
AFRL-RI-RS-TR-2022-026

# DATA-EFFICIENT ACTIVE MACHINE LEARNING

UNIVERSITY OF WISCONSIN-MADISON

*FEBRUARY 2022*

FINAL TECHNICAL REPORT

***APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED***

STINFO COPY

**AIR FORCE RESEARCH LABORATORY**
**INFORMATION DIRECTORATE**

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

# NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09. This report is available to the general public, including foreign nations. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2022-026 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

**/ S /**
AMY G. WONG
Work Unit Manager

**/ S /**
JULIE BRICHACEK
Chief, Information Systems Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

# REPORT DOCUMENTATION PAGE

| 1. REPORT DATE | 2. REPORT TYPE | 3. DATES COVERED | |
|---|---|---|---|
| | | START DATE | END DATE |
| FEBRUARY 2022 | FINAL TECHNICAL REPORT | AUGUST 2017 | AUGUST 2021 |

**4. TITLE AND SUBTITLE**

DATA-EFFICIENT ACTIVE MACHINE LEARNING

| 5a. CONTRACT NUMBER | 5b. GRANT NUMBER | 5c. PROGRAM ELEMENT NUMBER |
|---|---|---|
| FA8750-17-2-0262 | N/A | 62788F |
| **5d. PROJECT NUMBER** | **5e. TASK NUMBER** | **5f. WORK UNIT NUMBER** |
| S2DL | UO | WM |

**6. AUTHOR(S)**

Robert Nowak

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

University of Wisconsin-Madison
1415 Engineering Drive
Madison WI 53715-1218

**8. PERFORMING ORGANIZATION REPORT NUMBER**

| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) | 10. SPONSOR/MONITOR'S ACRONYM(S) | 11. SPONSOR/MONITOR'S REPORT NUMBER(S) |
|---|---|---|
| Air Force Research Laboratory/RISC<br>525 Brooks Road<br>Rome NY 13441-4505 | RI | AFRL-RI-RS-TR-2022-026 |

**12. DISTRIBUTION/AVAILABILITY STATEMENT**

Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09.

**13. SUPPLEMENTARY NOTES**

**14. ABSTRACT**

Data-efficient machine learning (DEML) is critical to AF/DoD operations for the following reasons. First, training machine learning algorithms generally requires a large and completely labeled training dataset. Human labeling of raw data is an expensive and time-consuming process, especially with a limited pool of expert analysts. Therefore, machine learning algorithms must produce accurate predictive models from limited labeled training data. Moreover, mission environments and objectives can be varied and rapidly changing, and so machine learning models must be quickly adaptable to the situation at hand. The quality of the raw data available to a machine learning system (and to human analysts) is also often unpredictable. It may often happen that not all of the desired features for making predictions and decisions are available. Therefore, machine learning algorithms must be robust to missing or partially unobserved data.

**15. SUBJECT TERMS**

Machine Learning, Data-efficient machine learning (DEML), Support Vector Machine (SVM)

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES |
|---|---|---|---|---|
| a. REPORT | b. ABSTRACT | C. THIS PAGE | | |
| U | U | U | SAR | 41 |

| 19a. NAME OF RESPONSIBLE PERSON | 19b. PHONE NUMBER *(Include area code)* |
|---|---|
| AMY G. WONG | N/A |



PREVIOUS EDITION IS OBSOLETE.

## TABLE OF CONTENTS

## 1.0 SUMMARY

Data-efficient machine learning (DEML) is critical to AF/DoD operations for the following reasons. First, training machine learning algorithms generally requires a large and completely labeled training dataset. Human labeling of raw data is an expensive and time-consuming process, especially with a limited pool of expert analysts. Therefore, *machine learning algorithms must produce accurate predictive models from limited labeled training data.* Moreover, mission environments and objectives can be varied and rapidly changing, and so *machine learning models must be quickly adaptable to the situation at hand.* The quality of the raw data available to a machine learning system (and to human analysts) is also often unpredictable. It may often happen that not all of the desired features for making predictions and decisions are available. Therefore, *machine learning algorithms must be robust to missing or partially unobserved data.* The objective of this effort is to develop new tools for DEML that address these challenges.

## 2.0 INTRODUCTION

The scope of this effort is to create new tools for DEML in the following key areas: 1) develop data-efficient active learning algorithms for classification and search problems involving rich and high-dimensional feature spaces; 2) develop new interactive tools that allow human analysts to quickly and accurately label large datasets; 3) develop a new framework for enriched human annotation, where explanations and feature relevance feedback are provided in addition to labels; 4) prototype algorithms in software. These goals will require basic mathematical research and analysis of DEML problems, algorithmic development and prototyping, and testing and experimentation with real and synthetic datasets.

## 3.0 METHODS, ASSUMPTIONS, AND PROCEDURES

State-of-the-art ML methods can be extremely powerful, given a sufficiently large amount of labeled training data. However, in many applications it is prohibitive, costly, or simply impossible (e.g., due to time constraints) to generate a large training dataset. For example, the best image classification algorithms are trained on the Imagenet dataset http://www.image-net.org, which consists of over 10 million hand-annotated images that required tens of thousands of hours of human work, if not more. Enormous datasets like this are key to the success state-of-the-art methods, but the very same methods can perform arbitrarily poorly if trained with a relatively small dataset. There is a great need for new mathematical approaches to the design of ML algorithms that can provide good performance from small training datasets. Addressing this challenge is the focus of our effort.

End Results and Deliverables. Design DEML algorithms. Mathematical analyze and assess performance of DEML algorithms. Prototype DEML algorithm in software.

Software Requirements. Engineer software prototypes in open-source software systems, Python and NEXT (nextml.org).

Testing and Demonstrations. Test software and algorithms with real and synthetic datasets. Conduct a demonstration of the NEXT in order to verify the results of this effort.

## 4.0 RESULTS AND DISCUSSION

The project produces the following results; more detailed technical descriptions appear in the referenced papers. The papers included in the Appendix to this report.

1. Developed data-efficient active learning algorithms for classification and search problems involving rich and high-dimensional feature spaces. Generating labeled training datasets has become a major bottleneck in Machine Learning (ML) pipelines. Active ML aims to address this issue by designing learning algorithms that automatically and adaptively select the most informative examples for labeling so that human time is not wasted labeling irrelevant, redundant, or trivial examples. This project developed a new approach to active ML with nonparametric or overparameterized models such as kernel methods and neural networks. In the context of binary classification, the new approach is shown to possess a variety of desirable properties that allow active learning algorithms to automatically and efficiently identify decision boundaries and data clusters. This work was published in the papers

Karzand, Mina, and Robert D. Nowak. "Maximin active learning in overparameterized model classes." *IEEE Journal on Selected Areas in Information Theory* 1, no. 1 (2020): 167-177.

Karzand, Mina, and Robert D. Nowak. "Maximin Active Learning with Data-Dependent Norms." In *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 871-878. IEEE, 2019.

2. Developed new interactive tools that allow human analysts to quickly and accurately label large datasets. This work focused on best-arm identification in multi-armed bandits with bounded rewards. We developed an algorithm that is a fusion of lil-UCB and KL-LUCB, offering the best qualities of the two algorithms in one method. This is achieved by proving a novel anytime confidence bound for the mean of bounded distributions, which is the analogue of the LIL-type bounds recently developed for sub-Gaussian distributions. We corroborated our theoretical results with numerical experiments based on real data from human-machine interactions. This work was reported in the publication

Tanczos, Ervin, Robert Nowak, and Bob Mankoff. "A KL-LUCB bandit algorithm for large-scale crowdsourcing." In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 5896-5905. 2017.

3. Developed a new framework for enriched human annotation, where explanations and feature relevance feedback are provided in addition to labels. This work developed a new form of the linear bandit problem in which the algorithm receives the usual stochastic rewards as well as stochastic feedback about which features are relevant to the rewards, the latter feedback being the novel aspect. The focus was the development of new theory and algorithms for linear bandits with feature feedback. We show that linear bandits with feature feedback can achieve regret over time horizon T that scales like k sqrt(T), without prior knowledge of which features are relevant nor the number k of relevant features. In comparison, the regret of traditional linear bandits is d sqrt(T), where d is the total number of (relevant and irrelevant) features, so the improvement can be dramatic if $k \ll d$. The computational complexity of the new algorithm is proportional to k rather than d, making it much more suitable for real-world applications compared to traditional linear

bandits. We demonstrate the performance of the new algorithm with synthetic and real human-labeled data. This work was reported in the paper

Oswal, U., Bhargava, A., & Nowak, R. (2020, April). Linear bandits with feature feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 34, No. 04, pp. 5331-5338).

4. Extended the capabilities of the NEXT system for interactive learning (nextml.org) to incorporate the new tools and algorithms proposed above. In addition, a new system called Salmon (https://docs.stsievert.com/salmon/) was developed that allows for streamlined, large-scale interactive learning applications.

5. We investigated interactive learning in the realizable setting and developed a general framework to handle problems ranging from best arm identification to active classification. We begin our investigation with the observation that agnostic algorithms cannot be minimax-optimal in the realizable setting. Hence, we design novel computationally efficient algorithms for the realizable setting that match the minimax lower bound up to logarithmic factors and are general-purpose, accommodating a wide variety of function classes including kernel methods, H{ö}lder smooth functions, and convex functions. The sample complexities of our algorithms can be quantified in terms of well-known quantities like the extended teaching dimension and haystack dimension. However, unlike algorithms based directly on those combinatorial quantities, our algorithms are computationally efficient. To achieve computational efficiency, our algorithms sample from the version space using Monte Carlo "hit-and-run" algorithms instead of maintaining the version space explicitly. Our approach has two key strengths. First, it is simple, consisting of two unifying, greedy algorithms. Second, our algorithms have the capability to seamlessly leverage prior knowledge that is often available and useful in practice. In addition to our new theoretical results, we demonstrate empirically that our algorithms are competitive with Gaussian process UCB methods. This work was published in the paper

Katz-Samuels, Julian, Blake Mason, Kevin G. Jamieson, and Rob Nowak. "Practical, Provably-Correct Interactive Learning in the Realizable Setting: The Power of True Believers." *Advances in Neural Information Processing Systems* 34 (2021).

6. We considered the problem of learning the nearest neighbor graph of a dataset of n items. The metric is unknown, but we can query an oracle to obtain a noisy estimate of the distance between any pair of items. This framework applies to problem domains where one wants to learn people's preferences from responses commonly modeled as noisy distance judgments. In this paper, we propose an active algorithm to find the graph with high probability and analyze its query complexity. In contrast to existing work that forces Euclidean structure, our method is valid for general metrics, assuming only symmetry and the triangle inequality. Furthermore, we demonstrate efficiency of our method empirically and theoretically, needing only O(n log(n)Delta^-2) queries in favorable settings, where Delta^-2 accounts for the effect of noise. Using crowd-sourced data collected for a subset of the UT Zappos50K dataset, we apply our algorithm to learn which shoes people believe are most similar and show that it beats both an active baseline and ordinal embedding. This work was published in the paper Mason, Blake, Lalit Jain, and Robert Nowak. "Learning Nearest Neighbor Graphs from Noisy Distance Samples." *Advances in neural information processing systems* 1, no. 1 (2019).

7. We studied the problem of adaptively sampling from K distributions (arms) in order to identify the largest gap between any two adjacent means. We call this the MaxGap-bandit problem. This problem arises naturally in approximate ranking, noisy sorting, outlier detection, and top-arm identification in bandits. The key novelty of the MaxGap-bandit problem is that it aims to adaptively determine the natural partitioning of the distributions into a subset with larger means and a subset with smaller means, where the split is determined by the largest gap rather than a pre-specified rank or threshold. Estimating an arm's gap requires sampling its neighboring arms in addition to itself, and this dependence results in a novel hardness parameter that characterizes the sample complexity of the problem. We propose elimination and UCB-style algorithms and show that they are minimax optimal. Our experiments show that the UCB-style algorithms require 6-8x fewer samples than non-adaptive sampling to achieve the same error. These results were published in the paper

Katariya, Sumeet, Ardhendu Tripathy, and Robert Nowak. "MaxGap Bandit: Adaptive Algorithms for Approximate Ranking." *Advances in Neural Information Processing Systems* 32 (2019): 11047-11057.

8. We developed new concentration inequalities for the Kullback-Leibler (KL) divergence between the empirical distribution and the true distribution. Applying a recursion technique, we improve over the method of types bound uniformly in all regimes of sample size n and alphabet size k, and the improvement becomes more significant when k is large. We discuss the applications of our results in obtaining tighter concentration inequalities for L1 deviations of the empirical distribution from the true distribution, and the difference between concentration around the expectation or zero. We also obtain asymptotically tight bounds on the variance of the KL divergence between the empirical and true distribution, and demonstrate their quantitatively different behaviors between small and large sample sizes compared to the alphabet size. This work was published in the paper

Mardia, Jay, Jiantao Jiao, Ervin Tánczos, Robert D. Nowak, and Tsachy Weissman. "Concentration inequalities for the empirical distribution of discrete distributions: beyond the method of types." *Information and Inference: A Journal of the IMA* 9, no. 4 (2020): 813-850.

9. Visual representations are prevalent in DoD applications and STEM instruction. To benefit from visuals, students need representational competencies that enable them to see meaningful information. Most research has focused on explicit conceptual representational competencies, but implicit perceptual competencies might also allow students to efficiently see meaningful information in visuals. Most common methods to assess students' representational competencies rely on verbal explanations or assume explicit attention. However, because perceptual competencies are implicit and not necessarily verbally accessible, these methods are ill-equipped to assess them. We address these shortcomings with a method that draws on similarity learning, a machine learning technique that detects visual features that account for participants' responses to triplet comparisons of visuals. In Experiment 1, 614 chemistry students judged the similarity of Lewis structures and in Experiment 2, 489 students judged the similarity of ball-and-stick

models. Our results showed that our method can detect visual features that drive students' perception and suggested that students' conceptual knowledge about molecules informed perceptual competencies through top-down processes. Furthermore, Experiment 2 tested whether we can improve the efficiency of the method with active sampling. Results showed that random sampling yielded higher accuracy than active sampling for small sample sizes. Together, the experiments provide the first method to assess students' perceptual competencies implicitly, without requiring verbalization or assuming explicit visual attention. These findings have implications for the design of instructional interventions that help students acquire perceptual representational competencies. This work was published in the paper

Mason, Blake, Martina A. Rau, and Robert Nowak. "Cognitive Task Analysis for Implicit Knowledge About Visual Representations With Similarity Learning Methods." *Cognitive science* 43, no. 9 (2019): e12744.

10. Trained a graduate student, one undergraduate students, and four postdoctoral researchers in advanced AF/DoD-relevant machine learning.

## 5.0 CONCULSIONS

The research results and findings in this project demonstrate the potential of large-scale interactive machine learning algorithms and systems for AF/DoD applications. The question of designing active learning algorithms in the regime of nonparametric and overparameterized models become more essential as we look at larger models which require bigger training sets. To reduce the human cost of labeling all samples, we can use a pool-based active learning algorithm to avoid labeling non-informative examples. An important direction for future work is generalization of the algorithms developed in this project to multi-class settings and regression problems. The computational complexity of some of the methods can also be a serious bottleneck in applications with bigger datasets and should be addressed in future.

## 6.0 REFERENCES


Karzand, Mina, and Robert D. Nowak. "Maximin active learning in overparameterized model classes." *IEEE Journal on Selected Areas in Information Theory* 1, no. 1 (2020): 167-177.

Oswal, U., Bhargava, A., & Nowak, R. (2020, April). Linear bandits with feature feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 34, No. 04, pp. 5331-5338).

Katz-Samuels, Julian, Blake Mason, Kevin G. Jamieson, and Rob Nowak. "Practical, Provably-Correct Interactive Learning in the Realizable Setting: The Power of True Believers." *Advances in Neural Information Processing Systems* 34 (2021).

Mason, Blake, Lalit Jain, and Robert Nowak. "Learning Nearest Neighbor Graphs from Noisy Distance Samples." *Advances in neural information processing systems* 1, no. 1 (2019).

Katariya, Sumeet, Ardhendu Tripathy, and Robert Nowak. "MaxGap Bandit: Adaptive Algorithms for Approximate Ranking." *Advances in Neural Information Processing Systems* 32 (2019): 11047-11057.

Mardia, Jay, Jiantao Jiao, Ervin Tánczos, Robert D. Nowak, and Tsachy Weissman. "Concentration inequalities for the empirical distribution of discrete distributions: beyond the method of types." *Information and Inference: A Journal of the IMA* 9, no. 4 (2020): 813-850.

Tanczos, Ervin, Robert Nowak, and Bob Mankoff. "A KL-LUCB bandit algorithm for large-scale crowdsourcing." In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 5896-5905. 2017.

Karzand, Mina, and Robert D. Nowak. "Maximin Active Learning with Data-Dependent Norms." In *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 871-878. IEEE, 2019.

Mason, Blake, Martina A. Rau, and Robert Nowak. "Cognitive Task Analysis for Implicit Knowledge About Visual Representations With Similarity Learning Methods." *Cognitive science* 43, no. 9 (2019): e12744.

## APPENDIX – PUBLICATIONS

# MaxiMin Active Learning in Overparameterized Model Classes

Mina Karzand and Robert D. Nowak

***Abstract*—Generating labeled training datasets has become a major bottleneck in Machine Learning (ML) pipelines. Active ML aims to address this issue by designing learning algorithms that automatically and adaptively select the most informative examples for labeling so that human time is not wasted labeling irrelevant, redundant, or trivial examples. This paper proposes a new approach to active ML with nonparametric or overparameterized models such as kernel methods and neural networks. In the context of binary classification, the new approach is shown to possess a variety of desirable properties that allow active learning algorithms to automatically and efficiently identify decision boundaries and data clusters.**

***Index Terms*—Active learning, overparameterized learning, neural networks, reproducing Kernel Hilbert spaces, pool-based learning.**

### I. Introduction

THE FIELD of Machine Learning (ML) has advanced considerably in recent years, but mostly in well-defined domains using huge amounts of human-labeled training data. Machines can recognize objects in images and translate text, but they must be trained with more images and text than a person can see in nearly a lifetime. The computational complexity of training has been offset by recent technological advances, but the cost of training data is measured in terms of the human effort in labeling data. People are not getting faster nor cheaper, so generating labeled training datasets has become a major bottleneck in ML pipelines. Active ML aims to address this issue by designing learning algorithms that automatically and adaptively select the most informative examples for labeling so that human time is not wasted labeling irrelevant, redundant, or trivial examples. This paper explores active ML with nonparametric or overparameterized models such as kernel methods and neural networks.

Deep neural networks (DNNs) have revolutionized machine learning applications, and theoreticians have struggled to explain their surpising properties. DNNs are highly overparameterized and often fit perfectly to data, yet remarkably the learned models generalize well to new data. A mathematical understanding of this phenomenom is beginning to emerge [1], [2], [3], [4], [5], [6], [7], [8]. This work suggests that among all the networks that could be fit to the training data, the learning algorithms used in fitting favor networks with smaller weights, providing a sort of implicit regularization. With this in mind, researchers have shown that shallow (but wide) networks and classical kernel methods fit to the data but regularized to have small weights (e.g., minimum norm fit to data) can generalize well [2], [8], [9], [10].

Despite the recent success and new understanding of these systems, it still is a fact that learning good neural network models can require an enormous number of labeled data. The cost of obtaining labels can be prohibitive in many applications. This has prompted researchers to investigate active ML for kernel methods and neural networks [11], [12], [13], [14], [15], [16]. None of this work, however, directly addresses overparameterized and interpolating regime, which is the focus in this paper. Active ML algorithms have access to a large but unlabeled dataset of examples and sequentially select the most "informative" examples for labeling [17], [18] . This can reduce the total number of labeled examples needed to learn an accurate model.

Broadly speaking, active ML algorithms adaptively select examples for labeling based on two general strategies [19]. The first is to select examples that rule-out as many (incompatible) classifiers as possible at each step. In effect, this leads to algorithms that tend to label examples near decision boundaries. The second strategy involves discovering cluster structure in unlabeled data and labeling representative examples from each cluster. We show that our new MaxiMin active learning approach automatically exploits both these strategies, as depicted in Figure 1.

This paper builds on a new framework for active learning in the overparameterized and interpolationg regime, focusing on kernel methods and two-layer neural networks in the binary classification setting. The approach, called *MaxiMin Active Learning*, is based on mininum norm interpolating models. Roughly speaking, at each step of the learning process the maximin criterion requests a label for the example that is most difficult to interpolate. A minimum norm interpolating model is constructed for each possible example and the one yielding the largest norm indicates which example to label next. The rationale for the maximin criterion is that labeling the most challenging examples first may eliminate the need to label many of the other examples.

Manuscript received October 16, 2019; revised April 23, 2020; accepted April 26, 2020. Date of publication April 30, 2020; date of current version June 8, 2020. This work was supported in part by the Air Force Machine Learning Center of Excellence under Grant FA9550-18-1-0166. Parts of this article were presented at the 57th Annual Allerton Conference on Communication, Control, and Computing, 2019. *(Corresponding author: Mina Karzand.)*

Mina Karzand is with the Wisconsin Institute of Discovery, University of Wisconsin–Madison, Madison, WI 53706 USA (e-mail: karzand@wisc.edu).

Robert D. Nowak is with the Department of Electrical and Computer Engineering, University of Wisconsin–Madison, Madison, WI 53706 USA (e-mail: rdnowak@wisc.edu).

Digital Object Identifier 10.1109/JSAIT.2020.2991518

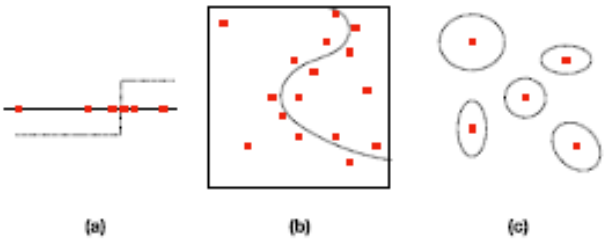



Fig. 1. MaxiMin Active Learning strategically selects examples for labeling (red points). (a) reduces to binary search in simple 1-d threshold problem setting; (b) labeling is focused near decision boundary in multidimensional setting; (c) automatically discovers clusters and labels representative examples from each.

