leq T} \mathcal{E}_{a,t}(x)$

The following theorems operate under the assumption the event $\mathcal{E}$ holds. We provide two properties of the terms in the algorithm that will be of help in the proofs:

- $B_{J_t(x)} = U_{j_t(x),t}(x) - L_{J_t(x),t}(x)$
- $U_{a,t}(x) = L_{a,t}(x) + s_{a,t}(x)$

Lemma 29. *Over event $\mathcal{E}$, for any sub-optimal arm $a(x) \neq a^*(x)$ at any time $t \leq T$, the simple regret of pulling that arm is upper bounded by the $B_{a,t}(x)$,*

Proof.

$$\begin{aligned} B_{a,t}(x) &= \max_{i \neq a} U_{i,t}(x) - L_{a,t}(x) \\ &\geq \max_{i \neq a} f_i(x) - f_{a,t}(x) = f^*(x) - f_a(x) = R_a(x). \end{aligned}$$

The first inequality holds due to the definition of event $\mathcal{E}$ and the equality holds since we are only considering sub-optimal arms. □

Note that the preceding lemma need not hold for the optimal arm, for which $R_a(x) = 0$ and it is not necessary that $B_{a,t}(x) \geq 0$.

Lemma 30. *Consider the contextual bandit setting proposed in the chapter 4. Over event $\mathcal{E}$, for any time t and context $x \in \mathcal{X}$, the following statements hold for the arm $a = a_t$ to be selected:*

$$\text{if } a = j_t(x), \text{ then } L_{j_t(x),t}(x) \leq L_{J_t(x),t}(x),$$

$$\text{if } a = J_t(x), \text{ then } U_{j_t(x),t}(x) \leq U_{J_t(x),t}(x).$$

Proof. We consider two cases based on which of the two candidate arms $j_t(x), J_t(x)$ is selected.

Case 1: $a = j_t(x)$ is selected. The proof works by contradiction. Assume that $L_{j_t(x),t}(x) > L_{J_t(x),t}(x)$. From the arm selection rule we have $s_{j_t(x),t}(x) \geq s_{J_t(x),t}(x)$. Based on this we can deduce that $U_{j_t(x),t}(x) \geq U_{J_t(x),t}(x)$. As a result,

$$\begin{aligned} B_{j_t(x),t}(x) &= \max_{i \neq j_t(x)} U_{i,t}(x) - L_{j_t(x),t}(x) \\ &< \max_{i \neq J_t(x)} U_{i,t}(x) - L_{J_t(x),t}(x) = B_{J_t(x),t}(x). \end{aligned}$$

The above inequality holds because the arm $j_t(x)$ must necessarily have the highest upper bound over all the arms. However, this contradicts the definition of $B_{J_t(x),t}(x)$ and as a result it must hold that $L_{j_t(x),t}(x) \leq L_{J_t(x),t}(x)$.

Case 2: $a = J_t(x)$ is selected. The proof works by contradiction. Assume that $U_{j_t(x),t}(x) > U_{J_t(x),t}(x)$. From the arm selection rule we have $s_{J_t(x),t}(x) \geq s_{j_t(x),t}(x)$. Based on this we can deduce that $L_{J_t(x),t}(x) \leq L_{j_t(x),t}(x)$. As a result, similar to Case 1,

$$\begin{aligned} B_{j_t(x),t}(x) &= \max_{j \neq j_t(x)} U_{j,t}(x) - L_{j_t(x),t}(x) \\ &< \max_{j \neq J_t(x)} U_{j,t}(x) - L_{J_t(x),t}(x) = B_{J_t(x),t}(x). \end{aligned}$$

The above inequality holds because the arm $j_t(x)$ must necessarily be have the highest upper bound over all the arms. However, this contradicts the definition of $B_{J_t(x),t}(x)$ and as a result it must hold that $U_{j_t(x),t}(x) \leq U_{J_t(x),t}(x)$. □

