Bandit Learning for Sequential Decision Making

A practical way to address the trade-off between exploration and exploitation

Meng Fang

Faculty of Engineering and Information Technology

University of Technology, Sydney

This dissertation is submitted for the degree of

Doctor of Philosophy

October 2015

To my loving parents.

Declaration

I hereby declare that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text. I also declare that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Meng Fang
October 2015

Acknowledgements

There are so many people to thank for helping me during my PhD candidate. So many have made my study in Sydney a lot easier and happier than I thought it was going to be. I wish to express appreciation to all of them.

First, I would like to thank Professor Dacheng Tao for his guidance, encouragement, and patience. Thank you so much for encouraging me to look at research and my work in different ways and for opening my mind. I am so lucky to have Professor Dacheng Tao as my adviser. His great support was essential to my success here.

I would like to thank Professor Xingquan Zhu and Scientist Jie Yin for helping my research and taking time to talk with me on many occasions. I would like to thank Professor Shuliang Wang for introducing Artifical Intelligence research to me.

I would like to thank the members of my faculty: Prof. Chengqi Zhang, Dr. Bin Li, Dr. Lin Chen, Dr. Lu Qin. I learned a lot from discussion with them.

I have been fortunate to work in a group gathering the most brilliant researchers and best friends in the past 4 years: Dr. Wei Bian, Tianyi Zhou, Zhibin Hong, Maoying Qiao, Tongliang Liu, Mingming Gong, Nannan Wang, Associate Professor Weifeng Liu, Associate Professor Bo Du, Ruxin Wang, Qiang Li, Changxin Ding, Zhe Xu, Shaoli Wang, Shujuan Hou, Chang Xu, Chen Gong, Baosheng Yu, Yali Du.

I am also grateful to all the other friends who made my four years at Sydney unforgettable: Chunyang Liu, Bozhong Liu, Shirui Pan, Mingsong Mao, Hongshu Chen, Guodong Long, Jing Jiang, Yifan Fu, Lianhua Chi, Jia Wu, Dianshuang Wu, Yin Song, Can Wang, and my friend Dong Fang, Junhan Gao and Lanfeng Wen since middle school. I would like to especially thank Zhaofeng Su, Allan Yin, Nancy Nan, Hong Man, Hehua Chi, Shaoyuan Li, Xiaoxi Hu, Yinan Li, Wenlin Chen, Shengqi Yang. They are the ones who have given me support during both joyful and stressful times, to whom I will always be thankful.

Finally, it is my greatest honor to thank my family: my dearest parents. They are always believing in me, keeping encouraging me, giving me indispensable suggestions, and fully supporting all my final decisions. No words could possibly express my deepest gratitude for their endless love, self-sacrifice and unwavering help.

To them I dedicate the dissertation.

Abstract

The sequential decision making is to actively acquire information and then make decisions in large uncertain options, such as recommendation systems and the Internet. The sequential decision becomes challenging since the feedback is often partially observed. In this thesis we propose new algorithms of "bandit learning", whose basic idea is to address the fundamental trade-off between exploration and exploitation in sequence. The goal of bandit learning algorithms is to maximize some objective when making decision. We study several novel methodologies for different scenarios, such as social networks, multi-view, multi-task, repeated labeling and active learning. We formalize these adaptive problems as sequential decision making for different real applications. We present several new insights into these popular problems from the perspective of bandit. We address the trade-off between exploration and exploitation using a bandit framework.

In particular, we introduce "networked bandits" to model the multi-armed bandits with correlations, which exist in social networks. The "networked bandits" is a new bandit model that considers a set of interrelated arms varying over time and selecting an arm invokes the other arms. The objective is still to obtain the best cumulative payoffs. We propose a method that considers both the arm and its relationships between arms. The proposed method selects an arm according to the integrated confidence sets constructed from historical data.

We study the problem of view selection in stream-based multi-view learning, where each view is obtained from a feature generator or source and is embedded in a reproducing kernel Hilbert space (RKHS). We propose an algorithm that selects a near-optimal subset of m views of n views and then makes the prediction based on the subset. To address this problem, we define the multi-view simple regret and study an upper bound of the expected regret for our algorithm. The proposed algorithm relies on the Rademacher complexity of the co-regularized kernel classes.

We address an active learning scenario in the multi-task learning problem. Considering that labeling effective instances across different tasks may improve the generalization error of all tasks, we propose a new active multi-task learning algorithm based on the multi-armed bandits for effectively selecting instances. The proposed algorithm can balance the trade-off

between exploration and exploitation by considering both the risk of multi-task learner and the corresponding confidence bounds.

We study a popular annotation problem in crowdsourcing systems: repeated labeling. We introduce a new framework that actively selects the labeling tasks when facing a large number of labeling tasks. The objective is to identify the best labeling tasks from these noisy labeling tasks. We formalize the selection of repeated labeling tasks as a bandit framework. We consider a labeling task as an arm and the quality of a labeling task as the payoff. We introduce the definition of ε-optimal labeling task and use it to identify the optimal labeling task. Taking the expected labeling quality into account, we provide a simple repeated labeling strategy. We then extend this to address how to identify the best m labeling tasks, and in doing so propose the best m labeling algorithm by indexing the labeling tasks using the expected labeling quality.

We study active learning in a new perspective of active learning. We build the bridge between the active learning and multi-armed bandits. Active learning aims to learn a classifier by actively acquiring the data points, whose labels are hidden initially and incur querying cost. The multi-armed bandit problem is a framework that can adapt the decision in sequence based on rewards that have been observed so far. Inspired by the multi-armed bandits, we consider active learning so as to identify the best hypothesis in an optimal candidate set of hypotheses by involving querying the labels of points as few as possible. Our algorithms are proposed to maintain the candidate set of hypotheses using the error or the corresponding general lower and upper error bounds to help select or eliminate hypotheses. To maintain the candidate set of hypotheses, in the realizable PAC setting, we directly use the error. In the agnostic setting, we use the lower and upper error bounds of the hypotheses. To label the data points, we use the uncertainty strategy based on the candidate set of hypotheses.

Table of contents

List of figures **xv**

List of tables **xvii**

1 Introduction **1**

1.1 Multi-armed bandits . . . 1

1.1.1 Stochastic multi-armed bandit . . . 1

1.2 Networked bandits . . . 4

1.3 Multi-view bandits . . . 5

1.4 Multi-task . . . 6

1.5 Repeated labeling . . . 7

1.6 Active learning . . . 7

1.7 Summary of contributions . . . 8

1.8 Publications . . . 9

2 Networked bandits **11**

2.1 Introduction . . . 11

2.2 Related work . . . 14

2.3 Networked bandits . . . 15

2.4 Algorithm . . . 17

2.5 Regret analysis . . . 21

2.6 Practical issues . . . 25

2.6.1 Dynamic network . . . 25

2.6.2 Static network . . . 26

2.6.3 Neighborhood or group . . . 26

2.7 Experiments . . . 26

2.7.1 Illustrative example . . . 26

2.7.2 Baselines and performance metric . . . 27

2.7.3 Simulation experiments . . . 29
2.7.4 Real-world datasets experiments . . . 31
2.8 Conclusion and future work . . . 33

3 Multi-view bandits 35
3.1 Introduction . . . 35
3.2 Related work . . . 39
3.3 Multi-view bandits . . . 42
3.4 CoRLSUB . . . 43
3.4.1 View subset calculation . . . 44
3.5 Regret analysis of CoRLSUB . . . 48
3.6 Experiments . . . 50
3.6.1 Toy example . . . 51
3.6.2 The robot navigation example . . . 52
3.6.3 Public datasets . . . 53
3.6.4 Stream-based multi-view learning . . . 56
3.7 Proofs . . . 57
3.7.1 Proof of Lemma 3.1 . . . 57
3.7.2 Proof of Lemma 3.2 . . . 58
3.7.3 Proof of Theorem 3.1 . . . 59
3.7.4 Proof of Proposition 3.1 . . . 60
3.8 Conclusion . . . 60

4 Active multi-task learning via bandits 63
4.1 Introduction . . . 63
4.2 Related work . . . 65
4.3 Problem definition . . . 66
4.4 Algorithm . . . 68
4.4.1 Confidence bounds . . . 70
4.4.2 Active multi-task learning via bandits . . . 72
4.5 Analysis . . . 72
4.6 Experiments . . . 74
4.6.1 Synthetic data . . . 75
4.6.2 Restaurant & consumer data . . . 75
4.6.3 Dermatology data . . . 76
4.6.4 School data . . . 77
4.7 Conclusion . . . 78

5 Selective repeated labeling via bandits **79**
5.1 Introduction . . . 79
5.2 Related work . . . 82
5.3 General framework . . . 83
5.4 Algorithm . . . 84
5.4.1 Repeated labeling strategies . . . 85
5.4.2 Selective repeated labeling strategies . . . 89
5.5 Experiments . . . 92
5.5.1 Data sets . . . 92
5.5.2 Labeling strategies . . . 93
5.5.3 Integration methods . . . 93
5.5.4 Results of the selective repeated labeling strategies . . . 93
5.5.5 Comparison between the selective repeated labeling and the single labeling . . . 95
5.5.6 Comparison between the **Best** *m* **Labeling** and the **Improved Best** *m* **Labeling** . . . 96
5.5.7 Study on the size of selected labeling tasks . . . 97
5.6 Conclusion . . . 97

6 Active learning via bandits **99**
6.1 Introduction . . . 99
6.2 Related work . . . 101
6.3 Methodology . . . 103
6.3.1 Realizable PAC setting . . . 104
6.3.2 Agnostic setting . . . 105
6.4 Theoretical analysis . . . 107
6.5 Experiments . . . 110
6.5.1 Experimental results of realizable setting . . . 110
6.5.2 Experimental results of agnostic setting . . . 111
6.6 Proofs . . . 112
6.6.1 Proof of Theorem 6.1 . . . 112
6.6.2 Proof of Theorem 6.2 . . . 114
6.6.3 Proof of Theorem 6.3 . . . 115
6.6.4 Proof of Theorem 6.4 . . . 116
6.6.5 Proof of Theorem 6.6 . . . 116
6.6.6 Proof of Theorem 6.7 . . . 117
6.7 Conclusion . . . 119

7 Conclusion **121**

References **125**

List of figures

2.1 An overview of networked bandits at different rounds. The network is changing over time. An arm (user) can invoke other arms (relations) and has different relations at different rounds. Given the contextual information, the arm is chosen by the decision algorithm for getting multiple payoffs (feedback). The algorithm updates the selection strategy after collecting new payoff information. 12

2.2 An example of the upper bound B in 10-arm networked bandits when $t = 120$. Bar denotes the payoff estimation and vertical line denotes the penalty of the estimation. 21

2.3 An example of the regret value in 10-arm networked bandits. The experiments are repeated 100 times and the average regrets are shown. $y = x$ is provided for comparison. 24

2.4 Illustrative synthetic example of exploration-exploitation trade-off. Bottom, arms with networked topology. Second row: the upper bound B for each arm computed using NetBandits. Third row: the expected estimation v, where bar denotes the estimation and vertical line denotes the penalty of estimation. Fourth row: the real payoff of each arm. 28

2.5 The average payoff at each round in dynamic networks. 30

2.6 The cumulative payoff at each round in dynamic networks. 31

2.7 The average payoff and cumulative payoff for two real-world datasets. . . . 32

3.1 An example of the application of SMVL to the automatic navigation control of a robot. 36

3.2 Using bandit framework to model stream-based multi-view learning. 37

3.3 A toy dataset with different views. 51

3.4 Performance comparison on the toy example. 52

3.5 Performance comparison on robot motion example. 53

3.6 Example views selected by different strategies in the automatic navigation control of a robot. 54
3.7 Performance comparison on (a) G50C and (b) PCMAC. 55
3.8 A comparison of the multi-view bandit strategy with other strategies on Caltech 256. 55
3.9 A comparison of the multi-views bandit strategy with other strategies on VOC 2006. 55
3.10 A comparison of the multi-view bandit strategy with other strategies on ImageNet. 56

4.1 Performance comparison on the synthetic data. 75
4.2 Performance comparison on the Restaurant & consumer data. 76
4.3 Performance comparison on the Dermatology data. 76
4.4 Performance comparison on the School data. 77

5.1 An example of selective repeated labeling. There are a large number of labeling tasks, where each task corresponds to multiple repeated labels and an integrated label (using a majority voting/average ratings). Our goal is to design a selective repeated labeling strategy that identifies the best *m* labeling tasks. 80
5.2 A comparison of test accuracy between the **Best** *m* **Labeling** and the Random on different data sets. 94
5.3 A comparison of test accuracy between the **Best** *m* **Labeling** and the **Improved Best** *m* **Labeling** on different data sets. 95
5.4 The test accuracy of the **Best** *m* **Labeling** and the Random on different data sets. 96

6.1 Labeled data points rate. 111
6.2 Test error rates for the classification experiments. 111
6.3 Labeled data points rate. 112
6.4 The locations of label queries. The x-axis is the unit interval and the y-axis is the rate of numbers in the corresponding interval. The top histogram shows the locations of label requests at the early stage; the bottom histogram is for all label queries. 113
6.5 Test error rates for the classification experiments. 114

List of tables

2.1 Running time results of NetBandits on four synthetic datasets. 30

5.1 The 9 data sets used in the experiments, including the numbers of attributes, and examples in each, and the split into positive and negative examples. . . 93

5.2 The test accuracy of the **Best** *m* **Labeling** and the Single Labeling. 97

Chapter 1

Introduction

We present an overview of multi-armed bandit algorithms for different areas, such as social networks, multi-view, multi-task, repeated labeling and active learning. We also present the contributions of this dissertation.

1.1 Multi-armed bandits

Multi-armed bandit problem is a sequential decision problem involving a set of actions or arms. The decision is made according to the exploration and exploitation trade-off.

The multi-armed bandit was a lottery game originally. The term bandit is from a Casino's slot machine, which can be called a one-armed bandit. The player can pull the arm of the machine and then obtain the payoff from the machine. In the multi-armed bandit problem, there are a finite number of slot machines or arms. The player always faces these arms and decides which arm to pull at each round. The player allocates his/her money on different slot machines or arms sequentially and earns money or rewards from the machine depending on the machine he/she selected. The goal is to obtain as much money as possible. An important problem of this model is the assumption on the slot machines' reward generation process.

Originally there is one basic assumption about the reward generation process. That is the payoffs are based on the stochastic assumption. In the stochastic setting, the reward is sampled from an unknown probability distribution on $[0, 1]$.

1.1.1 Stochastic multi-armed bandit

The stochastic multi-armed bandit is originally formulated by Robbins (1952), where each arm is associated with an unknown probability distribution on $[0, 1]$. At each time step t, the player selects an arm I_t, and then the player receives a payoff X_t from the distribution

associated with the selected arm and independently from the past given that arm. The goal of the player is to maximize the sum of payoffs $\sum_{t=1}^{n} X_t$ where n is the time horizon. If the time horizon is not known for the player in advance we say that the player is anytime.

In order to analyze the behavior of a player, we compare its performance with the best strategy. However, the best strategy is hard to calculate because these distributions for arms are unknown. If the distributions were known, the player would always select the arm with the highest mean reward in order to maximize the cumulative rewards. In such assumption, this optimal strategy should be the best one. However, this assumption does not exist. We study the **regret** of the player for not playing optimally. Formally, given K arms and sequences $X_{i,1}, X_{i,2}, \cdots$ of unknown rewards associated with each arm, the player receives the associated reward $X_{I_t,t}$. The regret after n time steps $I_1, \cdots, I_n$ is defined by

$$R_n = \max_{i=1,\cdots,K} \sum_{t=1}^{n} X_{i,t} - \sum_{t=1}^{n} X_{I_t,t}. \tag{1.1}$$

For $i = 1, \cdots, K$ we denote by μ_i the mean of v_i (mean payoff of arm i). Let

$$\mu^* = \max_{i=1,\cdots,K} \mu_i. \tag{1.2}$$

We can define the pseudo-regret as

$$\bar{R}_n = n\mu^* - \sum_{t=1}^{n} \mathbb{E}[\mu_{I_t}]. \tag{1.3}$$

Bandits are traditionally used to analyze medical trials. It has been realized in medical trials in the past decades that bandit models a number of more sophisticated and relevant applications. We introduce the modern motivating examples that the bandit learning can address. We describe five examples with different objectives, ranging from theoretical to applied, where the bandit model has been used or is currently under investigation, including social networks, multi-view problem, multi-task problem, repeated labeling and active learning.

Networked bandits: The social networks become popular nowadays and enable an increasing number of applications, such as recommendation, advertisement and so on. In the network, an important thing is that the users are connected by relationships. That means a message posted by one user can be seen by other users or a recommendation posted to one user can be expanded to other users.

We consider a real-application problem that posts an advertisement on the social networks. In the social networks, it is observed that even when a user is randomly selected for promotion,

other users close to the selected user in the network will be influenced. These correlations motivate a new bandit model containing relationships. Previously, in the bandit model, the algorithms only address the single arm and it is naturally used for the usual recommendation problem. Based on our observations, we introduce the new bandit framework, named as "networked bandits", where a set of interrelated arms varies over time and, given the contextual information that selects one arm, invokes other correlated arms.

Multi-view: Multi-view problem is to learn from the multi-view data by considering the diversity of different views. In recent years, there have been lots of methods addressing the problem. In the multi-view problem, a person can be identified by face, finger print, signature or iris with information obtained from multiple sources. In the multi-view problem, it is generally difficult to model the compatibility and temporal changes independently based on each view. In practice, a forecaster usually explores the unknown by collecting the prediction feedback based on the view subset in real time to evaluate the compatibility of views.

Considering that redundant and noisy views seriously affects the prediction, and regarding different views of different examples affects the prediction in different ways, pool-based multi-view learning algorithms ignore the environment feedback and thus cannot perform well for stream-based multi-view learning tasks. In addition, given limited budget or computational resources, a predictor can only exploit a small number of observed views. It thus becomes indispensable to design a decision strategy for sequentially selecting a subset of views to deal with changed environments.

We formalize the problem under a new bandit framework, i.e. a multi-view bandit, in which each of the n views $\{V_1, \cdots, V_n\}$ is defined as an arm, and at each time step t, a set of context vectors $\{x_{(1,t)}, \cdots, x_{(n,t)}\}$ represents an example defined by n views. The context vector $x_{(i,t)}$ for the i-th view V_i corresponds to the i-th arm. The forecaster is allowed to select m views ($m \leq n$), then exploits them for prediction and thus may suffer a loss. The loss is defined as payoff.

Multi-task: Multi-task learning is important in a variety of practical situations. Multi-task learning is to address the problem that the data representations are common across multiple related supervised learning tasks. The goal of multi-task learning is to improve the performance of the learner by learning a supervised classifier for multiple tasks jointly.

We introduce a new active multi-task learning paradigm, which selectively samples effective instances for multi-task learning. For all of these multi-task learning algorithms, we often first collect a significant quantity of data that is randomly sampled from the underlying population distribution and then induce a learner or model. However, the most time-consuming and costly task is usually the collecting of data. Thus, it is particularly valuable to determine ways in which we can make use of these resources as much as possible.

Furthermore, in multi-task learning, the simultaneously multiple related tasks allow each one to benefit from the learning of all of the others, and labeling instances for one task can also affect other tasks especially when the task has a small number of labeled data. Thus our work focuses on how to guide the sampling process for multi-task learning.

Repeated labeling: Labeling is important in real-world applications. Repeated labeling, where multiple data labels are repeatedly obtained from multiple sources, is often available via crowdsourcing. Data collection involves various preprocessing costs, including costs associated with acquiring features, formulating data, cleaning data and obtaining labels.

We first present a new framework for repeated labeling using the multi-armed bandit model. In the repeated labeling problem, each example's labeling is a labeling task in which multiple labels are repeatedly obtained from multiple noisy labelers. We assume that there is no more information about the labelers. From the bandit perspective, a labeling task can be considered as an arm and the uncertainty of labels for the corresponding labeling task can be considered as the payoff. We are often confronted with a large number of labeling tasks; however, due to cost or budget constraints, we would rather select a small or fixed number of these labeling tasks with high expected labeling quality.

Active learning: Active learning is a popular algorithm in machine learning. There is always a pool of data and we do not use all of them for training a learner because there is a budget for labeling or labeling all data is infeasible. The active learner is to selectively pay for the label of any example in the pool in order to obtain a good classifier with significantly fewer labels by making the query.

We study active learning from a bandit perspective, i.e. active learning is a process that we select the hypothesis depending on the sequential data and labeling. We treat a set of hypotheses as a set of arms and consider the error of a hypothesis as the reward of an arm. Similar to the problem of exploration in stochastic multi-armed bandits, our objective is to select the best hypothesis which has the lowest expected error.

1.2 Networked bandits

We introduce networked bandit to address a special kind of multi-armed bandit problem, where there exists correlations between the arms. We do not calculate the payoffs based on only one arm. Instead, we consider the payoffs over the network. In the networked bandit problem, we select an arm at each round and receive the associate payoffs over the network.

After selecting a series $a_1, a_2, \ldots, a_n$, we define the regret as follows:

$$R_n = \max_{a=1,\cdots,k} \sum_{t=1}^{n} g_{a,t} - \sum_{t=1}^{n} g_{a_t,t}, \tag{1.4}$$

where $g_{a_t,t} = \sum_{a \in N_t(a_t)} y_a$. Here we use $N_t(a)$ to indicate both a and its relations.

Our strategy is to consider the arms and the network topology. The proposed strategy is to optimally select an arm at each round based on the contextual information and the network topology information of arms.

We provide new bounds for our strategy. We assume that the forecaster does not care about the detail of the network but consider the invoked arms directly. Thus the confidence bound generated by the confidence sets of parameters, defined by

$$B_{a,t} = \nu_{a,t} + \xi_a(t), \tag{1.5}$$

where $\nu_{a,t}$ indicates the expected value, and $\xi_a(t)$ is considered as confidence and indicates the penalty of the estimation.

Thus in each round, our algorithm selects an arm based on the estimation from the confidence bound, such that the predicted payoff is maximized.

As shown in our bound, we are mainly interested in the interrelated arms. We show that our regret bound depends on the number of invoked arms $N_t(a)$ or loose K. Our algorithm keeps the regret as low as possible, and can reach $\to 0$ with high probability when t is large enough.

1.3 Multi-view bandits

The views selection in the multi-view problem is an important issue in stream-based multi-view learning. It is to sequentially identify the most appropriate views for the forecaster, in which each example represented by different views is drawn at the same time from the data sources. We formalize the problem under a new bandit framework, i.e. multi-view bandit. In multi-view bandit, we select m views, then exploit them for prediction and thus may suffer a loss corresponding to the prediction. The loss is defined as payoff. The multi-view bandit selects a subset of views depending on the joint context vectors (arms) and conducts the prediction based on the combination of all the selected views.

We propose a randomized algorithm CoRLSUB which depends on the confidence analysis of the generalization of the co-regularized least squares. We introduce the multi-view regret

as follows:

$$R_t = L(\varphi_{\mathcal{S}_t}(x), y) - L(\varphi_{\mathcal{S}_*}(x), y), \tag{1.6}$$

where $\mathcal{S}_t$ indicates a subset of arms and $\varphi_{\mathcal{S}_*}(x) = \min_{\mathcal{S}} L(\varphi_{\mathcal{S}}(x), y)$. Then the expected regret is

$$\mathcal{R}_t = \mathbb{E}[L(\varphi_{\mathcal{S}_t}(x), y) - L(\varphi_{\mathcal{S}_*}(x), y)]. \tag{1.7}$$

We provide the analysis of CoRLUB and show that the upper bound of multi-view simple regret scales $O(1/\sqrt{t})$. We also show that the consistency of different views improves the simple regret bound. We provide the algorithm to show how to choose the view subset.

1.4 Multi-task

In the multi-task problem, labeling instances is much more special because labeling for one task can also affect the other tasks. Considering that labeling informative instances across different learning tasks may help the performance of all tasks, we address the labeling problem for multiple tasks through a bandit approach. We consider both the risk of multi-task learner and the corresponding confidence bounds. We then consider these two aspects and use the bandit to address the trade-off between them.

Considering both the risk and the corresponding confidence, we want to find a hypothesis which can be

$$h = \arg\min_{h \in H} R(h) + C(h), \tag{1.8}$$

where $R(h)$ is the risk of h and $C(h)$ is the confidence of the risk. Our active learning algorithm aims to return a hypothesis which has the lowest expected error and also is closed to the best hypothesis on the dataset.

We propose an adaptive sampling algorithm, AMLB, which at each round queries the label for an instance according to this distribution. At each round, we maintain a distribution on the pool of data and this distribution will be updated when a new multi-task learner is trained with new instances and involves the risk and the confidence. We sample the data from the pool based on the distribution. The hypothesis can be considered as an arm. As different instances are acquired, we can select different hypotheses for our optimization function and finally filter the good candidates which are close to the ideal hypothesis.

1.5 Repeated labeling

Repeated labeling is a popular labeling problem in real crowdsourcing systems, where for each example there are many labels for one specific class. However, the quality of repeated labeling is not addressed well. When confronted with a large number of labeling examples with repeated labels, we aim to identify the best labeling examples to improve the performance of learning. For instance, in real applications, the examples which have lots of labels may contain labeling noise. This noise will make the performance worse. We try to answer the question of how many labels are good enough for the labeling tasks.

In the repeated labeling framework, identifying the best labeling task can be formalized to identify a subset of labeling tasks which are ε-optimal. That is, for a labeling task x_i, $\forall \varepsilon > 0, \forall m \in \{1,2,\cdots,n\}$, if

$$q_a \geq q_m - \varepsilon, \tag{1.9}$$

then the labeling task x_i is called an (ε, m)-optimal task. The best subset labeling is a set of labeling tasks that are ε-optimal.

Following the bandit framework, we formalize the repeated labeling problem as a bandit model, where each labeling task can be considered as an arm and the labeling quality the payoff. The problem is that after a lot of labeling we need to select a subset of labeling tasks which has a high quality of labeling. We first introduce a simple repeated labeling strategy with theoretical guarantee. Similar to the subset selection for multi-armed bandit, we propose two algorithms for actively selecting the labeling tasks: Best m Labeling and Improved Best m Labeling. Both two algorithms rely on the quality of labeling. The Best m Labeling simply selects the top m qualified labeling tasks. Then Improved Best m Labeling improves this algorithm by splitting the labeling process into several phases and some unqualified labeling tasks are eliminated during the phases.

1.6 Active learning

We study active learning in the bandit framework. In this framework, we define a hypothesis as an arm and the error of a hypothesis as the reward of an arm. The error of a hypothesis $h : X \to Y$, where X is the input space and Y are the possible labels, is

$$\mathrm{err}(h) := \Pr(h(x) \neq y), \tag{1.10}$$

where $x \in X$ and $y \in Y$. Then the empirical error of h with respect to a labeled sample L is defined as $\text{err}_L(h) = \frac{1}{|L|} \sum_{(x,y)\in L} \mathbb{1}[h(x) \neq y]$, representing the fraction of points in L on which h makes mistakes.

Let $h^* = \arg\min\{\text{err}(h) : h \in \mathcal{H}\}$ be the hypothesis associated with the minimum error in $\mathcal{H}$. Simply we assume that the minimum error always exists. The goal of the learner is to obtain a hypothesis $h \in \mathcal{H}$ with error $\text{err}(h)$ not much more than $\text{err}(h^*)$.

We propose three active learning algorithms for the realizable PAC and agnostic settings. The active learning process is re-explained as that there are a set of hypotheses and the objective is to return the best hypothesis by making the queries as few as possible. The proposed algorithms are guaranteed to find a hypothesis with the lowest error with a high probability. We resolve two issues. One issue is how to select the best hypothesis. Regarding the realizable PAC setting, which assumes there always exists at least one correct hypothesis that can correctly classify all the examples, we exploit the error of this hypothesis. For the agnostic setting, which assumes even the target hypothesis cannot correctly classify all the data points, we consider the error bounds of different hypotheses. We maintain a candidate set of hypotheses by selecting good hypotheses having low errors and eliminating bad hypotheses having high errors. Another issue is which data point we should label. We query the labels of points according to the disagreement for the candidate set of hypotheses.

1.7 Summary of contributions

We introduce the bandit framework for several important areas based on the idea of exploration and exploitation trade-off, such as network application, multi-view problem, multi-task problem, repeated labeling and active learning. We begin by studying approaches involving bandit based on the idea of exploration and exploitation. In Chapter 2, we formalize a new problem for networked bandits. We provide a novel solution for it and analysis of its regret. This analysis turns out to be a guaranteed bounds. We compare the results of experiments for traditional multi-armed bandits.

Multi-armed bandits can be used to solve some practical situations, such as multi-view and multi-task problems. In Chapter 3, firstly, to the best of our knowledge, we are the first to propose subset selection in the multi-view problem. Secondly, we are the first to address subset selection in the stream-based multi-view learning (SMVL) setting. Thirdly, we propose the multi-view bandit algorithm CoRLSUB and prove the multi-view simple regret bound. In Chapter 4, firstly, we propose a new active learning algorithm for the multi-task learning problem, named active multi-task learning via bandits, which is a general active learning framework. Secondly, we provide an implementation of our algorithm based

on the trace-norm regularization method. Thirdly, we verify our algorithm's effectiveness and efficiency by comparing its evaluation results to some passive learning and other active learning strategies for multi-task learning empirically.