The maximin selection criterion is studied through experiments and mathematical analysis. We prove that the criterion has a number of desirable properties:

- It tends to label examples near the current (estimated) decision boundary and close to oppositely labeled examples, allowing the active learning algorithm to focus on learning decision boundaries.
- It reduces to optimal bisection in the one-dimensional linear classifier setting.
- A data-based form of the criterion also provably discovers clusters and also automatically generates labeled coverings of the dataset.

Experimentally, we show that these properties generalize in several ways. For example, we find that in multiple dimensions the maximin criterion leads to a multidimensional bisection-like process that automatically finds a portion of the decision boundary and then locally explores to efficiently identify the complete boundary. We also show that MaxiMin Active Learning can learn hand-written digit classifiers with far fewer labeled examples than traditional passive learning based on labeling a randomly selected subset of examples.

## II. A New Active Learning Criterion

At each iteration of the active learning algorithm, looking at the currently labeled set of samples, a new unlabeled point is selected to be labeled. The criterion we are proposing to pick the samples to be labeled is based on a 'maximin' operator. We will describe the criterion in its most general form along with the intuition behind this choice of criterion. In the remainder of the paper, we will go through some theoretical results about the properties of variations of this criterion in various setups along with some additional descriptive numerical evaluations and simulations.

### A. Nonparametric Pool-Based Active Learning

At each time step, the algorithm has access to a pool of labeled samples and a set of unlabaled samples. In other words, we have a partially labeled training set. Let $\mathcal{L} = \{(x_1, y_1), \ldots, (x_L, y_L)\}$ be the set of labeled examples so far. We assume $x_i \in \mathcal{X}$ where $\mathcal{X}$ is the input/feature space and binary valued labels $y_i \in \{-1, +1\}$. Let $\mathcal{U} \subseteq \mathcal{X}$ be the set of unlabeled samples.

*In the interpolating regime, the goal is to correctly label all the points in $\mathcal{U}$ so that the training error is zero:* Passive learning generally requires labeling every point in $\mathcal{U}$. Active learning sequentially selects points in $\mathcal{U}$ for labeling with the aim of learning a correct classifier without necessarily labeling all of $\mathcal{U}$. Our setting can be viewed as an instance of *pool-based* active learning.

At each iteration, one unlabeled sample, $u^* \in \mathcal{U}$ is selected, labeled and added to the pool of labeled samples. The selection process is designed to pick the samples which are most *informative* upon being labeled. The proposed notion of score is the measure of informativeness of each sample $u \in \mathcal{U}$ at each time: the score of each unlabaled sample is computed, and the sample with the largest score is selected to be labeled.

$$u^* = \text{argmax}_{u \in \mathcal{U}} \textsf{score}(u). \tag{1}$$

If there are multiple maximizers, then one is selected uniformly at random. Note that for any unlabeled sample $u \in \mathcal{U}$, the value of $\textsf{score}(u)$ depends implicitly on the set of currently labeled points, $\mathcal{L}$. That is, information gained by labeling $u$ depends on the current knowledge of the learner. To define our proposed notion of score, we define minimum norm interpolating function and introduce some notations next.

### B. Minimum Norm Interpolating Function

Let $\mathcal{F}$ be a class of functions mapping $\mathcal{X}$ to $\mathbb{R}$, where $\mathcal{X}$ is the input/feature space,. We assume the class $\mathcal{F}$ is rich enough to interpolate the training data. For example, $\mathcal{F}$ could be a nonparametric infinite dimensional Reproducing Kernel Hilbert Space (RKHS) or an overparameterized neural network representation.

Given the set of labeled samples, $\mathcal{L}$, and a class of functions $\mathcal{F}$, let $f \in \mathcal{F}$ be the interpolating function such that $f(x_i) = y_i$ for all $(x_i, y_i) \in \mathcal{L}$. Note that there may be many functions that interpolate a discrete set of points such as $\mathcal{L}$. Among these, we choose $f$ to be the minimum norm interpolator:

$$\begin{aligned} f(x) &\coloneqq \text{argmin}_{g \in \mathcal{F}} \ \ \|g\|_{\mathcal{F}} \\ \text{s.t.} \quad & g(x_i) = y_i, \ \text{for all } (x_i, y_i) \in \mathcal{L}. \end{aligned} \tag{2}$$

Clearly, the definition of $f$ depends on the set of currently labeled samples $\mathcal{L}$ and the function norm $\|\cdot\|_{\mathcal{F}}$, although we omit these dependencies for ease of notation. The choice of $\mathcal{F}$ and the norm $\|\cdot\|_{\mathcal{F}}$ is application dependent. In this paper, we focus on (1) function classes represented by an overparameterized neural network representation with the $\ell^2$ norm of the weight vectors and (2) reproducing kernel Hilbert spaces with the corresponding Hilbert norm.

For unlabeled points $u \in \mathcal{U}$ and $\ell \in \{-1, +1\}$, define $f_\ell^u(x)$ is the minimum norm interpolating function based on current set of labeled samples $\mathcal{L}$ and the point $u \in \mathcal{U}$ with label $\ell$:

$$\begin{aligned} f_\ell^u(x) &\coloneqq \text{argmin}_{g \in \mathcal{F}} \ \ \|g\|_{\mathcal{F}} \\ \text{s.t.} \quad & g(x_i) = y_i, \ \text{for all } (x_i, y_i) \in \mathcal{L} \\ & g(u) = \ell. \end{aligned} \tag{3}$$

We use this definition in the next subsection to define the notion of score.

### C. Definition of Proposed Notion of score

Roughly speaking, we want our selection criterion to prioritize labeling the most "informative" examples. Since the ultimate goal is to correctly label every example in $\mathcal{U}$, we design $\mathsf{score}(u)$ to measure the how hard it is to interpolate after adding $u$ to the set of labeled points. The intuition is that attacking the most challenging points in the input space first may eliminate the need to label other 'easier' examples later.

Note that we need to compute $\mathsf{score}(u)$ without knowing the label of $u$. To do so, we come up with an estimate of label of $u$, denoted by $\ell(u) \in \{-1, +1\}$ and compute $\mathsf{score}(u)$ assuming that upon labeling, $u$ will be labeled $\ell(u)$. We propose the following criterion for choosing $\ell(u)$:

$$\ell(u) := \text{argmin}_{\ell \in \{-1,+1\}} \|f_\ell^u(x)\|_{\mathcal{F}}. \tag{4}$$

Operating in the interpolating regime, we estimate the label of any unlabeled sample, $u$, to be the one that yields the minimum norm interpolant (i.e., the "smoother" of the two interpolants among the two possible functions $f_+^u(x)$ and $f_-^u(x)$).

Define

$$f^u(x) := f^u_{\ell(u)}(x) \tag{5}$$

to be the interpolating function after adding the sample $u$ with the label $\ell(u)$, defined in (4).

We propose two notions of $\mathsf{score}$. For $u \in \mathcal{U}$, define

$$\mathsf{score}_{\mathcal{F}}(u) = \left\|f^u(x)\right\|_{\mathcal{F}} \tag{6}$$

$$\mathsf{score}_{\square}(u) = \left\|f^u(x) - f(x)\right\|_{\square} \tag{7}$$

where $\|\cdot\|_{\mathcal{F}}$ is the norm associated the function space $\mathcal{F}$. The function $f$ is the minimum norm interpolator of the labeled examples in $\mathcal{L}$ (defined in (2)), and $f^u(x)$ is defined (5) as the minimum norm interpolator after adding $u$ with the estimated label $\ell(u)$ to the set of labeled points. Also, define

$$\|g\|_{\square} = \int_{\mathcal{X}} |g(x)|^2 \, \mathrm{d}P_X(x), \tag{8}$$

where $P_X$ is the distribution of $x$. In practice, $P_X$ is the empirical distribution of $\mathcal{U}$. We refer to the (6) as the *function norm score* and (7) as the *data-based norm score*.[1]

The distinction between the two definitions of the $\mathsf{score}$ function is as follows. Scoring unlabeled points according to the definition $\mathsf{score}_{\mathcal{F}}$ priotorizes labeling the examples which result in minimum norm interpolating functions with largest norm. Since the norm of the function can be associated with its smoothness, roughly speaking, this means that this criterion picks the points which give the least smooth interpolating functions. However, $\mathsf{score}_{\mathcal{F}}$ is insensitive to the distribution of data. The data-based $\mathsf{score}_{\square}$, in contrast, is sensitive to the distribution of the data. Measuring the difference between the new interpolation $f^u$ and the previous one makes this also sensitive to the structure of the function class.

With these definitions in place, we state the MaxiMin Active Learning criterion as follows. Given labeled data $\mathcal{L}$, the next example $u^* \in \mathcal{U}$ to label is selected according to

$$f^u = \arg \min_{f \in \{f_+^u, f_-^u\}} \|f\|_{\mathcal{F}}, \quad \forall u \in \mathcal{U}$$

$$u^* = \arg \max_{u \in \mathcal{U}} \mathsf{score}(u)$$

with either $\mathsf{score}_{\mathcal{F}}$ or $\mathsf{score}_{\square}$.

## III. MaxiMin Active Learning With Neural Networks

### A. Overparameterized Neural Networks and Interpolation

Neural networks are often highly overparameterized and exactly fit to training data, yet remarkably the learned models generalize well to new data. A mathematical understanding of this phenomenom is beginning to emerge [1], [2], [3], [4], [5], [6], [7], [8]. This work suggests that among all the networks that could be fit to the training data, the learning algorithms used in training favor networks with smaller weights, providing a sort of implicit regularization. With this in mind, researchers have shown that even shallow networks and classical kernel methods fit to the data but regularized to have small weights (e.g., minimum norm fit to data) can generalize well [2], [8], [9], [10]. The functional mappings generated by wide, two-layer neural networks with Rectified Linear Unit (ReLU) activation functions were studied in [20]. It is shown that exactly fitting such networks to training data subject to minimizing the $\ell^2$-norm of the network weights results in a linear spline interpolation. This result was extended to a broad class of interpolating splines by appropriate choices of activation functions [21]. Our analysis of the MaxiMin active learning with neural networks will leverage these connections.

### B. Neural Network Regularization

It has been long understood that the size of neural network weights, rather than simply the number of weights/neurons, characterizes the complexity of neural networks [22]. Here we focus on two-layer neural networks with ReLU activation functions in the hidden layer. If $\boldsymbol{x} \in \mathbb{R}^d$ is input to the network, then the output is computed by the function

$$f_{\boldsymbol{w},\boldsymbol{b},c}(\boldsymbol{x}) = \sum_{n=1}^{N} v_n \sigma\left(\boldsymbol{u}_n^T \boldsymbol{x} + b_n\right) + c, \tag{9}$$

where $\sigma(\cdot) = \max\{0, \cdot\}$ is the ReLU activation, $\boldsymbol{w} := \{v_n, \boldsymbol{u}_n\}_{n=1}^N$ are the "weights" of the network, and $\boldsymbol{b} := \{b_n\}$ and $c$ are constant "bias" terms. The "norm" of $f_{\boldsymbol{w},\boldsymbol{b},c}$ is defined as $\|f_{\boldsymbol{w},\boldsymbol{b},c}\| := \|\boldsymbol{w}\|_2$, the $\ell_2$-norm of the vector of network weights. We use the term norm in quotes because technically the weight norm does not correspond to a true norm on the function $f_{\boldsymbol{w},\boldsymbol{b},c}$ since, for example, constant functions $f_{\boldsymbol{w},\boldsymbol{b},c} = c$ have $\|\boldsymbol{w}\|_2 = 0$. From now on we will drop the subscripts and just write $f$ for ease of notation. Let $\{(x_i, y_i)\}_{i=1}^M$ be a set of training data. The minimum "norm" neural network

[1] Operationally, to compute the data-based norm of any function, the algorithm uses the probability mass function of set of unlabeled points as a proxy for the input probability density function over the feature space $\mathcal{X}$. In particular, the algorithm approximates $\|g\|_{\square}$ by the average of the function over the set of unlabeled points: $\|g\|_{\square} \approx \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} |g(u)|^2$. High density of set of unlabeled points and some mild regularity conditions guarantee that this is a good approximation. Throughout the paper, we use (8) to prove theoretical statements and its approximation in the numerical simulations.

interpolation of these data is the solution to the optimization

$$\min_{w} \|w\|_2 \text{ subject to } f(x_i) = y_i, \quad i = 1, \ldots, M.$$

A solution exists if the number of neurons $N$ is sufficiently large (see [23, Th. 5.1]).

In Section V we explore the behavior of MaxiMin active learning through numerical experiments using both the function "norm" score and the data-based norm score. In all our experiments and theory, we assume the binary classification setting where $y_i = \pm 1$. Broadly speaking, we observe the following behaviors.

- With the function "norm" score the MaxiMin active learning algorithm tends to sample aggressively in the vicinity of the boundary, prefering to gather new labels between the closest oppositely labeled examples.
- The data-based norm score is sensitive to the distribution of the data. It strikes a balance between exploiting regions between oppositely labeled examples (as in the function-based case) and exploring regions further away from labeled examples. Thus we see evidence that the data-based norm can effectively seek out the decision boundary and explore data clusters.

These behaviors are supported by a formal analysis of MaxiMin active learning in one dimension, discussed next.

### C. MaxiMin Active Learning in One-Dimension

Our analysis of MaxiMin active learning with neural networks will focus on the behavior in one-dimension. We show that MaxiMin active learning with a two-layer ReLU netwok recovers optimal bisection learning strategies. The following characterization of minimum "norm" neural network interpolation in one-dimension follows from [20], [21] (see [21, Th. 4.4 and Proposition 6.1]).

*Theorem 1:* Let $f : \mathbb{R} \to \mathbb{R}$ be a two-layer neural network with ReLU activation functions and $N$ hidden nodes as in (9). Let $\{(x_i, y_i)\}_{i=1}^M$ be a set of training data. If $N \geq M$, then a solution to the optimization

$$\min_{w} \|w\|_2 \text{ subject to } f(x_i) = y_i, \quad i = 1, \ldots, M$$

is a minimal knot linear spline interpolation of the points $\{(x_i, y_i)\}_{i=1}^M$.

In our analysis, we exploit the equivalence between minimum "norm" neural networks and linear splines. Specifically, a solution to the optimization is an interpolating function that is linear between each pair of neighboring points. This ensures that given a pair of neighboring labeled points $x_1$ and $x_2$ and any unlabeled point $x_1 < u < x_2$, adding $u$ to the set of labeled points can only potentially change the interpolating function between $x_1$ and $x_2$. To eliminate uncertainty in the boundary conditions of the interpolation, we assume that the neural network is initialized by labeling the leftmost and rightmost points in the dataset and forced to have a constant extension to the left and right of these points (this can be accomplished by adding two artificial points to the left and right with the same labels as the true endpoints).

The main message of our analysis is that MaxiMin active learning with two-layer ReLU networks recovers optimal bisection (binary search) in one-dimension. This is summarized by the next corollary which follows in a straightforward fashion from Theorems 2 and 3.

*Corollary 1:* Consider $N$ points uniformly distributed in the interval $[0, 1]$ labeled according to a $k$-piecewise constant function $f$ so that $y_i = f(x_i) \in \{-1, +1\}$, $i = 1, \ldots, N$, and length of the pieces are $\Theta(1/K)$. Then after labeling $O(k \log N)$ examples, the MaxiMin active learning with a two-layer ReLU network correctly labels all $N$ examples (i.e., the training error is zero).

The corollary follows from the fact that the MaxiMin criteria (both function norm and data-based norm) selects the next example to label at the midpoint between neighboring and oppositely labeled examples (i.e., at a bisection point). This is characterized in the next two theorems. First we consider the function "norm" criterion. The proof of the following theorem appears in Appendix A.1.

*Theorem 2:* Let $\mathcal{L}$ be a set of labeled examples and let $u$ be an unlabeled example. Let $f_+^u$ be the minimum "norm" interpolator of $\mathcal{L} \cup (u, +1)$ and let $f_-^u$ be the minimum "norm" interpolator of $\mathcal{L} \cup (u, -1)$. Define the score of an unlabeled example $u$ as $\mathsf{score}_{\mathcal{F}}(u) = \min\{\|f_+^u\|, \|f_-^u\|\}$, where $\|f\| = \|w\|_2$, the neural network weight norm. Then, the selection criterion based on $\mathsf{score}_{\mathcal{F}}$ has the following properties

1) Let $x_1$ and $x_2$ be two oppositely labeled neighboring points in $\mathcal{L}$, i.e., no other points between $x_1$ and $x_2$ have been labeled and $y_1 \neq y_2$. Then for all $x_1 < u < x_2$, $\mathsf{score}_{\mathcal{F}}(\frac{x_1+x_2}{2}) \geq \mathsf{score}_{\mathcal{F}}(u)$.
2) Let $x_1 < x_2$ and $x_3 < x_4$ be two pairs of oppositely labeled neighboring points (i.e., $y_1 \neq y_2$ and $y_3 \neq y_4$) such that $x_2 - x_1 \geq x_4 - x_3$. Then,
$$\mathsf{score}_{\mathcal{F}}\left(\frac{x_1 + x_2}{2}\right) \leq \mathsf{score}_{\mathcal{F}}\left(\frac{x_3 + x_4}{2}\right).$$
3) Let $x_5$ and $x_6$ be two identically labeled neighboring points in $\mathcal{L}$, i.e., $y_5 = y_6$. Then for all $x_5 < u < x_6$, the function $\mathsf{score}_{\mathcal{F}}(u)$ is constant.
4) For any pair of neighboring oppositely labeled points $x_1$ and $x_2$, any pair of neighboring identically labeled points $x_5$ and $x_6$, any $x_1 < u < x_2$ and any $x_5 < v < x_6$, we have
$$\mathsf{score}_{\mathcal{F}}(v) \leq \mathsf{score}_{\mathcal{F}}(u).$$

Now we turn to the data-based norm. Here we observe the effect of the data distribution on the bisection properties. The properties mirror those in Theorem 2 except in the case of the second property. The data-based norm criterion tends to sample in the *largest* (most data-massive) interval between oppositely labeled points, whereas the function-based norm criterion favors points in the *smallest* interval.

*Theorem 3:* Let the distribution $\mathbb{P}(X)$ be uniform over an interval. Let $\mathcal{L}$ be a set of labeled examples and let $u$ be an unlabeled example. Let $f_+^u$ be the minimum "norm" interpolator of $\mathcal{L} \cup (u, +1)$ and let $f_-^u$ be the minimum "norm" interpolator of $\mathcal{L} \cup (u, -1)$ and let $f^u = \arg_{g \in [f_+^u, f_-^u]} \|g\|$ consistent with notations in (3) and (5). Then $\mathsf{score}_{\mathrm{D}}(u) = \int |f^u(x) - f(x)|^2 \, dP_X(x)$, where $f$ is the minimum "norm"

interpolator based on the labeled data $\mathcal{L}$. Then, the selection criterion based on $\mathsf{score}_{\mathrm{D}}$ has the following properties.

1) Let $x_1$ and $x_2$ be two oppositely labeled neighboring points in $\mathcal{L}$, i.e., $y_1 \neq y_2$. Then for all $x_1 < u < x_2$ $\mathsf{score}_{\mathrm{D}}(\frac{x_1+x_2}{2}) \geq \mathsf{score}_{\mathrm{D}}(u)$.
2) Let $x_1 < x_2$ and $x_3 < x_4$ be two pairs of oppositely labeled neighboring labeled points (i.e., $y_1 \neq y_2$ and $y_3 \neq y_4$) such that $x_2 - x_1 \geq x_4 - x_3$. If the unlabeled points are uniformly distributed in each interval and the number of points is in $(x_1, x_2)$ is less than the number in $(x_4, x_3)$, then

$$\mathsf{score}_{\mathrm{D}}\left(\frac{x_1+x_2}{2}\right) \geq \mathsf{score}_{\mathrm{D}}\left(\frac{x_3+x_4}{2}\right).$$

3) Let $x_5$ and $x_6$ be two identically labeled neighboring points in $\mathcal{L}$, i.e., $y_5 = y_6$. Then for all $x_5 < v < x_6$, we have $\mathsf{score}_{\mathrm{D}}(v) = 0$.
4) For any pair of neighboring oppositely labeled points $x_1$ and $x_2$, any pair of neighboring identically labeled points $x_5$ and $x_6$, any $x_1 < u < x_2$ and any $x_5 < v < x_6$, we have

$$\mathsf{score}_{\mathrm{D}}(v) \leq \mathsf{score}_{\mathrm{D}}(u).$$

The proof appears in Appendix A.2.

## IV. Interpolating Active Learners in an RKHS

In this section, we will focus on minimum norm interpolating functions in a Reproducing Kernel Hilbert Space (RKHS). We present theoretical properties for general RKHS settings, detailed analytical results in the one-dimensional setting, and numerical studies in multiple dimensions. Broadly speaking, we establish the following properties: the proposed score functions

- tend to select examples near the decision boundary of $f$, the current interpolator;
- the score is largest for unlabeled examples near the decision boundary *and* close to oppositely labeled examples, in effect searching for the boundary in the most likely region of the input space;
- in one dimension the interpolating active learner coincides with an optimal binary search procedure;
- using data-based function norms, rather than the RKHS norm, the interpolating active learner executes a tradeoff between sampling near the current decision boundary and sampling in regions far away from currently labeled examples, thus exploiting cluster structure in the data.