Corollary 2. *For context x, if arm $a = a_t(x)$ is pulled at time t, then $B_{J_t(x),t}(x)$ is bounded above by the uncertainty of arm a, i.e.,*

$$B_{J_t(x),t}(x) \leq s_{a,t}(x).$$

Proof. By construction of the algorithm $a \in \{j_t(x), J_t(x)\}$. If $a = j_t(x)$, then using the definition of $B_{J_t(x),t}(x)$ and Lemma 30, we can write

$$\begin{aligned} B_{J_t(x),t}(x) &= U_{j_t(x),t}(x) - L_{J_t(x),t}(x) \\ &\leq U_{j_t(x),t}(x) - L_{j_t(x),t}(x) = s_{a,t}(x). \end{aligned}$$

Similarly, for $a = J_t(x)$,

$$\begin{aligned} B_{J_t(x),t}(x) &= U_{j_t(x),t}(x) - L_{J_t(x),t}(x) \\ &\leq U_{J(x),t}(x) - L_{J_t(x),t}(x) = s_{a,t}(x). \end{aligned}$$

□

Lemma 31. *On event $\mathcal{E}$, for any time $t \leq T$ and for arm $a = a_t(x)$ the following bounds hold for the minimal gap*

$$B_{J_t(x),t}(x) \leq \min(0, s_{a,t}(x) - \Delta_a(x)) + s_{a,t}(x).$$

Proof. The arm to be pulled is restricted to $a \in \{j_t(x), J_t(x)\}$. The optimal arm for the context x at time t can either belong to $\{j_t(x), J_t(x)\}$ or be equal to some other arm. This results in 6 cases:

1. $a = j_t(x), a^* = j_t(x)$

2. $a = j_t(x), a^* = J_t(x)$

3. $a = j_t(x), a^* \notin \{j_t(x), J_t(x)\}$

4. $a = J_t(x), a^* = j_t(x)$

5. $a = J_t(x), a^* = J_t(x)$

6. $a = J_t(x), a^* \notin \{j_t(x), J_t(x)\}$

We define $f^*(x) := f_{a^*}(x)$ as the expected reward associated with the best arm and $f_{(a)}(x)$ as the expected reward of the a^{th} best arm.

Case 1: The following sequence of inequalities holds:

$$\begin{aligned} f_{(2)}(x) &\geq f_{J_t(x)}(x) \\ &\geq L_{J_t(x),t}(x) \\ &\geq L_{j_t(x),t}(x) \\ &\geq f_a(x) - s_{a,t}(x). \end{aligned}$$

The first inequality follows from the assumption that $a = a^* = j_t(x)$, the chosen and optimal arm has the highest upper confidence bound, and therefore, the expected reward of arm $J_t(x)$ can be at most that of the second best arm. The second inequality follows from event $\mathcal{E}$, the third inequality follows from 30. The last inequality follows from event $\mathcal{E}$. Using the above string of inequalities and the definition of $\Delta_a(x)$, we can write

$$s_{a,t} - (f_a(x) - f_{(2)}(x)) = s_{a,t} - \Delta_a(x) \geq 0.$$

The result holds for case 1 with the application of Corollary 2.

Case 2: $a = j_t(x), a^* = J_t(x)$. We can write

$$\begin{aligned}
B_{J_t(x),t}(x) &= U_{j_t(x),t}(x) - L_{J_t(x),t}(x) \\
&\leq f_{j_t(x)}(x) + s_{j_t(x),t}(x) \\
&- f_{J_t(x)}(x) + s_{J_t(x),t}(x) \\
&\leq f_a(x) - f^*(x) + 2s_{a,t}(x).
\end{aligned}$$

The first inequality follows from event $\mathcal{E}$ and the second inequality holds because the selected arm has a larger uncertainty. From the definition of $\Delta_a(x)$,

$$\begin{aligned}
B_{J_t(x),t}(x) &\leq 2s_{a,t}(x) - \Delta_a(x) \\
&\leq s_{a,t}(x) + \min(0, s_{a,t} - \Delta_a(x)).
\end{aligned}$$

Where the inequality follows from Corollary 2.

Case 3: $a = j_t(x), a^* \notin \{j_t(x), J_t(x)\}$. We can write the following sequence of inequalities

$$f_{j_t(x)}(x) + s_{j_t(x),t}(x) \geq U_{j_t(x),t}(x) \geq U_{a^*} \geq f^*.$$

The first and third inequalities hold due to event $\mathcal{E}$, the second inequality holds by definition as $j_t(x)$ has the highest upper bound on any arm other than $J_t(x)$ neither of which is the optimal arm in this case. From the first and last inequalities, we obtain

$$s_{a,t}(x) - (f^* - f_{a,t}(x)) \geq 0,$$

or $s_{a,t}(x) - \Delta_a(x) \geq 0$. The result follows from Corollary 2.