In Chapter 5, firstly, we introduce a new framework for the repeated labeling problem using the multi-armed bandit model. Secondly, we design a simple repeated labeling algorithm, Naive Repeated Labeling, to repeatedly acquire labels for an example. Thirdly, we propose two algorithms, the Best *m* Labeling and the Improved Best *m* Labeling, to selectively obtain the labeling tasks. Fourthly, we provide a theoretical guarantee for our algorithm and demonstrate the effectiveness of our algorithms empirically.

In Chapter 6, firstly, we formalize active learning as a bandit problem. Secondly, we introduce two algorithms ALB-1 and ALB-2 for active learning in the realizable PAC setting. The difference between ALB-1 and ALB-2 is their uncertainty strategies. Thirdly, we propose ALB-LUB for active learning in the agnostic setting. Fourthly, we theoretically show the correctness of our algorithms and analyze their label complexities.

1.8 Publications

My publications include active learning [51–54, 56–58], transfer learning [55, 59] and sequential decision [49, 50]. List of my publications:

Meng Fang, Dacheng Tao. *Active Multi-task Learning via Bandits*. In the SIAM International Conference on Data Mining (SDM), Oral presentation, 2015

Meng Fang, Jie Yin, Xingquan Zhu, Chengqi Zhang. *TrGraph: Cross-Network Transfer Learning via Common Signature Subgraphs*. IEEE Transactions on Knowledge and Data Engineering, 2015

Meng Fang, Jie Yin, Xingquan Zhu. *Active Exploration for Large Graphs*. Data Mining and Knowledge Discovery, 2015

Meng Fang, Dacheng Tao. *Networked Bandits with Disjoint Linear Payoffs*. In the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), Oral presentation, 2014

Meng Fang, Jie Yin, Dacheng Tao. *Active Learning for Crowdsourcing Using Knowledge Transfer*. In the AAAI Conference on Artifical Intelligence (AAAI), Oral presentation, 2014

Meng Fang, Xingquan Zhu. *Active learning with uncertain labeling knowledge*. Pattern Recognition Letters, 2014

Meng Fang, Jie Yin, Xingquan Zhu. *Active exploration: simultaneous sampling and labeling for large graphs*. In the ACM International Conference on Information and Knowledge Management (CIKM), Oral presentation, 2013

Meng Fang, Jie Yin, Xingquan Zhu. *Transfer Learning across Networks for Collective Classification*. In the IEEE International Conference on Data Mining (ICDM), Oral presentation, 2013

Meng Fang, Jie Yin, Xingquan Zhu. *Knowledge Transfer for Multi-labeler Active Learning*. In the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD), Oral presentation, 2013

Meng Fang, Jie Yin, Chengqi Zhang, Xingquan Zhu. *Active Class Discovery and Learning for Networked Data*. In the SIAM International Conference on Data Mining (SDM), Oral presentation, 2013

Meng Fang, Xingquan Zhu, Bin Li, Wei Ding, Xindong Wu. *Self-Taught Active Learning from Crowds*. In the IEEE International Conference on Data Mining (ICDM), Oral presentation, 2012

Meng Fang, Xingquan Zhu. *I don't know the label: Active learning with blind knowledge*. In the International Conference on Pattern Recognition (ICPR), Oral presentation, 2012 (***Best Student Paper Award***)

Chapter 2

Networked bandits

In this work, we study 'networked bandits', a new bandit problem where a set of interrelated arms varies over time and, given the contextual information that selects one arm, invokes other correlated arms. This problem remains under-investigated, in spite of its applicability to many practical problems. For instance, in social networks, an arm can obtain payoffs from both the selected user and its relations since they often share the content through the network. We examine whether it is possible to obtain multiple payoffs from several correlated arms based on the relationships. In particular, we formalize the networked bandit problem and propose an algorithm that considers not only the selected arm, but also the relationships between arms. Our algorithm is 'optimism in face of uncertainty' style, in that it decides an arm depending on integrated confidence sets constructed from historical data. We analyze the performance in simulation experiments and on two real-world offline datasets. The experimental results demonstrate our algorithm's effectiveness in the networked bandit setting.

2.1 Introduction

A multi-armed bandit problem (or bandit problem) is a sequential decision problem defined by a set of actions (or arms). The term 'bandit' originates from the colloquial term for a casino slot machine ('one-armed bandit'), in which a player (or a forecaster) faces a finite number of slot machines (or arms). The player sequentially allocates coins (one at a time) to different machines and earns money (or payoff) depending on the machine selected. The goal is to earn as high a payoff as possible.

Robbins formalized this problem in 1952 [114]; in the multi-armed bandit problem, K arms exist that are associated with unknown payoff distributions, and a forecaster can select an arm sequentially. In each round of play, a forecaster selects one arm and then receives

the payoff from the selected arm. The forecaster's aim is to maximize the total cumulative payoff, i.e., the sum of the payoffs of the chosen arms in total. Since the forecaster does not know the process of generating the payoff but has historical payoff information, the bandit problem highlights the fundamental difficulty of decision making in the face of uncertainty: balancing the decision of whether to exploit past choices or make new choices with the hope of discovering a better one.

The bandit problem has been studied for many years, with works primarily focusing on the theory and designing different algorithms based on different settings, such as the stochastic setting, adversarial setting, and contextual setting [31]. In real-world applications, the multi-armed bandit problem is an effective way of solving situations where one encounters an exploration-exploitation dilemma. It has historically been used to decide which clinical trial is better when multiple treatments are available for a given disease and there is a need to decide which treatment to use on the next patient.

Modern technologies have created many opportunities for the use of the bandit algorithm, and it has a wide range of applications including advertising, recommendation systems, online systems, and games. For example, an advertising task may be the choice of which advertisement to display to the next visitor to a web page, where the payoff is associated with the visitor's action. More recently, bandit algorithm has been used in personalized recommendation tasks [88], where a user visits a website and the system collects the user's feedback. The system selectively provides content from a content pool through the user's current and past behaviors analyzing to best satisfy the user's needs, and the payoff is based on user-click feedback.

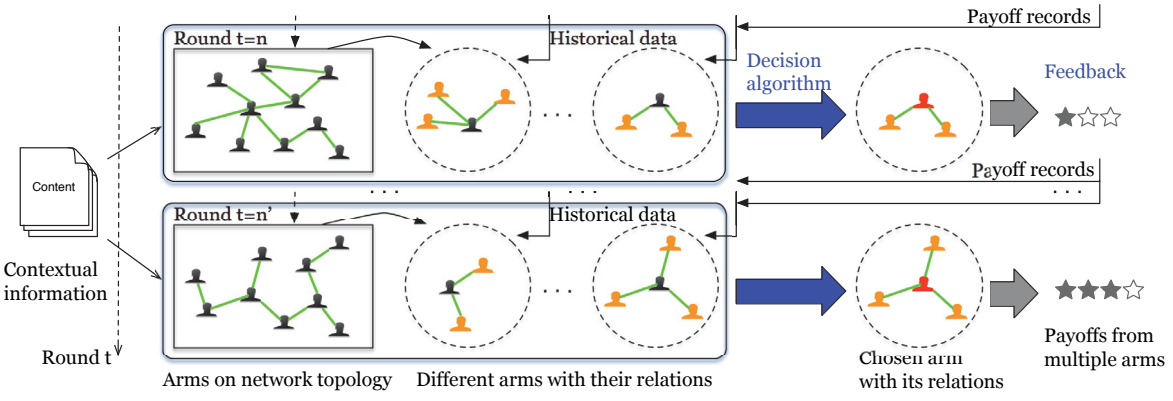


Fig. 2.1 *An overview of networked bandits at different rounds. The network is changing over time. An arm (user) can invoke other arms (relations) and has different relations at different rounds. Given the contextual information, the arm is chosen by the decision algorithm for getting multiple payoffs (feedback). The algorithm updates the selection strategy after collecting new payoff information.*

All the above bandit problems have the major underlying assumption that all the arms are independent, which is inappropriate for web-based social network applications. In a network, including social networks, the users are connected by relationships [5, 128]. Contextual information can be obtained from other users and can be spread via these relationships. Content promoted to one user provides feedback, not only from that user, but also from his/her relations. For example, a user of Twitter or Facebook can read a tweet/message and can re-post someone else's tweet/message, allowing the user to quickly share it with his/her followers. Impact can be assessed by counting the number of 'favorites/likes' from different users' pages. Therefore, careful selection of a user for tweet/message posting can maximize the number of 'favorites/likes'.

Our study is motivated by the observation that even when a user is randomly selected for promotion, other users close to the selected user in the network will be influenced [101, 128]. Specifically, as shown in Figure 2.1, in a social network, if we promote a content to a user, the user may share it with others and the payoff can be collected from the user and its relations. The goal is to gain higher payoffs. The process is similar to share-then-like, which occurs daily in social networks and needs to be considered for personalized recommendation and advertising tasks. An important point is that the content is expanded from the selected user to all other invoked users. There are several challenges to realizing this problem. First, only partial information is available about the chosen users when content is posted, and the information of other users is unknown. Therefore, there is a dilemma of whether the system should select a user with the best payoff history or a new user in order to explore more possibilities. Second, the content may frequently change and few overlapping historical records may exist. Furthermore, relationships exist between users and these relationships may change over time. These challenges inspire us to formalize the networked bandit problem.

The above problem can be considered a balance of the trade-off between exploration (discovering a new user) and exploitation (using the current best user) when network topology is known.

We formalize a well-defined but simple setting for the networked bandit problem, in which there exist K arms connected by network topology G. We propose an approach in which a learning algorithm optimally selects an arm at each round based on contextual information and the network topology information of arms. The networked bandit problem can be considered as an extension of the contextual multi-armed bandit problem. However, the difference is that in our problem the arm can be connected to other arms and the payoff comes from the multiple arms.

Our contribution is three-fold: firstly, we formalize a new networked bandit problem motivated by real network applications; secondly, we provide an algorithm based on confidence

sets to solve it along with theoretical analysis; and thirdly, we design a set of experiments to test and evaluate the algorithm. To the best of our knowledge, we define and solve this problem for the first time and answer the fundamental question of how to define regret when payoffs come from interrelated multiple arms. We design an effective strategy to select arms in order to increase payoffs over time, known as NetBandits, which provides a solution to this problem. Our approach is an 'optimism in the face of uncertainty'-style algorithm that considers the integrated confidence sets and we prove a regret bound for it. Finally, we analyze empirically the performance, which shows that our algorithm is effective in the networked bandit setting.

2.2 Related work

The traditional multi-armed bandit problem does not assume that side information is observed. The forecaster's goal is to maximize the sum of payoffs over time based on the historical payoff information. There are two basic settings. In the first, the stochastic setting, the payoff is i.i.d. drawn from an unknown distribution. The upper confidence bound (UCB) strategy has been used to explore the exploration-exploitation trade-off [11, 14, 86], in which an upper bound estimate is constructed on the mean of each arm at a fixed confidence level, and then the arm with the best estimate is selected. In the second, the adversarial setting, the i.i.d. assumption does not exist. Auer et al. [15] proposed the EXP3 algorithm for the adversarial setting, which was later improved by Bubeck and Audibert [10].

The contextual multi-armed bandit problem is a natural extension of the original bandit problem. Our setting addresses bandit problem with contextual information. Compared to the traditional K-armed bandit problem, the forecaster may use action features to infer the payoff in the contextual setting. This problem largely considers the linear model assumption about payoff of action [1, 13, 37, 40, 117]. Auer [13] proposed the LinRel algorithm, a UCB-style algorithm that has a regret of $\tilde{O}(\sqrt{Td})$. Dani et al. [40] studied the LinRel and provided an $\tilde{O}(d\sqrt{T})$ regret bound and proved this upper bound is tight. Chu et al. [37] provided the LinUCB and SupLinUCB algorithms, and proved an $O(\sqrt{Td\log^3(KT\log(T)/\delta)})$ regret bound for SupLinUCB that holds with probability $1-\delta$. Abbasi-Yadkori et al. [1] proposed an algorithm that modified the UCB-style algorithm based on the confidence sets, and showed a regret of $O(d\log(1/\delta)/\Delta)$.

Recently, the bandit problem has been used in real-life problems, such as recommendation systems and advertising. Li et al. [88] first introduced the bandit problem to recommendation systems by considering a personalized recommendation as a feature-based exploration-exploitation problem. This problem was formalized as a contextual bandit problem with

disjoint linear payoffs and by focusing on the article-selection strategy based on user-click feedback, maximizing the total number of clicks. The features of the users and articles were defined as contextual information, and the expected payoff of an arm was assumed to be a linear function of its contextual features, including the user and article information. Finally, the LinUCB algorithm was proposed to solve this problem and attained a good empirical regret. They further extended the algorithm as SupLinUCB and provided the theoretical analysis [37].

There are limited studies that consider the networked bandit problem or that combine bandit problem and network. Buccapatnam et al. [32] considered the bandit problem in social networks, and assumed that the forecaster can take advantage of side observations of neighbors, except for the selected user (arm). The side observations were used to update the sample mean of other related users and the payoff of the selected arm was collected each time, the goal once again being to maximize the total cumulative payoff of selected arms. Bnaya et al. [26] considered a bandit view for network exploration and proposed VUCB1 to handle the dynamic changes in arms when crawling the network. More recently, Cesa-Bianchi et al. [34] considered the recommendation problem by taking advantage of the relationships between users in the network. They proposed GOB.Lin, which models the similarity between users and used this similarity to help predict the behavior of other users.

Our work belongs to the contextual bandit setting. However, in contrast to these previous studies we assume that the arms (actions) are correlated in the network. The selected arm can invoke other related arms and the forecaster obtains multiple payoffs from these arms. It is a more general setting in networked bandit problem.

2.3 Networked bandits

We consider a network G. Let $\mathcal{V}$ indicate the nodes in the network and $\mathcal{E}$ indicate the edges of the network. We can then use $G = (\mathcal{V}, \mathcal{E})$ to represent the networked bandits, where $v \in \mathcal{V}$ is considered as an arm and $e \in \mathcal{E}$ indicates the relationship between arms; nodes here are correlated. Thus, given the network G and a node v, it is possible to obtain the information for the node v and its relations $N(v)$.

In our formulation, we consider a sequential decision problem with contextual information. At round t, except for contextual information x_t, we have a network topology of arms denoted by G_t. Given v we let $N_t(v)$ be its relations and $N_t(v)$ may change over time. If v is selected then $N_t(v)$ will also be invoked. We define this setting as networked bandits. Formally, a networked bandit algorithm $\mathcal{A}$ proceeds as follows: at each round t, the algorithm observes a set of arms $\mathcal{K}_t = \{1, 2, \cdots, k\}_t$, contextual information x_t, and the

network topology G_t of arms associated with the relationships of arms. The set of relations of arm a is denoted by $N_t(a)$. If we also consider the information of arms, we can redefine the context as a set $\mathcal{C}_t = \{x_{1,t}, \cdots, x_{k,t}\}$ by adding arms' information. When the algorithm selects an arm a_t, then a_t invokes other related arms $N_t(a_t)$. $N_t(a_t)$ are observed based on the network topology of arms. Before the decision algorithm selects the arm, it observes G_t, $\mathcal{C}_t$, and $\mathcal{K}_t$. Based on historical payoff records, the algorithm selects an arm a_t and receives a set of payoffs $\{y_{a_t}\} \cup \{y_a | a \in N_t(a_t)\}$. The algorithm will improve the selection strategy after collecting new payoff information. It then proceeds to the next round $t+1$. Note that traditional contextual bandit problems usually assume that the arms are independent. However, in our problem, we assume that the correlation exists between the chosen arm and its relations. After a total T rounds the cumulative payoff is defined as $\sum_{t=1}^{T} g_{a_t,t}$, where $g_{a_t,t} = \sum_{a \in N_t(a_t)} y_a + y_{a_t}$ and y_a is the payoff from arm a. For simplicity, we use $N_t(a)$ to indicate both a and its relations. We rewrite $g_{a_t,t}$ as $g_{a_t,t} = \sum_{a \in N_t(a_t)} y_a$.

For this networked bandit problem, the algorithm $\mathcal{A}$ selects an arm a_t at each round $t = 1, 2, \cdots$ and receives the associate payoff $g_{a_t,t}$. After n selections $a_1, a_2, \cdots, a_n$ we define the regret as follows:

$$\mathcal{R}_n = \max_{a=1,\cdots,k} \sum_{t=1}^{n} g_{a,t} - \sum_{t=1}^{n} g_{a_t,t}. \tag{2.1}$$

The regret can now be used to compare the best decision with the algorithm $\mathcal{A}$. In this problem, $\mathcal{R}_n$ is a random variable; therefore, the goal is to calculate the expectation of $\mathcal{R}_n$ with high probability, and it is not easy to obtain expectation directly since its search space is large. Normally we try to bound the pseudo-regret, i.e.,

$$\overline{\mathcal{R}}_n = \max_{a=1,\cdots,k} \mathbb{E} \sum_{t=1}^{n} g_{a,t} - \mathbb{E} \sum_{t=1}^{n} g_{a_t,t}, \tag{2.2}$$

where the pseudo-regret competes against the optimal action in the expectation.

There are two important issues in the networked bandit problem: arms and their network topology. In the context of a social network, the users in the pool may be viewed as arms, the provided message or article as context, and the user's information as additional contextual information. The new context vector then summarizes information of both user and context. A payoff of 1 is incurred when a provided message is 'favorited' or 'liked'; otherwise, the payoff is 0. The network topology of a social network naturally constructs the relationships between users. When a message is posted to a user, the message can be seen by relations (followers). The payoff can be collected from the user's page (selected arm). Furthermore, any 'like', 'share', or 'comment' action by a follower allows the message to be reposted on the follower's page and to be seen by the follower's friends. The payoff can then be collected

from the followers' pages (invoked arms). In the special case that the follower does not repost the message, $N_t(a)$ can be considered as 0 or the arm is not invoked. For simplicity, we only consider the selected arm and its relations. With these definitions of payoff, arm, and invoked arms, the collected payoff after selecting an arm involves the selected user and his/her relations. Thus, the payoff at round t is defined as $g_{a_t,t} = \sum_{a \in N_t(a_t)} y_a$.

It is assumed that algorithm $\mathcal{A}$ can observe the network topology prior to make a decision. This is intuitive, since network structure information between users can easily be collected or the network structure information can be obtained in advance. In practice, given an arm, we only need concern itself with the invoked arms, and therefore knowledge of full network topology is unnecessary. The invoked arms depend on how we define $N_t(a)$. The worst case scenario is that the whole network needs to be searched to find the invoked arms and feedback; however, we do not concern how to constrict $N_t(a)$ using such a network propagation model since, as stated above, we only focus on the selection strategy and we simplify the problem by only observing the invoked arms.

2.4 Algorithm

In this work, we propose an algorithm to solve the networked bandit problem and show that an integrated confidence bound can efficiently be computed in a closed form when the payoff model of an arm is linear. As with previous contextual bandit work [88], we assume that the expected payoff of an arm a is linear in context x_t and coefficient w_a. At round t, for arm a given context $x_{a,t}$, we assume that the expected payoff of the arm a is a linear function:

$$\mathbb{E}[y_{a,t}|x_{a,t}] = x_{a,t}^{\top} w_a + \varepsilon_a, \tag{2.3}$$

where different arms have different w_a and ε_a is conditionally R-sub-Gaussian when $R \leq 0$ is a fixed constant. Formally, this means that $\forall \lambda$ and we have

$$\mathbb{E}\left[e^{\lambda \varepsilon_{a,t}} | x_{a,1:t}, \varepsilon_{a,1:t-1}\right] \leq \exp\left(\frac{\lambda^2 R^2}{2}\right), \tag{2.4}$$

where $x_{a,1:t}$ denotes the sequence $x_{a,1}, x_{a,2}, \cdots, x_{a,t}$ and, similarly $\varepsilon_{a,1:t-1}$ denotes the sequence $\varepsilon_{a,1}, \cdots, \varepsilon_{a,t-1}$. The arms therefore have disjoint linear payoffs. The decision of the algorithm lies on w with distribution ε. Based on our R-sub-Gaussian assumption of the noise, we can obtain meaningful upper bound on the regret. According to this sub-Gaussian condition, we know that $\mathbb{E}[\varepsilon_{a,t}|x_{a,1:t}, \varepsilon_{a,1:t-1}] = 0$ and $\mathbb{VAR}[\varepsilon_{a,t}|x_{a,1:t}, \varepsilon_{a,1:t-1}] \leq R^2$. The

conditions therefore show that $\varepsilon_{a,t}$ is bounded by a zero-mean noise lying in an interval of length of at most $2R$.

As the networked bandit problem, the algorithm faces a set of uncertainties of arms which involve $N_t(a)$. We design a new algorithm which is the 'optimism in the face of uncertainty' principle, by maintaining confidence of parameter w for each arm. The basic idea is to construct the confidence sets for parameters of each disjoint payoff function and then provide an integrated upper bound.

We use technology from the 'self-normalized bound for vector-valued martingales' [105] and confidence sets [1]. For each arm $\hat{w}_a$ is defined as the L^2-regularized least-squares estimate of w_a^* with regularization parameter $\lambda > 0$:

$$\hat{w}_a = (X_a^\top X_a + \lambda)^{-1} X_a^\top Y_a, \tag{2.5}$$

where X_a is the matrix whose rows are $x_1, \cdots, x_{n_a(t)}$ corresponding to historical contexts of an arm a and $Y_a \in \mathbb{R}^{n_a(t)}$ is the corresponding historical payoff vector. For a positive definite self-adjoint operator V, we define $\|x\|_V = \sqrt{\langle x, Vx \rangle}$ as the weighted norm of vector x. It can be proved that $\hat{w}$ lies with high probability in an ellipsoid centered at w^* as follows:

Theorem 2.1. [1, 105] *According to the 'self-normalized bound for vector-valued martingales', let $V = \lambda I, \lambda > 0$, and $\overline{V}_t = V + \sum_{n=1}^{t-1} x_n \otimes x_n$ be the regularized design matrix underlying the covariates. Define $y_t = x_t^\top w^* + \varepsilon_t$ and assume that $\|w^*\|_2 \leq S$. Then, for any $0 < \delta < 1$, with probability at least $1 - \delta$, for all $t \geq 1$ we can bound w^* in such a confidence set:*

$$C_t = \Bigg\{ w^* \in \mathbb{R}^d : \\ \|\hat{w}_t - w^*\|_{\overline{V}_t} \leq R\sqrt{2\log\left(\frac{|\overline{V}_t|^{1/2}|\lambda I|^{-1/2}}{\delta}\right)} + \lambda^{1/2} S \Bigg\}. \tag{2.6}$$

In addition, if $\|x_t\| \leq L$ then with probability at least $1 - \delta$, for all $t \geq 1$, we can bound w^* in a new confidence set:

$$C_t' = \Bigg\{ w^* \in \mathbb{R}^d : \\ \|\hat{w}_t - w^*\|_{\overline{V}_t} \leq R\sqrt{d\log\left(\frac{1 + tL^2/\lambda}{\delta}\right)} + \lambda^{1/2} S \Bigg\}. \tag{2.7}$$

The above bound provides the confidence region at time t. It shows that with good choice of the right parts of the equation, w^* always remains inside this ellipsoid for all times t with probability $1-\delta$. Next, we show the bound of the arm with a single linear payoff.

Theorem 2.2. *Let* $(x_1,y_1),\cdots,(x_{t-1},y_{t-1})$, $x_i \in \mathbb{R}^d, y_i \in \mathbb{R}$ *satisfy the linear model assumption. Furthermore, we have the same assumption as Theorem 2.1. Then, for any* $0<\delta<1$, *with probability at least* $1-\delta$, *for all* $t \geq 1$ *we can have:*

$$\|x^\top \hat{w} - x^\top w^*\| \leq \|x\|_{\overline{V}_t^{-1}} \left(R\sqrt{2\log\left(\frac{|\overline{V}_t|^{1/2}|\lambda I|^{-1/2}}{\delta}\right)} + \lambda^{1/2} S \right). \tag{2.8}$$

In addition, if $\|x_t\| \leq L$ *then for all* $t \geq 1$, *with probability* $1-\delta$ *we can have:*

$$\|x^\top \hat{w} - x^\top w^*\| \leq \|x\|_{\overline{V}_t^{-1}} \left(R\sqrt{d\log\left(\frac{1+tL^2/\lambda}{\delta}\right)} + \lambda^{1/2} S \right). \tag{2.9}$$

Proof.

$$\begin{aligned}\|x^\top \hat{w} - x^\top w^*\| &= \|x^\top(\hat{w}-w^*)\| \\ &\leq \|x\|\|\hat{w}-w^*\| = \|x\|_{\overline{V}_t^{-1}}\|\hat{w}-w^*\|_{\overline{V}_t}.\end{aligned}$$

According to (2.6), with probability at least $1-\delta$, for all $t \geq 1$, we have:

$$\|x^\top \hat{w} - x^\top w^*\| \leq \|x\|_{\overline{V}_t^{-1}} \left(R\sqrt{2\log\left(\frac{|\overline{V}_t|^{1/2}|\lambda I|^{-1/2}}{\delta}\right)} + \lambda^{1/2} S \right).$$

According to (2.7), with probability at least $1-\delta$, for all $t \geq 1$ we have:

$$\|x^\top \hat{w} - x^\top w^*\| \leq \|x\|_{\overline{V}_t^{-1}} \left(R\sqrt{d\log\left(\frac{1+tL^2/\lambda}{\delta}\right)} + \lambda^{1/2} S \right).$$

□

Lemma 2.1. *Given an arm $a \in \mathcal{K}_t$ with the context feature x, let $(x_{a,1}, y_{a,1}), (x_{a,2}, y_{a,2}), \cdots, (x_{a,n_a(t-1)}, y_{a,n_a(t-1)})$ be history records of arm a before t and $x_a \in X_a$ and $y_a \in Y_a$, and let $\hat{w}_a = (X_a^\top X_a + \lambda I)^{-1} X_a^\top Y_a$. We have*

$$x^\top w_a^* \leq x^\top \hat{w}_a + \\ \|x\|_{\overline{V}_t^{-1}} \left(R\sqrt{2\log\left(\frac{|\overline{V}_t|^{1/2}|\lambda I|^{-1/2}}{\delta}\right)} + \lambda^{1/2} S \right). \tag{2.10}$$

As shown in (2.10), we have a possible upper bound of $x^\top w_a^*$, which has two parts. The first term can be deemed as empirical expected estimation of payoff of the arm, and the second term can be considered as a penalty. This penalty is typically a high probability upper confidence bound on the payoff of the arm.

Thus, given an arm a and its relations $N_t(a)$, we face the exploration-exploitation problem. We use the integrated confidence bound on the payoffs of these invoked arms.

Lemma 2.2. *In the networked bandits, given an arm $a \in \mathcal{K}_t$ and the network relationship $N_t(a)$, we obtain:*

$$\sum_{a \in N_t(a)} x_a^\top w_a^* \leq \sum_{a \in N_t(a)} x_a^\top \hat{w}_a + \\ \sum_{a \in N_t(a)} \|x_a\|_{\overline{V}_t^{-1}} \left(R\sqrt{2\log\left(\frac{|\overline{V}_t|^{1/2}|\lambda I|^{-1/2}}{\delta}\right)} + \lambda^{1/2} S \right). \tag{2.11}$$

We believe that the confidence bound can be successfully applied to this situation with the exploitation-exploration trade-off. We use the confidence bound generated by the confidence sets of parameters, defined by:

$$B_{a,t} = \nu_{a,t} + \xi_a(t), \tag{2.12}$$

$\nu_{a,t} = \sum_{a \in N_t(a)} x^\top \hat{w}_{a,t}$ indicates the expected value, and $\xi_a(t)$ is the last term of (2.11) and indicates the penalty of the estimation. Figure 2.2 shows the upper bound of arms from our illustrative example. Each arm has the empirical payoff and a potential value. Thus, in each round, our algorithm selects an arm based on the estimation from the confidence bound, such that the predicted payoff is maximized. Our algorithm is shown in Algorithm 2.1.

Algorithm 2.1. NetBandits

Input: $\mathcal{K}_t, G_t, \mathcal{C}_t, t = 1, \cdots, T$

1: **for** round $t = 1, 2, \cdots, T$ **do**
2: For each arm we can observe the features $x_{a,t}, a \in \mathcal{K}_t$, and the invoked arms $N_t(a)$ based on G_t
3: **for** each $a \in \mathcal{K}_t$ **do**
4: Compute $\hat{w}_a$ according to (2.5)
5: Compute the quality

$$B_{a,t} = \sum_{a \in N_t(a)} x_{a,t}^{\top} \hat{w}_a + \sum_{a \in N_t(a)} \|x_{a,t}\|_{\overline{V}_t^{-1}} \left(R\sqrt{2\log\left(\frac{|\overline{V}_t|^{1/2}|\lambda I|^{-1/2}}{\delta}\right)} + \lambda^{1/2} S \right)$$

6: **end for**
7: Choose arm $a_t = \arg\max_{a \in \mathcal{K}_t} B_{a,t}$
8: Observe the multiple payoffs $\{y_{a,t} | a \in N_t(a_t)\}$
9: **for** each node $a \in N_t(a_t)$ **do**
10: Update X_a, Y_a
11: **end for**
12: **end for**

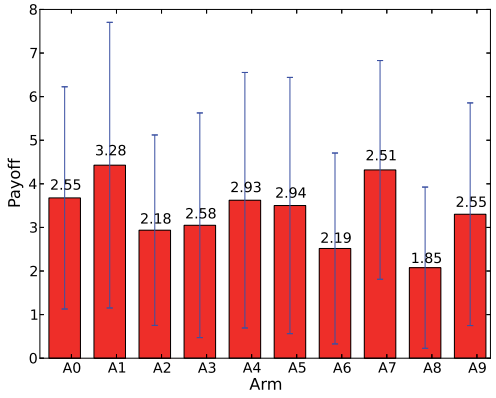


Fig. 2.2 *An example of the upper bound B in 10-arm networked bandits when* $t = 120$*. Bar denotes the payoff estimation and vertical line denotes the penalty of the estimation.*

2.5 Regret analysis

We next provide a bound on the regret of our algorithm when run through the confidence sets constructed in Theorem 2.1. We assume that the expected estimation of payoff is bounded.