### A. Kernel Methods

A Hilbert space $\mathcal{H}$ is associated with an inner product: $\langle f, g\rangle_{\mathcal{H}}$ for $f, g \in \mathcal{H}$. This induces a norm defined by $\|f\|_{\mathcal{H}} = \sqrt{\langle f, f\rangle_{\mathcal{H}}}$. A symmetric bivariate function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is positive semidefinite if for all $n \geq 1$, and points $\{x_i\}_{i=1}^n$, the matrix $\mathbf{K}$ with element $\mathbf{K}_{i,j} = K(x_i, x_j)$ is positive semidefinite (PSD). These functions are called PSD kernel functions. A PSD kernel constructs a Hilbert space, $\mathcal{H}$ of functions on $f : \mathcal{X} \to \mathbb{R}$. For any $x \in \mathcal{X}$ and any $f \in \mathcal{H}$, the function $K(\cdot, x) \in \mathcal{H}$ and $\langle f, K(\cdot, x)\rangle_{\mathcal{H}} = f(x)$. Throughout this section, we assume $K(x, x) = 1$.

For the set of labeled samples $\mathcal{L} = \{(x_1, y_1), \ldots, (x_L, y_L)\}$ with $y_i \in \{-1, +1\}$, let the function $f(x)$ be decomposed as

$$f(x) = \sum_{i=1}^{L} \alpha_i k(x_i, x) \quad \text{with} \quad \alpha = \mathbf{K}^{-1}\mathbf{y}, \tag{10}$$

where $\mathbf{K} = [\mathbf{K}_{i,j}]_{i,j}$ is the $L$ by $L$ matrix such that $\mathbf{K}_{i,j} = k(x_i, x_j)$ and $\mathbf{y} = [y_1, \ldots, y_L]^T$. Using reproducible kernels implies that $f(x) \in \mathcal{H}$ for the a RKHS $\mathcal{H}$. Then, $f(x)$ defined above is the minimum Hilbert norm interpolating function defined in (2). Using the property $\langle K(x_i, \cdot), K(x_j, \cdot)\rangle = K(x_i, x_j)$, we have

$$\|f(x)\|_{\mathcal{H}}^2 = \alpha^T \mathbf{K} \alpha = \mathbf{y}^T \mathbf{K}^{-1} \mathbf{y}.$$

For $u \in \mathcal{U}$ and $\ell \in \{-1, +1\}$, the minimum norm interpolating unction $f_\ell^u(x)$, defined in (3) (based on currently labeled samples $\mathcal{L}$ and sample $u$ with label $\ell$) is derived similarly:

$$f_\ell^u(x) = \sum_{i=1}^{L} \widetilde{\alpha}_i k(x_i, x) + \widetilde{\alpha}_{L+1} k(u, x) \quad \text{with} \quad \widetilde{\alpha} = \widetilde{\mathbf{K}}_u^{-1} \widetilde{\mathbf{y}}_\ell, \tag{11}$$

where

$$\widetilde{\mathbf{K}}_u = \begin{bmatrix} \mathbf{K} & \mathbf{a}_u \\ \mathbf{a}_u^T & b \end{bmatrix}, \quad \mathbf{a}_u = \begin{bmatrix} k(x_1, u) \\ \vdots \\ k(x_L, u) \end{bmatrix}, \quad \widetilde{\mathbf{y}}_\ell = \begin{bmatrix} \mathbf{y} \\ \ell \end{bmatrix},$$
$$\text{and } b = K(u, u). \tag{12}$$

Throughout this paper, we use kernel such that $K(x, x) = 1$ for all $x \in \mathcal{X}$.

### B. Properties of General Kernels for Active Learning

We first show that using kernel based function spaces for interpolation, $\ell(u)$ defined in (4) coincides with the sign of value of current interpolator at $u$.

*Proposition 1:* For $u \in \mathcal{X}$ and $\ell \in \{-, +\}$, define $f(x)$ and $f_\ell^u(x)$ according to (2) and (3) in Section II. Then, $\ell(u)$ defined in (4) satisfies

$$\ell(u) = \begin{cases} +1 & \text{if } f(u) \geq 0 \\ -1 & \text{if } f(u) < 0. \end{cases}$$

*Proof:* Let $\widetilde{\mathbf{y}}_\ell = [y_1, \ldots, y_n, \ell]^T$, $\mathbf{a}_u = [K(x_1, u), \ldots, K(x_n, u)]^T$ and $b = K(u, u) = 1$. Let $\mathbf{K}$ be the kernel matrix for the elements in $\mathcal{L}$ and $\widetilde{\mathbf{K}}_u$ be the kernel matrix for the elements in $\mathcal{L} \cup \{u\}$, as defined in (12). Then, for $\ell \in \{-1, +1\}$

$$\begin{aligned} \left\|f_\ell^u(x)\right\|_{\mathcal{H}}^2 &= \widetilde{\mathbf{y}}_\ell^T \widetilde{\mathbf{K}}_u^{-1} \widetilde{\mathbf{y}}_\ell \overset{(a)}{=} \mathbf{y}^T \left(\mathbf{K} - \mathbf{a}_u \mathbf{a}_u^T\right)^{-1} \\ &\quad \times \mathbf{y} - 2\ell\, \mathbf{y}^T \left(\mathbf{K} - \mathbf{a}_u \mathbf{a}_u^T\right)^{-1} \mathbf{a} + \left(1 - \mathbf{a}_u^T \mathbf{K}^{-1} \mathbf{a}_u\right)^{-1} \\ &\overset{(b)}{=} \mathbf{y}^T \mathbf{K}^{-1} \mathbf{y} + \frac{\left(1 - \ell\, \mathbf{y}^T \mathbf{K}^{-1} \mathbf{a}_u\right)^2}{1 - \mathbf{a}_u^T \mathbf{K}^{-1} \mathbf{a}_u} \overset{(c)}{=} \|f(x)\|_{\mathcal{H}}^2 \\ &\quad + \frac{\left[1 - \ell f(u)\right]^2}{1 - \mathbf{a}_u^T \mathbf{K}^{-1} \mathbf{a}_u}. \end{aligned}$$

where Schur's complement formula gives (a) and Woodbury Identity with some algebra gives (b). We are using the property that $K(x,x) = 1$ and the diagonal elements of matrix $\widetilde{\mathbf{K}}_u$ are equal to one. (c) uses (10) for the minimum norm interpolating function based on $\mathcal{L}$, i.e., $f(x)$. Hence, $\|f_+^u(x)\|_{\mathcal{H}} > \|f_-^u(x)\|_{\mathcal{H}}$ if and only if $f(u) < 0$ which gives the statement of proposition. ■

### C. Radial Basis Kernels

From here on, we will focus on minimum norm interpolating functions with radial basis kernels. The kernel functions we use have the following form: For $x, x' \in \mathbb{R}^d$, $h > 0$ and $p > 1$, let

$$k_{h,p}(x,x') = \exp\left(-\frac{1}{h}\|x - x'\|_p\right), \tag{13}$$

where $\|x\|_p \coloneqq (\sum_{i=1}^d x_i^p)^{1/p}$ is the $\ell_p$ norm and $\|x - x'\|_p$ is the Minkowski distance satisfying the triangle inequality. For $p = 1, 2$ this category of kernels construct Reproducing Kernel Hilbert Spaces. When the parameters $h$ and $p$ are specified, we denote the kernel function $k_{h,p}(x, y)$ by $k(x, y)$.

### D. Laplace Kernel in One Dimension

To develop some intuition, we consider active learning in one-dimension. The sort of target function we have in mind is a multiple threshold classifier. Optimal active learning in this setting coincides with binary search. We now show that the proposed selection criterion based on $\mathsf{score}_{\mathcal{H}}$ with Hilbert norm associated with the Laplace kernels result in an optimal active learning in one dimension (proof in Appendix B.1).

*Proposition 2 (Maximin Criteria in One Dimension With Laplace Kernel):* Define $K(x,x') = \exp(-|x - x'|/h)$ to be the Laplace kernel in one dimension and the minimum norm interpolator function defined in Section IV-A. Let the selection criterion be based on $\mathsf{score}_{\mathcal{H}}(u)$ function defined in (6) with the Laplace kernel Hilbert norm. Then the following statements hold for any value of $h > 0$:

1) Let $x_1$ and $x_2$ be two neighboring labeled points in $\mathcal{L}$. Then $\mathsf{score}_{\mathcal{H}}(\frac{x_1+x_2}{2}) \geq \mathsf{score}_{\mathcal{H}}(u)$ for all $x_1 < u < x_2$.
2) Let $x_1 < x_2$ and $x_3 < x_4$ be two pairs of neighboring labeled points such that $x_2 - x_1 \geq x_4 - x_3$, then
   - if $y_1 \neq y_2$ and $y_3 = y_4$. Then $\mathsf{score}_{\mathcal{H}}(\frac{x_1+x_2}{2}) \geq \mathsf{score}_{\mathcal{H}}(\frac{x_3+x_4}{2})$.
   - if $y_1 = y_2$ and $y_3 \neq y_4$. Then $\mathsf{score}_{\mathcal{H}}(\frac{x_1+x_2}{2}) \leq \mathsf{score}_{\mathcal{H}}(\frac{x_3+x_4}{2})$.
   - if $y_1 \neq y_2$ and $y_3 \neq y_4$. Then $\mathsf{score}_{\mathcal{H}}(\frac{x_1+x_2}{2}) \leq \mathsf{score}_{\mathcal{H}}(\frac{x_3+x_4}{2})$.
   - if $y_1 = y_2$ and $y_3 = y_4$. Then $\mathsf{score}_{\mathcal{H}}(\frac{x_1+x_2}{2}) \geq \mathsf{score}_{\mathcal{H}}(\frac{x_3+x_4}{2})$.

The key conclusion drawn from these properties is that the midpoints between the closest oppositely labeled neighboring examples have the highest score. If there are no oppositely labeled neighbors, then the score is largest at the midpoint of the largest gap between consecutive samples. Thus, the score results in a binary search for the thresholds defininining the classifier. Using the proposition above, it is easy to show the following result, proved in the Appendix B.3.

*Corollary 2:* Consider $N$ points uniformly distributed in the interval $[0, 1]$ labeled according to a $k$-piecewise constant function $g(x)$ so that $y_i = g(x_i) \in \{-1, +1\}$ and length of the pieces are roughly on the order of $\Theta(1/K)$. Then by running the proposed active learning algorithm with Laplace Kernel and any bandwidth, after $O(k \log N)$ queries the sign of the resulting interpolant $f$ correctly labels all $N$ examples (i.e., the training error is zero).

This statement is true for $N > 5/h$. The proof is provided in Appendix B.3.

### E. General Radial-Basis Kernels in One Dimension

In the next proposition, we look at the special case of radial basis kernels, defined in Equation(13) applied to one dimensional functions with only three initial points. We show how maximizing $\mathsf{score}_{\mathcal{H}}$ with the appropriate Hilbert norm is equivalent to picking the zero-crossing point of our current interpolator.

*Proposition 3 (One Dimensional Functions With Radial Basis Kernels):* Assume that for any pair of samples $x, x' \in \mathcal{L}$ we have $|x - x'| \geq \Delta$. Assume $\Delta h^{-1/p} \geq D$ for a constant value of $D$. Let $x_1 < x_2 < x_3 \in \mathbb{R}$, $y_1 = y_2 = +1$ and $y_3 = -1$. For $u$ such that $x_2 + \Delta/2 < u < x_3 - \Delta/2$, we have $\mathsf{score}_{\mathcal{H}}(u) \leq \mathsf{score}_{\mathcal{H}}(u^*)$ where $u^*$ is the point satisfying $f(u^*) = 0$.

The proof is rather tedious and appears in Appendix C.1. But the idea is based on showing that with small enough bandwidth, $\|f_+^u\|$ is increasing in $u$ in the interval $[x_2 + \Delta/2, x_3 - \Delta/2]$ and $\|f_-^u\|$ is decreasing in $u$ in the same interval. This shows that $\max_u \min_{\ell \in \{-1,+1\}} \|f_\ell^u\|$ occurs at $u^*$ such that $\|f_+^{u^*}\| = \|f_-^{u^*}\|$. We showed that this is equivalent to the condition $f(u^*) = 0$.

### F. Properties of Data Based-Norm Criterion

Intuitively, $\mathsf{score}_{\square}$ measures the expected change in the squared norm over all unlabeled examples if $u \in \mathcal{U}$ is selected as the next point. This norm is sensitive to the particular distribution of the data, which is important if the data are clustered. This behavior will be demonstrated in the multidimensional setting discussed next.

In this section, we present two theoretical results on the properties of data-based norm selection criterion. To do so, we will prove the properties of the selected examples based on the data-based norm in the context of the clustered data. In particular, if the support of the generative distribution $P_X(x)$ is composed of several disjoint clusters, the data-based norm criterion prioritizes labeling samples from bigger clusters first. Subsequently, it selects a sample from each cluster to be labeled. If the clustering in the dataset is aligned with their labels (most of the samples in the same cluster are in the same class), labeling one sample in each cluster ensures rapid decay in the probability of error of the classifier as a function of number of labeled samples. This behavior is consistent with numerical simulations presented in Section V.

The next theorem will show that if the clusters are well-separated (the distance between the clusters are sufficiently

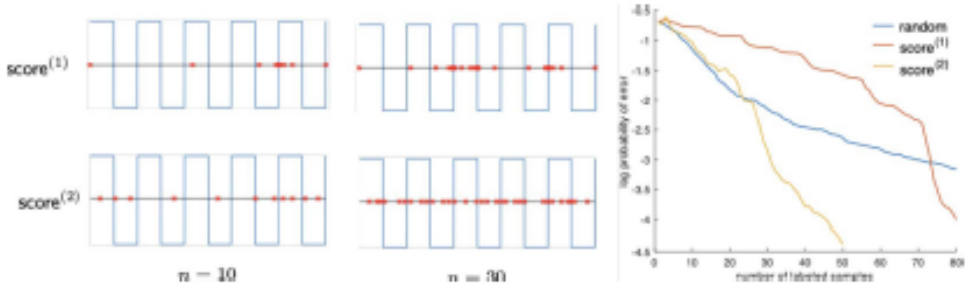



Fig. 2. Uniform distribution of samples in unit interval, multiple thresholds between ±1 labels, and active learning using Laplace Kernel, Bandwidth= 0.1. Probability of error of the interpolated function shown on right.

large), then the first example to be selected to for labeling is in the biggest cluster.

*Theorem 4 (First Point in Clustered Data):* Fix $p > 1$ and $h > 0$. Let the distribution $\mathbb{P}(X)$ be uniform over $M$ disjoint sets $B_1, \ldots, B_M$ such that $B_i$ is an $\ell_p$ ball with radius $r_i$ and center $c_i$, i.e.,

$$B_i = \mathcal{B}_{d,p}(r_i; c_i) := \left\{ x \in \mathbb{R}^d : \|x - c_i\|_p \leq r_i \right\}. \qquad (14)$$

Without loss of generality, assume $r_1 > r_2 > \cdots > r_K$. Define $D = \min_{i \neq j} \|c_i - c_j\|_p - 2r_1$ as an upper bound for the minimum distance between the clusters.

Assume $\mathcal{L} = \varnothing$ and let the interpolating functions $f$ be defined in (10) with $k_{h,p}$ (defined in (13)). The selection criterion is based on the $\mathsf{score}_{\mathcal{D}}$ function defined in (7). If

$$D > \frac{h}{2}\left[\ln M - \ln\left(1 - (r_2/r_1)^d\right)\right] \qquad \text{and} \qquad r_1 \leq h/2,$$

then the first point to be labeled is in the biggest ball, $B_1$.

The proof is presented in Appendix C.1.

The next theorem shows that if the distance between the clusters are sufficiently large and the radius of the clusters are not too large, then the active learning algorithm based on the notion of $\mathsf{score}$ with data-based norm labels one sample from each cluster before zooming in inside the clusters.

*Theorem 5 (Cluster Exploration):* Let $S$ be the support of $P_X$. Assume $S = \cup_{i=1}^{M} B_i$ where $B_i$'s are $\ell_p$-balls with radii $r$ and centers $c_i$. Define $D := \min_{i \neq j} \|c_i - c_j\|_p - 2r_1$ to be the minimum distance between the clusters. Let $\mathcal{L} = \{x_1, x_2, \ldots, x_L\}$ be $L < M$ labeled points such that $x_1 \in B_1, x_2 \in B_2, \ldots, x_L \in B_L$. Let the selection criterion be based on the $\mathsf{score}_{\mathcal{D}}$ function defined in (7). If $r < h/3$ and $D \geq 12h \ln(2M)$, then the next point to be labeled is in a new ball ($\cup_{i=L+1}^{M} B_i$) containing no labeled points.

As a corollary of the above theorem, one can see that if the ratio of the distance between the clusters to the radius of clusters is sufficiently large ($D/r > 36ln(2M)$), then one can use a kernel with proper bandwidth which picks one sample from each cluster initially. The proof is presented in Appendix C.2.

## V. Numerical Simulations of Kernel Based

In this Section, we present the outcome of numerical simulations of the proposed selection criteria on synthetic and real data. In this section, $\mathsf{score}_{\mathcal{H}}$ is used to denoted the $\mathsf{score}$ function defined in (6) with the Hilbert norm associated with the Laplace Kernel. Similarly, $\mathsf{score}_{\mathcal{D}}$ is the $\mathsf{score}$ function defined in (7) with the data-based norm.

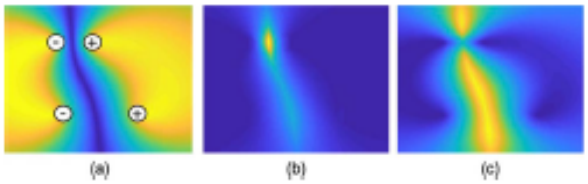



Fig. 3. Data selection of Laplace kernel active learner. (a) Magnitude of output map kernel machine trained to interpolate four data points as indicated (dark blue is 0 indicating the learned decision boundary). (b) Max-Min RKHS norm selection of next point to label. Brightest yellow is location of highest score and selected example. (c) Max-Min selection of next point to label using data-based norm. Both select the point on the decision boundary, but the RKHS norm favors points that are closest to oppositely labeled examples.

### A. Bisection in One Dimension

The bisection process is illustrated experimentally in the Figure 2 below. $\mathsf{score}_{\mathcal{H}}$ uses the RKHS norm. For comparison, we also show the behavior of the algorithm using $\mathsf{score}_{\mathcal{D}}$ and the data-based norm. Data selection using either score drives the error to zero faster than random sampling (as shown on the left). We clearly see the bisection behavior of $\mathsf{score}_{\mathcal{H}}$, locating one decision boundary/threshold and then another, as the proof corollary above suggests. Also, we see that the data-based norm does more exploration away from the decision boundaries. As a result, the data-based norm has a faster and more graceful error decay, as shown on the right of the figure. Similar behavior is observed in the multidimensional setting shown in Figure 5.

### B. Multidimensional Setting With Smooth Boundary

The properties and behavior found in the one dimensional setting carry over to higher dimensions. In particular, the max-min norm criterion tends to select unlabeled examples near the decision boundary *and* close to oppositely labeled examples, This is illustrated in Figure 3 below. The inputs points (training examples) are uniformly distributed in the square $[-1, 1] \times [-1, 1]$. We trained an Laplace kernel machine to perfectly interpolate four training points with locations and binary labels as depicted in Figure 3(a). The color depicts the magnitude of the learned interpolating function: dark blue is 0 indicating the "decision boundary" and bright yellow is approximately 3.5.

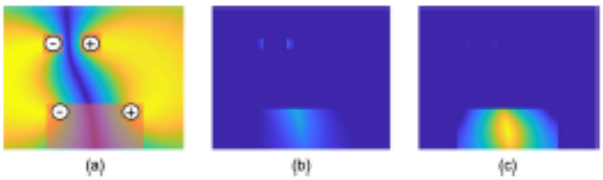



Fig. 4. Data selection of Laplace kernel active learner. (a) Unlabeled examples are only available in magenta shaded regions. (b) Max-Min selection map using RKHS norm (6). (c) Max-Min selection map using data-based norm defined in Equation (7).

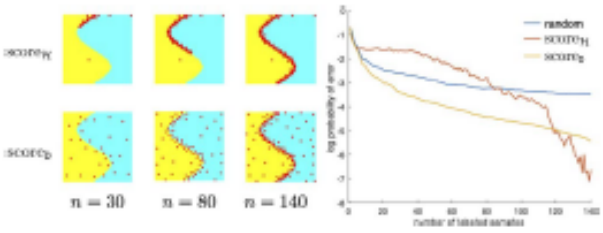



Fig. 5. Uniform distribution of samples, smooth boundary, Laplace Kernel, Bandwidth = 0.1. On left, sampling behavior of $\text{score}_{\mathcal{H}}$ and $\text{score}_{\square}$ at progressive stages (left to right). On right, error probabilities as a function of number of labeled examples.

Figure 3(b) denotes the score for selecting a point at each location based on RKHS norm criterion. Figure 3(c) denotes the score for selecting a point at each location based on data-based norm criterion discussed above. Both criteria select the point on the decision boundary, but the RKHS norm favors points that are closest to oppositely labeled examples whereas the data-based norm favors points on the boundary further from labeled examples.

Next we present a modified scenario in which the examples are not uniformly distributed over the input space, but instead concentrated only in certain regions indicated by the magenta highlights in Figure 4(a). In this setting, the example selection criteria differ more significantly for the two norms. The weight norm selection criterion remains unchanged, but is applied only to regions where there are examples. Areas with out examples to select are indicated by dark blue in Figure 4(b)-(c). The data-based norm is sensitive to the non-uniform input distribution, and it scores examples near the lower portion of the decision boundary highest.

The distinction between the max-min selection criterion using the RKHS vs. data-based norm is also apparent in the experiment in which a curved decision boundary in two dimensions is actively learned using a Laplace kernel machine, as depicted in Figure 5 below. $\text{score}_{\mathcal{H}}$ is the max-min RKHS norm criterion at progressive stages of the learning process (from left to right). The data-based norm is used in $\text{score}_{\square}$ defined in Equation (7). Both dramatically outperform a passive (random sampling) scheme and both demonstrate how active learning automatically focuses sampling near the decision boundary between the oppositely labeled data (yellow vs. blue). However, the data-based norm does more exploration away from the decision boundary. As a result, the data-based norm requires slightly more labels to perfectly predict all unlabeled examples, but has a more graceful error decay, as shown on the right of the figure.