Case 4: $a = J_t(x), a^* = j_t(x)$. We can write

$$\begin{aligned} B_{J_t(x),t}(x) &= U_{j_t(x),t}(x) - L_{J_t(x),t}(x) \\ &\leq f_{j_t(x)}(x) + s_{j_t(x),t}(x) \\ &- f_{J_t(x)}(x) + s_{J_t(x),t}(x) \\ &\leq f_a(x) - f^*(x) + 2s_{a,t}(x). \end{aligned}$$

The first inequality follows from event $\mathcal{E}$ and the second inequality holds because the selected arm has a larger uncertainty. From the definition of $\Delta_a(x)$,

$$\begin{aligned} B_{J_t(x),t}(x) &\leq 2s_{a,t}(x) - \Delta_a(x) \\ &\leq s_{a,t}(x) + \min(0, s_{a,t} - \Delta_a(x)). \end{aligned}$$

Where the inequality follows from Corollary 2.

Case 5: $a = J_t(x), a^* = J_t(x)$. The following sequence of inequalities holds:

$$\begin{aligned} f_a(x) + s_{a,t}(x) &\geq U_{J_t(x),t}(x) \\ &\geq U_{j_t(x),t}(x) \\ &\geq f_{j_t(x)}(x) \\ &\geq f_{(2)}(x). \end{aligned}$$

The first and third inequalities follow from event $\mathcal{E}$, the second inequality is a consequence of Lemma 30, the fourth inequality follows from the fact that since $J_t(x)$ is the optimal arm, the upper bound and the arm selected should be as good as the second arm. Using the above chain of inequalities, we can write

$$s_{a,t}(x) - (f_{(2)}(x) - f_a(x)) = s_{a,t}(x) - \Delta_a(x) \geq 0.$$

Case 6: $a = J_t(x), a^* \notin \{j_t(x), J_t(x)\}$. We can write the following sequence of inequalities

$$f_{J_t(x)}(x) + s_{J_t(x),t}(x) \geq U_{J_t(x),t}(x) \geq U_{a^*,t}(x) \geq f^*.$$

The first and third inequalities hold due to event $\mathcal{E}$, the second inequality holds by definition as $J_t(x)$ has the highest upper bound on any arm when $a = J_t(x)$ due to Lemma 30 and $J_t(x)$ is not optimal in this case. From the first and last inequalities, we obtain

$$s_{a,t}(x) - (f^* - f_{a,t}(x)) \geq 0,$$

or $s_{a,t}(x) - \Delta_a(x) \geq 0$. The result follows from Corollary 2.

□

CHAPTER V

Conclusion and Future Work

In this thesis, I explored three challenges of limited labeled data and proposed novel kernel based solution to address these challenges.

As discussed in the introduction, in domain generalization setting, the learner is given unlabeled data to classify, and must do so by leveraging labeled data sets from similar yet distinct classification problems. In other words, label training data drawn from the same distribution as the test data are not available, but are available from several related tasks (which may have slightly different distribution).

In chapter 2, I provide an efficient way to solve an existing domain generalization approach and extend the analysis to multi-class classification. I also give empirical evidence based on two medical datasets and one satellite dataset demonstrating the superiority of proposed algorithms over state-of-the-art.

Deep learning extension to the algorithm proposed in chapter 2 could be useful for representation learning and embed feature engineering part into domain generalization algorithm. An idea is to learn domain/distribution specific embedding which is a good representation of that domain. Preliminary investigation suggests that one can learn these domain specific embeddings and can learn decision functions that generalize to new domains or tasks using deep neural network [103]. But like all deep learning methods, even this idea is data hungry. In future work, I intend to work on addressing the issue of limited labeled data in deep learning for domain generalization and giving generalization error analysis.