We can view this as a bound on parameters and the bound on the arms set. We state a bound on the regret of the algorithm as follows:

Theorem 2.3. *On the networked bandits, assume that each arm's payoff function satisfies the linear model, and assume that the contextual vector is $x_{a,t}$ for each arm $a \in \mathcal{K}_t, |\mathcal{K}_t| \leq K$ and $t = 1, \cdots, T$. Then, for any $0 < \delta < 1$, with probability at least $1-\delta$, the cumulative regret satisfies*

$$\overline{\mathcal{R}}_T \leq 2K\sqrt{2\beta_T(\delta)T\log|I+XX^\top/\lambda|}, \tag{2.13}$$

where

$$\beta_T(\delta) = \left(R\sqrt{2\log\left(\frac{|I+XX^\top/\lambda|^{1/2}}{\delta}\right)} + \lambda^{1/2}S\right)^2.$$

Proof. Considering the instantaneous regret at round t, we select an optimal arm according to our algorithm. Thus, we have optimistic $(a_t, \tilde{w}_a)$ and $\hat{w}_a$ for the $N(a_t)$. For round t, we rely on [1] to have:

$$\begin{aligned}
r_t &= \sum_{a\in N(a_t)} x_{a,t}^\top w_a^* - \sum_{a\in N(a^*)} x_{a,t}^\top w_a^* \\
&\leq \sum_{a\in N(a_t)} x_{a,t}^\top w_a^* - \sum_{a\in N(a_t)} x_{a,t}^\top \tilde{w}_a \\
&= \sum_{a\in N(a_t)} x_{a,t}^\top w_a^* - \sum_{a\in N(a_t)} x_{a,t}^\top \hat{w}_a \\
&+ \sum_{a\in N(a_t)} x_{a,t}^\top \hat{w}_a - \sum_{a\in N(a_t)} x_{a,t}^\top \tilde{w}_a \\
&= \sum_{a\in N(a_t)} x_{a,t}^\top (w_a^* - \hat{w}_a) + \sum_{a\in N(a_t)} x_{a,t}^\top (\hat{w}_a - \tilde{w}_a) \\
&= \sum_{a\in N(a_t)} \|x_t\|_{\overline{V}_t^{-1}} \|w^* - \hat{w}_t\|_{\overline{V}_t^{-1}} \\
&+ \sum_{a\in N(a_t)} \|x_t\|_{\overline{V}_t^{-1}} \|\hat{w}_t - \tilde{w}_t\|_{\overline{V}_t^{-1}} \\
&\leq \sum_{a\in N(a_t)} 2\sqrt{\beta_t(\delta)}\|x_t\|_{\overline{V}_t^{-1}}.
\end{aligned} \tag{2.14}$$

For each arm $a \in N_t(a)$, we define

$$r_{a,t} = x_{a,t}^\top w_a^* - x_{a,t}^\top \hat{w}_a + x_{a,t}^\top \hat{w}_a - x_{a,t}^\top \tilde{w}_a, \tag{2.15}$$

and we have

$$r_{a,t} \leq 2\sqrt{\beta_t(\delta)}\|x_{a,t}\|_{\overline{V}_t^{-1}}. \tag{2.16}$$

We then rewrite the instantaneous regret (2.14) as

$$r_t \leq \sum_{a \in N(a_t)} r_{a,t}. \tag{2.17}$$

Regarding to the fact that $r_{a,t} \leq 2$, we have

$$\begin{aligned} r_{a,t} &\leq 2\min\left(\sqrt{\beta_t(\delta)}\|x_{a,t}\|^2_{\overline{V}_t^{-1}}, 1\right) \\ &\leq 2\sqrt{\beta_t(\delta)}\min\left(\|x_{a,t}\|^2_{\overline{V}_t^{-1}}, 1\right). \end{aligned} \tag{2.18}$$

Given arms $N_t(a)$, we define $a^{'} = \arg\max_a \|x_{a,t}\|^2_{\overline{V}_t^{-1}}$. Then we have

$$\begin{aligned} r_t &\leq |N_t(a)|2\sqrt{\beta_t(\delta)}\|x_{a^{'},t}\|_{\overline{V}_t^{-1}} \\ &\leq |N_t(a)|2\sqrt{\beta_t(\delta)}\min\left(\|x_{a^{'},t}\|^2_{\overline{V}_t^{-1}}, 1\right). \end{aligned} \tag{2.19}$$

Thus, with probability at least $1-\delta$, for any $T \geq 1$,

$$\begin{aligned} \overline{\mathcal{R}}_T &= \sum_{t=1}^{T} r_t \leq \sum_{t=1}^{T} |N_t(a)|2\sqrt{\beta_t(\delta)}\left(\|x_{a^{'},t}\|_{\overline{V}_t^{-1}} \wedge 1\right) \\ &\leq 2K\sum_{t=1}^{T}\sqrt{\beta_t(\delta)}\left(\|x_{a^{'},t}\|_{\overline{V}_t^{-1}} \wedge 1\right) \\ &\leq 2K\sqrt{T\sum_{t=1}^{T}\left(\sqrt{\beta_t(\delta)}\left(\|x_{a^{'},t}\|_{\overline{V}_t^{-1}} \wedge 1\right)\right)^2} \\ &\leq 2K\sqrt{T\beta_T(\delta)\sum_{t=1}^{T}\left(\|x_{a,t}\|^2_{\overline{V}_t^{-1}} \wedge 1\right)}. \end{aligned} \tag{2.20}$$

According to $\log(1+z) \leq z$, we have:

$$\log|I + XV^{-1}X| \leq \sum_{k=1}^{t-1}\|x_k\|^2_{\overline{V}_k^{-1}}. \tag{2.21}$$

Then according to $z \leq 2\log(1+z), z \in [0,1]$, we have

$$\sum_{k=1}^{t-1} \left(\|x_k\|^2_{\overline{V}_k^{-1}} \wedge 1 \right) \leq 2\log|I + XV^{-1}X|. \tag{2.22}$$

We choose $V = \lambda I$, then we rewrite $\overline{\mathcal{R}}_T$ as

$$\overline{\mathcal{R}}_T \leq 2K\sqrt{2T\beta_T(\delta)\log|I + XX^\top/\lambda|}. \tag{2.23}$$

□

Lemma 2.3. *Assume that $x \in \mathbb{R}^d$ and $V = \lambda I$. Then, for any $0 < \delta < 1$, with probability at least $1-\delta$, the bound is*

$$\overline{\mathcal{R}}_T \leq 2K\sqrt{2Td\log\left(\lambda + \frac{(T-1)L}{d}\right)} \cdot \left(\lambda^{1/2}S + R\sqrt{2\log\left(\frac{1}{\delta}\right) + d\log\left(1 + \frac{(T-1)L}{\lambda d}\right)}\right). \tag{2.24}$$

We are mainly interested in the interrelated arms. Our regret bound depends on the number of invoked arms $|N_t(a)|$ or loose K. Figure 2.3 shows experimentation of our bound applied to the networked bandit problem. Our algorithm keeps the regret as low as possible, and can reach $\overline{\mathcal{R}}_t/t \to 0$ with high probability when t is large enough.

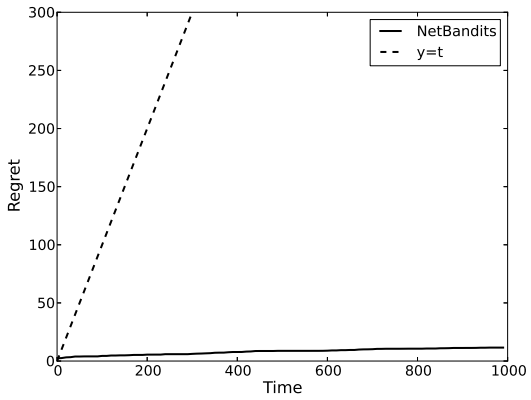


Fig. 2.3 *An example of the regret value in 10-arm networked bandits. The experiments are repeated 100 times and the average regrets are shown. $y = x$ is provided for comparison.*

2.6 Practical issues

In real-world applications, according to different assumptions about the network topology of arms, we can consider special cases of the networked bandit problem. We focus on $N_t(a)$. In our algorithm, we make the very loose assumption that $N_t(a)$ varies over time. However, the network topology is sometimes stable over a fixed duration. For example, a school social network is stable for the duration of a semester. This means that $N_t(a) = N_0(a)$, which is a special case of network bandits. In other cases, for example when inquiring users in the same company, we only need to consider their colleagues or a group of the same interest.

2.6.1 Dynamic network

In the networked bandit problem the network topology of arms is usually dynamic over time, which means for each round t we have different $N_t(a)$. Although we assume that $N_t(a)$ will be active after the forecaster selects a, we omit how to generate $N_t(a)$ and how arm a invokes $N_t(a)$, which are not our primary concerns. Instead, we simplify our problem using the simple setting of selecting an arm and then receiving payoffs from invoked arms. We assume that we can observe invoked arms $N_t(a)$. In practice, we can directly obtain $N_t(a)$ by predefining the arms, for example as neighbors or groups, or by observing feedback and collecting arms which provide feedback. We focus on how to select arms in order to maximize the total payoff, and therefore we concern the arms of $N_t(a)$ and the forecaster can obtain the invoked bandits over the course of collecting the payoffs from the network. In particular, at each round the algorithm observes the network topology of arms; it then decides which arm to select using the knowledge of the network topology and historical payoff information.

In Algorithm 2.2, we provide a pseudo-code for the selection at each round in a dynamic network.

Algorithm 2.2. Selection at round t in dynamic network

1: For each arm we have $\hat{w}_a$ and observe the context $x_{a,t}$
2: For each arm we collect $N_t(a)$
3: **for** each $a \in \mathcal{K}_t$ **do**
4: Compute $B_{a,t}$
5: **end for**
6: Select arm $a_t = \arg\max_{a \in \mathcal{K}_t} B_{a,t}$
7: Observe the payoffs $\{y_{a,t} | a \in N_t(a_t)\}$ from the network
8: For each arm $a \in N_t(a_t)$ update X_a, Y_a and $\hat{w}_a$

2.6.2 Static network

We make the simple assumption that the network topology is fixed. In other words, the relationships between arms do not change. For all t, we have $G_t = G_0$, $\mathcal{K}_t = \mathcal{K}_0$ and $N_t(a) = N_0(a)$. For example, DBLP, Last.FM, and many offline social network datasets are of fixed duration. This is a degenerate version of our problem and can be solved using our algorithm.

In Algorithm 2.3, we provide a pseudo-code for the selection at each round in a static network.

Algorithm 2.3. Selection at round t in static network

1: For each arm we have $\hat{w}_a$ and $N_0(a)$, and observe the context $x_{a,t}$
2: **for** $a \in \mathcal{K}_0$ **do**
3: Compute $B_{a,t}$
4: **end for**
5: Select arm $a_t = \arg\max_{a \in \mathcal{K}_0} B_{a,t}$
6: Observe the payoffs $\{y_{a,t} | a \in N_0(a_t)\}$ from the network
7: For each arm $a \in N_0(a_t)$ update X_a, Y_a and $\hat{w}_a$

2.6.3 Neighborhood or group

In networks, especially in social networks, a simple yet common assumption is that a node largely influences its neighborhoods or community [39, 76]. Moreover, some applications only focus on people who have the same interest or are in a group. This makes it possible to assume that the selected arm only invokes its neighbors or a group; that is, $N_t(a) = Neig_t(a)$, where $Neig_t(a)$ indicates the neighbors of a. We can only collect the payoffs of neighbors of an arm, and therefore $N_t(a)$ is appropriate. Although there are also two cases in this situation - static and dynamic - here we focus only on the neighborhood in dynamic network.

In Algorithm 2.4, we provide a pseudo-code for the selection at each round with specific $N_t(a)$.

2.7 Experiments

2.7.1 Illustrative example

We first illustrate our model by a synthetic example (Figure 2.4), which contains 10 arms (A0-A9) randomly connected at each round. At different rounds, the networks are different. At rounds $t = 11$ and $t = 20$, the upper bound B (the second row, blue) is large; however, the

Algorithm 2.4. Selection at round t with neighborhood

For each arm we have $\hat{w}_a$ and observe the context $x_{a,t}$
For each arm we collect $Neig_t(a)$
for $a \in \mathcal{K}_t$ **do**
    Compute $B_{a,t}$
end for
Select arm $a_t = \arg\max_{a \in \mathcal{K}_t} B_{a,t}$
Observe the payoffs $\{y_{a,t} | a \in Neig_t(a_t)\}$ from the network
For each arm $a \in Neig_t(a_t)$ update X_a, Y_a and $\hat{w}_a$

expected estimation is small (the third row, red) because the variance is large. Our algorithm selects the arm with maximal upper bound. We also show the real payoffs of all the arms, which are not known to the algorithm. At an early stage, the selection is poor compared to the real payoff (the fourth row, green). At round $t = 20$, our algorithm chooses A0, however, the best is A1, illustrating that the expected estimation is small and the algorithm can try another arm that has potential, but with uncertainty. Later, selection becomes efficient and at $t = 120$ the algorithm chooses A1 since more information has been learned and the upper bound becomes more stable with lower penalty. This is close to the real situation and provides a good estimation.

2.7.2 Baselines and performance metric

In this section we evaluate the proposed NetBandits strategy on four synthetic datasets and two public real-world datasets. We perform two types of experiments: simulation experiments and offline evaluation of two real applications.

We compare our proposed method against two baselines: a state-of-the-art algorithm for the contextual bandit problem, referred as TraBandits, and the random strategy. Since there is no existing method for the networked bandit problem, these methods are altered a little for networked bandits. The details are follows:

- **TraBandits:** a state-of-the-art method for contextual bandits with linear payoff models [88]. The algorithm is a UCB style method with the linear payoff assumption that always selects the arm using highest UCB at each round.
- **Random:** a simple strategy that just randomly selects an arm.

We use two methods to assess the performance of our algorithm. We first analyze the average payoff at each round, and then we analyze the cumulative payoff at each round, which ignores the performance of the algorithm at each fixed round but gives an overall view of the lifetime performance of the algorithm.

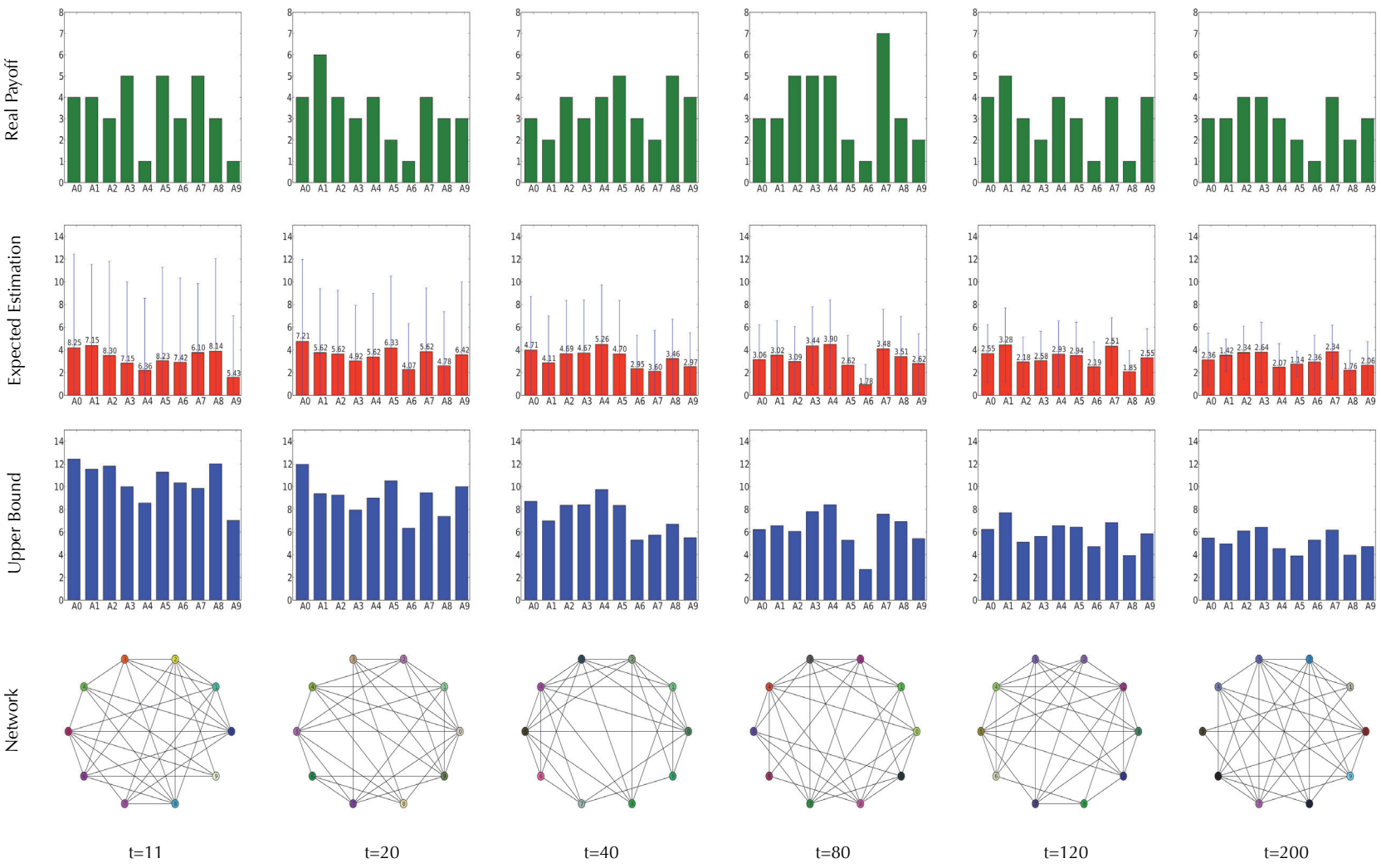


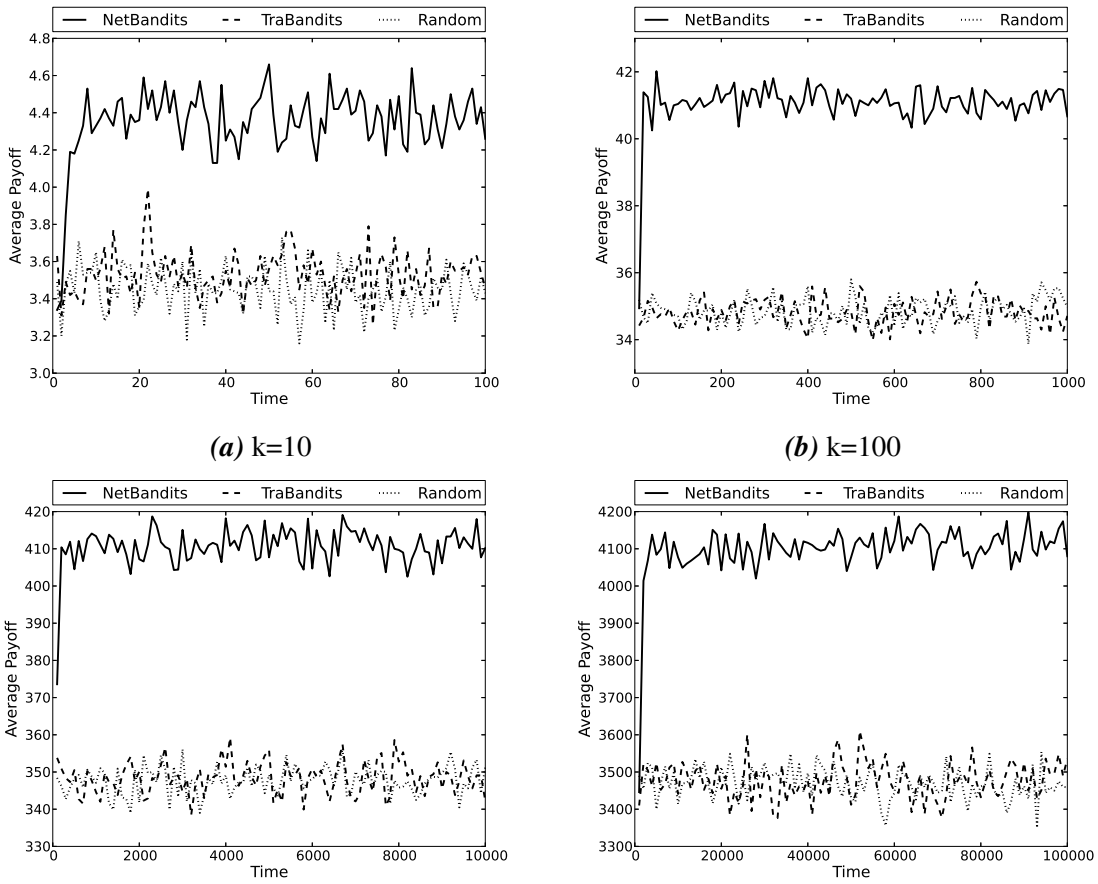
Fig. 2.4 *Illustrative synthetic example of exploration-exploitation trade-off. Bottom, arms with networked topology. Second row: the upper bound B for each arm computed using NetBandits. Third row: the expected estimation* ν*, where bar denotes the estimation and vertical line denotes the penalty of estimation. Fourth row: the real payoff of each arm.*

2.7.3 Simulation experiments

We test our algorithm on a series of synthetic datasets. In contrast to previous work, we need to construct the network topology, which can be either static or dynamic. Static network is a special case of dynamic situation and static network is generated in advance and remains unchanged. We therefore construct the networked bandits based on the dynamic network as follows: we first construct a fixed number k of nodes, which are considered as arms. We then randomly create edges between them, which are used to generate the relations for each arm. Neighborhood is considered as relationship. For every node, we assign different norm random vector $u_i, u_i \in \mathbb{R}^{10}$ and we use the following stochastic model to generate its payoffs: $y_a(x) = x^\top u_a + \varepsilon_a$, where ε_a is uniformly distributed in a bounded interval centered around zero and u_a and ε_a are not known to the decision algorithm. For contextual information, at each round t we randomly create a set of context vectors $\{x_{1,t}, \cdots, x_{k,t}\}, x_{k,t} \in \mathbb{R}^{10}$. The network topology does not have strict assumptions and is created simply: in the dynamic situation, we generate the network topology at each round (relationships between the nodes change at each round). We randomly create $k^2/3$ edges between the nodes, and therefore for most nodes the relations will be no greater than $k/3$.

We present the results for k =10, 100, 1000, and 10000 arms with dynamic network topology. In Figure 2.5 and Figure 2.6 we present the results of average payoff and cumulative payoff; our NetBandits outperforms the other baselines. TraBandits does not work well, indicating that best single arm does not always have the best payoff in a network but also depends on its relations. As per the network construction, the average payoff for each node is around $k/3$ if the node and its relations provide feedback, and the average payoff of TraBandits and Random is around $k/3$. For example, as shown in Figure 2.5, when $k = 10$ the payoff ranges from 3.2 to 3.6; when $k = 100$ the payoff ranges from 34 to 35; when $k = 1000$ the payoff ranges from 340 to 350; when $k = 10000$ the payoff ranges from 3450 to 3550. However, NetBandits usually performs better except the earliest time points, and its value is greater than 4.2 when $k = 10$, 40 when $k = 100$, 400 when $k = 1000$, and 4500 when $k = 10000$. This is because NetBandits performs more exploration than exploitation to begin with. Figure 2.6 shows that our algorithm obtains the best cumulative payoff over all rounds. As average payoff improves, NetBandits also exhibits higher cumulative payoff. This indicates that more early exploration improves later selections, leading to a fairer assessment of the performance of the different algorithms.

The running time of NetBandits according to different numbers of arms and network topology is also shown in Table 2.1, which demonstrates the running time increases rapidly with the scale of the networks. For example running time with $k = 100$ is slower than the time with $k = 10$ by more than the time with k^2 but less than the time k^3. The time taken depends



(a) k=10 ***(b)*** k=100

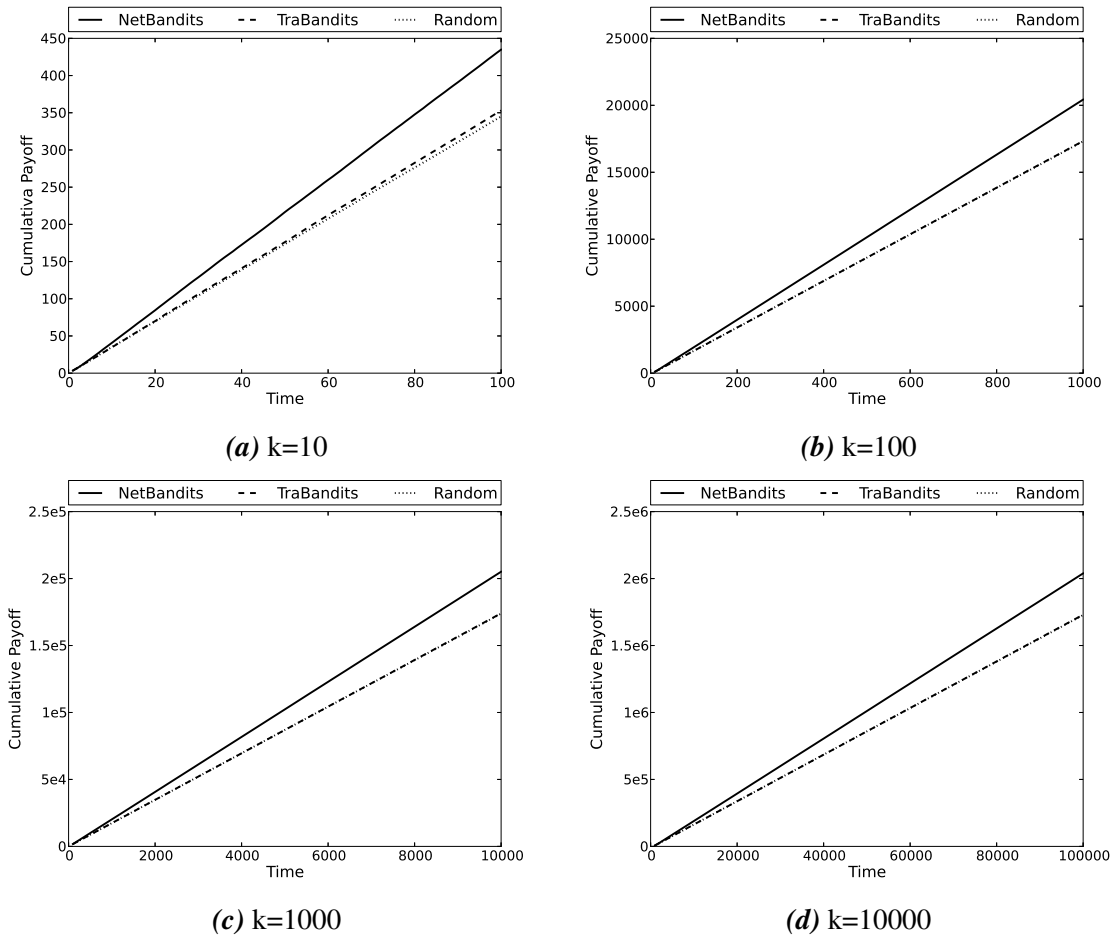
(c) k=1000 ***(d)*** k=10000

Fig. 2.5 *The average payoff at each round in dynamic networks.*

on the size of the network, including the number of nodes and edges. The time complexity of NetBandits is $O(TKN\Omega)$, where T is the total number of rounds, K is the number of arms, N indicates the average number of invoked arms, and Ω indicates the time taken to compute the parameters; it is no more than $O(TK^2\Omega)$ where $N = K$. It can be improved by calculating each arm in parallel for a large number of arms.