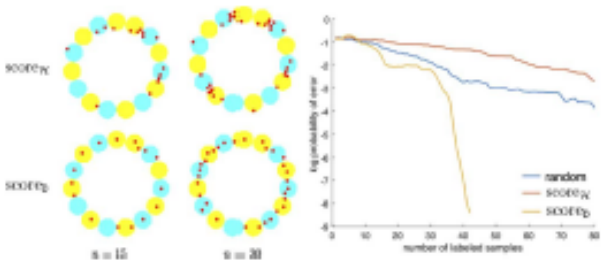



Fig. 6. Uniform distribution of samples, smooth boundary, Laplace Kernel, Bandwidth = 0.1. On left, sampling behavior of $\text{score}_{\mathcal{H}}$ and $\text{score}_{\square}$ at progressive stages (left to right). On right, error probabilities as a function of number of labeled examples.

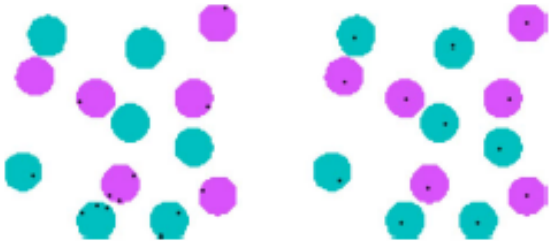


Fig. 7. Points in blue and yellow clusters are labeled +1 and −1, respectively. The left figure uses $\text{score}_{\mathcal{H}}$ to be the score function defined in (6) with the Hilbert norm associated with the Laplace Kernel. Similarly, $\text{score}_{\square}$ is the score function defined in (7) with the data-based norm. The first 13 samples selected by $\text{score}_{\mathcal{H}}$ and $\text{score}_{\square}$ are depicted as black dots. $\text{score}_{\square}$ has labeled one sample from each cluster, but $\text{score}_{\mathcal{H}}$ has not labeled any samples from 5 clusters. Note that $\text{score}_{\mathcal{H}}$ has spent some of the sample budget to discriminate between nearby clusters with opposite labels.

### C. Multidimensional Setting With Clustered Data

To capture the properties of the proposed selection criteria in clustered data, we implemented the algorithm on synthetic clustered data in Figures 6 and 7. We demonstrate how the data-based norm also tends to automatically select representative examples from clusters when such structure exists in the unlabeled dataset. Figure 6 compares the behavior of selection based on $\text{score}_{\mathcal{H}}In$ with the RKHS norm and $\text{score}_{\square}$ with data-based norm, when data are clusters and each cluster is homogeneously labeled. We see that the data-based norm quickly identifies the clusters and labels a representative from each, leading to faster error decay as shown on the right.

In the setup in Figure 7, the samples are generated based on a uniform distribution on 13 clusters. Points in blue and yellow clusters are labeled +1 and −1, respectively. We run the two variations of proposed active learning algorithms and compare their sampling strategy in this setup. The left figure uses $\text{score}_{\mathcal{H}}$ to be the score function defined in (6) with the Hilbert norm associated with the Laplace Kernel. Similarly, $\text{score}_{\square}$ is the score function defined in (7) with the data-based norm.

The selection criterion based on $\text{score}_{\mathcal{H}}$ prioritizes sampling on the decision boundary of the current classifier where the currently oppositely labeled samples are close to each other. This behavior of the algorithm based on $\text{score}_{\mathcal{H}}$ in one dimension is proved in Sections IV-D and IV-E. Alternatively, $\text{score}_{\square}$ prioritizes labeling at least one sample from each

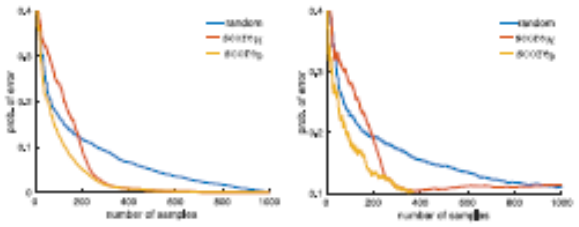


Fig. 8. Probability of error for learning a classification task on MNIST data set. The performance of three selection criteria for labeling the samples: random selection, active selection based on $\mathsf{score}_{\mathcal{H}}$, and active selection based on $\mathsf{score}_{\mathbb{D}}$. The first curve depicts the probability of error on the training set and the second curve is the probability of error on the test set.

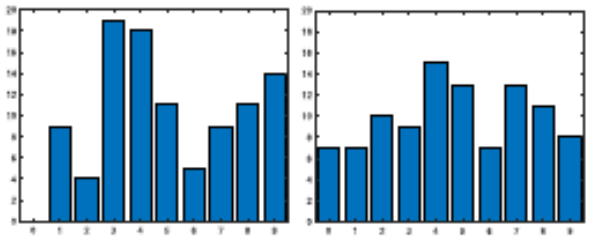


Fig. 9. The histogram of the handwritten digits associated with the labeled samples after labeling 100 samples. The first histogram is for the selection criterion $\mathsf{score}_{\mathcal{H}}$ and the second histogram is for the selection criterion $\mathsf{score}_{\mathbb{D}}$. Notably, $\mathsf{score}_{\mathcal{H}}$ has not labeled any of the images of the digit 0.

cluster. Hence, after labeling 13 samples, the active learning algorithm based on $\mathsf{score}_{\mathbb{D}}$ has one sample in each cluster, but the active learning algorithm based on $\mathsf{score}_{\mathbb{D}}$ has not labeled any samples in 5 clusters.

### D. MNIST Experiments

Here we illustrate the performance of the proposed active learning method on the MNIST dataset. We ran algorithms based on our proposed selection criteria for a binary classification task on MNIST dataset. The binary classification task used in this experiment assigns a label $-1$ to any digit in set $\{0, 1, 2, 3, 4\}$ and label $+1$ to $\{5, 6, 7, 8, 9\}$. The goal of the classifier is detecting whether an image belongs to the set of numbers greater or equal to 5 or not. We used Laplace kernel as defined in (13) with $p = 2$ and $h = 10$ on the vectorized version of a dataset of 1000 images. In Figures 8, $\mathsf{score}_{\mathcal{H}}$ is the $\mathsf{score}$ function defined in (6) with the Hilbert norm associated with the Laplace Kernel. Similarly, $\mathsf{score}_{\mathbb{D}}$ is the $\mathsf{score}$ function defined in (7) with the data-based norm.

To asses the quality of performance of each of the selection criteria, we compare the probability of error of the interpolator at each iteration. In particular, we plot the probability of error of the interpolator as a function of number of labeled samples, using the $\mathsf{score}_{\mathcal{H}}$ and $\mathsf{score}_{\mathbb{D}}$ functions on the training set and test set separately. For comparison, we also plot the probability of error when the selection criterion for picking samples to be labeled is random.

Figure 8 (a) shows the decay of probability of error in the training set. When the number of labeled samples is equal to the number of samples in the training set, it means that all the samples in training set are labeled and used in constructing the interpolator. Hence, the probability of error on the training set for any selection criterion is zero when number of labeled samples is equal to the number of samples in the training set. Figure 8 (b) shows the probability of error on the test set as a function of the number of labeled samples in the training set selected by each selection criterion.

*1) Clustering in MNIST:* The binary classification task used in the MNIST experiment assigns a label $-1$ to any digit in set $\{0, 1, 2, 3, 4\}$ and label $+1$ to $\{5, 6, 7, 8, 9\}$. We expect that the images are clustered where each cluster would correspond to the images of a digit. We expect that the advantageous behavior of using data-based norm criterion in clustered data is one of the reasons for faster decay of probability of error of the $\mathsf{score}_{\mathbb{D}}$ in Figure 8.

To verify this intuition, we look at the samples that were chosen by each criterion and the digit corresponding to that sample. Note that this digit is the number represented in the image and not the label of the sample since the label of each sample is $+1$ or $-1$ depending whether the number is greater than 4 or not. After labeling 100 samples, we look at histogram of the digits associated with the labeled samples with each criterion $\mathsf{score}_{\mathcal{H}}$ and $\mathsf{score}_{\mathbb{D}}$. If samples of each cluster are chosen to be labeled uniformly among clusters, we would see about 10 labeled samples in each cluster. Figure 9 shows the histogram described above for two variations of the selection criteria based on $\mathsf{score}_{\mathcal{H}}$ or $\mathsf{score}_{\mathbb{D}}$. We observe that selecting samples based on $\mathsf{score}_{\mathbb{D}}$ is much more uniform among the clusters. On the contrary, selecting samples based on $\mathsf{score}_{\mathcal{H}}$ gives much less uniform samples among clusters. In the particular example given in Figure 9, we see that even after selecting 100 samples to be labeled, no sample in the cluster of images of number 0 has been labeled in this instance of execution of the selection algorithm based on notion of $\mathsf{score}_{\mathcal{H}}$.

To quantify the uniformity of selecting samples in different clusters, we ran this experiment 20 times and estimated the standard deviation of number of labeled samples in each cluster after labeling 100 samples. Note that since we have 10 clusters, the mean of the number of labeled samples in each cluster is 10. The standard deviation using $\mathsf{score}_{\mathcal{H}}$ is 4.1 whereas standard deviation using $\mathsf{score}_{\mathcal{H}}$ is 2.7. This shows that selection criterion based on $\mathsf{score}_{\mathbb{D}}$ samples more uniformly among the clusters.

## VI. Interpolating Neural Network Active Learners

Here we briefly examine the extension of the max-min criterion and its variants to neural network learners. Neural network complexity or capacity can be controlled by limiting magnitude of the network weights [24], [25], [26]. A number of weight norms and related measures have been recently proposed in the literature [27], [28], [29], [30], [31]. For example, ReLU networks with a single hidden layer and minimum $\ell_2$ norm weights coincide with linear spline interpolation [32]. With this in mind, we provide empirical evidence showing that defining the max-min criterion with the norm of the network weights yields a neural network active learning algorithm with properties analagous to those obtained in the RKHS setting.

Consider a single hidden layer network with ReLU activation units trained using MSE loss. In Figure 10 we show the

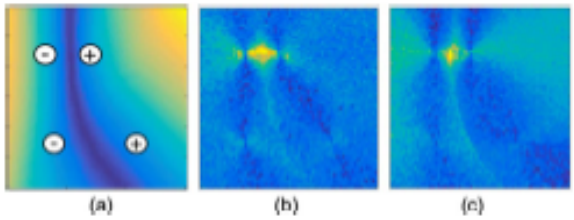



Fig. 10. Data selection of neural network active learner. (a) Magnitude of output map of single hidden layer ReLU network trained to interpolate four data points as indicated (dark blue is 0 indicating the learned decision boundary). (b) Max-Min selection of next point to label using network weight norm. (c) Max-Min selection of next point to label using data-based norm. Both select the point on the decision boundary that is closest to oppositely labeled examples.

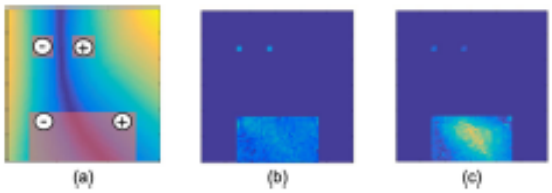



Fig. 11. Data selection of neural network active learner. (a) Unlabeled examples are only available in magenta shaded regions. (b) Max-Min selection map using network weight norm. (c) Max-Min selection map using data-based norm.

results of an experiment implemented in PyTorch in the same settings considered above for kernel machines in Figures 3 and 4. We trained an overparameterized network with 100 hidden layer units to perfectly interpolate four training points with locations and binary labels as depicted in Figure 10(a). The color depicts the magnitude of the learned interpolating function: dark blue is 0 indicating the "decision boundary" and bright yellow is approximately 3.5. Figure 10(b) denotes the $\mathsf{score}_{\mathcal{H}}$ with the weight norm (i.e., the $\ell_2$ norm of the resulting network weights when a new sample is selected at that location). The brightest yellow indicates the highest score and the location of the next selection. Figure 10(c) denotes the $\mathsf{score}_{\square}$ with the data-based norm defined in Equation (7). In both cases, the max occurs at roughly the same location, which is near the current decision boundary *and* closest to oppositely labeled points. The data-based norm also places higher scores on points further away from the labeled examples. Thus, the data selection behavior of the neural network is analagous to that of the kernel-based active learner (compare with Figure 3).

Next we present a modified scenario in which the examples are not uniformly distributed over the input space, but instead concentrated only in certain regions indicated by the magenta highlights in Figure 11(a). In this setting, the example selection criteria differ more significantly for the two norms. The weight norm selection criterion remains unchanged, but is applied only to regions where there are examples. Areas without examples to select are indicated by dark blue in Figure 11(b)-(c). The data-based norm is sensitive to the non-uniform input distribution, and it scores examples near the lower portion of the decision boundary highest. Again, this is quite similar to the behavior of the kernel active learner (compare with Figure 4).

## VII. Conclusion and Future Work

The question of designing active learning algorithms in the regime of nonparametric and overparameterized models become more essential as we look at larger models which require bigger training sets. To reduce the human cost of labeling allyl samples, we can use a pool-based active learning algorithm to avoid labeling non-informative examples.

Our algorithm does not exploit any assumption about the underlying classifier in selecting the samples to label. Yet, for a wide range of classifiers, it performs well with provable guarantees. It is designed for the extreme case of the nonparametric setting in which no assumption about the smoothness of the boundary between different classes is made by the learner.

There are many interesting questions remaining: the behavior of our proposed criterion applied to other classifiers such as kernel SVM instead of minimum norm interpolators, generalization of the criterion to multi-class settings and regression algorithms. The computational complexity of our criterion can also be a serious bottleneck in applications with bigger data-sets and should be addressed in future. Additional numerical simulations, especially with more complex architecture of Neural Networks can also be insightful.

## References

[1] S. Ma, R. Bassily, and M. Belkin, "The power of interpolation: Understanding the effectiveness of SGD in modern over-parametrized learning," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 3331–3340.

[2] M. Belkin, S. Ma, and S. Mandal, "To understand deep learning we need to understand kernel learning," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 540–548.

[3] S. Vaswani, F. Bach, and M. Schmidt, "Fast and faster convergence of SGD for over-parameterized models and an accelerated perceptron," 2018. [Online]. Available: arXiv:1810.07288.

[4] M. Belkin, A. Rakhlin, and A. B. Tsybakov, "Does data interpolation contradict statistical optimality?" 2018. [Online]. Available: arXiv:1806.09471.

[5] M. Belkin, D. Hsu, S. Ma, and S. Mandal, "Reconciling modern machine learning and the bias-variance trade-off," 2018. [Online]. Available: arXiv:1812.11118.

[6] S. Arora, S. S. Du, W. Hu, Z. Li, and R. Wang, "Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks," 2019. [Online]. Available: arXiv:1901.08584.

[7] M. Belkin, D. J. Hsu, and P. Mitra, "Overfitting or perfect fitting? Risk bounds for classification and regression rules that interpolate," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 2300–2311.

[8] T. Hastie, A. Montanari, S. Rosset, and R. J. Tibshirani, "Surprises in high-dimensional ridgeless least squares interpolation," 2019. [Online]. Available: arXiv:1903.08560.

[9] M. Belkin, D. Hsu, and J. Xu, "Two models of double descent for weak features," 2019. [Online]. Available: arXiv:1903.07571.

[10] T. Liang and A. Rakhlin, "Just interpolate: Kernel 'ridgeless' regression can generalize," 2018. [Online]. Available: arXiv:1808.00387.

[11] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *J. Mach. Learn. Res.*, vol. 2, pp. 45–66, Nov. 2001.

[12] D. A. Cohn, L. E. Atlas, and R. E. Ladner, "Improving generalization with active learning," *Mach. Learn.*, vol. 15, no. 2, pp. 201–221, 1994.

[13] O. Sener and S. Savarese, "Active learning for convolutional neural networks: A core-set approach," 2017. [Online]. Available: arXiv:1708.00489.

[14] Y. Gal, R. Islam, and Z. Ghahramani, "Deep Bayesian active learning with image data," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 1183–1192.

[15] K. Wang, D. Zhang, Y. Li, R. Zhang, and L. Lin, "Cost-effective active learning for deep image classification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 12, pp. 2591–2600, Jan. 2017.

[16] Y. Shen, H. Yun, Z. C. Lipton, Y. Kronrod, and A. Anandkumar, "Deep active learning for named entity recognition," 2017. [Online]. Available: arXiv:1707.05928.
[17] B. Settles, "Active learning literature survey," Dept. Comput. Sci., Univ. Wisconsin-Madison, Madison, WI, USA, Rep. 1648, 2009.
[18] B. Settles, "Active learning," *Synth. Lectures Artif. Intell. Mach. Learn.*, vol. 6, no. 1, pp. 1–114, 2012.
[19] S. Dasgupta, "Two faces of active learning," *Theor. Comput. Sci.*, vol. 412, no. 19, pp. 1767–1781, 2011.
[20] P. H. P. Savarese, I. Evron, D. Soudry, and N. Srebro, "How do infinite width bounded norm networks look in function space?" in *Proc. Conf. Learn. Theory (COLT)*, Phoenix, AZ, USA, 2019, pp. 2667–2690.
[21] R. Parhi and R. D. Nowak, "Minimum 'norm' neural networks are splines," 2019. [Online]. Available: arXiv:1910.02333.
[22] P. L. Bartlett, "The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network," *IEEE Trans. Inf. Theory*, vol. 44, no. 2, pp. 525–536, Mar. 1998.
[23] A. Pinkus, "Approximation theory of the MLP model in neural networks," *Acta numerica*, vol. 8, pp. 143–195, Jan. 1999.
[24] P. L. Bartlett, "For valid generalization the size of the weights is more important than the size of the network," in *Proc. Adv. Neural Inf. Process. Syst.*, 1997, pp. 134–140.
[25] B. Neyshabur, R. Tomioka, and N. Srebro, "In search of the real inductive bias: On the role of implicit regularization in deep learning," 2014. [Online]. Available: arXiv:1412.6614.
[26] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," 2016. [Online]. Available: arXiv:1611.03530.
[27] B. Neyshabur, R. Tomioka, and N. Srebro, "Norm-based capacity control in neural networks," in *Proc. Conf. Learn. Theory*, 2015, pp. 1376–1401.
[28] P. L. Bartlett, D. J. Foster, and M. J. Telgarsky, "Spectrally-normalized margin bounds for neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6240–6249.
[29] N. Golowich, A. Rakhlin, and O. Shamir, "Size-independent sample complexity of neural networks," in *Proc. Conf. Learn. Theory*, 2018, pp. 297–299.
[30] S. Arora, R. Ge, B. Neyshabur, and Y. Zhang, "Stronger generalization bounds for deep nets via a compression approach," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 254–263.
[31] B. Neyshabur, S. Bhojanapalli, and N. Srebro, "A PAC-Bayesian approach to spectrally-normalized margin bounds for neural networks," 2017. [Online]. Available: arXiv:1707.09564.
[32] P. Savarese, I. Evron, D. Soudry, and N. Srebro, "How do infinite width bounded norm networks look in function space?" 2019. [Online]. Available: arXiv:1902.05040.

# Linear Bandits with Feature Feedback

**Urvashi Oswal,[1] Aniruddha Bhargava,* Robert Nowak[2]**
[1]Oracle Labs, Redwood Shores, CA
[2]University of Wisconsin-Madison, Madison, WI
urvashi.oswal@oracle.com, aniruddha.j@gmail.com, rdnowak@wisc.edu

**Abstract**

This paper explores a new form of the linear bandit problem in which the algorithm receives the usual stochastic rewards as well as stochastic feedback about which features are relevant to the rewards, the latter feedback being the novel aspect. The focus of this paper is the development of new theory and algorithms for linear bandits with feature feedback which can achieve regret over time horizon $T$ that scales like $k\sqrt{T}$, without prior knowledge of which features are relevant nor the number $k$ of relevant features. In comparison, the regret of traditional linear bandits is $d\sqrt{T}$, where $d$ is the total number of (relevant and irrelevant) features, so the improvement can be dramatic if $k \ll d$. The computational complexity of the algorithm is proportional to $k$ rather than $d$, making it much more suitable for real-world applications compared to traditional linear bandits. We demonstrate the performance of the algorithm with synthetic and real human-labeled data.

## 1 Introduction

Linear stochastic bandit algorithms are used to sequentially select actions to maximize rewards. For instance, (Deshpande and Montanari 2012) propose to model recommendation systems that help users navigate through a large collection of items (products, videos, documents) using linearly parameterized multi-armed bandits. This model strikes a balance by allowing the user to explore the space of available items and probing the user's preferences. The linear bandit model assumes that the expected reward of each action is an (unknown) linear function of a (known) finite-dimensional feature associated with the action. Mathematically, if $\mathbf{x}_t \in \mathbf{R}^d$ is the feature associated with the action chosen at time $t$, then the stochastic reward is

$$y_t = \mathbf{x}_t^\top \boldsymbol{\theta}_* + \eta_t, \tag{1}$$

where $\boldsymbol{\theta}_*$ is the unknown linear functional (representing the user's preferences) and $\eta_t$ is a zero mean random variable. The goal is to adaptively select actions to maximize the rewards (corresponding to user's assessment of the chosen item's value). This involves (approximately) learning $\boldsymbol{\theta}_*$ and exploiting this knowledge. Linear bandit algorithms that exploit this special structure have been extensively studied and applied (Rusmevichientong and Tsitsiklis 2010; Abbasi-Yadkori, Pal, and Szepesvari 2011).