One can also extend the domain generalization idea in chapter 2 to semisupervised setup where one can learn more with unlabeled data. One such idea is presented in [104] without theoretical analysis. I intend to continue working on the idea of semisupervised learning and its theoretical analysis.

In Chapter 3, I discuss the problem of multi-task learning of contextual bandits. I propose an upper confidence bound-based multi-task learning algorithm for contextual bandits and establish a corresponding regret bound. I also describe an effective scheme for estimating task similarity from data and demonstrate my algorithm's performance using several data sets.

One can investigate this idea of multi-task learning through arm similarity to predict and learn from the rewards of arms that were not picked during any given trial. This would allow us to address limited feedback setting and may speed up the learning process. One can also extend the idea to learn representation of arms or actions using bandit feedback. One of the major limitation of the study presented in chapter 3 is that regret analysis assumes that task or arm similarity is given. A good future direction is to analyze the regret where arm similarity is learnt on the fly.

One more challenge that I faced during my Ph.D work and specifically chapter 3 is how to select hyperparameters (e.g. exploration parameter, kernel widths and regularization in contextual bandits) for bandit and reinforcement learning problems. We have some preliminary ideas and one such idea is described in the technical report Deshmukh et. al. [105]. More work is needed to standardize computationally efficient hyperparameter selection with theoretical guarantees.

In chapter 4, I formulate and present performance guarantees on the simple regret for contextual bandits in the fixed budget framework and present experimental results for adaptive sensor selection in nano-satellites. One future idea is to work on similar concept of simple regret but in the reinforcement learning setting. Idea is to give guarantees on identifying the best policy using pure exploration strategy in Reinforcement learning.

The approaches I proposed to solve the three challenges mentioned have theoretical guarantees, and their effectiveness is proven using empirical evidence. I plan to continue to use both theory and experiments to guide my work in the future.

BIBLIOGRAPHY

[1] R. Tsunoda, “Tilts and wave structure in the bottomside of the low-latitude f layer: Recent findings and future opportunities,” in *AGU Fall Meeting Abstracts*, 2016.

[2] N. England, J. W. Cutler, and S. Sharma, “Tandom beacon experiment-tbex design overview and lessons learned,” *Cubesat Developer’s Workshop*, 2018.

[3] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, *et al.*, “Deep speech 2: End-to-end speech recognition in english and mandarin,” in *International Conference on Machine Learning*, pp. 173–182, 2016.

[4] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning.,” 2017.

[5] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, *et al.*, “Google’s neural machine translation system: Bridging the gap between human and machine translation,” *arXiv preprint arXiv:1609.08144*, 2016.

[6] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, *et al.*, “Mastering the game of go without human knowledge,” *Nature*, vol. 550, no. 7676, p. 354, 2017.

[7] OpenAI, “Openai five.” https://blog.openai.com/openai-five/, 2018.

[8] J. Baxter, “A Bayesian/information theoretic model of learning to learn via multiple task sampling,” *Machine Learning*, vol. 28, no. 1, pp. 7–39, 1997.

[9] J. Baxter, “A model of inductive bias learning,” *Journal of Artificial Intelligence Research*, vol. 12, pp. 149–198, 2000.

[10] A. A. Deshmukh, U. Dogan, and C. Scott, “Multi-task learning for contextual bandits,” in *Advances in Neural Information Processing Systems*, pp. 4848–4856, 2017.

[11] R. Weber *et al.*, “On the gittins index for multiarmed bandits,” *The Annals of Applied Probability*, vol. 2, no. 4, pp. 1024–1033, 1992.

[12] S. Bubeck and N. Cesa-Bianchi, “Regret analysis of stochastic and nonstochastic multi-armed bandit problems,” *Machine Learning*, vol. 5, no. 1, pp. 1–122, 2012.

[13] H. Robbins, “Some aspects of the sequential design of experiments,” in *Herbert Robbins Selected Papers*, pp. 169–177, Springer, 1985.

[14] V. Kuleshov and D. Precup, “Algorithms for multi-armed bandit problems,” *arXiv preprint arXiv:1402.6028*, 2014.

[15] J. Langford and T. Zhang, “The epoch-greedy algorithm for multi-armed bandits with side information,” in *Advances in neural information processing systems*, pp. 817–824, 2008.