Arms	10	100	1,000	10,000
Avg of invoked arms	3	33	333	3,333
Total rounds	100	1,000	10,000	100,000
Time (second)	0.1	23.4	2,034.2	173,628.3

Table 2.1 *Running time results of NetBandits on four synthetic datasets.*



(a) k=10 ***(b)*** k=100

(c) k=1000 ***(d)*** k=10000

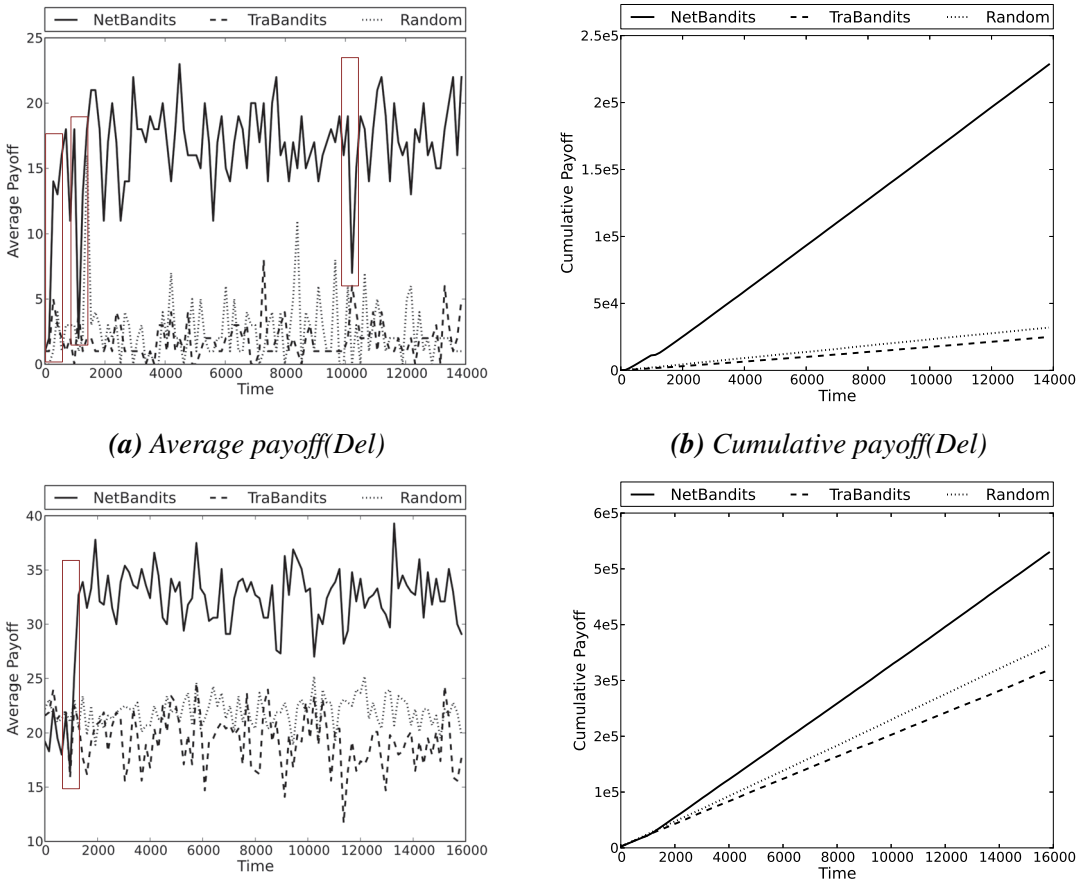
Fig. 2.6 *The cumulative payoff at each round in dynamic networks.*

2.7.4 Real-world datasets experiments

We also test our algorithm on two publicly available real-world datasets[1]: Delicious Bookmarks, a dynamic dataset, denoted by Del; and Last.FM, a static dataset, denoted by LFM.

Delicious Bookmarks is a social network for storing, sharing, and discovering web bookmarks. The Del dataset contains 1,861 nodes and 7,668 edges and 69,226 URLs described by 53,388 tags. Payoffs are created using the information about the bookmarked URLs for each user: the payoff is 1 if the user bookmarked the URL, otherwise the payoff is 0. Pre-processing is performed by breaking the tags down into smaller fragile items made up of single words, ignoring the underscores, hyphens, and dashes. Each word is represented using the TF-IDF context vector based on the words of all tags, i.e., these feature vectors are the context vectors. PCA was performed on the dataset and the first 16 principle components were selected as context vectors building a linear function based on payoff records for each

[1] http://grouplens.org/node/462



(a) *Average payoff(Del)*

(b) *Cumulative payoff(Del)*

(c) *Average payoff(LFM)*

(d) *Cumulative payoff(LFM)*

Fig. 2.7 *The average payoff and cumulative payoff for two real-world datasets.*

user. This linear function generates a payoff when given a new context. At each round t, we provided $x_{k,t} \in \mathbb{R}^{16}$ for all k users.

The Last.FM dataset is a music website that builds a detailed profile of each user's musical taste by recording details of the tracks that the user listened to from a range of digital devices. LFM contains 1,892 nodes and 12,717 edges and has 17,632 artists described by 11,946 tags. We used the listened-to artists information to construct payoffs: if the user listened to an artist at least once the payoff is 1, otherwise the payoff is 0. Similar pre-processing was performed as Delicious Bookmarks. Compound tags were broken down into several corresponding single words resulting in 6,036 words. We represented context features using the TF-IDF features, and after PCA the first 16 principle components were selected as context vectors. For each user we then built a linear function based on payoff records. This linear function can generate a payoff when given a new context. At each round, we provided $x_{k,t} \in \mathbb{R}^{16}$ for all users k.

We constructed the network topology according to the social network of the users. Neighborhood is considered as relationship. The linear payoff function for each user is learned in advance and unknown to the algorithm, which decides its next selection according to previous feedback.

For the Del dataset, there exists the timestamp information that records when contact relationships were created, and we can therefore construct a dynamic network according to the timestamps. Timestamps are from 1146752335000 to 1288104100000; we therefore divide them into 14 groups according to the first three numbers (114, 115, ..., 128). We set the total rounds $T = 14000$ and update the network every 1000 rounds. For the Last.FM dataset, there is no time information, thus we constructed a static network.

The results of average payoff and cumulative payoff are shown in Figure 2.7. Our algorithm outperforms the other baselines. Although the two networks have a similar number of users, LFM has more relationships and the average and cumulative payoff results are higher than Del. For the average payoff of Del, there exist three low intervals marked by (red) rectangles in Figure 2.7(a). These occurred at the beginning and close to round $t = 1000$ and $t = 10000$. Since many new nodes and edges are added at these rounds and NetBandits performs more exploration than exploitation. The payoffs improve after exploration. For the average payoff results of LFM, there is a low interval at the start, denoted by the (red) rectangle in Figure 2.7(c), because NetBandits is trying to select possible better arms and perform exploration; then later the performance improves. Figures 2.7(b)(d) show that the cumulative payoff results of NetBandits increase faster, and are much greater than the other algorithms, and demonstrate that exploration does not hurt the total performance.

2.8 Conclusion and future work

In this work we formalized a new bandit problem, termed networked bandits. We presented the novel problem of how to select the arm with multiple payoffs in networked bandits by considering a multi-armed bandit of interconnected arms, one of which can invoke other related arms at each round. After selecting an arm, we can obtain payoffs from this arm and its relations.

We considered this approach in the contextual bandit setting and assumed disjoint linear payoffs for arms. We proposed a new networked bandit algorithm NetBandits that considers the uncertainty of the payoffs using integrated confidence sets. We also provided a regret bound for our solution. Our experiments show that it is better to consider both the network topology and the payoffs of arms, and it is observed that our approach performs well in this setting.

The networked bandit problem requires further work. Some interesting problems still remain, such as how to model $N_t(a)$. In our work we do not make any assumption about the structure of the network topology; for example the hub may have higher priority, and it is possible to find a more efficient method for some fixed structures.

Another problem is arm complexity. We assume that one arm invokes other arms, which in turn can invoke other arms sequentially, with processing occurring at the same time. However in some real applications, the structure is possible to be much more complex and evolve over time, which is likely to delay the payoffs.

Chapter 3

Multi-view bandits

In stream-based multi-view learning, each sequentially received example is represented by n views, each of which is obtained from a feature generator or source and embedded in a reproducing kernel Hilbert space (RKHS). We examine the problem of selecting a near-optimal subset of m views from n views, which will be exploited to make the prediction incurring an observable payoff. Unlike subset selection in stochastic bandit settings, the forecaster can observe the changing environment information and the prediction may perform very differently with different view subsets. We therefore propose a multi-view bandit framework to address this problem, in which we introduce the multi-view simple regret, provide an upper bound of the expected regret for our algorithm, and study the generalization bounds. The proposed multi-view bandit algorithm relies on the Rademacher complexity of the co-regularized kernel classes. We prove a simple regret bound using the Rademacher complexity and show that the consistency of different views improves the simple regret. Experimental results on real and synthetic datasets demonstrate the effectiveness of our algorithm.

3.1 Introduction

In multi-view learning, examples are represented by different context vectors or abstractly different "views", in which each particular view is modeled by a function. Multi-view learning algorithms jointly optimize all the functions to simultaneously exploit redundant views of the same input examples and thus improve learning performance. For example, a multiple biometrics system collects various biometric traits such as face, fingerprint, signature and iris from a person to achieve high identification rates.

In recent years, a large number of multi-view learning algorithms have been developed to explore the consistency and complementarity of different views [139], and can be grouped

Fig. 3.1 *An example of the application of SMVL to the automatic navigation control of a robot.*

into the following three categories: (1) co-training [24, 123, 141], (2) multiple kernel learning [85, 87, 110, 126] and (3) subspace learning [4, 36, 74, 80]. Most existing algorithms are pool-based, i.e. all the labeled and unlabeled data are accessible in the training and testing stages, respectively.

Unlike pool-based multi-view learning (PMVL), stream-based multi-view learning (SMVL) aims to sequentially identify the most appropriate views for the forecaster, in which each example represented by different views is drawn at the same time from the data sources. Thus in SMVL, the forecaster can access all the views of the currently received example and make a prediction based on the selected views and the historical information (i.e. historically selected views and payoffs). The environment then returns feedback (payoff) in response to the forecaster's prediction. Figure 3.1 shows the application of SMVL to the automatic navigation control of a robot. In this example, a robot captures different directions using cameras located at different angles in each step. It then selects a small number of views to determine the direction of movement.

It is generally difficult to model the compatibility and temporal changes independently based on each view. In practice, a forecaster usually explores the unknown by collecting the prediction feedback based on the view subset in real time to evaluate the compatibility of views. Traditional PMVL algorithms consistently treat different views over all the examples (i.e. they do not consider that different views may play different roles on different examples) and ignore the environment feedback. Considering redundant and noisy views seriously affects the prediction, and regarding different views of different examples affects the prediction in different ways (i.e. the environment changes over time), PMVL algorithms ignore the environment feedback and thus cannot perform well for SMVL tasks. In addition, given limited budget or computational resources, a predictor can only exploit a small number of observed views. It thus becomes indispensable to design a decision strategy for sequentially selecting a subset of views to deal with changed environments.

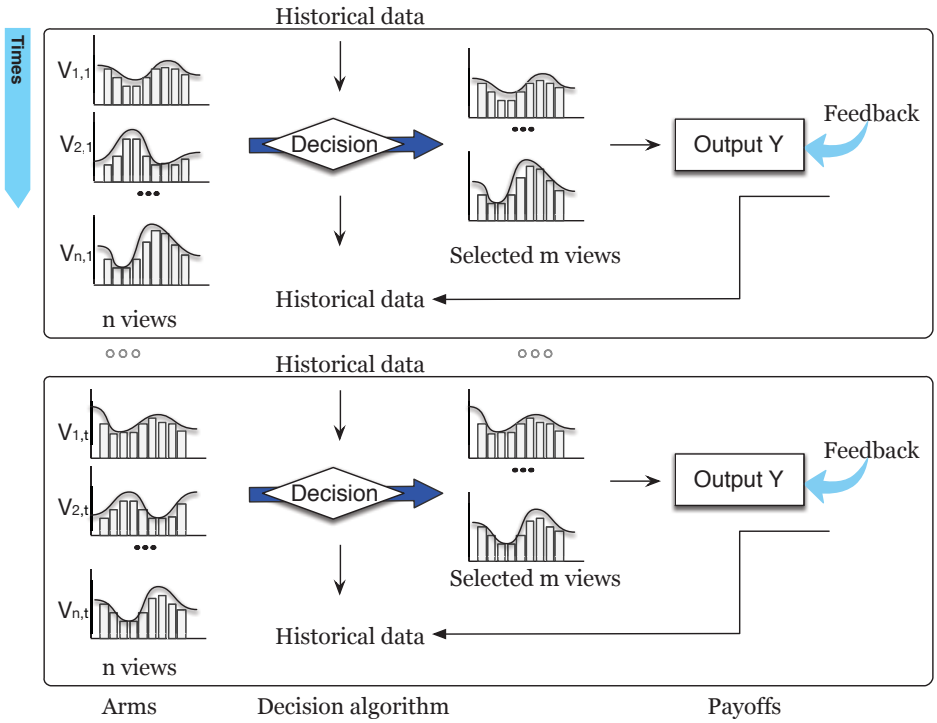


Fig. 3.2 *Using bandit framework to model stream-based multi-view learning.*

Sequential decision making has recently been extensively studied in the literature [31]. The multi-armed bandit is a classical decision theory and control model. It was first investigated by Robbins (1952), who proposed strategies that asymptotically attain an average reward that converges at the limit to the reward of the best arm. Bandits have been studied extensively in recent years [31] in a number of settings, such as the stochastic setting [11, 14, 66, 73, 91], adversarial setting [9, 10, 15, 33], and contextual setting [13, 117, 134]. In real-world applications, multi-armed bandits are effective for solving situations where an exploration-exploitation dilemma is encountered.

SMVL can be modeled under the bandit framework, as shown in Figure 3.2. At each time step t, the forecaster observes different views of the received example $\{x_{(1,t)}, \cdots, x_{(n,t)}\}$, in which each view is obtained from a feature generator or source and embedded in a reproducing kernel Hilbert space (RKHS). The forecaster selects a subset of views and then makes the prediction based on the selected view subset. The environment returns payoff after the forecaster makes the prediction. The forecaster simply aims to obtain a high payoff. Note that we examine a setting in which only a subset of all views can be used, and the environment provides payoff that corresponds to the current prediction.

The main challenge is how to make the decision in a changing environment. The forecaster relies on the currently selected views and historical information, including previously selected views, sequential predictions and the corresponding payoffs. For example, at various places

in robot navigation, such as a straight road or curved road, the forecaster's prediction depends on different views. The forecaster should acquire the information not only from the historical data but also from the new environment. It is important to make the decision by considering the exploration and exploitation of different views at each time step, since all the views potentially have different effectiveness in relation to the current or historical information. Bandit therefore provides an ideal platform for solving this problem.

In this work, we formalize the problem under a new bandit framework, i.e. multi-view bandit, in which each of the n views $\{V_1, \cdots, V_n\}$ is defined as an arm, and at each time step t, a set of context vectors $\{x_{(1,t)}, \cdots, x_{(n,t)}\}$ represents an example defined by n views. The context vector $x_{(i,t)}$ for the i-th view V_i corresponds to the i-th arm. The forecaster is allowed to select m views ($m \leq n$), then exploits them for prediction and thus may suffer a loss according to the prediction. The loss is defined as payoff. In contrast to the task of subset selection in bandits [30, 83] which tries to identify the best subset in the stochastic bandit setting, multi-view bandit selects a subset of views depending on the joint context vectors (arms) and conducts the prediction based on the combination of all the selected views.

Under the frame of multi-view bandit, we propose a randomized algorithm CoRLSUB which depends on the confidence analysis of the generalization of the co-regularized least squares (CoRLS) [115]. CoRLSUB first generates several candidate subsets, each of which contains m randomly selected views; it then estimates the upper bound of loss generalization for each candidate subset, which employs Rademacher complexity of the space of compatible predictions; and lastly, it chooses the subset associated with the lowest upper bound. We theoretically analyze CoRLSUB and the multi-view simple regret to show that its upper bound scales as $O(1/\sqrt{t})$. We also show that the consistency of different views improves the simple regret bound. The theoretical analyses explain how the choice of view subset affects the generalization performance.

Multi-view bandit can be applied in various important domains, such as the automatic navigation control of robots, autonomous driving and clinical decision analysis, in which an agent sequentially optimizes the decision by selecting the consecutively received types of environmental information. For example, in the automatic navigation control of robots, particular views can be used to make predictions about when the robot stops and decides where to move. Using CoRLSUB, the robot can actively select the views from several angles such as 45°, 90° and 135°, rather than from every 15° angle. In this way, the forecaster can make a successful prediction based on the selected views. In clinical decision analysis, the clinician collects different kinds of information, such as medical records, observations, medical imaging and so on, in a patient consultation. The clinician can then select the most valid information from several sources to determine the most appropriate treatment.

Our contributions are three-fold: (1) to the best of our knowledge, we are the first to propose subset selection in the multi-view problem; (2) we are the first to address subset selection in the SMVL setting; and (3) we propose the multi-view bandit algorithm CoRLSUB and prove the multi-view simple regret bound.

The rest of this work is organized as follows. In Section 3.2 we review related work about bandit and multi-view learning. In Section 3.3 we state the problem of multi-view bandit. Regarding multi-view bandit, Section 3.4 presents the core contribution of this work including a multi-view bandit algorithm CoRLSUB and its index calculation, and Section 3.5 theoretically analyzes the regret. Experiments in Section 3.6 verify the effectiveness of the proposed CoRLSUB for multi-view bandit and demonstrate that the selective strategy can improve performance. Section 3.7 provides detailed proofs of the theoretical results, and Section 3.8 concludes this work.

3.2 Related work

Inspired by research on multi-arm bandits and multi-view learning, we propose a multi-view bandit method which sequentially selects a subset of views in SMVL and thus is different from traditional bandit problems such as stochastic bandit, non-stochastic bandit and contextual bandit.

The multi-armed bandit problem was first studied in depth by Auer et al. [15], who defined the measure based on the cumulative regret bound and theoretically analyzed the selection strategy. The regret after n rounds is defined as

$$R_n = \max_{i=1,\cdots,K} \sum_{t=1}^{n} g_{i,t} - \sum_{t=1}^{n} g_{I_t,t}, \tag{3.1}$$

where g is the payoff, K is the number of arms and I_t is the selected arm at the round t. Many works have subsequently been developed to address this problem.

In the stochastic setting, Auer et al. [14] introduced an upper confidence bound algorithm and provided a finite-time regret bound. In the non-stochastic setting, Auer et al. [13] proposed the EXP3 algorithm and provided a regret of $O(\sqrt{KT\log T})$ bound. Audibert and Bubeck [9] provided a regret of $O(\sqrt{KT})$ algorithm. In the contextual setting, Auer et al. [15] proposed two algorithms LINREL and SUPLINREL based on the linear payoff assumption, and provided regret analysis for SUPLINREL with a regret of $\hat{O}(\log^{3/2} K\sqrt{dT})$ bound. Li et al. [88] and Chu et al. [37] proposed LinUCB and SupLinUCB algorithms and proved that the regret of SupLinUCB was $O(\sqrt{Td\log^3(KT\log(T)/\delta)})$.

Another natural measure is defined based on the simple regret bound [12, 28] regarding the stochastic setting, which is already used in the sub-optimality of an algorithm in the multi-armed bandit problem. The simple regret after t rounds is defined as

$$R_t = \max_{i=1,\cdots,K} g_{i,t} - g_{I_t,t}. \tag{3.2}$$

Bubeck et al. [28] defined simple regret as the regret on an one-shot instance of a game for the selected arm. They proposed the UCB(α) algorithm and showed that for distribution-dependent bounds the asymptotic optimal rate of decrease in the number of rounds is exponential. Audibert et al. [12] proposed a high exploration UCB strategy policy and a new algorithm based on the successive rejects. They showed that identifying the best arm requires a number of samples of $\sum_i 1/\Delta_i^2$, where Δ_i indicates the difference between the mean reward of the best arm and that of the arm i.

With respect to multi-view bandit, we define a new measure, multi-view simple regret, for SMVL, i.e.,

$$R_t = L(\varphi_{\mathcal{S}_t}(x), y) - L(\varphi_{\mathcal{S}_*}(x), y), \tag{3.3}$$

where $\mathcal{S}$ indicates a subset of arms, L is the loss function and $\varphi_{\mathcal{S}_*}(x) = \min_{\mathcal{S}} L(\varphi_{\mathcal{S}}(x), y)$. This measure is different from the previous two measures in the following three aspects: (1) the payoff or loss is generated by a set of arms $\mathcal{S}$ rather than one arm; (2) the decision is made by combining the selected arms; and (3) the regret depends only on the currently received context vectors of the selected views. These differences explain why existing bandit algorithms cannot be applied to SMVL tasks.

Several works have been developed to address the problem of subset selection in a stochastic setting by iteratively removing ineffective arms. Kalyanakrishnan and Stone [83] proposed several methods using the PAC model to retain the best-m arms (the LUCB method). Bubeck et al. [30] proposed a new method, introduced identification complexity, and exploited their SAR machinery to construct a parameter-free algorithm to identify the best m arms. These bandit algorithms filter arms and retain the optimal set which includes only independent arms and returns high payoffs. From the perspective of arm selection, multi-view bandit aims to select a subset of arms but is different from these works because (1) these works are not applicable to a changing environment and so cannot solve SMVL problems, and (2) the selected arms in the multi-view bandit can be dependent or independent, and both the decision and payoff are determined by combining all the selected arms. Therefore, this is the first work that solves the SMVL problem within the bandit framework.

In multi-view learning, different views of objects are obtained from a variety of sources or feature subsets and are incorporated to make the prediction. In contrast to single view

learning, multi-view learning simultaneously optimizes all the functions from different views to improve the learning performance by exploiting the consistency and complementarity of these views. Existing multi-view learning algorithms can be classified into four groups [139]: co-training, multiple kernel learning, subspace learning, and single view selection.

- Co-training maximizes the mutual agreement on two distinct views of the unlabeled data. Nigam and Ghani [102] generalized expectation-maximization by assigning changeable probabilistic labels to unlabeled data. Muslea et al. [98–100] combined active learning with co-training and proposed a robust semi-supervised learning algorithm. Sindhwani et al. [124] constructed a data-dependent "co-regularization" norm, where each view is associated with a particular reproducing kernel Hilbert space.

- Multiple kernel learning assumes that a kernel corresponds to a particular view and aims to improve generalization performance by either linearly or non-linearly combining multiple kernels. Lanckriet et al. [87] formulated MKL as a semi-definite programming problem and proved an estimation error bound $O(\sqrt{(k/\gamma^2)/n})$, where γ is the margin of the learned classifier. Sonnenburg et al. [125] developed an efficient semi-infinite linear program and made MKL applicable to large scale problems.

- Subspace learning-based algorithms obtain a latent subspace shared by multiple views under the assumption that the input views are generated from this latent subspace with perturbations. Canonical correlation analysis (CCA) [74] and its kernel extension [4] are applied to multi-view data to select the shared subspace. Quadrianto and Lampert [108] and Zhai et al. [142] studied multi-view metric learning by constructing embedding projections from multi-view data to a shared subspace. Salzmann et al. [118] aimed to find a latent subspace, in which the information is correctly factorized into shared and private parts across different views.

- Single view selection algorithms explore specific criteria to identify the optimal view for subsequent decision making. Paletta and Pinz [104] provided a near-optimal decision strategy in terms of sensorimotor mappings using reinforcement learning for active object recognition, and showed that ambiguous views may exist. A particular view was selected to achieve maximum discrimination. Jia et al. [78] introduced active view selection strategies for object and pose recognition. Borrowing the idea of AdaBoost and actively selecting optimal view to update the weights of instances and classifiers, the combined classifier was used to reduce classification error.

Multi-view bandit is different from conventional multi-view learning because: (1) existing multi-view learning algorithms are pool-based and developed for a static environment, while

multi-view bandit is stream-based and applied to a changing environment; (2) unlike view combination and single view selection in most existing results on multi-view learning, multi-view bandit is mainly concerned with how to select a subset of views to improve generalization; and (3) existing multi-view learning algorithms do not consider environment feedback, which is critical in multi-view bandit.

3.3 Multi-view bandits

Multi-view bandit considers the problem of selecting a near-optimal subset of m views($m < n$) from n views $\{V_1, \cdots, V_n\}$ of each sequentially received example $x_t = \{x_{(1,t)}, \cdots, x_{(n,t)}\}$ at time t, making the prediction by integrating the optimally selected views and the historical information, and receiving the payoff returned by the environment given the prediction. Thus, at different time steps, multi-view bandit may select different views to obtain optimal prediction.

At time step t, the prediction is defined by the linear combination of the functions over $\mathcal{S}_t = \{x_{(t_1,t)}, \cdots, x_{(t_m,t)}\}$ corresponding to the selected views $\{V_{t_1}, \cdots, V_{t_m}\}(1 \leq t_1 < t_2 < \cdots < t_m \leq n, m < n)$, i.e. for any $(f_{t_1}, \cdots, f_{t_m})$ the prediction function is given by

$$\varphi_{\mathcal{S}_t}(x_t) = w_{t_1} f_{t_1}(x_{(t_1,t)}) + \cdots + w_{t_m} f_{t_m}(x_{(t_m,t)}), \tag{3.4}$$

where the combination weights $w_{t_1}, \cdots, w_{t_m} \in \mathbb{R}$ are predefined, and functions f_{t_i} for $1 \leq i \leq m$ are learned from the retained historical examples and their corresponding payoffs.

The views $\{V_1, \cdots, V_n\}$ are considered as arms and each sequentially received example x is formed by all the context vectors of arms. The example x_t received at time step t can be naturally decomposed as $x_t = \{x_{(1,t)}, \cdots, x_{(n,t)}\}$, where each $x_{(i,t)}$ represents a particular "view" of the input x and corresponds to a specific arm embedded in an RKHS.

The sequential evaluation protocol is described as follows: at time step t, according to the historical information (including retained historical context vectors and payoffs) and the currently received example $x_t = \{x_{(1,t)}, \cdots, x_{(n,t)}\}$, the forecaster first selects a near-optimal set of m views $\mathcal{S}_t = \{V_{t_1}, \cdots, V_{t_m}\}$ and then makes the prediction based on the context vectors over the subset $\mathcal{S}_t$. After obtaining the prediction, the environment returns a payoff to the forecaster. The payoff is 1 when the prediction is correct, and otherwise is 0. The objective of the forecaster is to make a correct prediction or obtain a high payoff. In SMVL, we define the loss as the payoff. Let $L : Y \times Y$ be a nonnegative loss function, and define the expected

risk of a prediction function φ as

$$R(\varphi) = \mathbb{E}L(\varphi(x), y). \tag{3.5}$$

We define the multi-view simple regret as

$$R_t = L(\varphi_{\mathcal{S}_t}(x), y) - L(\varphi_{\mathcal{S}_*}(x), y), \tag{3.6}$$

where $\mathcal{S}_t$ indicates a subset of arms and $\varphi_{\mathcal{S}_*}(x) = \min_{\mathcal{S}} L(\varphi_{\mathcal{S}}(x), y)$. Then the expected regret is

$$\mathcal{R}_t = \mathbb{E}[L(\varphi_{\mathcal{S}_t}(x), y) - L(\varphi_{\mathcal{S}_*}(x), y)]. \tag{3.7}$$

We exploit the multi-view simple regret as the performance measure, because it is straightforward to measure the difference between the payoffs obtained by selecting $\mathcal{S}_*$ and those achieved by the forecaster in expectation.

3.4 CoRLSUB

Multi-view bandit aims to obtain the optimal view subset for prediction, which is NP hard (Proposition 3.1). We thus propose an approximate algorithm CoRLSUB, a variant of upper confidence bound algorithm based on CoRLS, in which a near-optimal view subset is selected.

Proposition 3.1. *It is NP hard to obtain the optimal view subset for the subsequent prediction. (Detailed proof is given in Section 3.7.)*

At time step t, given n views, there exist $\binom{n}{m}$ possible view subsets, and $T_{\mathcal{S}_i}(t)$ denotes the set of historically selected view subsets $\mathcal{S}_i$ for $1 \leq i \leq t$ and their corresponding payoffs at the first t time steps. Each view subset is associated with an index which is the sum of $\hat{R}_t(\varphi_{\mathcal{S}})$ and $\xi_t(\mathcal{S}, x)$, where $\hat{R}_t(\varphi_{\mathcal{S}})$ is the empirical loss of $\varphi_{\mathcal{S}}$ and $\xi_t(\mathcal{S}, x)$ is the penalty of the empirical loss defined by the Rademacher complexity of CoRLS. The selected view subset at each time step corresponds to the lowest index. The main step is the calculation of the index of a given view subset, which will be detailed in Section 3.5.1.

The forecaster subsequently makes the prediction according to (3.3) based on the selected view subset $\mathcal{S}_t$. The environment returns a payoff after the prediction. If the prediction is correct, the environment returns 1; if the prediction is wrong, the environment returns 0. The forecaster collects the historical information of selected views and the payoff, and then updates $T_{\mathcal{S}_t}(t)$. The above procedure for CoRLSUB is summarized in Algorithm 3.1.