Unfortunately, standard linear bandit algorithms suffer from the curse of dimensionality. The regret grows linearly with the feature dimension $d$. The dimension $d$ may be quite large in modern applications (*e.g.*, 1000s of features in NLP or image/vision applications). In the recommendations setting, the existing bounds easily require $T >$ 100s of ratings to be meaningful, which is not realistic. The high-dimensionality also makes it challenging to employ state-of-the-art algorithms since it involves maintaining and updating a $d \times d$ matrix at every stage. However, in many cases the linear function may only involve a sparse subset of the $k < d$ features, and this can be exploited to partially reduce dependence on $d$. In such cases, the regret of sparse linear bandit algorithms scales like $\sqrt{dk}$ (Abbasi-Yadkori, Pal, and Szepesvari 2012; Lattimore and Szepesvári 2018).

We tackle the problem of linear bandits from a new perspective that incorporates feature feedback in addition to reward feedback, mitigating the curse of dimensionality. Specifically, we consider situations in which the algorithm receives a stochastic reward *and* stochastic feedback indicating which, if any, feature-dimensions were relevant to the reward value. For example, consider a situation in which users rate recommended text documents and additionally highlight keywords or phrases that influenced their ratings. Figure 1(a) illustrates the idea. Obviously, the additional "feature feedback" may significantly improve an algorithm's ability to home-in on the relevant features. The focus of this paper is the development of new theory and algorithms for linear bandits with feature feedback. We show that the regret of linear bandits with feature feedback scales linearly in $k$, the number of relevant features, without prior knowledge of which features are relevant nor the value of $k$. This leads to large improvements in theory and practice.

The simple feedback model, where the user directly selects a subset of the relevant features, can be generalized by allowing for an indirect form of feedback. For example, the user can select a region of an image instead of a subset of the standard deep neural network features. This form of

*This research was performed when U.O. and A.B. were at University of Wisonsin-Madison.

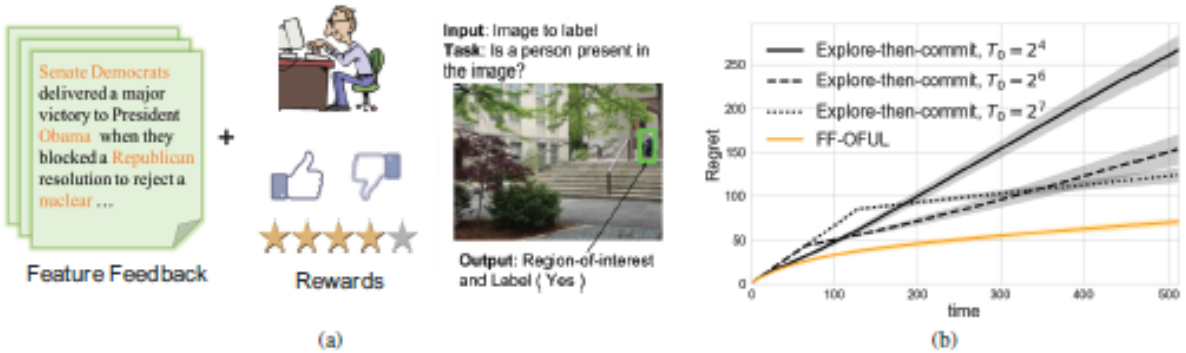



Figure 1: (a) (Left) Highlighted words for text-based applications and (Right) Region-of-interest feature feedback for image-based applications. (b) Comparison of the explore-then-commit strategy for different values of $T_0$ and our new FF-OFUL algorithm which combines exploration and exploitation steps (details of data generation in Section 5.1).

feedback could be used in different ways. For instance, we could use methods to map deep image features to image regions. However, in this paper we focus on the processes and benefits of incorporating direct feature feedback, and defer the development of indirect feedback models to future work.

Perhaps the most natural and simple way to leverage the feature feedback is an explore-then-commit strategy. In the first $T_0$ steps the algorithm selects actions at random and receives rewards and feature feedback. If $T_0$ is sufficiently large, then the algorithm will have learned all or most of the relevant features and it can then switch to a standard linear bandit algorithm operating in the lower-dimensional subspace defined by those features. There are two major problems with such an approach:

1. The correct choice of $T_0$ depends on the prevalence of relevant features in randomly selected actions, which generally is unknown. If $T_0$ is too small, then many relevant features will be missed and the long-run regret will scale linearly with the time horizon. If $T_0$ is too large, then the initial exploration period will suffer excess regret. This is depicted in Figure 1(b).
2. Regardless of the choice of $T_0$, the regret will grow linearly for $t < T_0$. The new FF-OFUL algorithm that we propose combines exploration and exploitation from the start and can lead to smaller regret initially and asymptotically as shown in Figure 1(b).

These observations motivate our proposed approach that dynamically adjusts the trade-off between exploration and exploitation. A key aspect of the approach is that it is automatically adaptive to the unknown number of relevant features $k$. Our theoretical analysis shows that its regret scales like $k\sqrt{T}$. Experimentally, we show the algorithm generally outperforms traditional linear bandits and the explore-then-commit strategy. This is due to the fact that the dynamic algorithm exploits knowledge of relevant features as soon as they are identified, rather than waiting until all or most are found. A key consequence is that our proposed algorithm yields significantly better rewards at early stages of the process, as shown in Figure 1(b) and in more comprehensive experiments later in the paper. The intuition for this is that estimating $\boldsymbol{\theta}_*$ on a fraction of the relevant coordinates can be exploited to recover a fraction of the optimal reward. Similar ideas are explored in linear bandits (without feature feedback) in (Deshpande and Montanari 2012).

### 1.1 Definitions

For round, $t$, let $\mathcal{X}_t \subseteq \mathbb{R}^d$ be the set of actions/items provided to the learner. We assume the standard linear model for rewards with a hidden weight vector $\boldsymbol{\theta}_* \in \mathbb{R}^d$. If the learner selects an action, $\mathbf{x}_t \in \mathcal{X}_t$, it receives reward, $y_t$, defined in (1) where $\eta_t$ is noise with a sub-Gaussian random distribution with parameter $R$. For the set of actions $\mathcal{X}_t$, the optimal action is given by, $\mathbf{x}_t^* := \text{argmax}_{\mathbf{x} \in \mathcal{X}_t} \mathbf{x}^\top \boldsymbol{\theta}_*$, which is unknown. We define regret as,

$$R_T = \sum_{t=1}^{T} \left( \mathbf{x}_t^{*\top} \boldsymbol{\theta}_* - \mathbf{x}_t^\top \boldsymbol{\theta}_* \right). \tag{2}$$

This is also called cumulative regret but, unless stated otherwise, we will refer to it as regret. We refer to the quantity $\mathbf{x}_t^{*\top} \boldsymbol{\theta}_* - \mathbf{x}_t^\top \boldsymbol{\theta}_*$ as the instantaneous regret which is the difference between the optimal reward and the reward received at that instant. We make the standard assumption that the algorithm is provided with an enormous action set which is only changing slowly over time, for instance, from sampling the actions without replacement ($\mathcal{X}_{t+1} = \mathcal{X}_t \backslash \mathbf{x}_t$).

### 1.2 Related Work

The area of contextual bandits was introduced by (Ginebra and Clayton 1995). The first algorithms for linear bandits appeared in (Abe and Long 1999) followed by those using the optimism in the face of uncertainty principle, (Auer and Long 2002), (Dani, Hayes, and Kakade 2008). (Rusmevichientong and Tsitsiklis 2010) showed matching upper and lower bounds when the action (feature) set is a unit hypersphere. Finally, (Abbasi-Yadkori, Pal, and Szepesvari

**Algorithm 1** OFUL

1: **for** $t = 1, 2, \ldots, T-1$ **do**
2: $(\mathbf{x}_t, \tilde{\boldsymbol{\theta}}_t) = \text{argmax}_{(\mathbf{x},\boldsymbol{\theta}) \in \mathcal{X}_t \times \mathcal{C}_{t-1}} \langle \mathbf{x}, \boldsymbol{\theta} \rangle$
3: Select action $\mathbf{x}_t$ and receive reward $y_t$.
4: Update $\overline{\mathbf{V}}_t = (\mathbf{X}_t^T \mathbf{X}_t + \lambda \mathbf{I})$ and $\widehat{\theta}_t = \overline{\mathbf{V}}_t^{-1} \mathbf{X}_t^T \mathbf{y}_t$
5: Update ellipsoidal confidence set $\mathcal{C}_t$ as
$$\mathcal{C}_t = \left\{ \boldsymbol{\theta} : \|\widehat{\theta}_t - \boldsymbol{\theta}\|_{\overline{\mathbf{V}}_t} \leq R\sqrt{2\log\left(\frac{\det(\overline{\mathbf{V}}_t)^{1/2}\det(\lambda I)^{-1/2}}{\delta}\right)} + \lambda^{1/2} S \right\}$$
6: **end for**

2011) gave a tight regret bound using new martingale techniques. We use their algorithm, OFUL, as a subroutine in our work. In the area of sparse linear bandits, regret bounds are known to scale like $\sqrt{kdT}$, (Abbasi-Yadkori, Pal, and Szepesvari 2012),(Lattimore and Szepesvári 2018), when operating in a $d$ dimensional feature space with $k$ relevant features. The strong dependence on the ambient dimension $d$ is unavoidable without further (often strong and unrealistic) assumptions. For instance, if the distribution of feature vectors is isotropic or otherwise favorably distributed, then the regret may scale like $k\log(d)\sqrt{T}$, e.g., by using incoherence based techniques from compressed sensing (Carpentier and Munos 2012). These results also assume knowledge of sparsity parameter $k$ and without it no algorithm can satisfy these regret bounds for all $k$ simultaneously. In contrast, we propose a new algorithm that automatically adapts to the unknown sparsity level $k$ and removes the dependence of regret on $d$ by exploiting additional feature feedback. In terms of feature feedback in text-based applications, (Croft and Das 1989) have proposed a method to reorder documents based on the relative importance of words using feedback from users. (Poulis and Dasgupta 2017) consider a similar problem but for learning a linear classifier. We use a similar feedback model but focus on the bandit setting where such feedback can be naturally collected along with rewards. The idea of allowing user's to provide richer forms of feedback has been studied in the active learning literature (Raghavan, Madani, and Jones 2006), (Druck, Settles, and McCallum 2009) and also been considered in other (interactive) learning tasks, such as cognitive science (Roads, Mozer, and Busey 2016), machine teaching (Chen et al. 2018), and NLP tasks (Yessenalina, Yue, and Cardie 2010).

## 2 Model for Feature Feedback

The algorithm presents the user with an item (*e.g.*, document) and the user provides feedback in terms of whether they like the item or not (logistic model) or how much they like it (inner product model). The user also selects a few features (*e.g.*, words), if they can find them, to help orient the search. We make the following assumptions.

**Assumption 1** (Sparsity). *The hidden weight vector $\boldsymbol{\theta}_\star \in \mathbb{R}^d$ is $k$-sparse and $k$ is unknown. In other words, $\boldsymbol{\theta}_\star$ has at most $k$ non-zero entries or if $supp(\boldsymbol{\theta}_\star) = \{i | \boldsymbol{\theta}_{\star i} \neq 0\}$ then $|supp(\boldsymbol{\theta}_\star)| = k \leq d$.*

**Assumption 2** (Discoverability). *For an action $\mathbf{x} \in \mathcal{X}$ selected uniformly at random, the probability that a relevant feature is present and is selected is at least $p > 0$ (unknown).*

**Assumption 3** (Noise). *Users may report irrelevant features. The number of reported irrelevant features (denoted by $0 \leq k' \leq d - k$) is unknown in advance.*

Assumption 1 ensures that there are at most $k$ relevant features, however we stress that the value of $k$ is unknown (it is possible that all $d$ features are relevant). Assumption 2 ensures that while every item may not have relevant features, we are able to find them with a non-zero probability when searching through items at random. This assumption can be viewed as a (possibly pessimistic) lower bound on the rate at which relevant features are discovered. For example, it is possible that exploitative actions may yield relevant features at a higher rate (e.g., relevant features may be correlated with higher rewards). We do not attempt to model such possibilities since this would involve making additional assumptions that may not hold in practice. Assumption 3 accounts for ambiguous features that are irrelevant but users erring on the side of marking as relevant.

The set up is as follows: we have a set of items, $\mathcal{X} \subseteq \mathbb{R}^d$ that we can propose to the users. There is a hidden weight vector $\boldsymbol{\theta}_\star \in \mathbb{R}^d$ that is $k$-sparse. We will further assume that $\|\boldsymbol{\theta}_\star\| \leq S$ and the action vectors are bounded in norm: $\forall \mathbf{x} \in \mathcal{X}, \|\mathbf{x}\| \leq L$. Besides the reward $y_t$, (1), at each time-step the learner gets $\mathcal{I}_t \subseteq [d]$ which is the relevance feedback information. The model further specifies that $\forall j \in \text{supp}(\boldsymbol{\theta}_\star), \Pr(j \in \mathcal{I}_t) \geq p$. That is, the probability a relevant feature is selected at random is at least $p$. We need this assumption to make sure that we can find all the relevant features.

## 3 Algorithm

In this section, we introduce an algorithm that uses feature relevance feedback by starting with a small feature space and gradually increasing the space over time without knowledge of $k$. We use the OFUL algorithm (stated as Algo. 1) based on the principle of optimism in face of uncertainty as a subroutine. The algorithm constructs ellipsoidal confidence sets centered around the ridge regression estimate, using observed data such that the sets contain the unknown $\theta_\star$ with high probability, and selects the action/item that maximizes the inner product with any $\theta$ from the confidence set.

All updates are made only in the dimensions that have been marked as relevant and the space is dynamically increased as new relevant features are revealed. If nothing is marked as relevant, then by default the actions are selected at random, potentially suffering the worst possible reward but, at the same time, increasing our chances of getting relevance feedback leading to a trade-off. Note that the algorithm is adaptive to the unknown number of relevant features $k$. If $k$ were known, we could stop looking for features when all relevant ones have been selected. We find that in practice, this algorithm has an additional benefit of being more robust to changes in the ridge parameter ($\lambda$) due to its intrinsic regularization of restricting the parameter space.

**Algorithm 2** Feature Feedback OFUL (FF-OFUL)

Let the set of relevant indices, $\mathcal{R}_0, \mathcal{I}_0 = \{\}$.
**while** $\mathcal{I}_0$ is empty **do**
  Select action at random, $\mathcal{I}_0 = \{$ indices revealed $\}$
**end while**
$\mathcal{R}_1 = \mathcal{R}_0 \bigcup \mathcal{I}_0$
Initialize $\mathcal{C}_0$ using actions sampled.
**for** $t = 1, 2, \ldots, T$ **do**
  Let $\boldsymbol{X}_t$ be the feature matrix restricted to $\mathcal{R}_t$.
  Set $\epsilon_t = 1/\sqrt{t}$. Draw $b_t$ from bernoulli($\epsilon_t$)
  **if** $b_t = 1$ **then**
    Pick an action $\mathbf{x}_t$ uniformly at random from $\mathcal{X}_t$.
  **else**
    Pick $(\mathbf{x}_t, \tilde{\theta}_t) = \text{argmax}_{(\mathbf{x},\boldsymbol{\theta}) \in \mathcal{X}_t \times \mathcal{C}_{t-1}} \langle \mathbf{x}, \boldsymbol{\theta} \rangle$
  **end if**
  With action $\mathbf{x}_t$ observe reward $y_t$ and indices, $\mathcal{I}_t$.
  Update $\mathcal{R}_t = \mathcal{R}_t \bigcup \mathcal{I}_t$
  **if** $\mathcal{I}_t$ is empty **then**
    Rank one update to $\overline{\mathbf{V}}_t, \widehat{\theta}_t, \mathcal{C}_t$ (see Algo. 1) using $(y_t, \mathbf{x}_t)$
  **else**
    Update $\boldsymbol{X}_t$ with features in $\mathcal{R}_t$.
    Recompute $\overline{\mathbf{V}}_t, \widehat{\theta}_t, \mathcal{C}_t$ with new feature set $\boldsymbol{X}_t$.
  **end if**
**end for**

(Abbasi-Yadkori, Pal, and Szepesvari 2011) provide a $\tilde{O}(d\sqrt{t})$ bound on the regret of OFUL stated as Algorithm 1 by ignoring constants and logarithmic terms. We prove a similar result but reduce the dependence on the dimension from $d$ to $k$. In order to do so, we must discover the support of $\boldsymbol{\theta}_*$. The idea being that we apportion a set of actions, with a form of $\epsilon$-greedy algorithm due to (Sutton and Barto 1998), to random plays in order to guarantee that we find all the relevant features, otherwise we run OFUL on the identified relevant dimensions. Reducing the proportion of random actions over time guarantees that the regret remains sub-linear in time. We propose Algorithm 2 to exploit feature feedback. Here, at each time $t$, with probability proportional to $1/\sqrt{t}$, the algorithm selects an action/item to present at random, otherwise it selects the item recommended by feature-restricted-OFUL.

## 4 Regret Analysis

In this section, we state regret bounds for the FF-OFUL algorithm along with a sketch of the proof and discuss approaches to improve the bounds while deferring proof details to supplementary material.

### 4.1 Regret Bound for Algorithm 2 (FF-OFUL)

Recall that $\forall \mathbf{x} \in \mathcal{X}, \|\mathbf{x}\| \leq L$ and $\|\boldsymbol{\theta}_*\| \leq S$. Therefore, for any action, the worst-case instantaneous regret can be derived using Cauchy-Schwarz as follows:

$$\begin{aligned} |\langle \mathbf{x}^*, \boldsymbol{\theta}_* \rangle - \langle \mathbf{x}, \boldsymbol{\theta}_* \rangle| &\leq |\langle \mathbf{x}^*, \boldsymbol{\theta}_* \rangle| + |\langle \mathbf{x}, \boldsymbol{\theta}_* \rangle| \\ &\leq \|\mathbf{x}^*\| \, \|\boldsymbol{\theta}_*\| + \|\mathbf{x}\| \, \|\boldsymbol{\theta}_*\| \\ &\leq 2SL \end{aligned}$$

We provide the main result that bounds the regret (2) of Algorithm 2 in the following theorem.

**Theorem 1.** *Assume that $\forall t > 0$ and $\mathbf{x} \in \mathcal{X}_t$, $\langle \mathbf{x}, \boldsymbol{\theta}_* \rangle \in [-1, 1]$, with the additional assumptions 1, 2 and 3 ($k' = 0$). Then with probability $\geq 1 - \delta$, the cumulative regret after $T$ steps for Algorithm 2,*

$$\begin{aligned} R_T \leq\ & \frac{8SL}{\log 6M/\delta} \left( \frac{\log 3k/\delta}{\log 1/(1-p)} \right)^2 \\ & + \log_2 \frac{T}{2} \left( 3SL \sqrt{T \log \frac{6M}{\delta}} \right) \\ & + 4 \log_2 \frac{T}{2} \sqrt{\frac{T}{2} k \log(\lambda + nL/k)} \left( \lambda^{1/2} S \right. \\ & \left. + R\sqrt{2 \log(3M/\delta) + k \log(1 + TL/(2\lambda k))} \right). \end{aligned}$$

*where $M = \log_2 \frac{T}{2}$, $\lambda > 0$ is the ridge regression parameter and $k$ is the (unknown) number of relevant features.*

In other words, with high probability, the regret of Algorithm 2 (FF-OFUL) scales like $\tilde{O}(k\sqrt{T} + \frac{1}{p^2})$, by ignoring constants and logarithmic terms and using the taylor series expansion of $-\log(1-p)$, over time horizon $T$ where $k$ is the number of relevant features and $p$ is the probability with which a relevant feature is marked in an action selected uniformly at random.

**Remark.** The values of $k$ and $p$ are unknown and the algorithm implicitly adapts to these problem-dependent parameters. Since the regret of any algorithm is trivially bounded by $O(T)$ (for bounded rewards), our new regret bound is non-trivial for $T > \max(k^2, p^{-2})$. In comparison, linear bandits without feature feedback have an $O(d\sqrt{T})$ regret, which is non-trivial only when $T > d^2$. So, our new algorithm enjoys a better regret bound if $p > d^{-1}$, which is a reasonable condition in high-dimensional settings (*e.g.*, $d = 10^4$).

The three terms in the total regret come from the following events. Regret due to: (1) exploration to guarantee observing all the relevant features (with high probability), (2) exploration after observing all relevant features (due to lack of knowledge of $p$ or $k$), and (3) exploitation and exploration running OFUL (after having observed all relevant features).

In practice, feature feedback may be noisy. Sometimes, features that are irrelevant may be marked as relevant. To account for this, we can relax our assumption to allow for subset of $k'$ irrelevant features that are mistakenly marked as relevant. Including these features will increase the regret but the theory goes through without much difficulty as stated in the following corollary.

**Corollary 1.** *With the same assumptions as Theorem 1, if a fixed set of $k'$ irrelevant features were indicated by the user (Assumption 3), then the regret of Algorithm 2 (FF-OFUL) scales like $\tilde{O}((k + k')\sqrt{T} + \frac{1}{p^2})$.*

The corollary follows since exploration is not affected by this noise and the regret of exploitation on the vector restricted to $k+k'$ dimensions scales like $(k+k')\sqrt{T}$. This accounts for having some features being ambiguous and users erring on the side of marking them as relevant. This only results in slightly higher regret so long as $k+k'$ is still smaller than $d$. One could improve this regret by making assumptions on the probabilities of feature selection to weed out the irrelevant features.

**Proof Sketch of Main Result** We provide a sketch of the proof here and defer details to supplementary material. Recall, the cumulative regret is summed over the instantaneous regrets for $t = 1, \ldots, T$. We divide the cumulative regret across epochs $s = 0, \ldots, M$ of doubling size $T_s = 2^s$ for $M = \log_2 \frac{T}{2}$.