[16] P. Auer, “Using confidence bounds for exploitation-exploration trade-offs,” *Journal of Machine Learning Research*, vol. 3, no. Nov, pp. 397–422, 2002.

[17] S. Agrawal and N. Goyal, “Analysis of Thompson sampling for the multi-armed bandit problem,” in *Conference on Learning Theory*, pp. 39–1, 2012.

[18] Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári, "Improved algorithms for linear stochastic bandits," in *Advances in Neural Information Processing Systems*, pp. 2312–2320, 2011.

[19] W. Chu, L. Li, L. Reyzin, and R. Schapire, "Contextual bandits with linear payoff functions," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 208–214, 2011.

[20] S. Kale, L. Reyzin, and R. E. Schapire, "Non-stochastic bandit slate problems," in *Advances in Neural Information Processing Systems*, pp. 1054–1062, 2010.

[21] J. White, *Bandit algorithms for website optimization.* O'Reilly Media, Inc., 2012.

[22] S. S. Villar, J. Bowden, and J. Wason, "Multi-armed bandit models for the optimal design of clinical trials: benefits and challenges," *Statistical science: a review journal of the Institute of Mathematical Statistics*, vol. 30, no. 2, p. 199, 2015.

[23] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proceedings of the 19th international conference on World wide web*, pp. 661–670, ACM, 2010.

[24] M. Valko, N. Korda, R. Munos, I. Flaounas, and N. Cristianini, "Finite-time analysis of kernelised contextual bandits," in *Uncertainty in Artificial Intelligence*, p. 654, Citeseer, 2013.

[25] N. Srinivas, A. Krause, M. Seeger, and S. M. Kakade, "Gaussian process optimization in the bandit setting: No regret and experimental design," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 1015–1022, 2010.

[26] V. Gabillon, M. Ghavamzadeh, and A. Lazaric, "Best arm identification: A unified approach to fixed budget and fixed confidence," in *Advances in Neural Information Processing Systems*, pp. 3212–3220, 2012.

[27] K. Jamieson and R. Nowak, "Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting," in *Information Sciences and Systems (CISS), 2014 48th Annual Conference on*, pp. 1–6, IEEE, 2014.

[28] A. Garivier and E. Kaufmann, "Optimal best arm identification with fixed confidence," in *Conference on Learning Theory*, pp. 998–1027, 2016.

[29] A. Carpentier and M. Valko, "Simple regret for infinitely many armed bandits," in *International Conference on Machine Learning*, pp. 1133–1141, 2015.

[30] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, vol. 47, no. 2-3, pp. 235–256, 2002.

[31] J.-Y. Audibert and S. Bubeck, "Best arm identification in multi-armed bandits," in *COLT-23th Conference on Learning Theory-2010*, pp. 13–p, 2010.

[32] G. Blanchard, G. Lee, and C. Scott, "Generalizing from several related classification tasks to a new unlabeled sample," in *Advances in neural information processing systems*, pp. 2178–2186, 2011.

[33] T. Evgeniou, C. A. Michelli, and M. Pontil, "Learning multiple tasks with kernel methods," *J. Machine Learning Research*, pp. 615–637, 2005.

[34] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola, "A kernel approach to comparing distributions," in *Proceedings of the 22nd AAAI Conference on Artificial Intelligence* (R. Holte and A. Howe, eds.), pp. 1637–1641, 2007.

[35] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola, "A kernel method for the two-sample-problem," in *Advances in Neural Information Processing Systems 19* (B. Schölkopf, J. Platt, and T. Hoffman, eds.), pp. 513–520, 2007.

[36] B. Sriperumbudur, A. Gretton, K. Fukumizu, B. Schölkopf, and G. Lanckriet, "Hilbert space embeddings and metrics on probability measures," *Journal of Machine Learning Research*, vol. 11, pp. 1517–1561, 2010.

[37] A. Christmann and I. Steinwart, "Universal kernels on non-standard input spaces," in *Advances in Neural Information Processing Systems 23* (J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, eds.), pp. 406–414, 2010.

[38] I. Steinwart and A. Christmann, *Support Vector Machines.* Springer, 2008.

[39] T. Joachims, "Making large scale svm learning practical," tech. rep., Universität Dortmund, 1999.