Algorithm 3.1. CoRLSUB

for round $t = 1, 2, \cdots$ **do**
  Observe context vectors $x_t = \{x_{(1,t)}, \cdots, x_{(n,t)}\}$
  for a randomly generated candidate view subset $\mathcal{S} \subseteq \{V_1, \cdots, V_n\}, |\mathcal{S}| = m$ **do**
    Compute the $\eta(\mathcal{S}, x_t) = \hat{R}_t(\varphi_{\mathcal{S}}) + \xi_t(\mathcal{S}, x_t)$, where

$$\hat{R}_t(\varphi_{\mathcal{S}}) = \frac{1}{|T_{\mathcal{S}}(t-1)|} \sum_{T_{\mathcal{S}}(t-1)} L(\varphi_{\mathcal{S}}(x_i), y_i)$$

    and

$$\xi_t(\mathcal{S}, x_t) = \frac{2r}{|T_{\mathcal{S}}(t)|} \sqrt{tr(\hat{k}_t)} + \frac{1}{\sqrt{|T_{\mathcal{S}}(t)|}} (2 + 3\sqrt{\ln(2/\delta)/2})$$

  end for
  Draw a view subset $\mathcal{S}_t$ by using $\arg\min_{\mathcal{S}} \eta(\mathcal{S}, x)$
  Make the prediction based on $\mathcal{S}_t$
  Observe the payoff from the environment: receive 1 when prediction is correct and 0 when prediction is wrong
  Update historical information $T_{\mathcal{S}_t}(t)$
end for

3.4.1 View subset calculation

We consider m randomly selected views $\mathcal{S}_t = \{V_{t_1}, \cdots, V_{t_m}\}$ corresponding to a set of observed context vectors $\{x_{(t_1,t)}, \cdots, x_{(t_m,t)}\}$ for each of the randomly generated candidate view subsets. Simply we use φ_t to abbreviate $\varphi_{\mathcal{S}_t}$. Let $\mathcal{H}_{t_1}, \cdots, \mathcal{H}_{t_m}$ be RKHS's of real-valued functions on X, associated with kernels $k_{t_1}, \cdots, k_{t_m} : X \times X \to \mathbb{R}$, respectively, and each k_{t_i} corresponds to a particular view V_{t_i}. Then, we have the product space $\mathcal{F}_t = \mathcal{H}_{t_1} \times \cdots \times \mathcal{H}_{t_m}$. The space of prediction $\hat{\mathcal{H}}_t$ is defined as the image of $\mathcal{F}_t$ under υ. That is

$$\hat{\mathcal{H}}_t = \upsilon_t(\mathcal{F}_t) = \{\upsilon_t(f_t) | f_t \in \mathcal{F}_t\}, \tag{3.8}$$

where $\upsilon_t(f_t) = \sum_{V_i \in \mathcal{S}_t} w_i f_i$ and $w_i \in \mathbb{R}$.

Lemma 3.1. *$\mathcal{F}_t$ is a Hilbert space.*

Lemma 3.2. *$\hat{\mathcal{H}}_t$ is a Hilbert space.*

We show $\mathcal{F}_t$ is a Hilbert space in Lemma 3.1 and $\hat{\mathcal{H}}_t$ is a Hilbert space in Lemma 3.2 (proofs can be found in Section 3.7). Moreover we can give an explicit expression of its reproducing kernel. Thus we can express the optimization on the $\hat{\mathcal{H}}_t$ as a finite-dimensional optimization problem by using the Representer Theorem.

We firstly formulate the optimization problem for solving the final prediction. Inspired by co-regularization in the two views setting [27, 124, 127], we consider the agreement

on context vectors between the predictors of different views. The disagreement term for compatibility can be defined as:

$$\gamma \sum_{V_a \neq V_b} [f_{t_a}(x_{(t_a,t)}) - f_{t_b}(x_{(t_b,t)})]^2, \tag{3.9}$$

where $\gamma \in \mathbb{R}^+$. The objective function for learning from m views considered in this work is then written as:

$$\underset{\varphi_t \in \hat{\mathcal{H}}_t}{\arg\min} \min_{(f_{t_1},\cdots,f_{tm}) \in \upsilon_t^{-1}(\varphi_t)} \hat{R}(\varphi_t) + \sum_{j=1}^{m} \alpha_j \|f_{t_j}\|^2_{\mathcal{H}_{t_j}} + \gamma \sum_{V_a \neq V_b} [f_{t_a}(x_{(t_a,t)}) - f_{t_b}(x_{(t_b,t)})]^2, \tag{3.10}$$

where $\alpha_1, \cdots, \alpha_m > 0$ are RKHS norm regularization parameters and $\gamma \geq 0$ is the disagreement norm regularization parameter.

We give a general form of the disagreement term. We define the following column vector of function evaluations on the context vectors:

$$\tilde{f}_t = (f_{t_1}(x_{(t_1,t)}), f_{t_2}(x_{(t_2,t)}), \cdots, f_{t_m}(x_{(t_m,t)}))^\top. \tag{3.11}$$

In multi-view regularization, the L^2-disagreement penalty $\sum_{V_a \neq V_b} [f_{t_a}(x_{(t_a,t)}) - f_{t_b}(x_{(t_b,t)})]^2$ is replaced by a more general form $\tilde{f}_t^T D \tilde{f}_t$, where $D \in \mathbb{R}^{m \times m}$ is a positive semidefinite matrix. The objective function (3.10) can be rewritten as follows:

$$\underset{\varphi_t \in \hat{\mathcal{H}}_t}{\arg\min} \min_{(f_{t_1},\cdots,f_{tm}) \in \upsilon_t^{-1}(\varphi_t)} \hat{R}(\varphi) + \sum_{i=j}^{m} \alpha_j \|f_{t_j}\|_{\mathcal{H}_{t_j}} + \gamma \tilde{f}_t^\top D \tilde{f}_t, \tag{3.12}$$

where D is an $m \times m$ matrix defined by

$$D = \begin{cases} m-1 & i = j \\ -1 & \text{otherwise} \end{cases}.$$

This objective function (3.12) can be expressed as a standard Tikhonov regularization problem over a new data-dependent RKHS.

Denote the point kernel matrix for the ath view by $K^a = (k_a(x_i, x_j))_{i,j=1}^{|T_{\mathcal{S}(t)}|}$, and define the block diagonal matrix $\mathcal{K}_t = diag(K_{t_1}, \cdots, K_{t_m}) \in R^{|T_{\mathcal{S}(t)}|m \times |T_{\mathcal{S}(t)}|m}$. Denote the diagonal matrices of these parameters as $\tilde{A}_{i,t} = diag(\underbrace{w_{t_1}, \cdots w_{t_1}}_{|T_{\mathcal{S}}(t)|}, \cdots, \underbrace{w_{t_m}, \cdots, w_{t_m}}_{|T_{\mathcal{S}}(t)|})$ and $\tilde{G} = diag(\underbrace{\alpha_1, \cdots, \alpha_1}_{|T_{\mathcal{S}}(t)|}, \cdots, \underbrace{\alpha_m, \cdots, \alpha_m}_{|T_{\mathcal{S}}(t)|})$. For each kernel, we then denote the column vector of

kernel evaluations between the history data and an arbitrary point $x \in X$ by

$$\tilde{k}_t(x) = (k_{t_1}(x_1,x),\cdots,k^1(x_{|T_S(t)|},x),\cdots,k^m(x_1,x),\cdots,k^m(x_{|T_S(t)|},x))^T. \tag{3.13}$$

We can show that $\hat{\mathcal{H}}_t$ is an RHKS with kernel $\hat{k}_t$ and it is proved in Theorem 3.1 (proof can be found in Section 3.7).

Theorem 3.1. *Let the kernel function be*

$$\hat{k}_t(z,x) = \sum_{j=1}^{m} \frac{w_{t_j}^2}{\alpha_{t_j}} k_{t_j}(z,x) - \gamma \tilde{k}_t^\top \tilde{A}_t \tilde{G}_t^{-1} (I + \gamma D \tilde{G}_t^{-1} \mathcal{K}_t)^{-1} D \tilde{G}_t^{-1} \tilde{A}_t \tilde{k}_t(z), \tag{3.14}$$

and define the norm of any $\varphi_t \in \hat{\mathcal{H}}_t$ as

$$\|\varphi_t\|_{\hat{\mathcal{H}}_t} = \sqrt{\min_{(f_{t_1},\cdots,f_{t_m}) \in \upsilon_t^{-1}(\varphi_t)} \sum_{j=1}^{m} \alpha_{t_j} \|f_{t_j}\|_{\mathcal{H}_{t_j}}^2 + \gamma \tilde{f}_t^\top D \tilde{f}_t}. \tag{3.15}$$

Then $\hat{\mathcal{H}}_t$ is an RHKS with kernel $\hat{k}_t$.

In RKHS, to let the prediction function lie in a certain subset of a function space, such as in a norm ball, we use a soft constraint $\Phi(f)$ on the function space. Our minimization problem can be rewritten as:

$$\min\{\hat{R}(f) + \lambda \Phi(f) : f \in \mathcal{F}\}, \tag{3.16}$$

where $\mathcal{F}$ is the function space over which we are optimizing, $\hat{R}$ is the empirical risk and λ is the regularization parameter. Thus according to Theorem 3.1 we can apply minimization (3.16) to solve the objective function (3.12). That is

$$\arg\min_{\varphi_t \in \hat{\mathcal{H}}_t} \hat{R}(\varphi) + \|\varphi_t\|_{\hat{\mathcal{H}}_t}^2, \tag{3.17}$$

which is a standard RKHS regularization problem, and the norm $\|.\|_{\hat{\mathcal{H}}_t}$ is written as

$$\|\varphi_t\|_{\hat{\mathcal{H}}_t} = \sqrt{\min_{(f_{t_1},\cdots,f_{t_m}) \in \upsilon_t^{-1}(\varphi_t)} \sum_{j=1}^{m} \alpha_j \|f_{t_j}\|_{\mathcal{H}_{t_j}}^2 + \gamma \tilde{f}_t^\top D \tilde{f}_t}. \tag{3.18}$$

Now we analyze the Rademacher complexity for the above algorithm according to the RKHS theory. We provide the definition of the empirical Rademacher complexity of a

function class $\mathcal{H}$ for a sample $x_1, \cdots, x_l \in \mathcal{X}$

$$\mathfrak{R}_l(\mathcal{H}) = \mathbb{E}_\sigma \left[\sup_{\varphi \in \mathcal{H}} \left| \frac{2}{l} \sum_{i=1}^{l} \sigma_i \varphi(x_i) \right| \right], \tag{3.19}$$

where the expectation is with respect to $\sigma = \{\sigma_1, \cdots, \sigma_l\}$, and σ_i are i.i.d. Rademacher random variables.

We try to find the final prediction function from a norm ball of a particular radius in the RKHS $\hat{\mathcal{H}}_t$. We denote the norm ball of radius r by

$$\hat{\mathcal{H}}_r = \{\varphi_t \in \hat{\mathcal{H}}_t |\ \|\varphi_t\|_{\hat{\mathcal{H}}_t} \leq r\} \tag{3.20}$$

and the norm is always bounded as $\|\varphi^*\|^2_{\hat{\mathcal{H}}_t} \leq r^2 := L(0)$, where 0 denotes the prediction function that always predicts 0.

Theorem 3.2 proves a lower bound and an upper bound for the empirical Rademacher complexity of the function class $\hat{\mathcal{H}}_r$ with a high probability.

Theorem 3.2. *Under our assumption, at time t, given the views subset $\mathcal{S}_t$, the empirical Rademacher complexity of $\hat{\mathcal{H}}_r$ on a sample $(x_1, y_1), \cdots, (x_{|T_\mathcal{S}(t)|}, y_{|T_\mathcal{S}(t)|})$ is bounded above and below by*

$$\frac{1}{2^{1/4}} \frac{2r}{t} \sqrt{tr(\hat{k}_t)} \leq \mathfrak{R}_t(\hat{\mathcal{H}}_r) \leq \frac{2r}{t} \sqrt{tr(\hat{k}_t)}, \tag{3.21}$$

where

$$\hat{k}_t(z,x) = \sum_{j=1}^{m} \frac{w_{t_j}^2}{\alpha_{t_j}} k_{t_j}(z,x) - \gamma \tilde{k}_t^\top \tilde{A}_t \tilde{G}_t^{-1} (I + \gamma D \tilde{G}_t^{-1} \mathcal{K}_t)^{-1} D \tilde{G}_t^{-1} \tilde{A}_t \tilde{k}_t(z). \tag{3.22}$$

The Rademacher complexity can be plugged into the generalization bound. As a particular example, we present a bound from the theory of Rademacher complexity. The generalization error bounds can therefore be approximated by the following theorem.

Theorem 3.3. *Generalization bounds: let L be one of the surrogate convex losses with Lipschitz constant $\Lambda = 1$. Given the views subset $\mathcal{S}$ and contextual information x_t, for any $\delta \in (0,1)$, with probability at least $1-\delta$ over the sample of historical data $(x_1, y_1), \cdots, (x_{|T_\mathcal{S}(t-1)|}, y_{|T_\mathcal{S}(t-1)|})$ drawn i.i.d. from $\mathcal{D}$, we have for any predictor $\varphi_t \in \hat{\mathcal{H}}$ that*

$$\mathbb{E}_\mathcal{D} L(\varphi_t(x), y) \leq \hat{R}_t(\varphi_t) + 2\mathfrak{R}_t(\hat{\mathcal{H}}_r) + \frac{1}{\sqrt{|T_\mathcal{S}(t-1)|}} (2 + 3\sqrt{\ln(2/\delta)/2}), \tag{3.23}$$

where $\hat{R}_t(\varphi_t) = \frac{1}{|T_\mathcal{S}(t-1)|} \sum_{T_\mathcal{S}(t-1)} L(\varphi_t(x_i), y_i)$.

Theorem 3.3 shows that different view subsets have different generalization performances. Thus we let the index correspond to the empirical loss $\hat{R}$ penalized by some quantity ξ, which can be defined as

$$\eta(\mathcal{S}, x_t) = \hat{R}_t(\varphi_t) + \xi_t(\mathcal{S}, x_t), \tag{3.24}$$

where $\mathcal{S}$ indicates the view subset.

Then we define the penalty using the upper bound of generalization based on Rademacher complexity given in Theorem 3.2 as

$$\xi_t(\mathcal{S}, x_t) = \frac{2r}{|T_{\mathcal{S}}(t)|}\sqrt{tr(\hat{k}_t)} + \frac{1}{\sqrt{|T_{\mathcal{S}}(t)|}}(2 + 3\sqrt{\ln(2/\delta)/2}). \tag{3.25}$$

3.5 Regret analysis of CoRLSUB

Under the assumptions of view compatibility, we show that the selected view subset can be indexed by the confidence of generalization of a variant of CoRLS. Then we provide a regret analysis for CoRLSUB in Theorem 3.4.

In this section, we prove Theorem 3.2 and Theorem 3.3 and provide an upper bound for the multi-view simple regret.

The proof of Theorem 3.2 is as follows:

Proof. According to Theorem 3.1, we have found a reproducing kernel $\hat{k}_t(.,.)$ for the Hilbert space $\hat{\mathcal{H}}_t$. The optimization problem is

$$\varphi^* = \underset{\varphi \in \hat{\mathcal{H}}_t}{\arg\min}\, \hat{R}(\varphi) + \|\varphi\|^2_{\hat{\mathcal{H}}_t}. \tag{3.26}$$

According to [115] we know that

$$\|\varphi^*\|^2_{\hat{\mathcal{H}}_t} \leq r^2 := L(0). \tag{3.27}$$

We can restrict our search for h^* to the norm ball in $\hat{\mathcal{H}}_t$ of radius $\sqrt{L(0)}$. According to the results of [115], the Rademacher complexity is a standard result by using the new reproducing kernel $\hat{k}_t$ of RKHS $\hat{\mathcal{H}}_t$. □

Let L be one of the surrogate convex losses and $L \in [0,1]$, such as hinge loss with Lipschitz Constant 1, logistic loss with Lipschitz Constant $\sup_j |\frac{-e^{-j}}{1+e^{-j}}| = 1$ or exponential loss with Lipschitz Constant $L \leq B = 1$. We first show the condition on the loss function L with Lipschitz constant $\Lambda = 1$:

Condition 3.1: The loss function $L(.,.)$ is Lipschitz in its first argument, i.e. there exists a constant Λ such that $\forall y, \hat{y_1}, \hat{y_2} : \|L(\hat{y}_1, y) - L(\hat{y}_1, y)\| \leq \Lambda \|\hat{y}_1 - \hat{y}_2\|$.

With this condition, we can prove Theorem 3.3.

Proof. Given views subset $\mathcal{S}$, we set the loss function class as $\mathcal{Q} = \{(\varphi_s(x), y) : \varphi_s \in \mathcal{J}\}$. According to Theorem 3.1 of [115], given any labeled data and $\varphi_s \in \mathcal{J}$, for any $\delta \in (0,1)$ with the probability at least $1-\delta$ we have

$$E_{\mathcal{D}} L(\varphi_s(x), y) \leq \frac{1}{|T_{\mathcal{S}}(t-1)|} \sum_{T_{\mathcal{S}}(t-1)} L(\varphi_s(x_i), y_i) + \mathfrak{R}_t(\mathcal{Q}) + 3\sqrt{\frac{\ln(2/\delta)}{2|T_{\mathcal{S}}(t-1)|}}.$$

We let $q_y = L(0, y)$ and $p_y(\hat{y}) = L(\hat{y}, y) - L(0, y)$. Now we have $L(\varphi(x), y) = q_y + p_y * (\varphi(x))$ and $\mathcal{Q} = q_y + p_y \circ \mathcal{J}$. Since $|q_y| \leq 1$ for all y, according to a property of Rademacher complexity [19] we have

$$\mathfrak{R}_t(\mathcal{Q}) \leq \mathfrak{R}_t(q_y \circ \hat{\mathcal{H}}_r) + \frac{2}{\sqrt{|T_{\mathcal{S}}(t-1)|}}. \tag{3.28}$$

Based on our loss condition there exists Lipschitz constant Λ and $p_y(0) = 0$. Then according to the Ledoux-Talagrand contraction inequality we have

$$\mathfrak{R}_t(p_y \circ \hat{\mathcal{H}}_r) \leq 2\Lambda \mathfrak{R}_t(\hat{\mathcal{H}}_r). \tag{3.29}$$

Thus we conclude the generalization bounds

$$\begin{aligned} &\mathbb{E}_{\mathcal{D}} L(\varphi_s(x), y) \\ &\leq \frac{1}{|T_{\mathcal{S}}(t-1)|} \sum_{T_{\mathcal{S}}(t-1)} L(\varphi_s(x_i), y_i) + 2\mathfrak{R}_t(\hat{\mathcal{H}}_r) + \frac{1}{\sqrt{|T_{\mathcal{S}}(t-1)|}} (2 + 3\sqrt{\ln(2/\delta)/2}). \end{aligned}$$

□

Theorem 3.4. *Multi-view bandits has an expected multi-view simple regret for any $\delta \in (0,1)$ that is with at least $1-\delta$ probability,*

$$\begin{aligned} &\mathbb{E}[L(\varphi_{\mathcal{S}_t}(x), y) - L(\varphi_{\mathcal{S}_*}(x), y)] \\ &\leq \sup_{\mathcal{S}} \left\{ \frac{4r}{|T_{\mathcal{S}}(t)|} \sqrt{tr(\hat{k}_{\mathcal{S}})} + \frac{2}{\sqrt{|T_{\mathcal{S}}(t)|}} (2 + 3\sqrt{\frac{ln(2/\delta)}{2}}) \right\}. \end{aligned} \tag{3.30}$$

Proof. We consider the upper bound of the simple regret in terms of the selected views. Let $\mathcal{S}_t$ denote a random set of all m-view subsets. We have

$$\begin{aligned}
&\mathbb{E}\left[L(\varphi_{\mathcal{S}_t}(x),y) - L(\varphi_{\mathcal{S}_*}(x),y)\right] \\
&= \mathbb{E}L(\varphi_{\mathcal{S}_t}(x),y) - \hat{R}_t(\varphi_{\mathcal{S}_t}) + \hat{R}_t(\varphi_{\mathcal{S}_t}) - \hat{R}_t(\varphi_{\mathcal{S}_*}) + \hat{R}_t(\varphi_{\mathcal{S}_*}) - \mathbb{E}L(\varphi_{\mathcal{S}_*}(x),y) \\
&\leq \left|\mathbb{E}L(\varphi_{\mathcal{S}_t}(x),y) - \hat{R}_t(\varphi_{\mathcal{S}_t})\right| + \left|\mathbb{E}L(\varphi_{\mathcal{S}_*}(x),y) - \hat{R}_t(\varphi_{\mathcal{S}_*})\right| \\
&\leq 2\sup_{\mathcal{S}_t}\left|\mathbb{E}L(\varphi_{\mathcal{S}_t}(x),y) - \hat{R}_t(\varphi_{\mathcal{S}_t})\right|.
\end{aligned}$$

The first inequality is derived from $\forall \mathcal{S}, \hat{R}_t(\varphi_{\mathcal{S}}) \leq \hat{R}_t(\varphi_{\mathcal{S}_*})$. According to Theorem 3.2, at time t, given $\delta \in (0,1)$ that is with probability at least $1-\delta$, we have

$$\begin{aligned}
&\mathbb{E}[L(\varphi_{\mathcal{S}_t}(x),y) - L(\varphi_{\mathcal{S}_*}(x),y)] \\
&\leq \sup_{\mathcal{S}}\left\{\frac{4r}{|T_{\mathcal{S}}(t)|}\sqrt{tr(\hat{k}_{\mathcal{S}})} + \frac{2}{\sqrt{|T_{\mathcal{S}}(t)|}}(2+3\sqrt{\frac{ln(2/\delta)}{2}})\right\}.
\end{aligned}$$

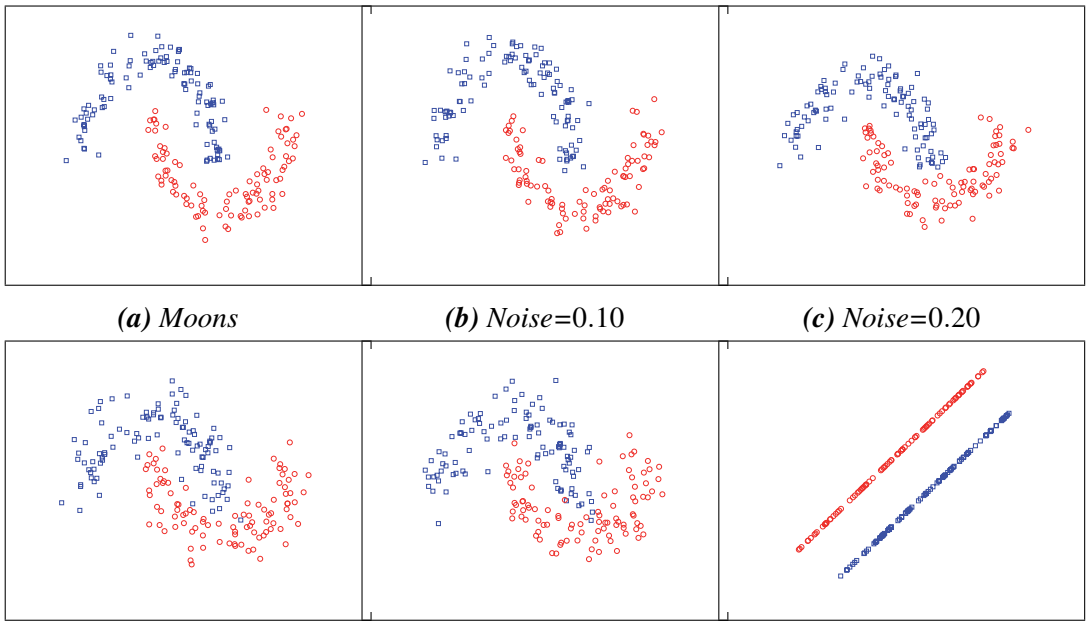
□

The problem dependent simple regret scales as $\mathcal{O}(1/\sqrt{t})$. As $t \to 0$, the simple regret can achieve the convergence. We examine $\sqrt{tr(\hat{k})}$ or $tr(\hat{k})$. The second term of $tr(\hat{k})$ depends on the disagreement norm regularization parameter γ. If other parameters are fixed, when γ increases to $+\infty$, $tr(\hat{k}) \to 0$ and the simple regret upper bound also decreases. Thus, when the consistency of the model based on the selected views becomes more constrained, the simple regret can be smaller and it is coherent with the intuition of multi-view learning.

3.6 Experiments

We empirically evaluate the effectiveness of the proposed multi-view bandit for view subset selection. Considering the lack of baseline algorithms which have a multi-view bandit setting, we define two naive baselines: (1) an all-view strategy, in which the forecaster employs all the available views for prediction; and (2) a random strategy, in which the forecaster randomly selects m views for prediction. The proposed CoRLSUB is developed based upon the state-of-the-art algorithm CoRLS [115]. For fair comparison, we employ the views chosen by either (1) or (2) for CoRLS. Briefly, the two baselines are termed CoRLS-All and CoRLS-Random.

Evaluations are conducted on a synthetic dataset - a toy example, and an example for the automatic navigation control of a robot [121]. Then we exploit five publicly available



(a) *Moons* ***(b)*** *Noise*=0.10 ***(c)*** *Noise*=0.20

(d) *Noise*=0.50 ***(e)*** *Noise*=0.60 ***(f)*** *Lines*

Fig. 3.3 *A toy dataset with different views.*

real-world datasets: G50C [35], PCMAC [122], Caltech 256 [68], VOC 2006 [47] and ImageNet [44].

3.6.1 Toy example

Several experiments were performed on the toy example, which is a synthetic dataset represented by $n = 5$ views and sampled from two classes. We extended the two-moons-two-lines dataset to six views with noise, as shown in Figure 3.3. As well as the original two-moons-two-lines views, we created a further four views by adding the standard deviation of Gaussian noise (level 0.10, 0.20, 0.50 and 0.60) to the two-moons view. Similar to two-moons-two-lines, we randomly associated the points on one moon with points on one line to construct the class conditional view. A few examples were labeled in each class, and the other examples sequentially arrived for prediction with their different view information. After prediction, the environment gave either 1 when the prediction was correct or 0 when the prediction was wrong as payoff to the forecaster. A Gaussian kernel was chosen for the two moons view and a linear kernel for the two lines view. The squared loss was chosen for $L(.,.)$. In the toy dataset, we let $m = 3$ and used the majority vote for the final prediction, that is $w_i = 1/m$. The experiments were repeated 100 times and the average accuracy of the various methods at each time step is shown in Figure 3.4. Our method outperforms the others, such as CoRLS-All and CoRLS-Random, which is logical since the noisy views can harm the prediction. Our view subset selection relies on the generalization performance of

the selected view subset, and all views strategy and random strategy do not try to isolate the noisy views.

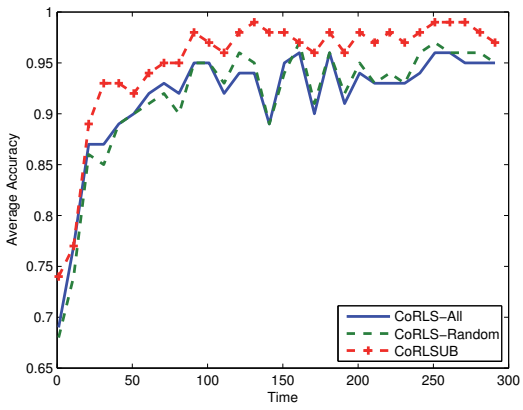


Fig. 3.4 *Performance comparison on the toy example.*

3.6.2 The robot navigation example

We designed a simple robot motion experiment to evaluate our algorithm. The robot received the images sequentially through multiple cameras. Five cameras at different angles provides five shots in each round. The shots have a 50° overlap. At each time step, the robot selects m shots and then makes a decision based on the selected shots. The decision in our problem is simply defined as turning right or turning left. Following prediction, the environment gives feedback as to whether the robot can turn right or left. The feedback is defined in a payoff way, such that if the robot takes the right action the payoff is 1, and if the robot takes the wrong action the payoff is 0. Images are represented by a histogram of oriented gradients calculated in each cell that is decomposed by the HOG. We let $m = 3$ for view subset selection and use the majority vote for the final prediction. We use Gaussian kernel for all views and choose the squared loss for $L(.,.)$. We report the accuracy of different methods at each time step and the results are shown in Figure 3.5. Our method performs better than the others. This indicates that the active selection strategy of our algorithm attempts to select views that will achieve better performance than other strategies. The view strategies of other methods were less effective than ours because of the existence of noisy and redundant views. The random strategy performs the worst because it uses useless shots which are unable to provide information about movement, and may also use noisy shots while ignoring important shots. We show two examples of view selection in Figure 3.6. The selected views by random strategy do not provide enough information for prediction, thus the prediction

may be wrong. However, the multi-view bandit algorithm CoLRSUB actively selects the view subset based on the feedback of previous selections. The selected views contained the necessary information to improve prediction performance.

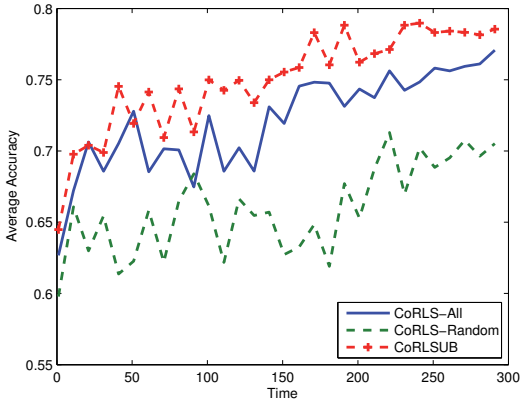


Fig. 3.5 *Performance comparison on robot motion example.*

3.6.3 Public datasets

The G50C dataset was generated from two unit covariance normal distributions with equal probabilities and contains 50 features. We split all features into 5 disjoint feature subsets (each feature subset corresponds to a context vector) as 5 views. We let $m = 3$ for selection and used the majority vote for the final prediction. We used Gaussian kernel for all views and chose the squared loss for $L(.,.)$.