This ensures that the last epoch dominates the regret and it allows for the evolving feature space. For each epoch, we bound the regret under two events, all relevant features have been identified (via user feedback) up to that epoch or not. First, we bound the regret conditioned on the event that all the relevant features have been identified in Lemma 3. This is further, in expectation, broken down into the $\epsilon_s$ portion of random actions for pure exploration (Lemma 1) and $1-\epsilon_s$ modified OFUL actions on the $k$-dimensional feature space for exploitation-exploration (Lemma 2). For pure exploration, we use the worst case regret bound but since $\epsilon_s$ is decreasing this does not dominate the OFUL term. Second, we bound the probability that some of the relevant features are not identified so far (Proposition 3), which is a constant depending on $k$ and $p$ since it becomes zero after enough epochs have passed. Pure exploration ensures the probability that some features are not identified decreases with each passing epoch. An issue of bounding regret of the actions selected by OFUL subroutine in each epoch is that, unlike OFUL, the confidence sets in our algorithm are constructed using additional actions from exploration rounds and past epochs. To accommodate this we prove a regret bound for this variation in Lemma 2. Putting all this together gives us the final result.

**Lower bound.** The arguments from (Dani, Hayes, and Kakade 2008),(Rusmevichientong and Tsitsiklis 2010) can be used get a lower bound of $O(k\sqrt{T})$. Suppose we knew the support, then any linear bandit algorithm that is run on that support must incur an order $k\sqrt{T}$ regret. We don't know the support but we estimate it with high probability and therefore the lower bound also applies here. Our algorithm is optimal up to log factors in terms of the dimension.

### 4.2 Better Early-Regret Bounds

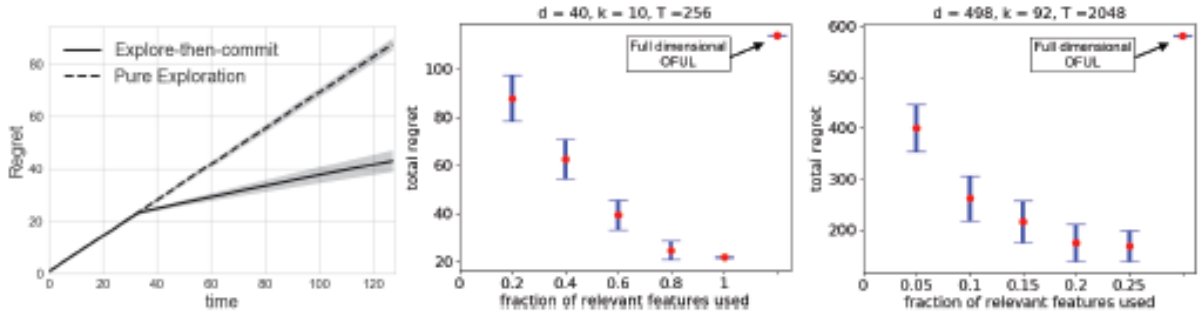
Our analysis bounds the regret of early rounds, before observing all relevant features, with the worst case regret which may be too pessimistic in practice. We present **results to support the idea of restricting the feature space in the short-term horizon and growing the feature space over time**. The results also suggest that an additional assumption on the behavior of early-regret could lead to better constants in our bounds. Any linear bandit algorithm restricted to the support of $\boldsymbol{\theta}_\star$ must incur an order $k\sqrt{T}$ regret so one can only hope to improve the constants of the bound.

In Figure 2(a) it can be seen that the average linear regret of pure exploration has a slope that is worse than OFUL restricted to a subset of the relevant features. The $N = 1000$ actions were randomly sampled from the unit sphere in $d = 40$ dimensions and $\theta_\star$ was generated with $k = 5$ sparsity. For a pure exploration algorithm that picks actions uniformly at random, independent of the problem instance, the regret can only be bound by $2SLT$ or $O(T)$. Let $R^{\mathcal{K}}_{alg}$ be the expected regret of algorithm $alg$ run on a subset of relevant features $\mathcal{K} \subseteq \{1, \ldots, k\}, |\mathcal{K}| = j \leq k$. For example, $alg$ could be the OFUL algorithm. Then $R^{\mathcal{K}}_{alg}$ represents the expected regret of OFUL restricted to features in $\mathcal{K}$. To discover relevant features ($\mathcal{K}$) we can employ an explore-then-commit strategy which first explores for $\sim \sqrt{T}$ time followed by an exploitation stage such as OFUL restricted to features in $\mathcal{K}$. The rewards in the latter exploitation stage can be divided in two parts,

$$\langle \mathbf{x}, \boldsymbol{\theta}_\star \rangle = \left\langle \mathbf{x}^{\mathcal{K}}, \boldsymbol{\theta}^{\mathcal{K}}_\star \right\rangle + \left\langle \mathbf{x}^{\mathcal{K}^c}, \boldsymbol{\theta}^{\mathcal{K}^c}_\star \right\rangle,$$

where $\mathbf{x}^{\mathcal{K}}$ is the portion of $\mathbf{x}$ restricted to $\mathcal{K}$ and $\mathcal{K} \cup \mathcal{K}^c = [d]$. Similarly, the regret $R^{\mathcal{K}}_{alg}$ can be divided in two parts. Roughly the regret on $\mathcal{K}$ can be bounded by $j\sqrt{T}$ under certain conditions using the OFUL regret bound. For the regret on $\mathcal{K}^c$, suppose each relevant component of $\boldsymbol{\theta}_\star$ has a mean square value of $S^2/k$ (this can be achieved with a sparse gaussian model such as those described in (Deshpande and Montanari 2012)). This yields $\mathbf{E}\|\boldsymbol{\theta}^{\mathcal{K}^c}_\star\|^2 \approx \frac{k-j}{k} S^2$ where $j = |\mathcal{K}|$. The worst-case instantaneous regret bound on $\mathcal{K}^c$ becomes $2\sqrt{(k-j)/k}SL$ leading to an improvement in the linear regret slope by a factor of $\sqrt{(k-j)/k}$ over pure exploration (see Figure 2).

Figure 2(b) shows average regret of OFUL restricted to feature subsets of different sizes with synthetic data ($N = 1000$ actions, $d = 40$ and $k = 10$). For $j \in \{2, 4, \ldots, 10\}$, we randomly picked 100 subsets of size $j$ from the support of $\boldsymbol{\theta}_\star$. We report the average regret of OFUL for a short horizon, $T = 2^8$, restricted to the 100 random subsets. We also plot average regret of OFUL on the full $d = 40$ dimensional data. Figure 2(c) depicts the same with real data from (Poulis and Dasgupta 2017) with $d = 498$ and sparsity, $k = 92$, we choose 100 random subsets of size $j \in \{5, 10, \ldots, 25\}$ from the set of relevant features marked by users (see Section 5 for details) and report the average regret of OFUL restricted to the feature subsets for a relatively short time horizon, $T = 2^{11}$. The plots show that, in the short horizon, it may be more beneficial to use a subset of the relevant features than using the total feature set which may include many irrelevant features. The intuition is that when OFUL has not seen many samples, it does not have enough information to separate the irrelevant dimensions from relevant ones. As time goes on (i.e., for longer horizons) OFUL's relative performance improves since it enjoys sublinear regret but would ultimately be a factor of $d/k$ worse than that of the low-dimensional model that includes all $k$ relevant features.




(a) Synthetic data (b) Synthetic data (c) Data from (Poulis and Dasgupta 2017)

Figure 2: (a) Regret of pure exploration versus explore-then-commit strategy (b,c) Average regret of OFUL restricted to feature subsets (red dots) with 95% confidence regions (blue). The short time horizon was chosen to make the case for restricting the feature space in early rounds. In the long horizon, with more information, the relative performance of OFUL improves, but would ultimately be a factor of $d/k$ worse than that of the low-dimensional model that includes all $k$ relevant features.

# 5 Experiments

## 5.1 Results with Synthetic Data

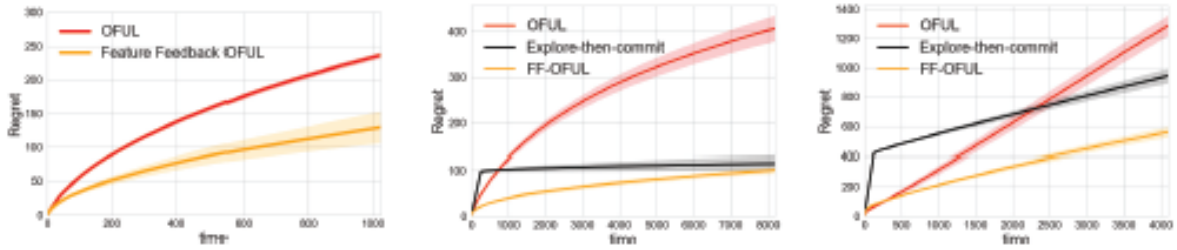
For synthetic data, we simulate a text categorization dataset as follows. Each action corresponds to an article. Generally an article contains only a small subset of the words from the dictionary. Therefore, to simulate documents we generate 1000 sparse actions in 40 dimensions. A 5-sparse reward generating vector, $\boldsymbol{\theta}_*$, is chosen at random. This represents the fact that in reality a document category probably contains only a few relevant words. The features represent word counts and hence are always positive. Here we have access to $\boldsymbol{\theta}_*$ therefore for any action $\mathbf{x}$, we use the standard linear model (1) for the reward $y_t$ with $\eta_t \sim \mathcal{N}(0, R^2)$. The support of $\boldsymbol{\theta}_*$ is taken as the set of oracle relevant words. For every round, each word from the intersection of the support of the action and oracle relevant words is marked as relevant with probability ($p' = 0.1$). Figure 3(a) shows the results averaged over 100 random trials for sparse $\boldsymbol{\theta}_*$ with $k = 5, d = 40$, and 1000 actions. As expected, the FF-OFUL algorithm outperforms standard OFUL significantly. Figure 6(b) in supplement also shows that the feedback does not hurt the performance much for non-sparse $\boldsymbol{\theta}_*$ with $k = d = 40$. Figure 1(b) compares the performance of FF-OFUL with an explore-then-commit strategy.

## 5.2 Results with 20Newsgroup Dataset

We use the 20Newsgroup (20NG) dataset from (Lang 1995). It has $2 \times 10^5$ documents covering 20 topics such as politics, sports. We choose a subset of 5 topics (misc.forsale, rec.autos, sci.med, comp.graphics, talk.politics.mideast) with approximately 4800 documents posted under these topics. For the word counts, we use the TF-IDF features for the documents which give us approximately $d = 47781$ features. For the sake of comparing our method with OFUL, we first report 500 and 1000 dimensional experiments and then on the full $47,781$ dimensional data. To do this, we use logistic regression to train a high accuracy sparse classifier to select 153 features. Then select an additional 847 features at random in order to simulate high dimensional features. We compared OFUL and FF-OFUL algorithms on this data. This is similar to the way (Poulis and Dasgupta 2017) ran experiments in the classification setting. We ran only our algorithm on the full 47781 dimension data since it was infeasible to run OFUL. For the reward model, we pick one of the articles from the database at random as $\boldsymbol{\theta}_*$ and the linear reward model in (1) or use the labels to generate binary, one vs many rewards to simulate search for articles from a certain category. In order to come close to simulating a noisy setting, we used the logistic model, with $q_t = 1/(1 - \exp(-\langle \mathbf{x}_t, \boldsymbol{\theta}_* \rangle))$, $P(y_t = +1) = q_t$.

**Oracle Feedback.** The support of one-vs-many sparse logistic regression is used to get an "oracle set of relevant features" for each class. Each word from the intersection of the support of an action and oracle relevant words was marked as relevant with probability $p'(= 0.1)$. In our theorem statements, $p$ is the probability that the feature is present in a random action *and* it is marked relevant. This depends on the distribution of the words, but typically $\boldsymbol{p \in (0.001, 0.01)}$ and $k \in (30, 100)$ relevant features for each category. Figure 3, compares OFUL, Explore-then-commit and FF-OFUL on the 20NG dataset with oracle feedback. In these simulations averaged over 100 random $\boldsymbol{\theta}_*$, FF-OFUL outperforms OFUL and Explore-then-commit significantly. OFUL parameter was tuned to $\lambda = 2^8$.

**Human Feedback.** (Poulis and Dasgupta 2017) took 50 20Newsgroup articles from 5 categories and had users annotate relevant words. These are the same categories that we used above. This is closer to simulating human feedback since we are not using sparse logistic regression to estimate the sparse vectors. We take the user indicated relevant words instead as the relevance dimensions. There were $k \in (30, 100)$ relevant features for each category. In Figure 4(a), we can see that FF-OFUL is already outperforming OFUL and Explore-then-commit. This is despite the fact that it is not a very sparse regime. Surprisingly, we found




(a) Synthetic data (b) Real data with replacement (c) Real data without replacement

Figure 3: **(a)** Synthetic data with sparse $\theta_*$ ($d = 40$, $k = 5$), FF-OFUL outperforms OFUL significantly. See Figure 1(b) for comparison with explore-then-commit strategy. Newsgroup dataset with oracle feedback: **(b)** This plot shows that FF-OFUL outperforms OFUL and Explore-then-commit when running in $d = 1000$ dimensions, sampling actions with replacement using binary rewards model. **(c)** sampling actions without replacement and using the numerical reward model. Smallest $T_0$ selected such that all relevant features are marked with high probability. Note shorter time horizon for without replacement sampling since $T$ must be less than the number of actions.

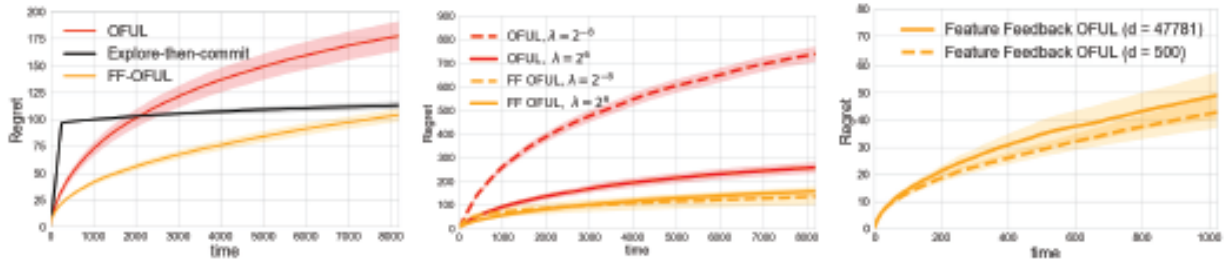



Figure 4: Newsgroup Dataset with Human Feedback: **(Left)** FF-OFUL outperforms OFUL and Explore-then-commit strategy in $d = 500$ dimensions. Both plots generated by tuning the parameter for OFUL. **(Center)** Sensitivity to tuning parameter $\lambda$ seen by the drastic difference in performance of OFUL. In contrast, our FF-OFUL has a relatively modest difference in performance showing its robustness to the ridge regression parameter $\lambda$. **(Right)** Our algorithm for $d = 47781$ and $d = 500$ with ridge parameter $\lambda = 1$, showing its robustness to changes in dimensions and tuning.

that tuning had little effect on the performance of FF-OFUL whereas it had a significant effect on OFUL (see Figure 4). This is possibly due to the implicit regularization provided by gradually growing the number of dimensions as we receive new feedback. FF-OFUL also yields significantly better rewards at early stages by exploiting knowledge of relevant features as soon as they are identified, rather than waiting until all or most are found.

**Parameter Tuning.** For OFUL the ridge parameter ($\lambda$) is tuned from $\{2^i\}_{i=-7}^{10}$ to pick the one with best performance. All the tuned parameters selected for OFUL were strictly inside this range (for $d = 40, k = 5$ , $\lambda = 2^{-5}$ and for $d = 10^3$ (Newsgroup), $\lambda = 2^8$). Figure 4(b) demonstrates the sensitivity of OFUL to change in tuning parameter. For FF-OFUL, the remarkable feature is that it does not require parameter tuning so $\lambda = 1$ for all experiments.

**Full dimension experiments.** Remarkably the performance of FF-OFUL barely drops in full ($d = 47781$) feature dimensions in Figure 4(c). Even though the ridge regression parameter ($\lambda$) for all experiments was not tuned and set to $\lambda = 1$. FF-OFUL is robust to changes in the ambient dimensions and the parameter $\lambda$. Recall that we do not compare with OFUL on 47781 dimensional data since it would require storing and updating a $d \times d$ matrix at each stage.

## Conclusion

In this paper we provide an algorithm that incorporates feature feedback in addition to the standard reward feedback. The framework is based on the following insight: In the short-term horizon, when the algorithm does not have enough feedback, it starts with a small subset of the dimensions and gradually grows the number of dimensions as it receives new feedback. This has three benefits: (1) it makes it possible to use the new algorithm in high-dimensional settings where conventional linear bandits are impractical, (2) it is more robust to choice of parameters because it grows the feature dimensions over time which provides implicit regularization, and (3) it leads to better early-regret. The choice of the dimensions is based on feature feedback provided by the user. This could be generalized in a number of ways, using ideas from compressed sensing and/or dimensionality

reduction, alleviating the need for feature feedback from the user in the future. The goal of this framework is to motivate this idea of growing the feature space over time.

**Acknowledgements** This work was partially supported by AF grant FA8750-17-2-0262 and a grant from American Family Insurance.

## References

Abbasi-Yadkori, Y.; Pal, D.; and Szepesvari, C. 2011. Improved Algorithms for Linear Stochastic Bandits. *Advances in Neural Information Processing Systems (NIPS)* 1–19.

Abbasi-Yadkori, Y.; Pal, D.; and Szepesvari, C. 2012. Online-to-Confidence-Set Conversions and Application to Sparse Stochastic Bandits. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Abe, N., and Long, P. M. 1999. Associative reinforcement learning using linear probabilistic concepts. In *Proceedings of the International Conference on Machine Learning (ICML)*, 3–11.

Auer, P., and Long, M. 2002. Using Confidence Bounds for Exploitation-Exploration Trade-offs. *Journal of Machine Learning Research* 3:2002.

Carpentier, A., and Munos, R. 2012. Bandit theory meets compressed sensing for high dimensional stochastic linear bandit. In *International Conference on Artificial Intelligence and Statistics*, 190–198.

Chen, Y.; Mac Aodha, O.; Su, S.; Perona, P.; and Yue, Y. 2018. Near-optimal machine teaching via explanatory teaching sets. In *International Conference on Artificial Intelligence and Statistics*, 1970–1978.

Croft, W. B., and Das, R. 1989. Experiments with query acquisition and use in document retrieval systems. In *Proceedings of the 13th annual international ACM SIGIR conference on Research and development in information retrieval*, 349–368. ACM.

Dani, V.; Hayes, T. P.; and Kakade, S. M. 2008. Stochastic linear optimization under bandit feedback.

Deshpande, Y., and Montanari, A. 2012. Linear bandits in high dimension and recommendation systems. In *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 1750–1754. IEEE.

Druck, G.; Settles, B.; and McCallum, A. 2009. Active learning by labeling features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, 81–90. Association for Computational Linguistics.

Ginebra, J., and Clayton, M. K. 1995. Response surface bandits. *Journal of the Royal Statistical Society. Series B (Methodological)* 771–784.

Lang, K. 1995. Newsweeder: Learning to filter netnews. In *Machine Learning Proceedings 1995*. Elsevier. 331–339.

Lattimore, T., and Szepesvári, C. 2018. Bandit algorithms.

Poulis, S., and Dasgupta, S. 2017. Learning with feature feedback: from theory to practice. In *Artificial Intelligence and Statistics*, 1104–1113.

Raghavan, H.; Madani, O.; and Jones, R. 2006. Active learning with feedback on features and instances. *Journal of Machine Learning Research* 7(Aug):1655–1686.

Roads, B.; Mozer, M. C.; and Busey, T. A. 2016. Using highlighting to train attentional expertise. *PloS one* 11(1):e0146266.

Rusmevichientong, P., and Tsitsiklis, J. N. 2010. Linearly Parameterized Bandits. *Math. Oper. Res.* 35(2):395–411.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.

Yessenalina, A.; Yue, Y.; and Cardie, C. 2010. Multi-level structured models for document-level sentiment classification. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, 1046–1056. Association for Computational Linguistics.

# MaxGap Bandit: Adaptive Algorithms for Approximate Ranking

**Sumeet Katariya** *
UW-Madison and Amazon
sumeetsk@gmail.com

**Ardhendu Tripathy** *
UW-Madison
astripathy@wisc.edu

**Robert Nowak**
UW-Madison
rdnowak@wisc.edu

## Abstract

This paper studies the problem of adaptively sampling from $K$ distributions (arms) in order to identify the largest gap between any two adjacent means. We call this the MaxGap-bandit problem. This problem arises naturally in approximate ranking, noisy sorting, outlier detection, and top-arm identification in bandits. The key novelty of the MaxGap bandit problem is that it aims to adaptively determine the natural partitioning of the distributions into a subset with larger means and a subset with smaller means, where the split is determined by the largest gap rather than a pre-specified rank or threshold. Estimating an arm's gap requires sampling its neighboring arms in addition to itself, and this dependence results in a novel hardness parameter that characterizes the sample complexity of the problem. We propose elimination and UCB-style algorithms and show that they are minimax optimal. Our experiments show that the UCB-style algorithms require 6-8x fewer samples than non-adaptive sampling to achieve the same error.

## 1 Introduction

Consider an algorithm that can draw i.i.d. samples from $K$ unknown distributions. The goal is to partially rank the distributions according to their (unknown) means. This model encompasses many problems including best-arms identification (BAI) in multi-armed bandits, noisy sorting and ranking, and outlier detection. Partial ranking is often preferred to complete ranking because correctly ordering distributions with nearly equal means is an expensive task (in terms of number of required samples). Moreover, in many applications it is arguably unnecessary to resolve the order of such close distributions. This observation motivates algorithms that aim to recover a partial ordering of groups of distributions having similar means. This entails identifying large "gaps" in the ordered sequence of means. The focus of this paper is the fundamental problem of finding the *largest gap* by sampling adaptively. Identification of the largest gap separates the distributions into two groups, and recursive application can identify any desired number of groupings in a partial order.