[40] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[41] C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S. S. Keerthi, and S. Sundararajan, "A dual coordinate descent method for large-scale linear svm," in *Proceedings of the 25th international conference on Machine learning*, pp. 408–415, ACM, 2008.

[42] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *Journal of machine learning research*, vol. 9, no. Aug, pp. 1871–1874, 2008.

[43] C. Williams and M. Seeger, "Using the nyström method to speed up kernel machines," in *Proceedings of the 14th Annual Conference on Neural Information Processing Systems*, no. EPFL-CONF-161322, pp. 682–688, 2001.

[44] P. Drineas and M. W. Mahoney, "On the nyström method for approximating a gram matrix for improved kernel-based learning," *The Journal of Machine Learning Research*, vol. 6, pp. 2153–2175, 2005.

[45] A. Rahimi and B. Recht, “Random features for large-scale kernel machines,” in *Advances in neural information processing systems*, pp. 1177–1184, 2007.

[46] R. Kohavi *et al.*, “A study of cross-validation and bootstrap for accuracy estimation and model selection,” in *IJCAI*, vol. 14, pp. 1137–1145, 1995.

[47] A. Tsanas, M. A. Little, P. E. McSharry, and L. O. Ramig, “Accurate telemonitoring of parkinson’s disease progression by noninvasive speech tests,” *IEEE transactions on Biomedical Engineering*, vol. 57, no. 4, pp. 884–893, 2010.

[48] S. Sharma and J. W. Cutler, “Robust orbit determination and classification: A learning theoretic approach,” *Interplanetary Network Progress Report*, vol. 203, p. 1, 2015.

[49] J. Toedling, P. Rhein, R. Ratei, L. Karawajew, and R. Spang, “Automated in-silico detection of cell populations in flow cytometry readouts and its application to leukemia disease monitoring,” *BMC Bioinformatics*, vol. 7, p. 282, 2006.

[50] N. Aghaeepour, G. Finak, H. Hoos, T. R. Mosmann, R. Brinkman, R. Gottardo, R. H. Scheuermann, F. Consortium, D. Consortium, *et al.*, “Critical assessment of automated flow cytometry data analysis techniques,” *Nature methods*, vol. 10, no. 3, pp. 228–238, 2013.

[51] A. Maurer, “A vector-contraction inequality for rademacher complexities,” *Algorithmic Learning Theory: 27th International Conference, ALT 2016, Bari, Italy, October 19-21, 2016, Proceedings*, pp. 3–17, 2016.

[52] G. Blanchard, G. Lee, and C. Scott, “Generalizing from several related classification tasks to a new unlabeled sample,” in *Advances in Neural Information Processing Systems 24* (J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, eds.), pp. 2178–2186, 2011.

[53] B. London, B. Huang, B. Taskar, and L. Getoor, “Collective stability in structured prediction: Generalization from one example,” in *International Conference on Machine Learning*, pp. 828–836, 2013.

[54] C. Cortes, V. Kuznetsov, M. Mohri, and S. Yang, “Structured prediction theory based on factor graph complexity,” in *Advances in Neural Information Processing Systems 29* (D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, eds.), pp. 2514–2522, Curran Associates, Inc., 2016.

[55] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, “A public domain dataset for human activity recognition using smartphones.,” in *21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2013.

[56] A. Rahimi and B. Recht, “Random features for large-scale kernel machines,” in *Advances in neural information processing systems*, pp. 1177–1184, 2008.

[57] I. Steinwart and A. Christmann, *Support vector machines.* Springer Science & Business Media, 2008.

[58] P. L. Bartlett and S. Mendelson, “Rademacher and gaussian complexities: Risk bounds and structural results,” *Journal of Machine Learning Research*, vol. 3, no. Nov, pp. 463–482, 2002.

[59] C. Scott, “Rademacher complexity.” http://web.eecs.umich.edu/~cscott/past_courses/eecs598w14/notes/10_rademacher.pdf.

[60] N. Cesa-Bianchi, C. Gentile, and G. Zappella, “A gang of bandits,” in *Advances in Neural Information Processing Systems*, pp. 737–745, 2013.