The PCMAC dataset was collected for a text binary classification problem drawn from the 20 newsgroups dataset; it contains 7511 features. We split all features into 11 disjoint feature subsets (each feature subset corresponds to a context vector) as 11 views. We let $m = 5$ for selection and used the majority vote for the final prediction. We used Gaussian kernel for all views and chose the squared loss for $L(.,.)$.

In the Caltech 256 database, there are 256 object categories and 30607 images. The images from each category were downloaded from both Google and PicSearch using scripts. The number of images in most categories is more than 100. Of the 256 categories, we used 10 categories: bear, camel, comet, dog, elephant, fire-truck, goose, hibiscus, kayak, and snake.

In the VOC 2006 database of 5301 images, there are 10 categories: bicycle, bus, car, cat, cow, dog, horse, motorbike, person, and sheep. The images were collected from Microsoft Research Cambridge and “Flickr”. We used all 10 categories.

ImageNet is an image database from internet the Internet and organized according to the WordNet hierarchy. There are more than 100000 synsets and on average 1000 images to

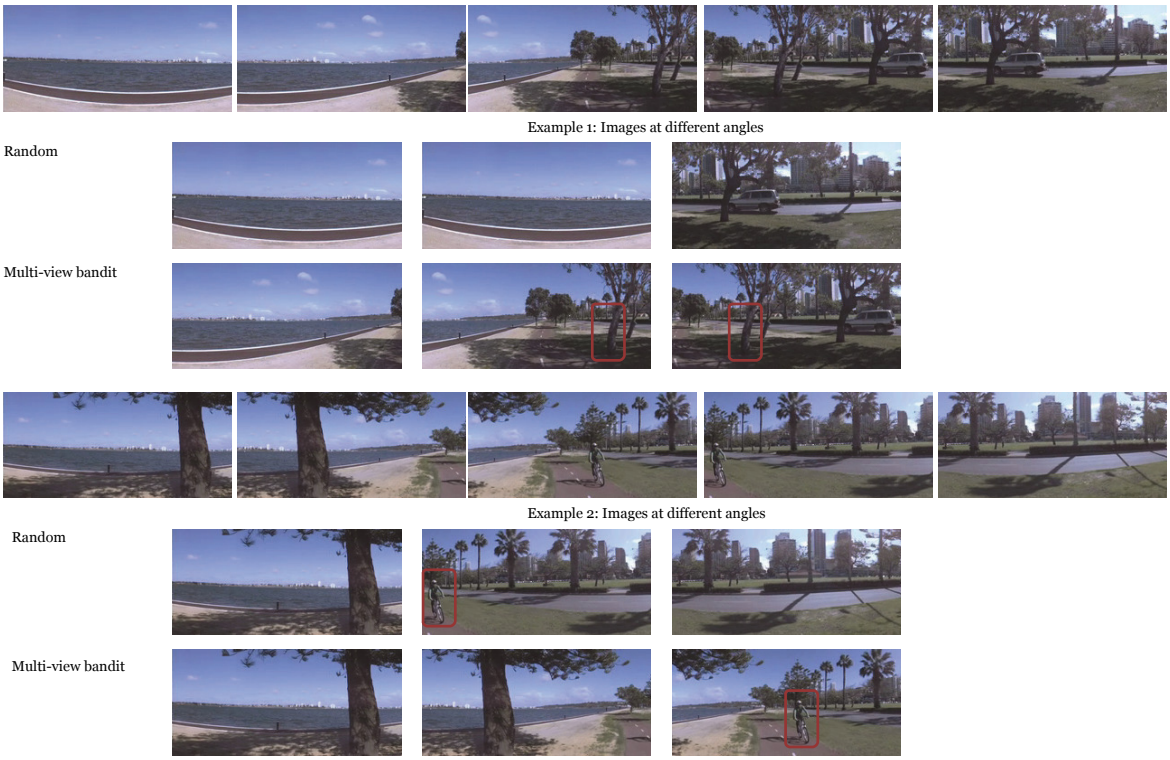
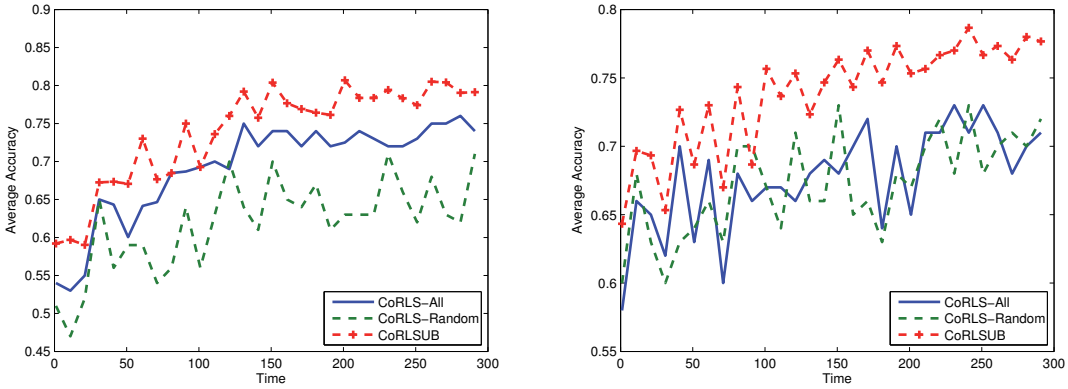


Fig. 3.6 *Example views selected by different strategies in the automatic navigation control of a robot.*

illustrate each synset in ImageNet. There are more than a thousand categories in ImageNet, of which and we used 10 categories: airliner, ambulance, bicycle, boat, calabash, cock, curassow, draft animal, jeep, and taxi.

To simulate the multi-view setting for image datasets (CalTech256, VOC2007 and ImageNet), we chose to represent each image by color histogram, histogram of oriented gradients, textons, Fisher coding and VLAD. The color histogram indicates the distribution of colors in an image. The histogram of oriented gradients calculates a histogram of oriented gradients in each cell which is decomposed by the HOG. We use a variant whereby the feature has 31 dimensions. Textons are the basis of texture gradation and are used to classify and manipulate textures. To summarize a number of local feature descriptors in a vectorial statistic, e.g. SIFT, we use two encodings, Fisher encoding and VLAD. The Fisher coding uses Gaussian mixture model (GMM) to construct a visual word dictionary. The VLAD encoding uses k-means instead of GMM to generate the feature vocabulary. For each dataset, we used these five different feature sets as different views to represent the image. We let $m = 3$ for selection and used the majority vote for the final prediction. We used Gaussian kernel for all views and chose the squared loss for $L(.,.)$.



(a) *G50C* ***(b)*** *PCMAC*

Fig. 3.7 *Performance comparison on (a) G50C and (b) PCMAC.*

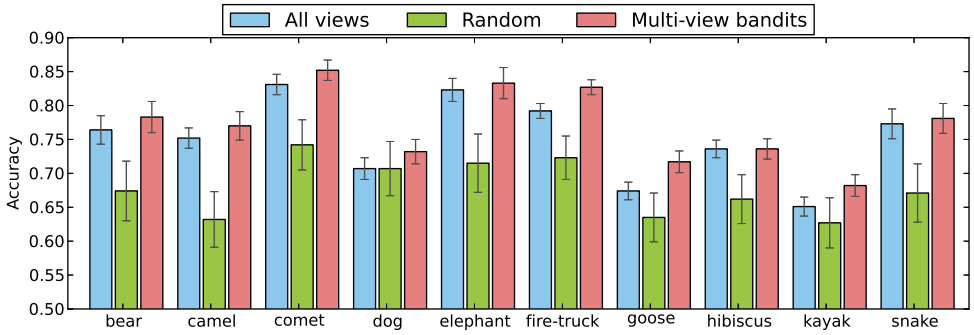


Fig. 3.8 *A comparison of the multi-view bandit strategy with other strategies on Caltech 256.*

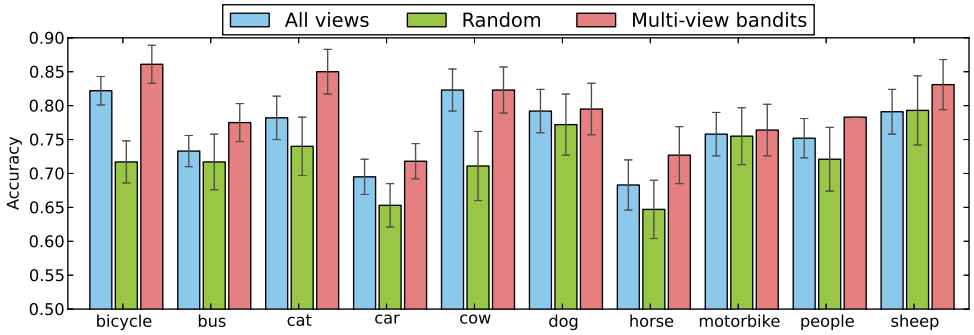


Fig. 3.9 *A comparison of the multi-views bandit strategy with other strategies on VOC 2006.*

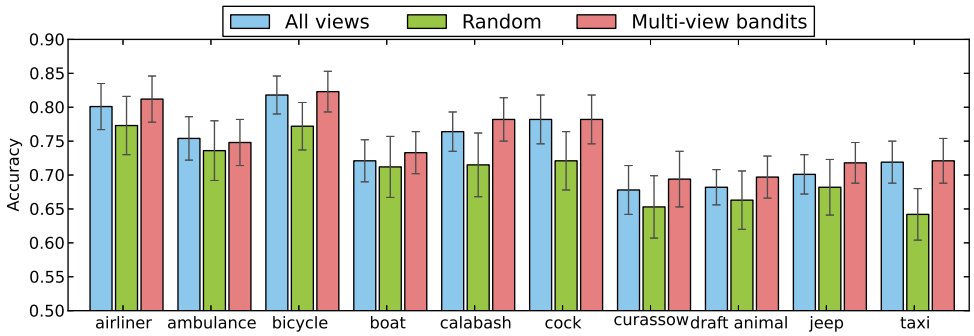


Fig. 3.10 *A comparison of the multi-view bandit strategy with other strategies on ImageNet.*

3.6.4 Stream-based multi-view learning

We simulated the stream-based multi-view learning setting on these five datasets. In the stream-based multi-view learning setting, the example is delivered one by one with multiple views information.

For the G50C, the environment randomly draws an example with 5 views at each round. We assume that the algorithm does not know the label of the example before it makes the prediction. The algorithm selects three views to make the prediction according to its strategy. After making the prediction, the algorithm will suffer a loss incurred by the difference between the prediction and the label provided by the environment and will receive the new example with the label provided by the environment. We record the accuracy of the prediction in each round. This simulation is repeated 100 times and the average results are reported.

Similar to the G50C, there is an example with 11 views for the PCMAC at each round, and the algorithm selects five views to make the prediction. The results of the average accuracy for the different algorithms are shown in Figure 3.7. The empirical results show that multi-view bandit algorithm performs better than CoRLS-All and CoRLS-Random. The results show that active selection works better than the passive selection in the stream-based multiple views setting.

For Caltech 256, we selected 10 categories and constructed 10 classification problems. For each category, we collected 100 positive examples and 300 negative examples. At each round the environment randomly draws an image which is represented using five views encoded by color histogram, edge direction histogram, textons, Fisher coding and VLAD. The algorithm selects three views from all views according to its strategy and then makes the prediction. The algorithm does not know the label prior to making the prediction. After making the prediction, the algorithm suffers a loss incurred by the difference between the prediction and label provided by the environment, and receives the new image with the label provided by the environment. We collected the average accuracy of the total number of predictions. We repeated the simulation 100 times and reported the average results.

As with Caltech 256, we conducted the same experiments on the VOC 2006 and ImageNet. We used 10 categories from each database and constructed classification problems. For each round, the environment draws an image represented by five views. The algorithm selects three of the five views according to its strategy and then makes the prediction. The algorithm may suffer a loss and receive the new image with its label provided by environment. We collected the average accuracy of the total number of predictions. We repeated the simulation 100 times and reported the average results.

In Figures 3.8, 3.9, 3.10, we show that our algorithm outperforms CoRLS-All and CoRLS-Random for the selected datasets from Caltech 256, VOC 2006 and ImageNet respectively. It suggests that the effectiveness of views is different. Noisy or redundant views may exist, and actively selecting the view subset can improve performance. CoLRSUB is a selective strategy according to the loss generalization bound and the results verify its effectiveness. The Multi-view bandit algorithm and All views strategy both offer more stable performance than Random. Comparing with Random strategy, Multi-view bandit algorithm CoRLSUB selects view subset according to historical records and current example and thus it maintains a better performance.

3.7 Proofs

In this section, we present the detailed proofs of the theoretical results.

3.7.1 Proof of Lemma 3.1

We prove $\mathcal{F}$ is complete. Let $f_1, f_2, \cdots$ be any Cauchy sequence in $\mathcal{F}$. Then for any $a, b \in 1, 2, \cdots$, we have

$$\|f_a - f_b\|_{\mathcal{F}}^2 = \sum_{i=1}^{m} \alpha_i \|f_a^i - f_b^i\|_{\mathcal{H}^i}^2 + \gamma(\tilde{f}_a - \tilde{f}_b)^T D(\tilde{f}_a - \tilde{f}_b). \tag{3.31}$$

Both the left hand side and right hand side are nonnegative. Recall that $\mathcal{H}^i$s are RKHS's, they are complete. Thus for each $i = 1, \cdots, m$, $\lim_{a\to\infty} f_a^i = f^i$, for some $f^i \in \mathcal{H}^i$. Let $f_a \to f$, we have

$$\|f_a - f\|_{\mathcal{F}}^2 = \sum_{i=1}^{m} \alpha_i \|f_a^i - f^i\|_{\mathcal{H}^i}^2 + \gamma(\tilde{f}_a - \tilde{f})^T D(\tilde{f}_a - \tilde{f}). \tag{3.32}$$

We define the largest eigenvalue of D is $\lambda_1(D)$, then by the variational characterization of eigenvalues we have

$$\begin{aligned} 0 &\leq \gamma(\tilde{f}_a - \tilde{f})^T D(\tilde{f}_a - \tilde{f}) \\ &\leq \gamma\lambda_1(D)(\tilde{f}_a - \tilde{f})^T(\tilde{f}_a - \tilde{f}) \\ &= \gamma\lambda_1(D)\sum_{i=1}^{m}(f_a^i(x) - f^i(x))^2. \end{aligned} \tag{3.33}$$

Since $\mathcal{H}^i$'s are RKHS's, their evaluation functions are continuous. Thus given $x \in \mathcal{X}$ if f_a^i and f^i are close in the norm $\|.\|_{\mathcal{H}^i}$, then $\|f_a(x)^i - f^i(x)\|$ is small. Then we can find large N that $a > N$ implies $\|f_a^i(x) - f^i(x)\| \leq \varepsilon$ for all $i = 1, \cdots, m$. Thus we conclude $\gamma(\tilde{f}_a - \tilde{f}) \to 0$ as $a \to \infty$.

It is straightforward that the summands in $\sum_{i=1}^m \alpha_i \|f_a^i - f^i\|_{\mathcal{H}^i}$ each go to 0 as $f_a^i \to f^i$ in $\mathcal{H}^i$.

Thus if the left hand side goes to 0, then all terms on the right hand side go to 0.

3.7.2 Proof of Lemma 3.2

Following the approach of [21] we show how to push the Hilbert space structure from $\mathcal{F}$ onto $\hat{\mathcal{H}}$.

Denote the nullspace of φ by $\mathcal{N} := \varphi^{-1}(0)$. $\mathcal{N}$ is a closed subspace of $\mathcal{F}$, and then its orthogonal complement $\mathcal{N}^\perp$ is also a closed subspace. Let $\psi : \mathcal{N}^\perp \to \hat{\mathcal{H}}$ be the restriction of φ to $\mathcal{N}^\perp$. We define an inner product on $\mathcal{N}^\perp$ as

$$\langle f, g\rangle_{\hat{\mathcal{H}}} = \left\langle \psi^{-1}(f), \psi^{-1}(g)\right\rangle_{\mathcal{F}}. \tag{3.34}$$

The $(\hat{\mathcal{H}}, \langle .,.\rangle_{\hat{\mathcal{H}}})$ is a Hilbert space isomorphic to $\mathcal{N}^\perp$.

Now we show the norm of $h \in \hat{\mathcal{H}}$ define as

$$\begin{aligned} \|h\|_{\hat{\mathcal{H}}} &= \|\psi^{-1}(h)\|_{\mathcal{F}}^2 \\ &= \min_{n\in\mathcal{N}}(\|\psi^{-1}(h)\|_{\mathcal{F}}^2 + \|n\|_{\mathcal{F}}^2) \\ &= \min_{n\in\mathcal{N}}(\|\psi^{-1}(h) + n\|_{\mathcal{F}}^2) \\ &= \min_{f\in\varphi^{-1}(h)}(\|f\|_{\mathcal{F}}^2) \\ &= \min_{(f^1,\cdots,f^m)\in\varphi^{-1}(h)} \sum_{i=1}^{m} \alpha_i \|f^i\|_{\mathcal{H}^i} + \gamma \tilde{f}^T D \tilde{f}. \end{aligned} \tag{3.35}$$

3.7.3 Proof of Theorem 3.1

We now construct a reproducing kernel $\hat{k}(.,.)$ for the Hilbert space $\hat{\mathcal{H}}$. That is, for each $z \in \mathcal{X}$, a tuple $g_z = (g_z^1, \cdots, g_z^m) \in \mathcal{F}$ such that the function $\varphi(g_z) \in \hat{\mathcal{H}}$ has the reproducing property:

$$\langle h, \varphi(g_z) \rangle_{\hat{\mathcal{H}}} = h(z). \tag{3.36}$$

If this property is suited for all $x \in \mathcal{X}$, then by definition $\hat{k}$ is a reproducing kernel for $\hat{\mathcal{H}}$.

We first define the maps u and v and have

$$v^{-1}(\hat{k}(z,.)) = g_z + n_z, \tag{3.37}$$

for some $n_z \in \mathcal{N}$.

Fix any $h \in \hat{\mathcal{H}}$, and let $f = (f^1, \cdots, f^m) = v^{-1}(h)$. Then

$$\begin{aligned}
&\left\langle h, \hat{k}(z,.) \right\rangle_{\hat{\mathcal{H}}} = \left\langle v^{-1}(h), v^{-1}(\hat{k}(z,.)) \right\rangle_{\mathcal{F}} \\
&= \left\langle v^{-1}(h), g_z + n_z \right\rangle_{\mathcal{F}} \\
&= \left\langle (f^1, \cdots, f^m), g_z \right\rangle_{\mathcal{F}} \\
&= \sum_{j=1}^{m} \alpha_j \left\langle f^j, g_z^j \right\rangle_{\mathcal{H}^j} + \gamma \tilde{f}^T M \tilde{g}_z,
\end{aligned}$$

where $\tilde{g}_z = (g_z^1(x_1), \cdots, g_z^1(x_t), \cdots, g_z^m(x_t), \cdots, g_z^m(x_t))$.

We define $\tilde{\beta}_x = (\beta_1^1(x), \cdots, \beta_n^m(x))^\top$ and $\tilde{h}_z = \tilde{G}^{-1}(\tilde{A}\tilde{k}_z + \gamma \mathcal{K}\tilde{\beta}_z)$. Then we have

$$\begin{aligned}
&\left\langle h, \hat{k}(z,.) \right\rangle_{\hat{\mathcal{H}}} = \sum_{j=1}^{m} \alpha_j f^j(z) + \gamma \sum_{j=1}^{m} \sum_{i=1}^{n} \beta_i^j(z) f^j(x_i) + \gamma \tilde{f}^\top D \tilde{g}_z \\
&= h(z) + \gamma \tilde{f}^\top \tilde{\beta}_z + \gamma \tilde{f}^\top D \tilde{G}^{-1}(\tilde{A}\tilde{k}_z + \gamma \mathcal{K}\tilde{\beta}_z) \\
&= h(z) + \gamma \tilde{f}^T [\tilde{\beta}_z + D \tilde{G}^{-1}(\tilde{A}\tilde{k}_z + \gamma \mathcal{K}\tilde{\beta}_z)] \\
&= h(z) + \gamma \tilde{f}^T [(I + \gamma D \tilde{G}^{-1}\mathcal{K})\tilde{\beta}_z + D \tilde{G}^{-1}\tilde{A}\tilde{k}_z].
\end{aligned}$$

We let $\tilde{\beta}_z = -(I + \gamma D\tilde{G}^{-1}\mathcal{K})^{-1} D\tilde{G}^{-1}\tilde{A}\tilde{k}_z$ and it can be shown that $I + \gamma D\tilde{G}^{-1}\mathcal{K}$ is full rank. Now we conclude that we have the reproducing property for z. Then we can have

$$\begin{aligned}
&\hat{k}(z,x) = \varphi((g_z^1, \cdots, g_z^m)) \\
&= \sum_{j=1}^{m} \frac{w_j}{\alpha_j}[a_j k^j(z,x) + \gamma \sum_{i=1}^{n} \beta_i^j(z) k^j(x_i,x)] \\
&= \sum_{j=1}^{m} \frac{w_j^2}{\alpha_j} k^j(z,x) + \gamma \frac{w_j}{\alpha_j} \sum_{i=1}^{n} \beta_i^j(z) k^j(x_i,x) \\
&= \sum_{j=1}^{m} \frac{w_j^2}{\alpha_j} k^j(z,x) + \gamma \tilde{k}_x^T \tilde{A}\tilde{G}^{-1}\tilde{\beta}_z \\
&= \sum_{j=1}^{m} \frac{w_j^2}{\alpha_j} k^j(z,x) - \gamma \tilde{k}_x^T \tilde{A}\tilde{G}^{-1}(I + \gamma D\tilde{G}^{-1}\mathcal{K}) D\tilde{G}^{-1}\tilde{A}\tilde{k}_z.
\end{aligned}$$

We conclude that $\hat{\mathcal{H}}$ is an RKHS with kernel $\hat{k}$.

3.7.4 Proof of Proposition 3.1

We reduce the Knapsack Problem to a special multi-view bandit problem where the view subsets have different payoffs, and discrete priors with non-zero probability at feedback 0 and 1. It shows that maximizing the profit in the Knapsack instance is equivalent to maximizing the probability of finding a perfect view subset, which is shown to be equivalent to minimizing the regret. The reduction reveals the packing aspect of the multi-view bandit. To find an optimal solution in the Knapsack Problem is NP-hard. Thus we can see that in multi-view bandit there is no known polynomial algorithm which can tell whether a given solution is the optimal solution.

3.8 Conclusion

In this work, we have defined and examined the problem of view subset selection for multi-view learning in the sequential environment. Unlike a traditional bandit problem, the forecaster aims to select a subset $m(m \geq 2)$ from n views after observing the view information generated from different views to make the final prediction and then obtain payoff. We formalized this problem as a multi-view bandit subset selection in the steam-based multi-view setting. Few previous studies have examined multiple view selection in multi-view learning, and there have been no studies on subset selection in the multi-view bandit setting. We first introduced the multi-view bandit and its corresponding multi-view

simple regret. Incorporating the Rademacher complexity, which plugs into the generalization bound, we proposed the multi-view bandit algorithm by considering this generalization bound, which can theoretically bound the multi-view simple regret. We provided an upper bound of expected multi-view simple regret using the Rademacher complexity and proved an upper bound of order $1/\sqrt{t}$. We also showed that the consistency of predictions based on different views can improve the simple regret bound. Our experiments verified that our view subset selection method is efficient.

Chapter 4

Active multi-task learning via bandits

In multi-task learning, the multiple related tasks allow each one to benefit from the learning of the others, and labeling instances for one task can also affect the other tasks especially when the task has a small number of labeled data. Thus labeling effective instances across different learning tasks is important for improving the generalization error of all tasks. In this work, we propose a new active multi-task learning paradigm, which selectively samples effective instances for multi-task learning. Inspired by the multi-armed bandits, which can balance the trade-off between the exploration and exploitation, we introduce a new active learning strategy and cast the selection procedure as a bandit framework. We consider both the risk of multi-task learner and the corresponding confidence bounds and our selection tries to balance this trade-off. Our proposed method is a sequential algorithm, which at each round maintains a sampling distribution on the pool of data, queries the label for an instance according to this distribution and updates the distribution based on the newly trained multi-task learner. We provide an implementation of our algorithm based on a popular multi-task learning algorithm that is trace-norm regularization method. Theoretical guarantees are developed by exploiting the Rademacher complexities. Comprehensive experiments show the effectiveness and efficiency of the proposed approach.

4.1 Introduction

Multi-task learning is to address the problem that the data representations are common across multiple related supervised learning tasks. The goal of multi-task learning is to improve the performance of learner for multiple tasks jointly. This problem is interesting in many research areas. For example, for predicting the therapeutic success of a given combination of drugs for a given strain of the HIV-1 [23], they designed a multi-task learning model to handle arbitrarily different data distributions for different tasks. Another example is to detect

different objects in images [130]. Each specific object detecting can be considered as a single supervised learning task. They proposed a method to learn the shared features from the pixel representation of images.

For all of these multi-task learning algorithms, we often first collect a significant quantity of data that is randomly sampled from the underling population distribution and then induce a learner or model. However, the most time-consuming and costly task is usually the collecting of data. Thus, it is particularly valuable to determine ways in which we can make use of these resources as much as possible. Furthermore, in multi-task learning, the simultaneously multiple related tasks allow each one to benefit from the learning of all of the others, and labeling instances for one task can also affect the other tasks especially when the task has a small number of labeled data. Thus our work focuses on how to guide the sampling process for multi-task learning.

However, currently, there is seldom work to address selectively labeling instances in multi-task learning. We first show some work about active learning for multi-task learning problem. Harpale and Yang [72] designed an active learning strategy for multi-task adaptive filtering approaches. Reichart et al. [113] considered the active learning algorithm for multi-task problem in linguistic annotations and combined rankings/scores from all single-task active learnings. Acharya et al. [3] used a topic-modeling framework for the dimension reduction to adapt the expected error reduction strategy. However, these methods all addressed specific problems. We try to develop a general framework for active multi-task learning by considering the risk of multi-task learner and the confidence bounds on the risk.

In this work, we propose a new active multi-task learning algorithm via bandits. Our work is built on a bandit framework for actively selecting instances from a pool of data. The multi-armed bandit is a well-known model on sequential decision problem. It tries to address the fundamental trade-off between exploration and exploitation in sequential decision making problems [31]. Inspired by these, we define the active learning process as a sequential instance decision problem. Our proposed strategy addresses the trade-off between the risk and the confidence bounds and tries to select instances to filter out good hypothesis. We maintain a designed distribution on the pool of data. This distribution will be updated when a new multi-task learner is trained with new instances and involves the risk and the confidence. We sample the data from the pool based on the distribution and try to select an instance to improve the performance of learner. The hypothesis can be considered as an arm. As different instances acquired, we can select different hypotheses for our optimization function and finally filter the good candidates which are close to the ideal hypothesis.

We provide an implementation of our approach based on multi-task learning with trace-norm regularization method. Trace-norm regularization method is one of the popular algo-

rithms for multi-task problems [106]. It is to solve the problem of minimizing the rank of a matrix variable subject to certain constraints. In problems where multiple related tasks are learned simultaneously, the models for different tasks can be constrained to share certain information. Recently this constraint has been studied using the trace-norm regularization method [2, 8, 103]. There is very few work to address the active learning strategy based on multi-task learning with trace-norm regularization method. Our implementation uses this regularization method as an example and is the first work to use the risk and the confidence bounds for active learning.

The main contributions of this work are three-fold. Firstly, we proposed a new active learning algorithm for multi-task learning problem, named active multi-task learning via bandits, which is a general active learning framework. Secondly, we provided an implementation of our algorithm based on trace-norm regularization method. Thirdly, we verified our algorithm's effectiveness and efficiency by comparing its evaluation results to such passive learning and other active learning strategies for multi-task learning empirically.

The rest of this work is organized as follows. In Section 4.2, we introduce the related work. In Section 4.3 we formalize our problem definition. We present our algorithm in Section 4.4. In Section 4.5, we give some theoretical results. The experiments on synthetic and real datasets are demonstrated in Section 4.6. Finally, we make the conclusion in Section 4.7.

4.2 Related work

Active learning is very popular and studied in many research areas [119]. According to the query strategies used, there are three categories of active learning techniques: (1) uncertainty sampling [90, 129], which focuses on selecting the instance that the classifier is most uncertain about; (2) query by committee [61, 95], in which the most informative instance is the one that a committee of classifiers find most disagreement; and (3) expected error reduction [116], which aims to query instance that minimizes the expected error of the classifier. Most existing studies have focused on a single domain and have assumed that an omniscient oracle always exists that provides an accurate label for each query.