As an illustration, consider a subset of images from the Chicago streetview dataset [17] shown in Fig. 1. In this study, people were asked to judge how safe each scene looks [18], and a larger mean indicates a safer looking scene. While each person has a different sense of how safe an image looks, when aggregated there are clear trends in the safety scores (denoted by $\mu_{(i)}$) of the images. Fig. 1 schematically shows the distribution of scores given by people as a bell curve below each image. Assuming the sample means are close to their true means, one can nominally classify them as 'safe', 'maybe unsafe' and 'unsafe' as indicated in Fig. 1. Here we have implicitly used the large gaps $\mu_{(2)} - \mu_{(3)}$ and $\mu_{(4)} - \mu_{(5)}$ to mark the boundaries. Note that finding the safest image (BAI) is hard as we need a lot of human responses to decide the larger mean between the two rightmost distributions; it is also arguably unnecessary. A common way to address this problem is to specify a tolerance $\epsilon$ [7],

* Authors contributed equally and are listed alphabetically.

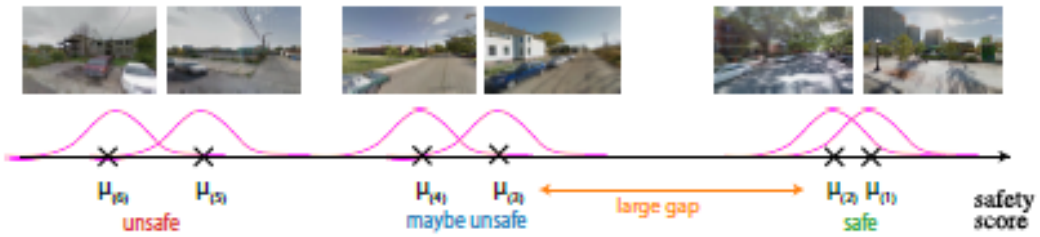



Figure 1: Six representative images from Chicago streetview dataset and their safety (Borda) scores.

and stop sampling if the means are less than $\epsilon$ apart; however determining this can require $\Omega(1/\epsilon^2)$ samples. Distinguishing the top 2 distributions from the rest is easy and can be efficiently done using top-$m$ arm identification [15], however this requires the experimenter to prescribe the location $m = 2$ where a large gap exists which is unknown. *Automatically identifying natural splits in the set of distributions is the aim of the new theory and algorithms we propose. We call this problem of adaptive sampling to find the largest gap the MaxGap-bandit problem.*

### 1.1 Notation and Problem Statement

We will use multi-armed bandit terminology and notation throughout the paper. The $K$ distributions will be called *arms* and drawing a sample from a distribution will be refered to as *sampling the arm*. Let $\mu_i \in \mathbb{R}$ denote the mean of the $i$-th arm, $i \in \{1, 2, \ldots, K\} =: [K]$. We add a parenthesis around the subscript $j$ to indicate the $j$-th largest mean, i.e., $\mu_{(K)} \leq \mu_{(K-1)} \leq \cdots \leq \mu_{(1)}$. For the $i$-th arm, we define its gap $\Delta_i$ to be the maximum of its left and right gaps, i.e.,

$$\Delta_i \;=\; \max\{\mu_{(\ell)} - \mu_{(\ell+1)}\,,\, \mu_{(\ell-1)} - \mu_{(\ell)}\} \quad \text{where } \mu_i = \mu_{(\ell)}. \tag{1}$$

We define $\mu_{(0)} = -\infty$ and $\mu_{(K+1)} = \infty$ to account for the fact that extreme arms have only one gap. The goal of the MaxGap-bandit problem is to (adaptively) sample the arms and return two clusters

$$C_1 = \{(1), (2), \ldots, (m)\} \quad \text{and} \quad C_2 = \{(m+1), \ldots, (K)\},$$

where $m$ is the rank of the arm with the largest gap between *adjacent* means, i.e.,

$$m \;=\; \arg\max_{j \in [K-1]} \mu_{(j)} - \mu_{(j+1)}. \tag{2}$$

The mean values are unknown as is the ordering of the arms according to their means. A solution to the MaxGap-bandit problem is an algorithm which given a probability of error $\delta > 0$, samples the arms and upon stopping partitions $[K]$ into two clusters $\hat{C}_1$ and $\hat{C}_2$ such that

$$\mathbb{P}(\hat{C}_1 \neq C_1) \leq \delta. \tag{3}$$

This setting is known as the fixed-confidence setting [10], and the goal is to achieve the probably correct clustering using as few samples as possible. In the sequel, we assume that $m$ is uniquely defined and let $\Delta_{\max} = \Delta_{i^*}$ where $\mu_{i^*} = \mu_{(m)}$.

### 1.2 Comparison to a Naive Algorithm: Sort then search for MaxGap

The MaxGap-bandit problem is not equivalent to BAI on $\binom{K}{2}$ gaps since the MaxGap-bandit problem requires identifying the largest gap between *adjacent* arm means (BAI on $\binom{K}{2}$ gaps would always identify $\mu_{(1)} - \mu_{(K)}$ as the largest gap). This suggests a naive two-step algorithm: we first sample the arms enough number of times so as to identify all pairs of adjacent arms (i.e., we sort the arms according to their means), and then run a BAI bandit algorithm [13] on the $(K-1)$ gaps between adjacent arms to identify the largest gap (an unbiased sample of the gap can be obtained by taking the difference of the samples of the two arms forming the gap).

We analyze the sample complexity of this naive algorithm in Appendix A, and discuss the results here for an example. Consider the arrangement of means shown in Fig. 2 where there is one large gap $\Delta_{\max}$ and all the other gaps are equal to $\Delta_{\min} \ll \Delta_{\max}$. The naive algorithm's sample complexity is $\Omega(K/\Delta_{\min}^2)$, as the first sorting step requires these many samples, which can be very large.

Is this sorting of the arm means necessary? For instance, we do not need to sort $K$ real numbers in order to cluster them according to the largest gap.[1] The algorithms we propose in this paper solve the MaxGap-bandit problem without necessarily sorting the arm means. For the configuration in Fig. 2 they require $\tilde{O}(K/\Delta_{\max}^2)$ samples, giving a saving of approximately $(\Delta_{\max}/\Delta_{\min})^2$ samples.

Figure 2: Arm means with one large gap.

The analysis of our algorithms suggests a novel hardness parameter for the MaxGap-bandit problem that we discuss next. We let $\Delta_{i,j} := \mu_j - \mu_i$ for all $i, j \in [K]$. We show in Section 5 that the number of samples taken from distribution $i$ due to its right gap is inversely proportional to the square of

$$\gamma_i^r := \max_{j:\Delta_{i,j}>0} \min\left\{\Delta_{i,j}\,,\, \Delta_{\max} - \Delta_{i,j}\right\}. \tag{4}$$

For the left gap of $i$ we define $\gamma_i^l$ analogously. The total number of samples drawn from distribution $i$ is inversely proportional to the square of $\gamma_i := \min\{\gamma_i^r, \gamma_i^l\}$. The intuition for Eq. (4) is that distribution $i$ can be eliminated quickly if there is another distribution $j$ that has a moderately large gap from $i$ (so that this gap can be quickly detected), but not too large (so that the gap is easy to distinguish from $\Delta_{\max}$), and (4) chooses the best $j$. We discuss (4) in detail in Section 5, where we show that our algorithms use $\tilde{O}\left(\sum_{i\in[K]/\{(m),(m+1)\}} \gamma_i^{-2}\log(K/\delta\gamma_i)\right)$ samples to find the largest gap with probability at least $1-\delta$. This sample complexity is minimax optimal.

### 1.3 Summary of Main Results and Paper Organization

In addition to motivating and formulating the MaxGap-bandit problem, we make the following contributions. First, we design elimination and UCB-style algorithms as solutions to the MaxGap-bandit problem that do not require sorting the arm means (Section 3). These algorithms require computing upper bounds on the gaps $\Delta_i$, which can be formulated as a mixed integer optimization problem. We design a computationally efficient dynamic programming subroutine to solve this optimization problem and this is our second contribution (Section 4). Third, we analyze the sample complexity of our proposed algorithms, and discover a novel problem-hardness parameter (Section 5). This parameter arises because of the arm interactions in the MaxGap-bandit problem where, in order to reduce uncertainty in the value of an arm's gap, we not only need to sample the said arm but also its neighboring arms. Fourth, we show that this sample complexity is minimax optimal (Section 6). Finally, we evaluate the empirical performance of our algorithms on simulated and real datasets and observe that they require 6-8x fewer samples than non-adaptive sampling to achieve the same error (Section 7).

## 2 Related Work

One line of related research is best-arm identification (BAI) in multi-armed bandits. A typical goal in this setting is to identify the top-$m$ arms with largest means, where $m$ is a *prespecified* number [15, 16, 1, 3, 9, 4, 14, 7, 20]. As explained in Section 1, our motivation behind formulating the MaxGap-bandit problem is to have an adaptive algorithm which finds the "natural" set of top arms as delineated by the largest gap in consecutive mean values. Our work can also be used to automatically detect "outlier" arms [23].

The MaxGap-bandit problem is different from the standard multi-armed bandit because of the local dependence of an arm's gap on other arms. Other best-arm settings where an arm's reward can inform the quality of other arms include linear bandits [22] and combinatorial bandits [5, 11]. In these problems, the decision space is known to the learner, i.e., the vectors corresponding to the arms in linear bandits and the subsets of arms over which the objective function is to be optimized in combinatorial bandits is known to the learner. However in our problem, we do not know the sorted order of the arm means, i.e., the set of all valid gaps is unknown *a priori*. Our problem does not reduce to these settings.

[1] First find the smallest and largest numbers, say $a$ and $b$ respectively. Divide the interval $[a, b]$ into $K+1$ equal-width bins and map each number to its corresponding bin, while maintaining the smallest and largest number in each bin. Since at least one bin is empty by the pigeonhole principle, the largest gap is between two numbers belonging to different bins. Calculate all gaps between bins and cluster based on the largest of those.

Another related problem is noisy sorting and ranking. Here the typical goal is to sort a list using noisy pairwise comparisons. Our framework encompasses noisy ranking based on Borda scores [1]. The Borda score of an item is the probability that it is ranked higher in a pairwise comparison with another item chosen uniformly at random. In our setting, the Borda score is the mean of each distribution. Much of the theoretical computer science literature on this topic assumes a bounded noise model for comparisons (i.e., comparisons are probably correct with a positive margin) [8, 6, 2, 21]. This is unrealistic in many real-world applications since near equals or outright ties are not uncommon. The largest gap problem we study can be used to (partially) order items into two natural groups, one with large means and one with small means. Previous related work considered a similar problem with prescribed (non-adaptive) quantile groupings [18].

## 3 MaxGap Bandit Algorithms

We propose elimination [7] and UCB [13] style algorithms for the MaxGap-bandit problem. These algorithms operate on the arm *gaps* instead of the arm *means*. The subroutine to construct confidence intervals on the gaps (denoted by $\mathtt{U}\Delta_a(t)$) using confidence intervals on the arm means (denoted by $[l_a(t), r_a(t)]$) is described in Algorithm 4 in Section 4, and this subroutine is used by all three algorithms described in this section.

### 3.1 Elimination Algorithm: `MaxGapElim`

At each time step, `MaxGapElim` (Algorithm 1) samples all arms in an active set consisting of arms $a$ whose gap upper bound $\mathtt{U}\Delta_a$ is larger than the global lower bound $\mathtt{L}\Delta$ on the maximum gap, and stops when there are only two arms in the active set.

**Algorithm 1** `MaxGapElim`

Initialize active set $A = [K]$
**for** $t = 1, 2, \ldots$ **do** *// rounds*
  $\forall a \in A$, sample arm $a$, compute $[l_a(t), r_a(t)]$ using (5). *//arm confidence intervals*
  $\forall a \in A$, compute $\mathtt{U}\Delta_a(t)$ using Algorithm 4. *// upper bound on arm max gap*
  Compute $\mathtt{L}\Delta(t)$ using (9). *// lower bound on max gap*
  $\forall a \in A$, if $\mathtt{U}\Delta_a(t) \leq \mathtt{L}\Delta(t), A = A \setminus a$. *// Elimination*
  If $|A| = 2$, stop. Return clusters using max gap in the empirical means. *// Stopping condition*

### 3.2 UCB algorithms: `MaxGapUCB` and `MaxGapTop2UCB`

`MaxGapUCB` (Algorithm 2) is motivated from the principle of "optimism in the face of uncertainty". It samples *all* arms with the highest gap upper bound. Note that there are at least two arms with the highest gap upper bound because any gap is shared by at least two arms (one on the right and one on the left). The stopping condition is akin to the stopping condition in Jamieson et al. [13].

**Algorithm 2** `MaxGapUCB`

Initialize $\mathcal{U} = [K]$.
**for** $t = 1, 2, \ldots$ **do**
  $\forall a \in \mathcal{U}$, sample $a$ and update $[l_a(t), r_a(t)]$ using (5).
  $\forall a \in [K]$, compute $\mathtt{U}\Delta_a(t)$ using Algorithm 4.
  Let $M_1(t) = \max_{j \in [K]} \mathtt{U}\Delta_j(t)$. Set $\mathcal{U} = \{a : \mathtt{U}\Delta_a(t) = M_1(t)\}$. *// highest gap-UCB arms*
  If $\exists i, j$ such that $T_i(t) + T_j(t) \geq c \sum_{a \notin \{i,j\}} T_a(t)$, stop. *// stopping condition*

Alternatively, we can use an LUCB [16]-type algorithm that samples arms which have the two highest gap upper bounds, and stops when the second-largest gap upper bound is smaller than the global lower bound $\mathtt{L}\Delta(t)$ . We refer to this algorithm as `MaxGapTop2UCB` (Algorithm 3).

**Algorithm 3** MaxGapTop2UCB

Initialize $\mathcal{U}_1 \cup \mathcal{U}_2 = [K]$.
**for** $t = 1, 2, \ldots$ **do**
  $\forall a \in \mathcal{U}_1 \cup \mathcal{U}_2$, sample $a$ and update $[l_a(t), r_a(t)]$ using (5).
  $\forall a \in [K]$, compute $\mathtt{U}\Delta_a(t)$ using Algorithm 4.
  Let $M_1(t) = \max_{j \in [K]} \mathtt{U}\Delta_j(t)$. Set $\mathcal{U}_1 = \{a : \mathtt{U}\Delta_a(t) = M_1(t)\}$. *// highest gap-UCB arms*
  Let $M_2(t) = \max_{j \in [K] \setminus \mathcal{U}_1} \mathtt{U}\Delta_j(t)$. Set $\mathcal{U}_2 = \{a : \mathtt{U}\Delta_a(t) = M_2(t)\}$. *// 2nd highest gap-UCB*
  Compute $\mathtt{L}\Delta(t)$ using (9). If $M_2(t) < \mathtt{L}\Delta(t)$, stop.

**Algorithm 4** Procedure to find $\mathtt{U}\Delta_a(t)$

Set $P_a^r = \{i : l_i(t) \in [l_a(t), r_a(t)]\}$.
$\mathtt{U}\Delta_a^r(t) = \max_{i \in P_a^r} \{G_a^r(l_i(t), t)\}$, where $G_a^r(x,t)$ is given by (7). *// eqn. (8)*
Set $P_a^l = \{i : r_i(t) \in [l_a(t), r_a(t)]\}$.
$\mathtt{U}\Delta_a^l(t) = \max_{i \in P_a^l} \{G_a^l(r_j(t), t)\}$, where $G_a^l(x,t)$ is given by (19). *// eqn. (20)*
**return** $\mathtt{U}\Delta_a(t) \leftarrow \max\{\mathtt{U}\Delta_a^r(t), \mathtt{U}\Delta_a^l(t)\}$

## 4 Confidence Bounds for Gaps

In this section we explain how to construct confidence bounds for the arm gaps (denoted by $\mathtt{U}\Delta_a$ and $\mathtt{L}\Delta$) using confidence bounds for the arm means (denoted by $[l_a, r_a]$). These bounds are key ingredients for the algorithms described in Section 3.

Given i.i.d. samples from arm $a$, an empirical mean $\hat{\mu}_a$ and confidence interval on the arm mean can be constructed using standard methods. Let $T_a(t)$ denote the number of samples from arm $a$ after $t$ time steps of the algorithm. Throughout our analysis and experimentation we use confidence intervals on the mean of the form

$$l_a(t) = \hat{\mu}_a(t) - c_{T_a(t)} \text{ and } r_a(t) = \hat{\mu}_a(t) + c_{T_a(t)}, \text{ where } c_s = \sqrt{\frac{\log(4Ks^2/\delta)}{s}}. \tag{5}$$

The confidence intervals are chosen so that [12]

$$\mathbb{P}(\forall t \in \mathbb{N}, \forall a \in [K], \mu_a \in [l_a(t), r_a(t)]) \geq 1 - \delta. \tag{6}$$

Conceptually, the confidence intervals on the arm means can be used to construct upper confidence bounds on the mean gaps $\{\Delta_i\}_{i \in [K]}$ in the following manner. Consider all possible configurations of the arm means that satisfy the confidence interval constraints in (5). Each configuration fixes the gaps associated with any arm $a \in [K]$. Then the maximum gap value over all configurations is the upper confidence bound on arm $a$'s gap; we denote it as $\mathtt{U}\Delta_a$. The above procedure can be formulated as a mixed integer linear program (see Appendix B.1). In the algorithms in Section 3, this optimization problem needs to be solved at every time $t$ and for every arm $a \in [K]$ before querying a new sample, which can be practically infeasible. In Algorithm 4, we give an efficient $O(K^2)$ time dynamic programming algorithm to compute $\mathtt{U}\Delta_a$. We next explain the main ideas used in this algorithm, and refer the reader to Appendix B.2 for the proofs.

Each arm $a$ has a right and left gap, $\Delta_a^r := \mu_{(\ell-1)} - \mu_{(\ell)}$ and $\Delta_a^l := \mu_{(\ell)} - \mu_{(\ell+1)}$, where $\ell$ is the rank of $a$, i.e., $\mu_a = \mu_{(\ell)}$. We construct separate upper bounds $\mathtt{U}\Delta_a^r(t)$ and $\mathtt{U}\Delta_a^l(t)$ for these gaps and then define $\mathtt{U}\Delta_a(t) = \max\{\mathtt{U}\Delta_a^r(t), \mathtt{U}\Delta_a^l(t)\}$. Here we provide an intuitive description for how the bounds are computed, focusing on $\mathtt{U}\Delta_a^r(t)$ as an example. To start, suppose the true mean of arm $a$ is known exactly, while the means of other arms are only known to lie within their confidence intervals. If there exist arms that cannot go to the left of arm $a$, one can see that the largest right gap for $a$ is obtained by placing all arms that can go to the left of $a$ at their leftmost positions, and all remaining arms at their rightmost positions, as shown in Fig. 3(a). If however all arms can go to the left of arm $a$, the configuration that gives the largest right gap for $a$ is obtained by placing the arm with the largest upper bound at its right boundary, and all other arms at their left boundaries, as illustrated in Fig. 3(b). We define a function $G_a^r(x,t)$ that takes as input a known position $x$ for the mean of arm $a$

(a) (b)

Figure 3: Computing maximum right gap of blue arm when its true mean is known (at position indicated by blue x), while the other means are known only to lie within their confidence intervals. (a) If there exist arms that cannot go to the left of blue (red, green, purple), the largest right gap for blue is obtained by placing all arms that can go to the left of blue at their left boundaries and the remaining arms at their rightmost positions. (b) If all arms can go to the left of blue, the largest right gap for blue is obtained by placing the arm with the largest right confidence bound (purple) at its right boundary and all other arms at their left boundaries.

and the confidence intervals of all other arms at time $t$, and returns the maximum right gap for arm $a$ using the above idea as follows.

$$G_a^r(x,t) = \begin{cases} \min_{j:l_j(t)>x} r_j(t) - x & \text{if } \{j : l_j(t) > x\} \neq \emptyset, \\ \max_{j\neq a} r_j(t) - x & \text{otherwise.} \end{cases} \tag{7}$$

However, the true mean of arm $a$ is not known exactly but only that it lies within its confidence interval. The insight that helps here is that $G_a^r(x,t)$ must achieve its maximum when $x$ is at one of the finite locations in $\{l_j(t) : l_a(t) \leq l_j(t) \leq r_a(t)\}$. We define $P_a^r := \{j : l_a(t) \leq l_j(t) \leq r_a(t)\}$ as the set of arms relevant for the right gap of $a$, and then the maximum possible right gap of $a$ is

$$\mathtt{U}\Delta_a^r(t) = \max\{G_a^r(l_j(t),t) : j \in P_a^r\}. \tag{8}$$

An upper bound for the left gap $\mathtt{U}\Delta_a^l$ can be similarly obtained. We explain this and give a proof of correctness in Appendix B.2.

The algorithms also use a single global lower bound on the maximum gap. To do so, we sort the items according to their empirical means, and find partitions of items that are clearly separated in terms of their confidence intervals. At time $t$, let $(i)_t$ denote the arm with the $i^{\text{th}}$-largest empirical mean, i.e., $\widehat{\mu}_{(K)_t}(t) \leq \ldots \widehat{\mu}_{(2)_t}(t) \leq \widehat{\mu}_{(1)_t}(t)$ (this can be different from the true ranking which is denoted by $(\cdot)$ without the subscript $t$). We *detect* a nonzero gap at arm $k$ if $\max_{a\in\{(k+1)_t,\ldots,(K)_t\}} r_a(t) < \min_{a\in\{(1)_t,\ldots,(k)_t\}} l_a(t)$. Thus, a lower bound on the largest gap is

$$\mathtt{L}\Delta(t) = \max_{k\in[K-1]} \left( \min_{a\in\{(1)_t,\ldots,(k)_t\}} l_a(t) - \max_{a\in\{(k+1)_t,\ldots,(K)_t\}} r_a(t) \right). \tag{9}$$

## 5 Analysis

In this section, we first state the accuracy and sample complexity guarantees for `MaxGapElim` and `MaxGapUCB`, and then discuss our results. The proofs can be found in the Supplementary material.