[61] S. Li, A. Karatzoglou, and C. Gentile, “Collaborative filtering bandits,” in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in*

Information Retrieval, pp. 539–548, ACM, 2016.

[62] A. Krause and C. S. Ong, “Contextual gaussian process bandit optimization,” in *Advances in Neural Information Processing Systems*, pp. 2447–2455, 2011.

[63] T. Evgeniou and M. Pontil, “Regularized multi–task learning,” in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 109–117, ACM, 2004.

[64] A. Christmann and I. Steinwart, “Universal kernels on non-standard input spaces,” in *Advances in neural information processing systems*, pp. 406–414, 2010.

[65] M. Lin, “Reversed determinantal inequalities for accretive-dissipative matrices,” *Math. Inequal. Appl*, vol. 12, pp. 955–958, 2012.

[66] B. Rajarama Bhat, A. Chattopadhyay, and G. S. R. Kosuru, “On submajorization and eigenvalue inequalities,” *Linear and Multilinear Algebra*, vol. 63, no. 11, pp. 2245–2253, 2015.

[67] R. A. Horn and C. R. Johnson, *Matrix analysis.* Cambridge university press, 2012.

[68] Y. Zi-Zong, “Schur complements and determinant inequalities,” *Journal of Mathematical Inequalities*, 2009.

[69] K. Azuma, “Weighted sums of certain dependent random variables,” *Tohoku Mathematical Journal, Second Series*, vol. 19, no. 3, pp. 357–367, 1967.

[70] A. W. Marshall, I. Olkin, and B. C. Arnold, *Inequalities: theory of majorization and its applications*, vol. 143. Springer.

[71] R. Bapat and V. Sunder, “On majorization and schur products,” *Linear algebra and its applications*, vol. 72, pp. 107–117, 1985.

[72] J. M. Steele, *The Cauchy-Schwarz master class: an introduction to the art of mathematical inequalities.* Cambridge University Press, 2004.

[73] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, “The nonstochastic multiarmed bandit problem,” *SIAM journal on computing*, vol. 32, no. 1, pp. 48–77, 2002.

[74] M. Y. Guan and H. Jiang, “Nonparametric stochastic contextual bandits,” in *The 32nd AAAI Conference on Artificial Intelligence*, 2018.

[75] C. Tekin and M. van der Schaar, “Releaf: An algorithm for learning and exploiting relevance,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 716–727, 2015.

[76] M. Hoffman, B. Shahriari, and N. Freitas, “On correlation and budget constraints in model-based bandit optimization with application to automatic machine learning,” in *Artificial Intelligence and Statistics*, pp. 365–374, 2014.

[77] M. Soare, A. Lazaric, and R. Munos, “Best-arm identification in linear bandits,” in *Advances in Neural Information Processing Systems*, pp. 828–836, 2014.

[78] P. Libin, T. Verstraeten, D. M. Roijers, J. Grujic, K. Theys, P. Lemey, and A. Nowé, “Bayesian best-arm identification for selecting influenza mitigation strategies,” *arXiv preprint arXiv:1711.06299*, 2017.

[79] L. Xu, J. Honda, and M. Sugiyama, “A fully adaptive algorithm for pure exploration in linear bandits,” in *International Conference on Artificial Intelligence and Statistics*, pp. 843–851, 2018.

[80] A. Sheinker and M. B. Moldwin, “Adaptive interference cancelation using a pair of magnetometers,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 1, pp. 307–318, 2016.

[81] E. L. Kepko, K. K. Khurana, M. G. Kivelson, R. C. Elphic, and C. T. Russell, "Accurate determination of magnetic field gradients from four point vector measurements. I. use of natural constraints on vector data obtained from a single spinning spacecraft," *IEEE Transactions on Magnetics*, vol. 32, no. 2, pp. 377–385, 1996.

[82] H. K. Leinweber, *In-flight calibration of space-borne magnetometers*. PhD thesis, Graz University of Technology, 2012.

[83] J. C. Springmann and J. W. Cutler, "Attitude-independent magnetometer calibration with time-varying bias," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 4, pp. 1080–1088, 2012.

[84] A. O. Hero and D. Cochran, "Sensor management: Past, present, and future," *IEEE Sensors Journal*, vol. 11, no. 12, pp. 3064–3075, 2011.