Multi-task is studied for addressing the dataset which contains multiple tasks. There are many different approaches to address this problem. Bakker and Heskes [16] used a common prior distribution in hierarchical Bayesian models to model task relatedness. Evgeniou et al. [48] worked on the kernel methods and regularization networks for multi-task problems. Argyriou et al. [8] and Jacob et al. [77] worked on the clustered tasks for multi-task learning. Recently, trace-norm regularization for multi-task learning has been proposed [2, 8, 103]. In

the foundations to a theoretical understanding of multi-ask learning, J. Baxter [20] used the covering numbers to expose the potential benefits of multi-task learning. Ando et al. [6] and Maurer [93] started to use Rademacher averages to give excess risk bounds for a method of multi-task subspace learning. Maurer [94] provided the excess risk bounds for the multi-task learning based on trace-norm regularization algorithm. Maurer [93] provided a general form of the bound of multi-task learning and their result is dimension independent. Kakade [82] introduced a general and elegant algorithm to derive bounds for the method which employs matrix norms as regularization.

In multi-armed bandit problem, the forecaster's goal is to maximize the sum of payoffs over time based on the historical payoff information. There are two basic settings. In the first, the stochastic setting, the payoffs are i.i.d. drawn from an unknown distribution. The upper confidence bound (UCB) strategy has been used to explore the exploration-exploitation trade-off [14, 31], in which an upper bound estimate is constructed on the mean of each arm at a fixed confidence level, and then the arm with the best estimate is selected. In the second, the adversarial setting, the i.i.d. assumption does not exist. Auer et al. [15] proposed the EXP3 algorithm for the adversarial setting, which was later improved by Bubeck and Audibert [10]. Fang and Tao introduced the networked bandits for the network setting [49].

There are some works about active learning for some specific multi-task learning problems. Reichart et al. [113] presented some huristics such as interatively selecting sample from different tasks or aggregating the selection scores from tasks. Qi et al. [107] proposed to estimate the correlation of labels directly from training data and used the resulting joint label distribution to guide active learning. Zhang [143] proposed a multi-task active learning algorithm with output constraints and considered the cross task value of information criteria. They used the value of information framework to measure the reward of a labeling assignment over all relevant tasks reachable through constraints. Acharya et al. [3] used a topic-modeling framework for the dimension reduction to adapt the expected error reduction strategy.

Our algorithm is the first work to consider the risk of multi-task learner and the confidence bounds for active learning and there is also few work about active learning based on multi-task learning with trace-norm regularization method.

4.3 Problem definition

We consider active learning for multi-task setting with unknown instance labeling $y_i \in \{-1,+1\}$ for each instance $x_i \in X$. First, we describe the multi-task learning setting. There are M tasks and a pool of data $\mathcal{P} = \{(x_1,y_1),\cdots,(x_n,y_n)\}$, drawn from an unknown distribution $\mathcal{D}$ defined on a domain $X \times \{-1,+1\}$. For each task, the unknown input-output

relationship is modeled by a distribution μ_m on $H \times \mathbb{R}$ with $\mu_m(x,y)$ and each task is modeled by a vector Z_m of n independent random variables $Z_m = (Z_{m,1}, \cdots, Z_{m,n})$, where each task $Z_{m,i} = (x_{m,i}, y_{m,i})$ is distributed according to μ_m. We assume bounded inputs, for simplicity $\|X\| \leq 1$.

Second, we introduce the popular method of multi-task learning algorithm – trace-norm regularization. For each task, we focus on linear predictors and the associated predictor predicts output of $\langle w,x \rangle$ for an observed input $x \in H$ using a specified weight vector $w \in H$. We then let $H = \mathbb{R}^d$ and $W = (w_1, \cdots, w_M)$. If the observed output is y then the loss is $\ell(h(x),y)$, where ℓ is a fixed loss function on $\mathbb{R}^2$ and is assumed that $\ell \in [0,1]$. We let ℓ be one of convex losses and we use squared loss. A classical learning strategy is empirical risk minimization, defined as

$$\arg\min_{h \in H} \hat{R}(h), \tag{4.1}$$

where

$$\hat{R}(h) = \frac{1}{M} \sum_{m=1}^{M} \frac{1}{n} \sum_{i=1}^{n} \ell(h(x_{m,i}), y_{m,i}) \tag{4.2}$$

indicates the average empirical risk.

We consider the trace-norm regularization method. The empirical risk minimization is defined as

$$\arg\min_{W} \frac{1}{M} \sum_{m=1}^{M} \frac{1}{n} \sum_{i=1}^{n} \ell(\langle w_m, x_{m,i} \rangle, y_{m,i}) \tag{4.3}$$

and the multi-task learning with trace-norm regularization can be defined as

$$W = \{W \in \mathbb{R}^{dM} : \|W\|_* \leq B\sqrt{M}\}, \tag{4.4}$$

where $\|W\|_* = tr((W^*W)^{1/2})$ and $B > 0$ is a regularization constant and $\sqrt{M}$ is a normalization factor.

These above are used as a basis for the design of our active sampling strategies. Unlike passive learning algorithm that takes as input a randomly chosen training data, active learning algorithm is to actively select an instance and ask for its label iteratively.

In the multi-task problem, more formally, given an unlabeled dataset $\mathcal{U}$, for each instance x_i, the label y_i is annotated when x_i is queried. The multi-task learning algorithm receives the selected instance with its label and adds it into the train dataset. Then the labeled dataset $\mathcal{L}$ is updated. The multi-task learning algorithm then evaluates itself using the new updated train dataset. Since the learner's performance is deemed to improve as more instances are used, the goal of active learning is to trade-off predictive performance and the number of queries.

Normally, we assume there exists a budget of querying. That is under this budget, we design a strategy to select instances to generate good multi-task learner.

4.4 Algorithm

We consider the active learning algorithm from the perspective of bandit. In the active learning, we try to find a hypothesis $h \in H$ with low risk by using portions of data. That is, we have many different hypotheses with different risks and we want to quickly discard suboptimal hypotheses. In the bandit problem, we try to find the best arm to obtain highest payoff. Given lots of arms with different payoffs, we want to filter out the good arms which have higher payoffs than the rest. Thus it is natural to consider the active learning as a bandit framework. The hypothesis can be considered as an arm and the corresponding loss of hypothesis can be considered as payoff.

Arm - hypothesis

In the multi-armed bandits, the player has many arm candidates to pull. Each arm is associated with unknown rewards. In the multi-task learning, we consider the hypothesis as the arm. Hypotheses are learned based on a dataset. Each hypothesis is associated with unknown risk.

Specially, the hypothesis is learned by using trace-norm regularization algorithm. We rewrite the multi-task learning formulation based on trace-norm regularization as the following optimization problem

$$h = \arg\min_{h \in H} \hat{R}(h) + \mu \|W\|_*. \tag{4.5}$$

The above optimization problem is a convex multi-task learning formulation and there are several efficient methods to solve it.

When the train dataset is different, the learned hypothesis will be different. In active learning procedure, at time step t, we collect the labeled dataset $\mathcal{L}_t$ and learn a new hypothesis h_t according to (4.5). We can compute the risk of h_t. By selecting an instance and querying its label from the dataset at time step $t+1$, and by updating the labeled dataset to $\mathcal{L}_{t+1}$, we can equate pulling a new hypothesis as solving the optimization problem $h_{t+1} = \arg\min_{h \in H} \hat{R}(h) + \mu \|W\|_*$. Then we can also compute the risk of h_{t+1}.

Payoff - risk

In the multi-armed bandits, the payoff is generated by an arm. When the player pulls an arm then the player receives the payoff. However, in the multi-task problem, the payoff

is not obvious. We define the risk of hypothesis as the payoff because the risk can be considered as the performance measure of the hypothesis. In active learning, as the trace-norm regularization algorithm receives new data, the hypothesis is computed again. The risk of hypothesis is straightforward and defined as

$$R(h) = \frac{1}{M}\sum_{m=1}^{M} \mathbb{E}_{(x,y)\sim\mu_m}[\ell(h(x),y)]. \tag{4.6}$$

Since the μ_m are unknown, the risk estimation is based on the finite sample of observations and normally uses average empirical risk. At time step t, the learned h_t has a risk. At time step $t+1$, after receiving the new instance with its querying label information, we learn a new hypothesis h_{t+1} and then compute the risk of h_{t+1}. When use different hypotheses, we may have different loss values.

Trade-off between exploration and exploitation

In multi-armed bandits, at each time step, the player needs to choose either to explore an arm which has not been pulled in the past, or to exploit the knowledge of cumulative losses of arms that have been pulled in the past. In the multi-armed bandits, it is popular to mitigate the exploration-exploitation trade-off using the upper confidence bound of estimation [31]. In active learning for multi-task problem, naturally, we can use a similar technique that we use a confidence bound on the risk of each hypothesis.

For the multi-task learning problem, firstly, we must learn a large enough candidate set to contain hypothesis set with low risk and it is defined as

$$h \in \arg\min_{h\in H} R(h), \tag{4.7}$$

where $R(h)$ is the risk of hypothesis (4.6). Furthermore, we should also learn a small enough hypothesis set that we can find such hypothesis from a finite number of samples. We define the confidence for each hypothesis as

$$C(h) = R(h) - R(h^*), \tag{4.8}$$

where

$$h^* = \arg\min_{h\in H} R(h). \tag{4.9}$$

It means the risk of the operator which we find by trace-norm regularization algorithm $R(h)$ based on the observed data, is not too different from the risk of h^*.

Considering both the risk and the corresponding confidence, we want to find a hypothesis which can be

$$h = \arg\min_{h \in H} R(h) + C(h). \tag{4.10}$$

It is not easy to compute (4.8) directly because h^* is unknown. However, we can provide a bound of $C(h)$. We define the confidence bound $CB(h)$ of hypothesis h as

$$CB(h) = \sup\{R(h) - R(h^*)\}. \tag{4.11}$$

This confidence bound $CB(h)$ is the least upper bound of the confidence on the risk of h. We use it to replace $C(h)$ and rewrite (4.10) as

$$h = \arg\min_{h \in H} R(h) + CB(h). \tag{4.12}$$

Then we want to minimize both the risk and the upper confidence bound.

4.4.1 Confidence bounds

We begin with the notation required to develop our confidence bounds, and provide a confidence bound which is data-dependent. The confidence bounds are for heterogeneous sample sizes with weighted trace-norm. The sample size for m-th task is denoted by n_m and we abbreviate $\overline{n}$ for the average sample size, $\overline{n} = (1/M)\sum_m n_m$, so that $\overline{n}M$ is the total number of data. We rewrite (4.4) as

$$W = \{W \in \mathbb{R}^{dM} : \|SW\|_* \leq B\sqrt{M}\}, \tag{4.13}$$

where $S = (s_1, \cdots, s_M)$ and $s_m = \sqrt{\overline{n}/n_m}$.

We rewrite (4.8) as

$$R(h) - R(h^*) = [R(h) - \hat{R}(h)] + [\hat{R}(h) - \hat{R}(h^*)] + [\hat{R}(h^*) - R(h^*)], \tag{4.14}$$

where $R(h)$ and $R(h^*)$ are the risks of h and h^*, respectively, and $\hat{R}(h)$ and $\hat{R}(h^*)$ are the empirical risks of h and h^*, respectively.

According to the Hoeffding's inequality, with the probability at least $1-\delta$, we have

$$\hat{R}(h^*) - R(h^*) < \sqrt{\frac{\ln(1/\delta)}{2\overline{n}M}}. \tag{4.15}$$

The above bound can be used to replace the third term of (4.14). It is obvious that the second term is always negative. Now we focus on the first term by analysing

$$\sup_{h\in H}\{R(h)-\hat{R}(h)\}. \tag{4.16}$$

Based on the trace-norm regularization method we define the empirical Rademacher complexity as

$$\Re(h)=\frac{2}{M}\mathbb{E}_\sigma \sup_{w\in W}\sum_{m=1}^{M}\frac{1}{n}\sum_{i=1}^{n_m}\sigma_i^m\ell(\langle w_m,x_{m,i}\rangle,y_{m,i}). \tag{4.17}$$

According to the standard technique from [19], we have with probability at least $1-\delta$ (4.8) is bounded by

$$\Re(h)+\sqrt{\frac{9\ln(1/\delta)}{2\overline{n}M}}. \tag{4.18}$$

We assume that the loss function ℓ is a Lipschitz loss function and let $\ell(.,.)<L$. According to standard results on Rademacher averages, we have

$$\begin{aligned}\Re(h)&\leq\frac{2L}{M}\mathbb{E}_\sigma\sup_{w\in W}\sum_{m=1}^{M}\frac{1}{n}\sum_{i=1}^{n_m}\sigma_i^m\ell\left(\langle w_t,\frac{x_{m,i}}{n_m}\rangle,y_{m,i}\right)\\&=\frac{2L}{T}\mathbb{E}_\sigma\sup tr(W^*D)\\&=\frac{2L}{T}\mathbb{E}_\sigma\sup tr(W^*SS^{-1}D),\end{aligned} \tag{4.19}$$

where the random operator $D:H\rightarrow R^T$ is defined for $v\in H$ by $(Dv)_t=\langle v,\sum_{i=1}^{n_m}\sigma_i^m X_{m,i}/n_m\rangle$, and the diagonal matrix S is as above. According to Holder's and Jensen's inequalities, we have

$$\Re(h)\leq\frac{2LB}{\sqrt{M}}\sqrt{\mathbb{E}_\sigma\|D^*S^{-2}D\|_\infty}. \tag{4.20}$$

According to [94], we rewrite $\mathbb{E}_\sigma\|D^*S^{-2}D\|_\infty$ as

$$\mathbb{E}_\sigma\|D^*S^{-2}D\|_\infty=\mathbb{E}_\sigma\left\|\left(\frac{1}{\overline{n}n_m}\right)\sum_{m,i}\langle v,\sigma_i^m x_{m,i}\rangle\sigma_i^m x_{m,i}\right\|_\infty. \tag{4.21}$$

Finally we have

$$\Re(h)\leq 2LB\left(\sqrt{\frac{\|\hat{C}\|_\infty}{\overline{n}}}+\sqrt{\frac{2(\ln(\overline{n}M)+1)}{\overline{n}M}}\right), \tag{4.22}$$

where $\hat{C} = \frac{1}{nM}\sum_{m,i}\langle x_{m,i}, w_m\rangle w_m$.

Combine (4.14), (4.15) and (4.22), we can have

$$\begin{aligned} & R(h) - R(h^*) \\ & \leq \sqrt{\frac{ln(1/\sigma)}{2\bar{n}M}} + 2LB\left(\sqrt{\frac{\|\hat{C}\|_\infty}{\bar{n}}} + \sqrt{\frac{2(\ln(\bar{n}M)+1)}{\bar{n}M}}\right). \end{aligned} \tag{4.23}$$

Now we provide an upper bound of $R(h) - R(h^*)$. Then we specify the definition of the confidence bounds of h and set

$$CB = \sqrt{\frac{ln(1/\sigma)}{2\bar{n}M}} + 2LB\left(\sqrt{\frac{\|\hat{C}\|_\infty}{\bar{n}}} + \sqrt{\frac{2(\ln(\bar{n}M)+1)}{\bar{n}M}}\right). \tag{4.24}$$

4.4.2 Active multi-task learning via bandits

We show the proposed algorithm in Algorithm 4.1. The proposed algorithm is to maintain a sampling distribution over the pool of data. At each time step, we select an instance according to the distribution. Our algorithm is to minimize the confidence bounds. According to the algorithm (Lines 14-22), our strategy encourages querying instances which have small margin w.r.t the current hypothesis, h_t, or instances which have already been queried for their labels, but on which the current hypothesis, h_t suffers a large loss. The selection procedure tries to balance the trade-off between minimizing the risk and finding the closest hypotheses to the optimal hypothesis.

4.5 Analysis

We define the associated class of linear functions $\mathcal{F}_W$ as $\mathcal{F}_W := \{x \to \langle w, x\rangle : w \in W\}$. We provide the bounds of the complexity of $\mathcal{F}$ for certain sets.

Theorem 4.1. *(Complexity Bounds) [81] Let S be a closed convex set and let $\mathcal{F} : S \to \mathbb{R}$ be σ-strongly convex w.r.t $\|.\|_*$. Further, we have $\|X\| \leq 1$. Define $W = \{w \in S : \mathcal{F}(w) \leq W_*^2\}$. Then, we have*

$$\mathfrak{R}_n(\mathcal{F}_w) \leq W_*\sqrt{\frac{2}{\sigma n}}. \tag{4.25}$$

Recall that our loss function is a Lispschitz loss function with Lipschitz constant L. Now we can obtain the generalization error bounds using Rademacher complexity.

Algorithm 4.1. Active Multi-task Learning via Bandits (AMLB)

Input: $\mathcal{P}$, $\mathcal{L}$, $p_0([0,1/N]$, we set $p_0 = 1/N)$, *Budget*

Set $t = 1$ and $w_i(1) = 1$ for $i = 1, \cdots, N$
while $t \leq$ *Budget* **do**
  for $x_i \in \mathcal{P}$ **do**
    Set $W_t = \sum_{i=1}^{N} w_i(t)$ and

$$p_i = (1 - np_0)\frac{w_i(t)}{W_t} + p_0.$$

  end for
  Select an instance x^* randomly according to the probabilities $p_1, \cdots, p_N$.
  Make the query for the instance x_*:
  if $x \notin \mathcal{L}$ **then**
    Ask for the label y_*.
    $\mathcal{L} \leftarrow \mathcal{L} + (x_*, y_*)$.
  else
    Use the y_* from $\mathcal{L}$.
  end if
  Solve

$$h = \arg\min_{h \in H} R(h) + CB(h).$$

  for $x_i \in \mathcal{P}$ **do**
    if $x_i \in \mathcal{L}$ **then**
      Update reward: $g_{i,t} = 1 - \ell(h(x_i), y_i)$.
    else
      Update reward: $g_{i,t} = 0$.
    end if
    Update weight: $w_i(t+1) = w_i(t)\exp(g_{i,t}p_0)$.
  end for
  $t \leftarrow t + 1$.
end while

Theorem 4.2. *[19] For a Lispchitz loss ℓ bounded by c, for any $\delta > 0$ with probability at least $1 - \delta$ simultaneously for all $f \in \mathcal{F}$ we have that,*

$$R(f) \leq \hat{R}(f) + 2L\mathfrak{R}_n(\mathcal{F}) + c\sqrt{\frac{\log(1/\delta)}{2n}}. \tag{4.26}$$

Now based on the above theorem and bounds on Rademacher complexity we provided a bound of our algorithm.

Theorem 4.3. *For the multi-task problem, there are M tasks, and each task is associated with a linear function class $\mathcal{F}_W$, for all $w \in W$, we define $W = \{w \in S : \mathcal{F}(w) \leq W_*^2\}$. For*

$h \in \mathcal{F}_W$ with probability at least $1-\delta$ we have

$$R(h) \leq \hat{R}(h) + 2MLW_* \sqrt{\frac{2\log(d)}{n}} + LMW_* \sqrt{\frac{\log(1/\delta)}{2n}}. \tag{4.27}$$

Proof. According to standard technique of Rademacher complexity, we have the expected risk,

$$R(h) \leq \hat{R}(h) + 2L\mathfrak{R}_n(\mathcal{F}) + c\sqrt{\frac{\log(1/\delta)}{2n}}.$$

Since there are M linear functions for all the tasks, according to Theorem 4.1 we have

$$R(h) \leq \hat{R}(h) + 2MLW_* \sqrt{\frac{2\log(d)}{n}} + LMW_* \sqrt{\frac{\log(1/\delta)}{2n}}.$$

□ □

4.6 Experiments

We evaluate our algorithm on a synthetic dataset and three real multi-task datasets: Restaurant & Consumer dataset, Dermatology dataset and School dataset. These datasets all contain multiple tasks. For each dataset, we use the trace-norm regularization algorithm to build the multi-task learner. Following the standard protocol, experiments on each dataset are performed in 10-fold cross-validation and the results of average accuracy of classification are reported. In the train dataset we use different sampling strategies to select instances for comparison. Then we test the learner on the test dataset.

We compare our algorithm **AMLB** with three baselines:

- **ERR**: expected error reduction based method [116]. It is also a variant of [3] and we do not use dimension reduction for the fairness of the comparison;
- **VIO**: value of information algorithm [143], which summaries the uncertainty of each task using traditional uncertainty strategy, defined as

  $$VIO(Y,x) = \sum_y p(Y=y|x) R(p, Y=y, x),$$

  where R is the rewards function and we use $R(p,Y=y,x) = -\log_p(Y=y|x)$. This strategy is to select the instance which has the most uncertainty information over all tasks;
- **Random**: passive learning algorithm, which randomly selects instances from dataset.

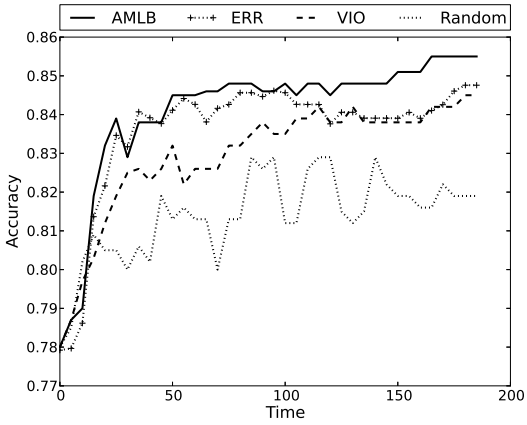


Fig. 4.1 *Performance comparison on the synthetic data.*

4.6.1 Synthetic data

We first illustrate our method on a synthetic dataset. We construct a dataset that contains multiple classification tasks. We defined a number of up to 10 tasks. Each of w parameters of these tasks was selected from a 6-dimensional Gaussian distribution which has zero mean and covariance Cov=Diag(1, 0.72, 0.63, 0.54, 0.45, 0.36). The training and test data were generated uniformly from $[0,1]^6$. The outputs $y_{m,i}$ were computed from the w_m and $x_{m,i}$ as $y_{m,i} = \langle w_m, x_{m,i} \rangle + \varepsilon$, where ε is zero-mean Gaussian noise which has standard deviation 0.1.

In Figure 4.1, it shows our AMLB outperforms others and ERR and VIO strategies are better than Random. The Random strategy works worst. These results show that the selectively choosing instances works better than passive learning. ERR does work well after $t = 100$ because it chooses some noise instances from unlabeled dataset. For VIO, an instance is uncertain to a task does not mean it is uncertain to other tasks. VIO computes a global uncertainty information and it is possible to select an instance that is not uncertain for some tasks. AMLB always try to select instance to reduce the risk. Unlike ERR, it also exploits the labeled dataset to build confident learner. AMLB does not work well at the very beginning because at that time it tries to explore new instances to build a good learner. Then it becomes better later.

4.6.2 Restaurant & consumer data

The Restaurant & Consumer dataset [132] consists of data to build a restaurant recommendation system and the aim of the recommendation is to predict consumers' ratings given to different restaurants. Each of 138 consumers gave three scores for food quality, service quality and overall quality. The dataset also consists of 44 various descriptive attributes of restaurants, including geographical position, cuisine type, price band, etc. We consider this

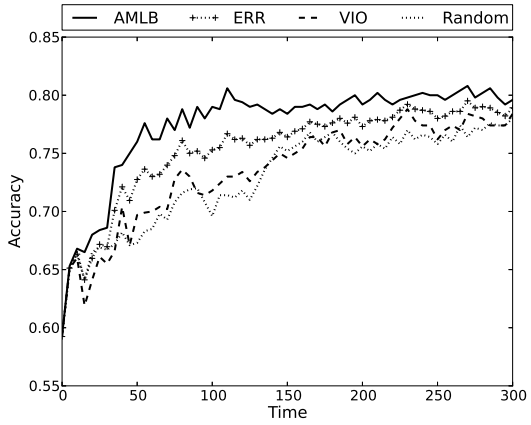


Fig. 4.2 *Performance comparison on the Restaurant & consumer data.*

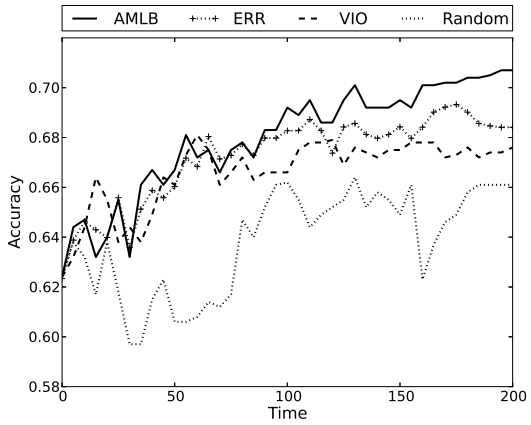


Fig. 4.3 *Performance comparison on the Dermatology data.*

to be a binary classification problem. We define median as the threshold and transform the scores to high or low labels. Our problem is to predict the labels given the attributes of a restaurant as an input query. Since there are 138 consumers, we construct 138*3 classification tasks.

In Figure 4.2, AMLB dominates all of the other methods and then ERR is better than the rest. Interestingly, the performance of VIO has close to Random strategy. Because a selected instance is uncertain for one task does not mean that it is uncertain for other tasks.

4.6.3 Dermatology data

Dermatology data is from the UCI datasets [89] and is used for the differential diagnosis of erythematous-squamous diseases. There are 6 diseases in this dataset: psoriasis, seboreic dermatitis, lichen planus, pityriasis rosea, cronic dermatitis, and pityriasis rubra pilaris. The dataset contains 34 attributes and 366 instances. The problem is to diagnose one of six

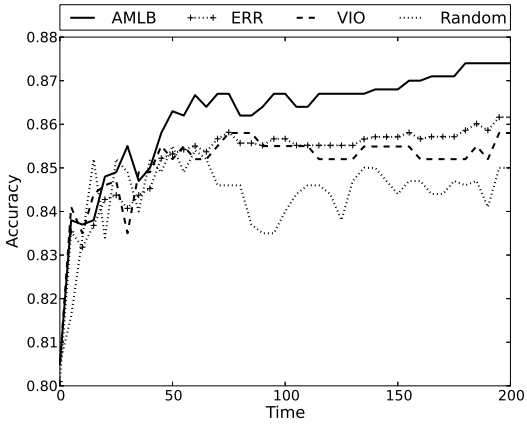


Fig. 4.4 *Performance comparison on the School data.*

dermatological disease based on these features. We transform this problem to a multi-task problem by constructing the six binary classification tasks.

In Figure 4.3, the performance of our method is better than other methods. At the beginning, the performance of our method is close to ERR and VIO. In this stage, AMLB tries to explore more instances to reduce the risk. Then its performance becomes increasingly stable. It demonstrates that exploration for improving confidence for the learner does not hurt the total performance.

4.6.4 School data

The school dataset is from the Inner London Education Authority [16]. This dataset contains examination scores of 15,362 students from 139 secondary schools in London during the years 1985, 1986 and 1987. The dataset consists of the year of examination (YR), 4 school-specific and 3 student-specific attributes. For a school, in each year, the attributes contain percentage of students eligible for free school meals, percentage of students in VR band one (highest band in a verbal reasoning test), school gender (S.GN.) and school denomination (S.DN.). Student-specific attributes contain gender, VR band and ethnic group. Similar to [48], we replaced categorical attributes with one binary variable for each possible attribute value and obtained 27 attributes. Our problem is to predict the student's performance in each school. Since there are 139 secondary schools, 139 classification tasks are constructed.

The average accuracies are shown in Figure 4.4. Our AMLB works better than others. At the beginning, the performance of all the methods is close. At this stage AMLB tries to explore the instances to reduce the risk thus its performance is a little variant. Later AMLB works much better than others. AMLB can exploit the instances to improve the confidence for the learner. The exploitation can benefit the performance of learner because these selected instances help the learner be close to the optimal learner.

4.7 Conclusion

In this work, we proposed a new active learning algorithm for multi-task learning, named active multi-task learning via bandits, which is a general active learning framework for multi-task learning problem. We cast the selection procedure as a bandit framework. We considered the hypothesis as an arm and selected the instances to filter out the hypotheses which have the better performance than others. Our active learning strategy balances the trade-off between minimizing the risk and improving the confidence bounds for the hypothesis. In our algorithm, at each round, we maintain a sampling distribution on data for selection, query the label of an instance according to the distribution and update the distribution based on the performance of newly trained multi-task learner. We also provided an implementation of our approach based on multi-task learning with trace-norm regularization method. Moreover, experimental results demonstrate the effectiveness and efficiency of our algorithm.

Chapter 5

Selective repeated labeling via bandits

Repeated labeling is an annotation problem in which, in a labeling task, the example labels are noisy but can be repeatedly acquired for multiple times. Our aim is to identify the best labeling tasks from a large number of labeling tasks with variable noise. In real applications, especially in the crowdsourcing setting, the task is often small, less-than-expert labeling can be obtained at low cost, and multiple repeated labels can be acquired for each example. However, preparing or processing the unlabeled part of an example can be even more expensive than labeling itself, and noisy examples decrease the performance of subsequent learning. In this work, we formalize the repeated labeling problem as a bandit model. Each labeling task can be considered as an arm and the labeling quality the payoff. We first introduce a simple repeated labeling strategy and an optimal labeling task based on the expected labeling quality, and then propose algorithms to select a proportion of the labeling tasks that have high expected labeling quality. The selection of labeling tasks for repeated labeling provides substantial benefit over the use of all labeling tasks with variable noise, and the expected labeling quality of the data is shown to be a good indicator of where to allocate labeling efforts. We show how many labels should be acquired for an example and which examples should be selected for learning when faced with a large number of labeling tasks. The proposed algorithms for repeated labeling via bandits are efficient, and we provide the theoretical guarantees of our algorithms and demonstrate their effectiveness and efficiency in comprehensive experiments.