**Theorem 1.** *With probability* $1-\delta$, `MaxGapElim`, `MaxGapUCB` *and* `MaxGapTop2UCB` *cluster the arms according to the maximum gap, i.e., they satisfy* (3).

The number of times arm $a$ is sampled by both the algorithms depends on $\gamma_a = \min\{\gamma_a^l, \gamma_a^r\}$ where

$$\gamma_a^r = \max_{j:0<\Delta_{a,j}<\Delta_{\max}} \min\{\Delta_{a,j}, (\Delta_{\max} - \Delta_{a,j})\} \tag{10}$$

$$\gamma_a^l = \max_{j:0<\Delta_{j,a}<\Delta_{\max}} \min\{\Delta_{j,a}, (\Delta_{\max} - \Delta_{j,a})\},. \tag{11}$$

The maxima is assumed to be $\infty$ in (10) and (11) if there is no $j$ that satisfies the constraint to account for edge arms. The quantity $\gamma_a$ acts as a measure of hardness for arm $a$; Theorem 2 states that `MaxGapElim` and `MaxGapUCB` sample arm $a$ at most $\tilde{O}(1/\gamma_a^2)$ number of times (up to log factors).

**Theorem 2.** *With probability* $1-\delta$*, the sample complexity of* MaxGapElim *and* MaxGapUCB *is bounded by*

$$O\left(\sum_{a\in[K]\setminus\{(m),(m+1)\}} \frac{\log(K/\delta\gamma_a)}{\gamma_a^2}\right)$$

Next, we provide intuition for why the sample complexity depends on the parameters in (10) and (11). In particular, we show that $O((\gamma_a^r)^{-2})$ (resp. $O((\gamma_a^l)^{-2})$) is the number of samples of $a$ required to rule out arm $a$'s right (resp. left) gap from being the largest gap.

Let us focus on the right gap for simplicity. To understand how (10) naturally arises, consider Fig. 4, which denotes the confidence intervals on the means at some time $t$. A lower bound on the gap $\mathtt{L}\Delta(t)$ can be computed between the left and right confidence bounds of arms 10 and 11 respectively as shown. Consider the computation of the upper bound $\mathtt{U}\Delta_7^r(t)$ on the right gap of arm $a=7$. Arm 4 lies to the right of arm 7 with high probability (unlike the arms with dashed confidence intervals), so the upper bound $\mathtt{U}\Delta_7^r(t) \le r_4(t) - l_7(t)$. Considering only the right gap for simplicity, as soon as $\mathtt{U}\Delta_7^r(t) < \mathtt{L}\Delta(t)$, arm 7 can be eliminated as a candidate for the maximum gap. Thus, an arm $a$ is removed from consideration as soon as we find a *helper* arm $j$ (arm 4 in Fig. 4) that satisfies two properties: (1) the confidence interval of arm $j$ is disjoint from that of arm $a$, and (2) the upper bound $\mathtt{U}\Delta_a^r(t) = r_j(t) - l_a(t) < \mathtt{L}\Delta(t)$. The first of these conditions gives rise to the term $\Delta_{a,j}$ in (10), and the second condition gives rise to the term $(\Delta_{\max} - \Delta_{a,j})$. Since any arm $j$ that satisfies these conditions can act as a helper for arm $a$, we take the maximum over all arms $j$ to yield the smallest sample complexity for arm $a$.

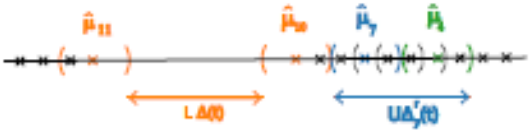


Figure 4: Arm $a=7$ is eliminated when a helper arm $j=4$ is found.

This also shows that if all arms are either very close to $a$ or at a distance approximately $\Delta_{\max}$ from $a$, then the upper bound $\mathtt{U}\Delta_7^r(t) = r_4(t) - l_7(t) > \mathtt{L}\Delta(t)$ and arm 7 cannot be eliminated. Thus arm $a$ could have a small gap with respect to its adjacent arms, but if there is a large gap in the vicinity of arm $a$, it cannot be eliminated quickly. This illustrates that the maximum gap identification problem is not equivalent to best-arm identification (BAI) on gaps. Section 6 formalizes this intuition.

**Key Differences compared to BAI Analysis:** The analysis of MaxGapUCB is very different from the standard UCB analysis. On a high-level, in BAI, the number of samples of a sub-optimal arm $i$ is bounded by observing that

$$\begin{aligned} \text{Arm } i \text{ is pulled} &\implies \mu_i + 2c_{T_i(t)} \ge \hat{\mu}_i + c_{T_i(t)} \ge \hat{\mu}_{(1)} + c_{T_{(1)}(t)} \ge \mu_{(1)} \\ &\implies 2c_{T_i(t)} \ge \mu_{(1)} - \mu_i = \Delta_i. \end{aligned} \tag{12}$$

The last inequality *directly* bounds the number of samples $T_i(t)$ of a sub-optimal arm $i$. In MaxGapUCB, the gap upper bound is obtained using the confidence intervals of two arms, and the fact that a sub-optimal gap $(i,j)$ has the highest gap-UCB implies that

$$\begin{aligned} (\mu_j + 2c_{T_j(t)}) - (\mu_i - 2c_{T_i(t)}) \ge (\hat{\mu}_j + c_{T_j(t)}) - (\hat{\mu}_i - 2c_{T_i(t)}) \ge \Delta_{\max} \\ \implies 2(c_{T_j(t)} + c_{T_i(t)}) \ge \Delta_{\max} - \Delta_{ij}. \end{aligned}$$

Thus unlike the reasoning in (12), the number of samples from arm $i$ is coupled to the number of samples from arm $j$, and $T_i(t) \to \infty$ if $j$ is not sampled enough. We show in our analysis that this cannot happen in MaxGapUCB. Furthermore, any arm $i$ is coupled with multiple other arms since the ordering of the arms is unknown, and may have to be sampled even if its own gap is small - a phenomenon absent in standard BAI analysis because of the independence of the arm means. In our proof, we account for all samples of an arm by defining states the arm can belong to (called levels), and arguing about the confidence intervals of the arms in unison.

## 6 Minimax Lower Bound

In this section, we demonstrate that the MaxGap problem is fundamentally different from best-arm identification (BAI) on gaps. We construct a problem instance and prove a lower bound on the number of samples needed by any probably correct algorithm. The lower bound matches the upper bounds in the previous section for this instance.

**Lemma 1.** *Consider a model $\mathcal{B}$ with $K = 4$ normal distributions $\mathcal{P}_i = \mathcal{N}(\mu_i, 1)$, where*

$$\mu_4 = 0, \quad \mu_3 = \epsilon, \quad \mu_2 = \nu + 2\epsilon, \quad \mu_1 = 2\nu + 2\epsilon,$$

*for some $\nu \gg \epsilon > 0$. Then any algorithm that is correct with probability at least $1 - \delta$ must collect $\Omega(1/\epsilon^2)$ samples of arm 4 in expectation.*

*Proof Outline:* The proof uses a standard change of measure argument [10]. We construct another problem instance $\mathcal{B}'$ which has a different maximum gap clustering compared to $\mathcal{B}$ (see Fig. 5, the maxgap clustering in $\mathcal{B}$ is $\{4,3\} \cup \{2,1\}$, while the maxgap clustering in $\mathcal{B}'$ is $\{4,3,2\} \cup \{1\}$), and show that in order to distinguish between $\mathcal{B}$ and $\mathcal{B}'$, any probably correct algorithm must collect at least $\Omega(1/\epsilon^2)$ samples of arm 4 in expectation (see Appendix E for details). From the definition of $\gamma_a$ using (10),(11), it is easy to check that $\gamma_4 = \epsilon$. Therefore, for problem instance $\mathcal{B}$ our algorithms find the maxgap clustering using at most $O(\log(\epsilon/\delta)/\epsilon^2)$ samples of arm 4 (*c.f.* Theorem 2). This essentially matches the lower bound above.

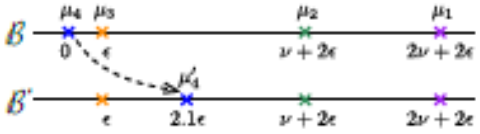



Figure 5: Changing the original bandit model $\mathcal{B}$ to $\mathcal{B}'$. $\mu_4$ is shifted to the right by $2.1\epsilon$. As a result, the maximum gap in $\mathcal{B}'$ is between green and purple.

This example illustrates why the maximum gap identification problem is different from a simple BAI on gaps. Suppose an oracle told a BAI algorithm the ordering of the arm means. Using the ordering it can convert the 4-arm maximum gap problem $\mathcal{B}$ to a BAI problem on 3 *gaps*, with distributions $\mathcal{P}_{4,3} = \mathcal{N}(\epsilon, 2)$, $\mathcal{P}_{3,2} = \mathcal{N}(\nu + \epsilon, 2)$, and $\mathcal{P}_{2,1} = \mathcal{N}(\nu, 2)$. The BAI algorithm can sample arms $i$ and $i+1$ to get a sample of the gap $(i+1, i)$. We know from standard BAI analysis [13] that the gap $(4,3)$ can be eliminated from being the largest by sampling it (and hence arm 4) $O(1/\nu^2)$ times, which can be arbitrarily lower than the $1/\epsilon^2$ lower bound in Lemma 1. Thus the ordering information given to the BAI algorithm is crucial for it to quickly identify the larger gaps. The problem we solve in this paper is identifying the maximum gap when the ordering information is *not* available.

## 7 Experiments

We conduct three experiments. First, we verify the validity of our sample complexity bounds in Section 7.1. We then study the performance of our adaptive algorithms on simulated data in Section 7.2, and on the Streetview dataset in Section 7.3. The code for all experiments is publicly available [19].

### 7.1 Sample Complexity

In Fig. 6(b) and Fig. 6(c), we plot the empirical stopping time against the theoretical sample complexity (Theorem 2) for different arm configurations. We choose the arm configuration in Fig. 6(a) containing $K = 15$ arms and three unique gaps - a small gap $\Delta_3$ and two large gaps $\Delta_2 < \Delta_1 = \Delta_{\max} = 0.4$. The hardness parameter is changed by increasing $\Delta_2$ (from 0.35 to 0.39) and bringing it closer to $\Delta_1$. The rewards are normally distributed with $\sigma = 0.05$. We see a linear relationship in Fig. 6(b) which suggests that the sample complexity expression in Theorem 2 is correct up to constants. In Fig. 6(c) we include random sampling and see that our adaptive algorithms require up to 5x fewer samples when run until completion. Fig. 6(c) also shows that our adaptive algorithms always outperform random sampling, and the gains increase with hardness. We used a lower bound based stopping condition for Random, Elimination, Top2UCB, and set $c = 5$ in the UCB stopping condition (value of $c$ chosen empirically as in [13]).

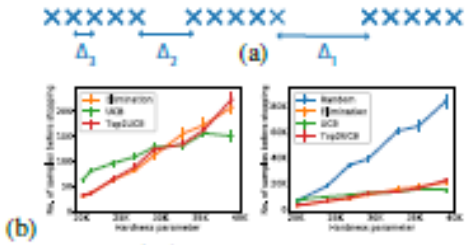



Figure 6: Stopping time experiments.

### 7.2 Simulated Data

In the second experiment, we study the performance on a simulated set of means containing two large gaps. The mean distribution plotted in Fig. 7(a) has $K = 24$ arms ($\mathcal{N}(\cdot, 1)$), with two large mean

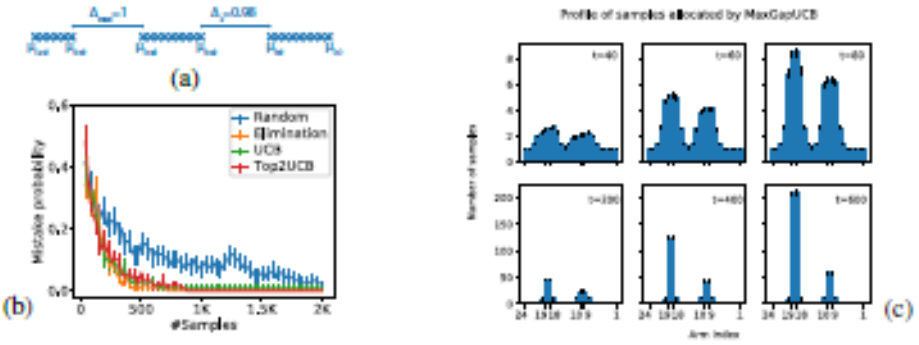



Figure 7: (a) Two large gaps. (b) Clustering error probability for means shown in Fig. 7(a). (c) The profile of samples allocated by MaxGapUCB to each arm in (a) at different time steps.

gaps $\Delta_{10,9} = 0.98, \Delta_{19,18} = 1.0$, and remaining small gaps ($\Delta_{i+1,i} = 0.2$ for $i \notin \{9, 18\}$). We expect to see a big advantage for adaptive sampling in this example because almost every sub-optimal arm has a *helper* arm (see Section 5) which can help eliminate it quickly, and adaptive algorithms can then focus on distinguishing the two large gaps. A non-adaptive algorithm on the other hand would continue sampling all arms. We plot the fraction of times $\hat{C}_1 \neq \{1, \ldots, 18\}$ in 120 runs in Fig. 7(b), and see that the active algorithms identify the largest gap in 8x fewer samples. To visualize the adaptive allocation of samples to the arms, we plot in Fig. 7(c) the number of samples queried for each arm at different time steps by MaxGapUCB. Initially, MaxGapUCB allocates samples uniformly over all the arms. After a few time steps, we see a bi-modal profile in the number of samples. Since all arms that achieve the largest UΔ are sampled, we see that several arms that are near the pairs $(10, 9)$ and $(19, 18)$ are also sampled frequently. As time progresses, only the pairs $(10, 9)$ and $(19, 18)$ get sampled, and eventually more samples are allocated to the larger gap $(19, 18)$ among the two.

### 7.3 Streetview Dataset

For our third experiment we study performance on the Streetview dataset [17, 18] whose means are plotted in Fig. 8(a). We have $K = 90$ arms, where each arm is a normal distribution with mean equal to the Borda safety score of the image and standard deviation $\sigma = 0.05$. The largest gap of 0.029 is between arms 2 and 3, and the second largest gap is 0.024. In Fig. 8(b), we plot the fraction of times $\hat{C}_1 \neq \{1, 2\}$ in 120 runs as a function of the number of samples, for four algorithms, viz., random (non-adaptive) sampling, MaxGapElim, MaxGapUCB, and MaxGapTop2UCB. The error bars denote standard deviation over the runs. MaxGapUCB and MaxGapTop2UCB require 6-7x fewer samples than random sampling.

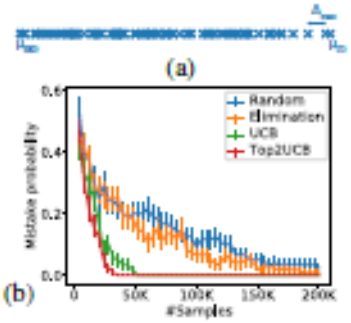



Figure 8: (a) Borda safety scores for Streetview images. (b) Probability of returning a wrong cluster.

## 8 Conclusion

In this paper, we proposed the MaxGap-bandit problem: a novel maximum-gap identification problem that can be used as a basic primitive for clustering and approximate ranking. Our analysis shows a novel hardness parameter for the problem, and our experiments show 6-8x gains compared to non-adaptive algorithms. We use simple Hoeffding based confidence intervals in our analysis for simplicity, but better bounds can be obtained using tighter confidence intervals [13]. Several extensions of this basic problem are possible. An $\epsilon$-relaxation of the MaxGap Bandit is useful when the largest and second-largest gaps are close to each other. Other possibilities include identifying the largest gap within a top quantile of the arms, or clustering with a constraint that the returned clusters are of similar cardinality. All of these extensions will likely require new ideas, as it is unclear how to obtain a lower bound for the gap associated with every arm. Finding an instance-dependent lower bound for MaxGap-bandit is an intriguing problem. Finally, one way to cluster the distributions into more than two clusters is to apply the max-gap identification algorithms recursively; however it would be interesting to come up with algorithms that can perform this clustering directly.

## Acknowledgments

Ardhendu Tripathy would like to thank Ervin Tánczos for helpful discussions. The authors would also like to thank the reviewers for their comments and suggestions. This work was partially supported by AFOSR/AFRL grants FA8750-17-2-0262 and FA9550-18-1-0166.

## References

[1] Arpit Agarwal, Shivani Agarwal, Sepehr Assadi, and Sanjeev Khanna. Learning with limited rounds of adaptivity: Coin tossing, multi-armed bandits, and ranking from pairwise comparisons. In *Conference on Learning Theory*, pages 39–75, 2017.

[2] Mark Braverman, Jieming Mao, and S Matthew Weinberg. Parallel algorithms for select and partition with noisy comparisons. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 851–862. ACM, 2016.

[3] Sebastien Bubeck, Tengyao Wang, and Nitin Viswanathan. Multiple identifications in multi-armed bandits. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, pages I–258–I–265. JMLR.org, 2013. URL http://dl.acm.org/citation.cfm?id=3042817.3042848

[4] Lijie Chen, Jian Li, and Mingda Qiao. Nearly instance optimal sample complexity bounds for top-k arm selection. In *Artificial Intelligence and Statistics*, pages 101–110, 2017.

[5] Shouyuan Chen, Tian Lin, Irwin King, Michael R Lyu, and Wei Chen. Combinatorial pure exploration of multi-armed bandits. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 379–387. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5433-combinatorial-pure-exploration-of-multi-armed-bandits.pdf

[6] Susan Davidson, Sanjeev Khanna, Tova Milo, and Sudeepa Roy. Top-k and clustering with noisy comparisons. *ACM Transactions on Database Systems (TODS)*, 39(4):35, 2014.

[7] Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of machine learning research*, 7(Jun):1079–1105, 2006.

[8] Uriel Feige, Prabhakar Raghavan, David Peleg, and Eli Upfal. Computing with noisy information. *SIAM Journal on Computing*, 23(5):1001–1018, 1994.

[9] Victor Gabillon, Mohammad Ghavamzadeh, and Alessandro Lazaric. Best arm identification: A unified approach to fixed budget and fixed confidence. In *Advances in Neural Information Processing Systems*, pages 3212–3220, 2012.

[10] Aurélien Garivier and Emilie Kaufmann. Optimal best arm identification with fixed confidence. In *Conference on Learning Theory*, pages 998–1027, 2016.

[11] Weiran Huang, Jungseul Ok, Liang Li, and Wei Chen. Combinatorial pure exploration with continuous and separable reward functions and its applications. In *IJCAI*, volume 18, pages 2291–2297, 2018.

[12] Kevin Jamieson and Robert Nowak. Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting. In *2014 48th Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6. IEEE, 2014.

[13] Kevin Jamieson, Matthew Malloy, Robert Nowak, and Sébastien Bubeck. lil'ucb: An optimal exploration algorithm for multi-armed bandits. In *Conference on Learning Theory*, pages 423–439, 2014.

[14] Kwang-Sung Jun, Kevin G Jamieson, Robert D Nowak, and Xiaojin Zhu. Top arm identification in multi-armed bandits with batch arm pulls. In *AISTATS*, pages 139–148, 2016.

[15] Shivaram Kalyanakrishnan and Peter Stone. Efficient selection of multiple bandit arms: Theory and practice. In *ICML*, volume 10, pages 511–518, 2010.

[16] Shivaram Kalyanakrishnan, Ambuj Tewari, Peter Auer, and Peter Stone. Pac subset selection in stochastic multi-armed bandits. In *ICML*, volume 12, pages 655–662, 2012.

[17] Sumeet Katariya, Lalit Jain, Nandana Sengupta, James Evans, and Robert Nowak. Chicago streetview dataset. 2018. URL https://github.com/sumeetsk/coarse_ranking/.

[18] Sumeet Katariya, Lalit Jain, Nandana Sengupta, James Evans, and Robert Nowak. Adaptive sampling for coarse ranking. In *International Conference on Artificial Intelligence and Statistics*, pages 1839–1848, 2018.

[19] Sumeet Katariya, Ardhendu Tripathy, and Robert Nowak. Code for maxgap bandit algorithms and experiments. 2019. URL https://github.com/sumeetsk/maxgap_bandit.

[20] Shie Mannor and John N Tsitsiklis. The sample complexity of exploration in the multi-armed bandit problem. *Journal of Machine Learning Research*, 5(Jun):623–648, 2004.

[21] Cheng Mao, Jonathan Weed, and Philippe Rigollet. Minimax rates and efficient algorithms for noisy sorting. In Firdaus Janoos, Mehryar Mohri, and Karthik Sridharan, editors, *Proceedings of Algorithmic Learning Theory*, volume 83 of *Proceedings of Machine Learning Research*, pages 821–847. PMLR, 07–09 Apr 2018. URL http://proceedings.mlr.press/v83/mao18a.html.

[22] Marta Soare, Alessandro Lazaric, and Rémi Munos. Best-arm identification in linear bandits. In *Advances in Neural Information Processing Systems*, pages 828–836, 2014.

[23] Honglei Zhuang, Chi Wang, and Yifan Wang. Identifying outlier arms in multi-armed bandit. In *Advances in Neural Information Processing Systems*, pages 5204–5213, 2017.

## LIST OF SYMBOLS, ABBERVIATIONS, AND ACRONYMS

| | |
|---|---|
| AF | Air Force |
| DEML | Data-Efficient Machine Learning |
| DoD | Department of Defense |
| ML | Machine Learning |
| SVM | Support Vector Machine |
| UCB | Upper Confidence Bound |