[85] D. A. Castanon, "Approximate dynamic programming for sensor management," in *Decision and Control, 1997., Proceedings of the 36th IEEE Conference on*, vol. 2, pp. 1202–1207, IEEE, 1997.

[86] J. Evans and V. Krishnamurthy, "Optimal sensor scheduling for hidden Markov model state estimation," *International Journal of Control*, vol. 74, no. 18, pp. 1737–1742, 2001.

[87] V. Krishnamurthy, "Algorithms for optimal scheduling and management of hidden Markov model sensors," *IEEE Transactions on Signal Processing*, vol. 50, no. 6, pp. 1382–1397, 2002.

[88] E. K. Chong, C. M. Kreucher, and A. O. Hero, "Partially observable Markov decision process approximations for adaptive sensing," *Discrete Event Dynamic Systems*, vol. 19, no. 3, pp. 377–422, 2009.

[89] A. Mahajan and D. Teneketzis, “Multi-armed bandit problems,” in *Foundations and Applications of Sensor Management*, pp. 121–151, Springer, 2008.

[90] A. Durand, O.-A. Maillard, and J. Pineau, “Streaming kernel regression with provably adaptive mean, variance, and regularization,” *Journal of Machine Learning Research*, vol. 19, no. August, 2018.

[91] C. Gentile, S. Li, and G. Zappella, “Online clustering of bandits,” in *International Conference on Machine Learning*, pp. 757–765, 2014.

[92] S. Li and S. Zhang, “Online clustering of contextual cascading bandits,” in *The 32nd AAAI Conference on Artificial Intelligence*, 2018.

[93] S. Tu and B. Recht, “Least-squares temporal difference learning for the linear quadratic regulator,” *arXiv preprint arXiv:1712.08642*, 2017.

[94] S. R. Chowdhury and A. Gopalan, “On kernelized multi-armed bandits,” in *International Conference on Machine Learning*, pp. 844–853, 2017.

[95] S. Sharma, *Machine Learning Applications in Spacecraft State and Environment Estimation.* PhD thesis, University of Michigan Ann Arbor, 2018.

[96] A. A. Deshmukh, S. Sharma, J. W. Cutler, M. Moldwin, and C. Scott, “Simple regret minimization for contextual bandits,” *arXiv preprint arXiv:1810.07371*, 2018.

[97] C. D. Norton, M. P. Pasciuto, P. Pingree, S. Chien, and D. Rider, “Spaceborne flight validation of NASA ESTO technologies,” in *Geoscience and Remote Sensing Symposium (IGARSS), 2012 IEEE International*, pp. 5650–5653, IEEE, 2012.

[98] J. W. Cutler, C. Lacy, T. Rose, S.-h. Kang, D. Rider, and C. Norton, “An update on the GRIFEX mission,” *Cubesat Developer’s Workshop*, 2015.

[99] R. Motwani and P. Raghavan, *Tail Inequalities*, p. 67–100. Cambridge University Press, 1995.

[100] H. Bercovici, W. Li, and D. Timotin, "The horn conjecture for sums of compact self-adjoint operators," *American Journal of Mathematics*, vol. 131, no. 6, pp. 1543–1567, 2009.

[101] S. Minsker, "On some extensions of Bernstein's inequality for self-adjoint operators," *Statistics & Probability Letters*, vol. 127, no. C, pp. 111–119, 2017.

[102] C. A. Micchelli, Y. Xu, and H. Zhang, "Universal kernels," *Journal of Machine Learning Research*, vol. 7, no. Dec, pp. 2651–2667, 2006.

[103] A. A. Deshmukh, A. Bansal, and A. Rastogi, "Domain2vec: Deep domain generalization," *arXiv preprint arXiv:1807.02919*, 2018.

[104] A. A. Deshmukh and E. Laftchiev, "Semi-supervised transfer learning using marginal predictors," in *2018 IEEE Data Science Workshop (DSW)*, pp. 160–164, IEEE, 2018.

[105] A. A. Deshmukh, F. Wei, and C. Scott, "Hyperparameter selection for multi-armed bandit problems," *The 3rd Annual Michigan Institute for Data Science Symposium*, 2017.