5.1 Introduction

Labeling is important in real-world applications. Repeated labeling, where multiple data labels are repeatedly obtained from multiple sources, is often available via crowdsourcing. Data collection involves various preprocessing costs, including costs associated with acquir-

ing features, formulating data, cleaning data, and obtaining labels [137]. Traditional data collection is therefore limited by the high cost of expert labeling; however, as modern crowdsourcing systems is increasingly popular (such as Amazon's Mechanical Turk), obtaining labels is becoming easier and cheaper. For examples, the Mechanical Turk allows thousands of human labelers to annotate objects via an online portal, where human labelers are provided with an interface to look at the objects or articles and asked to label them. In this way, huge amounts of labeling information can be obtained at very low cost.

However, in repeated labeling, noisy labeling can be a problem because the expertise of labelers are different and the difficulties of labeling problems are variable [75]. Another problem with repeated labeling is the cost of labelers; even though labeling is cheap, the payments to labelers may be different according to their expertise and the expense can become great when more labelers are used. Furthermore, acquiring features can incur a cost and the cost may increase when labeling needs more effort.

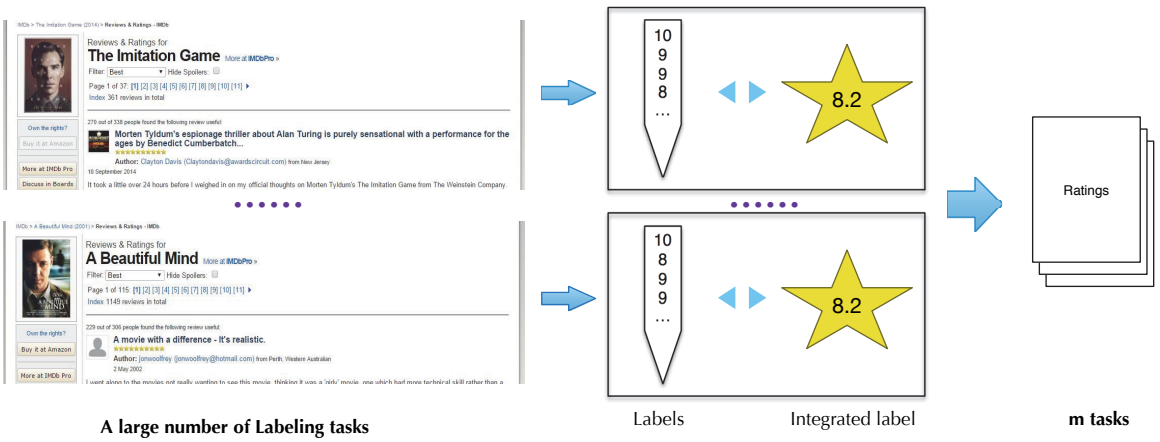


Fig. 5.1 *An example of selective repeated labeling. There are a large number of labeling tasks, where each task corresponds to multiple repeated labels and an integrated label (using a majority voting/average ratings). Our goal is to design a selective repeated labeling strategy that identifies the best m labeling tasks.*

It would therefore be useful to design smarter strategies for repeated labeling. Since there are often a large number of labeling tasks, it would be useful to select a small or fixed proportion of labeling tasks with best labeling quality to improve efficiency and efficacy. Here, we propose a selective repeated labeling strategy based on the well-known multi-armed bandit model. Multi-armed bandits is a well-studied model for sequential decision-making and is particularly suited to analysis in the context of repeated labeling.

A multi-armed bandit problem (or bandit problem) is a sequential decision problem defined by a set of actions (or arms). The term 'bandit' originates from the colloquial term for a casino slot machine ('one-armed bandit'), in which a player (or a forecaster) faces a finite

number of slot machines (or arms). The player sequentially allocates coins (one at a time) to different machines and earns money (or payoff) depending on the machine selected. The goal is to earn as high a payoff as possible. Since the forecaster does not know the process generating the payoff but has historical payoff information, the bandit problem highlights the fundamental difficulty of decision making in the face of uncertainty: balancing the decision of whether to exploit past choices or make new choices with the hope of discovering a better one.

We first present a new framework for repeated labeling using the multi-armed bandit model. In the repeated labeling problem, each example's labeling is a labeling task in which multiple labels are repeatedly obtained from multiple noisy labelers. We assume that there is no more information about the labelers. From the bandit perspective, a labeling task can be considered as an arm and the uncertainty of labels for an example can be considered as the payoff. As shown in Figure 5.1, we are often confronted with a large number of labeling tasks; however, due to cost or budget constraints, we would rather select a small or fixed number of these labeling tasks with high expected labeling quality.

Selecting a fixed number of labeling tasks is essentially a decision problem and can be regarded as subset selection in the multi-armed bandit problem. We introduce the definition of ε-optimal labeling task and use it to identify the optimal labeling task. Taking the expected labeling quality into account, we design a simple repeated labeling strategy. We then extend this to address how to identify the best m labeling tasks, and in doing so propose the best m labeling algorithm by indexing the labeling tasks using the expected labeling quality. Furthermore, we also propose an improved best m labeling algorithm. The theoretical guarantee is based on the study of multi-armed bandits.

Our main contributions are as follows. First, we introduce a new framework for the repeated labeling problem using the multi-armed bandit model; second, we design a simple repeated labeling algorithm, Naive Repeated Labeling, to repeatedly acquire labels for an example; third, we propose two algorithms, the **Best** m **Labeling** and the **Improved Best** m **Labeling** , to selectively obtain the labeling tasks; and fourth, we provide a theoretical guarantee for our algorithm and demonstrate the effectiveness of our algorithms empirically.

The remainder of this work is organized as follows. In Section 5.2, we briefly introduce the related literature. In Section 5.3, we present our new framework for repeated labeling under the multi-armed bandit model. In Section 5.4, we present Naive Repeated Labeling algorithm and propose two algorithms, the **Best** m **Labeling** and the **Improved Best** m **Labeling** , followed by their label complexity analyses. The experiments on benchmark data sets are demonstrated in Section 5.5. Finally, we conclude in Section 5.6.

5.2 Related work

Repeated labeling is studied recently, especially in the crowdsourcing systems. Sheng et al. [120] provided analysis for the impact of repeated labeling. They defined the quality of the resultant labels as a function of the number and the individual quality of the labelers. The repeated labeling often involves noisy labels and noisy labelers. There are lots of works to address the noisy labelers in the multiple weak labeler setting. Some studies worked on estimating the confidence of labelers [45, 111, 112, 136]. The uncertainty of examples has also been investigated. Some works studied estimating the true label from multiple uncertain labels [57, 140].

Another problem related to our work is data acquisition, which is important for real-world applications, such as feature obtaining and label obtaining in multiple labeler setting [67]. It inspires lots of works, including cost-sensitive learning [65], feature selection [133], and active learning [57, 140]. Gao et al. [65] estimated the profit of the crowdsourcing task so that those tasks with no future profit from multiple users can be terminated. Viappiani et al. [133] developed a Bayesian approach to concept learning for crowdsourcing applications. Yan et al. [140] employed a probabilistic model for learning from multiple labelers and actively selected labelers for labeling.

The traditional multi-armed bandit problem does not assume that side information is observed. The forecaster's goal is to maximize the sum of payoffs over time based on the historical payoff information. There are two basic settings: the stochastic and the adversarial settings. In the stochastic setting, the payoffs are i.i.d. drawn from an unknown distribution. The Upper Confidence Bound (UCB) strategy has been used to explore the exploration-exploitation trade-off [11, 14, 86], in which an upper bound estimate is constructed on the mean of each arm at a fixed confidence level, and then the arm with the best estimate is selected. In the adversarial setting, the i.i.d. assumption does not exist. Auer et al. [15] proposed the EXP3 algorithm for the adversarial setting, which was later improved by Bubeck and Audibert [10]. Fang and Tao [49] introduced the networked bandits for the network setting.

Recently, some extensions of exploration of multi-armed bandits to subset selection have been studied [29, 83, 84]. Bubeck et al. [29] proposed a variant of Successive Rejects algorithm that outputs a set of m-best arms under a fixed budget of pulls with high probability. Kalyanakrishnan and Stone [83] presented an explore-m algorithm under the PAC setting. Kaufmann and Kalyanakrishnan [84] drove a new bound using KL-divergence-based confidence intervals under the PAC and the fixed-budget setting.

The bandit problem has been widely applied in real-life applications, such as recommendation systems and advertising. Chu et al. [37] first introduced the bandit problem to

recommendation systems by considering a personalized recommendation as a feature-based exploration-exploitation problem. This problem was formalized as a contextual bandit problem with linear payoff. They focused on the article-selection strategy based on user-click feedback, and maximized the total number of clicks. The features of the users and articles were defined as contextual information, and the expected payoff of an arm was assumed to be a linear function of its contextual features, including the user and article information. Finally, the LinUCB algorithm was proposed to solve this problem and the method attained a good empirical regret. They further extended the algorithm as SupLinUCB, and theoretical analysis proved a $O(\sqrt{Td\log^3(KT\log(T)/\delta)})$ regret.

There are limited studies that consider the labeling task selection in repeated labeling problem. To our knowledge, this is the first work to formalize this problem as a bandit framework and present the selective repeated labeling algorithms as a solution.

5.3 General framework

In this work, we study the problem of repeated labeling. We consider a pool of data with labels, where each example has multiple repeated labels. Each training example (x_i, y_i) consists of the feature portion x_i, and the label portion y_i. Both procuring the feature portion x_i and labeling the training example with a label y_i incur a cost respectively. For simplicity, we assume that the cost is constant across all examples. Specially, given an example x_i without labels, we collect the labels for a specific class from multiple labelers. y_{ij} indicates the label of example x_i from labeler j. Then each example (x_i, y_i) has a final assigned label $\bar{y}_i$, which is inferred from the individual y_{ij}s by a strategy, such as majority voting or average ratings.

A labeling task is defined as labeling an example x_i for a specific class by multiple repeated labels $\{y_{i1}, \cdots, y_{in}\}$. Different labelers label the example based on their expertise. The confidence of labeler j is denoted by c_j, which is $P(y_{ij} = y_i)$; if we consider the case of binary classification, the probability that labeler j assigns an incorrect label is $1 - c_j$. However, it is hard to know the confidence of a labeler in real applications, such as crowdsourcing systems. Hence we do not use the information of labeler and focus on the labels themselves. We define the quality of a labeling task as the difference between the labels and the integrated label. Each example under repeated labeling is assigned a single "integrated" label $\bar{y}_i$, inferred from the individual y_{ij}s. Thus the labeling quality $\bar{q}_i$ depends on integrated label $\bar{y}$, which is $\bar{q}_i = P(y = \bar{y})$. The expected labeling quality for an example x_i is defined as $q_i = \mathbb{E}[\bar{q}_i]$.

We introduce three reasonable assumptions about the repeated labeling. These assumptions allow us to focus on a certain set of problems that arise when there are a large number of labels from multiple noisy labelers. Firstly, we assume that $P(y_{ij} = y_i|x_i) = P(y_{ij} = y_i) = p_j$, that is, individual labeling quality is independent of the specific example being labeled. Secondly, we assume labels are conditionally independent from each other. Note that each labeling task has an individual quality q_i instead of a uniform quality. Thirdly, we assume for simplicity that each labeler j only gives one label, but that is not a restrictive assumption in what follows. We do not care about how labeling comes and only consider the labels.

The objective of selective repeated labeling is to find a proportion of data with high quality from a large number of labeling tasks. Let m denote the size of this proportion of data. We formalize the repeated labeling under the multi-armed bandit framework. We consider each example labeling task as an arm and the current labeling quality as the reward. In multi-armed bandit problem, the goal is to pull the arm which has the cumulative maximal payoff. Similar to the multi-armed bandit problem, the goal of selective repeated labeling is to collect m examples with high expected labeling quality. In this work, we try to answer two questions: (1) How many labels should we obtain for an example? (2) Which example with multiple repeated labels should be selected for learning?

From the bandit perspective, we introduce the PAC style bounds for each labeling task. A high quality labeling task is defined as, with high probability, a near-optimal labeling task, namely, with probability at least $1-\delta$ output an ε-optimal labeling task. We need to identify the best m labeling tasks like that. It is studied in bandits that abstracts the case where the agent needs to choose one specific arm, and it is given only limited exploration initially.

Inspired by the bandits, first we focus on one specific labeling task and introduce the concept of ε-optimal labeling task. We show how to find an optimal labeling task and introduce the label complexity for repeated labeling. The label complexity of repeated labeling algorithm is the total number of labels it uses before termination. Then we adopt the idea to a more realistic setting that we actively select a proportion of m labeling tasks. We propose two algorithms, the **Best** m **Labeling** and the **Improved Best** m **Labeling** , for different settings and provide the theoretical analysis.

5.4 Algorithm

We have a set of labeling tasks denoted by X where $n = |X|$. Each labeling task is considered as a sequential process that the labels are received in sequence. At each time step t, a payoff for the labeling task is generated when a random label y_{it} is received. The payoff is the quality of labeling task at time t. In this work, we use majority voting to infer integrated

label, hence we redefined it as

$$\bar{q}_i = \frac{1}{t}\sum_{y_i} \mathbb{I}(y_i = \bar{y}_t), \tag{5.1}$$

where $\mathbb{I}$ is an indicator function. For every task $x \in X$, the labeling quality is bounded in $[0,1]$. Denote the labeling quality of the labeling tasks by $\bar{q}_1, \cdots, \bar{q}_n$ and $q_i = \mathbb{E}[\bar{q}_i]$. Without loss of generality and for the brevity of analysis we assume that

$$q_1 > q_2 > \cdots > q_n. \tag{5.2}$$

A labeling task with the best expected labeling quality is defined as the best labeling task, and denote by x^*, and its expected quality q^* is the optimal quality. A labeling task whose expected quality is strictly less than q^* is defined as a non-best labeling task. A labeling task is defined as an ε-optimal labeling task if its expected labeling quality differs at most ε from the optimal quality, i.e., $\mathbb{E}[q_i] > q^* - \varepsilon$.

In our repeated labeling framework, a labeling task receives one label at each time and the quality is also calculated after receiving a new label. When we collect a large number of examples and labels, we need to decide which examples with their labels should be selected for learning. When making its selection, the algorithm may depend on history, including the actions and payoffs, up to time $t-1$.

We first focus on a single labeling task. We indicates which labeling task can be considered as a high quality labeling task. From the PAC learning framework, we define that a labeling task is an ε-optimal for the repeated labeling task with label complexity T, if it outputs an ε-optimal labeling task, x, with probability at least $1-\delta$, when it terminates, and the number of time steps the algorithm perform until it terminates is bounded by T. Thus we need to identify the labeling task which is ε-optimal.

5.4.1 Repeated labeling strategies

First we propose a simple repeated labeling strategy for labeling. Then we provide the analysis based on the PAC learning framework.

We investigate an (ε,δ)-PAC algorithm for the repeated labeling problem. Such algorithms are required to output an ε-optimal labeling task with probability $1-\delta$. We start with a simple solution that collecting repeated labels for each labeling task at least $1/(\varepsilon/2)^2 \log(2n/\delta)$ times. Then we calculate an empirical quality for each labeling task. The repeated labeling strategies is described in Algorithm 5.1.

Algorithm 5.1. Naive Repeated Labeling

Input: $\varepsilon > 0$, $\delta > 0$, X.

1: **for** labeling task $x_i \in X$ **do**
2: Collect labels of this task for $\frac{4}{\varepsilon^2}\log(\frac{2n}{\delta})$ times.
3: Let $\hat{q}_i$ be the empirical labeling quality of the labeling task x_i.
4: **end for**
5: Output all the labeling tasks.

In order to analyze our algorithms, we use the large deviation bounds in this work and rely on the Hoeffding's inequality.

Lemma 5.1. *(Hoeffding, 1963) Let X be a set, D be a probability distribution on X of examples, and $f_1, \cdots, f_m$ be real-valued functions defined on X with $f_i : X \to [a_i, b_i]$ for $i = 1, \cdots, m$, where a_i and b_i are real numbers satisfying $a_i < b_i$ and $x_1, \cdots, x_m$ be independent identically distributed examples from D. Then we have the following inequality*

$$P\left[\frac{1}{m}\sum_{i=1}^{m} f_i(x_i) - \left(\frac{1}{m}\sum_{i=1}^{m}\int_{a_i}^{b_i} f_i(x_i)D(x)\right) \geq \varepsilon\right] \leq \exp\left\{-\frac{2\varepsilon^2 m^2}{\sum_{i=1}^{m}(b_i - a_i)^2}\right\} \tag{5.3}$$

and

$$P\left[\frac{1}{m}\sum_{i=1}^{m} f_i(x_i) - \left(\frac{1}{m}\sum_{i=1}^{m}\int_{a_i}^{b_i} f_i(x_i)D(x)\right) \leq -\varepsilon\right] \leq \exp\left\{-\frac{2\varepsilon^2 m^2}{\sum_{i=1}^{m}(b_i - a_i)^2}\right\}. \tag{5.4}$$

Note that the boundedness assumption is not essential and can be relaxed in certain situations. Sometimes tighter bounds can be obtained using the relative Chernoff bound.

In order to identify the labeling task which is ε-optimal, according to Algorithm 5.1, instead of returning all the labeling tasks, we return a labeling task which is

$$x' = \arg\max_{x_i \in X}\{\hat{q}_i\}, \tag{5.5}$$

where x' is ε-optimal.

Theorem 5.1. *In Algorithm 5.1, given $\varepsilon > 0, \delta > 0$, if it returns an example $x' = \arg\max_{x \in X}\{\hat{q}_i\}$, the labeling task for example x' is ε-optimal, with label complexity $O((n/\varepsilon^2)\log(n/\delta))$.*

Proof. The label complexity is immediate from the definition of the algorithm. We now prove it is an (ε, δ)-PAC algorithm. Let x' be the labeling task for which $\mathbb{E}[\bar{q}_i] < q^* - \varepsilon$. Let

l be $\frac{4}{\varepsilon^2}\log(\frac{2n}{\delta})$. We want to bound the probability of the event $\hat{q}_{x'} > \hat{q}_{x^*}$.

$$\begin{aligned}
&P(\hat{q}_{x'} > \hat{q}_{x^*}) \\
&\leq P(\hat{q}_{x'} > \mathbb{E}[\bar{q}_{x'}] + \varepsilon/2 \text{ or } \hat{q}_{x^*} < q^* - \varepsilon/2) \\
&\leq P(\hat{q}_{x'} > \mathbb{E}[\bar{q}_{x'}] + \varepsilon/2) + P(\hat{q}_{x^*} < q^* - \varepsilon/2) \\
&\leq 2\exp(-2(\varepsilon/2)^2 l),
\end{aligned}$$

where the last inequality uses the Hoeffding inequality. Choosing $l = (2/\varepsilon^2)\log(2n/\delta)$ assures that $P(\hat{q}_{x'} > \hat{q}_{x^*}) \leq \delta/n$. Summing over all possible x' we have that failure probability is at most $(n-1)(\delta/n) < \delta$. □

Algorithm 5.1 provides the idea to design the repeated labeling strategies. Then based on it we can easily select the optimal labeling task among a large number of labeling tasks. Note that the expected quality depends on the number of labels. Different labeling tasks have different expected qualities.

We have solved how to select an ε-optimal labeling task. A natural question is if we want to find half size of all the labeling tasks, which have high expected labeling quality, how to select the labeling tasks?

Median elimination

We design a general algorithm that eliminates the worst half of the labeling tasks at each iteration. We do not expect the best labeling task to be empirically "the best", we only expect an ε-optimal labeling task to be above the median of the empirical labeling qualities. Let $\mathcal{T}$ be the target value, such as $\frac{1}{2}|X|, \frac{1}{4}|X|, \cdots$. The new algorithm is described in Algorithm 5.2. It substitutes the term $O(\log(1/\delta))$ for $O(\log(n/\delta))$ of naive bound.

In Algorithm 5.2, given a fixed target value which is less than $\frac{1}{2}|X|$, it processes several phases to achieve the target.

Theorem 5.2. *In Algorithm 5.2, for the lth phase, we have that:*

$$P\left[\max_{j\in S_l} q_j \leq \max_{i\in S_{l+1}} q_i + \varepsilon_l\right] \geq 1 - \delta_l. \tag{5.6}$$

Proof. Without loss of generality we look at the first round and assume that q_1 is the payoff of the best labeling task. We bound the failure probability by looking at the event

$$E_1 = \{\hat{q}_1 < q_1 + \varepsilon_1/2\} \tag{5.7}$$

Algorithm 5.2. Median Elimination

Input: $\varepsilon > 0$, $\delta > 0$, $\mathcal{T}$, X.

Set $S_1 = X$, $\varepsilon_1 = \varepsilon/4$, $\delta_1 = \delta/2$, $l = 1$.
repeat
  for labeling task $x_i \in S_l$ **do**
    Collect labels of this task for $\frac{1}{(\varepsilon_l/2)^2}\log(3/\delta_l)$ times.
    Let $\hat{q}_i^l$ be the empirical labeling quality of the labeling task x.
  end for
  Find the median of $\hat{q}_i^l$, denoted by m_l.
  $S_{l+1} = S_l \setminus \{x_i : \hat{q}_i^l < m_l\}$.
  $\varepsilon_{l+1} = \frac{3}{4}\varepsilon_l$, $\delta_{l+1} = \delta_l/2$.
  $l = l + 1$.
until $|S_l| = \mathcal{T}$
Output S_l.

which is the case that the empirical estimate of the best labeling task is pessimistic. Since we collect labels sufficiently, we have that

$$P[E_1] \leq \delta_1/3. \tag{5.8}$$

In case E_1 does not hold, we calculate the probability that a labeling task j, which is not an ε_1-optimal labeling task, is empirically better than the best labeling task.

$$\begin{aligned} &P[\hat{q}_j \geq \hat{q}_1 | \hat{q}_1 \geq q_1 - \varepsilon_1/2] \\ &\leq P[\hat{q}_j \geq q_j + \varepsilon_1/2 | \hat{q}_1 \geq q_1 - \varepsilon_1/2] \\ &\leq \delta_1/3. \end{aligned} \tag{5.9}$$

Let *Bad* be the number of labeling tasks which are not ε_1-optimal but empirically better than the best labeling task. We have that

$$\mathbb{E}[Bad | \hat{q}_1 \geq q_1 - \varepsilon/2] \leq n\varepsilon_1/3. \tag{5.10}$$

Next we apply the Markov inequality to obtain

$$P[Bad \geq n/2 | \hat{q}_1 \geq q_1 - \varepsilon_1/2] \leq \frac{n\delta_1/3}{n/2} = 2\delta_1/3. \tag{5.11}$$

Using the union bound gives us that the probability of failure is bounded by δ_1. □

In the special case where $\mathcal{T} = 1$, it has similar result to the best labeling task.

Lemma 5.2. *In Algorithm 5.2, given $\mathcal{T} = 1$, then it returns a labeling task which is ε-optimal, with label complexity $O((n/\varepsilon^2)\log(1/\delta))$.*

5.4.2 Selective repeated labeling strategies

In real applications, we can not only select one labeling task. Usually we need to identify m examples with high labeling expected quality. We can extend the idea of naive repeated labeling strategy for m labeling task selection.

Recall that we assume

$$q_1 > q_2 > \cdots > q_n. \tag{5.12}$$

The labeling tasks are independent for each other, and the labels provided by labelers do not depend on the history labeling. A labeling task a is defined to be (ε, m)-optimal, $\forall \varepsilon > 0, \forall m \in \{1, 2, \cdots, n\}$, if

$$q_a \geq q_m - \varepsilon. \tag{5.13}$$

We denote T_m as the set of labeling tasks that are (ε, m)-optimal and denote B as the set of labeling tasks that are not (ε, m)-optimal. We assume that there are at least m (ε, m)-optimal labeling tasks. Then generally $|T_m| \geq m$.

The algorithm returns a set of m labeling tasks which are (ε, m)-optimal. We simply modify Algorithm 5.1 and the **Best m Labeling** is described in Algorithm 5.3.

Algorithm 5.3. Best m Labeling

Input: $\varepsilon > 0$, $\delta > 0$, X.

1: **for** labeling task $x_i \in X$ **do**
2: Collect labels of this task for $\frac{2}{\varepsilon^2}\log(\frac{n}{\delta})$ times.
3: Let $\hat{q}_i^l$ be the empirical labeling quality of the labeling task x.
4: **end for**
5: Find $S \subset X$ such that $|S| = m$, and $\forall i \in S \, \forall j \in (X - S) : (\hat{q}_i \leq \hat{q}_j)$.
6: Output S.

Theorem 5.3. *In Algorithm 5.3, given $\varepsilon > 0$, $\delta > 0$, the label complexity is $O(\frac{n}{\varepsilon^2}\log(\frac{n}{\delta}))$.*

Proof. Recall that B is the set of labeling tasks in T_n that are not (ε, m)-optimal. From (5.12) and (5.13) we can relate T_m and B as follows:

$$\forall i \in T_m \forall j \in B : (q_i - q_j > \varepsilon). \tag{5.14}$$

Since $|S| = m$, a labeling task j in B can occur in S only if there is some labeling task i in T_m such that $\hat{q}_i \leq \hat{q}_j$. (5.14) implies that the latter event can only occur if $\hat{q}_i \leq q_i - \frac{\varepsilon}{2}$

or $\hat{q}_j \geq q_j + \frac{\varepsilon}{2}$. Switching to the probability, applying the union bound and Hoeffding's inequality, we get:

$$\begin{aligned}
&P(\exists j \in B : (j \in S)) \\
&\leq P(\exists i \in T_m : (\hat{q}_i \leq q_i - \frac{\varepsilon}{2})) + P(\exists j \in B : (\hat{q}_j \geq q_j + \frac{\varepsilon}{2})) \\
&\leq \sum_{i \in T_m} P(\hat{q}_i \leq q_i - \frac{\varepsilon}{2}) + \sum_{j \in B} P(\hat{q}_j \geq q_j + \frac{\varepsilon}{2}) \\
&\leq |T_m| \exp\left\{-\frac{\varepsilon^2}{2}\left\lceil \frac{2}{\varepsilon^2}\log(\frac{n}{\delta})\right\rceil\right\} + |B| \exp\left\{-\frac{\varepsilon^2}{2}\left\lceil \frac{2}{\varepsilon^2}\log(\frac{n}{\delta})\right\rceil\right\} \\
&\leq (|T_m| + |B|)(\frac{\delta}{n}) \leq \delta. \qquad (5.15)
\end{aligned}$$

The labels of each labeling task is collected for $\left\lceil \frac{2}{\varepsilon^2}\log(\frac{n}{\delta})\right\rceil$ times exactly, thus the label complexity of the **Best** m **Labeling** is $O(\frac{2}{\varepsilon^2}\log(\frac{n}{\delta}))$. □

The **Best** m **Labeling** is straightforward and practical. Each labeling task collects enough number of labels. Then we can clearly filter the good labeling tasks.

We provide another algorithm to improve **Best** m **Labeling** by eliminating multiple labeling tasks every phase based on their inferior empirical labeling qualities. At each round, we have a set of labeling tasks remaining at the beginning of this round and we use half of them to proceed to next round (except that m proceed to the last round). Each labeling task remaining needs to collect enough labels such that at least m (ε, m)-optimal labeling tasks are likely to survive elimination. We describe the algorithm in Algorithm 5.4. Specifically, the lth phase is associated with parameters ε_l and δ_l, and we ensure that with probability at least $1 - \delta_l$ the mth highest expected labeling quality in R_{l+1} is not lower than the mth highest expected label quality in R_l by more than ε_l.

Theorem 5.4. *In Algorithm 5.4, given $\varepsilon > 0$, $\delta > 0$, the label complexity is $O(\frac{n}{\varepsilon^2}\log(\frac{m}{\delta}))$.*

Proof. Without loss of generality, let us sort the labeling tasks in R_l in decreasing order of their true quality. Let x_i^l be the ith labeling task in the sorted list and let q_i^l be its true quality. We say a "mistake" is made in the lth phase if $q_m^l - q_m^{l+1} \leq \varepsilon_l$. Note that (1) $q_m^1 = q_m$, (2) $\sum_{t=1}^{\lceil \log(\frac{n}{m})\rceil} \varepsilon_l < \varepsilon$, and (3) $\sum_{t=1}^{\lceil \log(\frac{n}{m})\rceil} \delta_l < \delta$. In fact, it suffices to show that the probability of making a mistake in round l is at most δ_1: this would establish that $P(q_m - q_m^{\lceil \log(\frac{n}{m})\rceil+1} > \varepsilon) < \delta$. That is, the labeling tasks returned by **Improved Best** m **Labeling** is (ε, m)-optimal.

Let $A_l = \{x_i^l, i \in 1, 2, \cdots, m\}$: A_l contains the m labeling tasks from R_l with the highest true quality. E_1 denotes the event $\exists x \in A_l : (\hat{q}_x \leq q_x - \frac{\varepsilon_l}{2})$. By applying Hoeffding's inequality

Algorithm 5.4. Improved Best m Labeling

Input: $\varepsilon > 0$, $\delta > 0$, X.

$R_1 = X$.
$\varepsilon_1 \leftarrow \frac{\varepsilon}{4}$; $\delta_1 \leftarrow \frac{\delta}{2}$.
for $l = 1$ to $\lceil \log(\frac{n}{m}) \rceil$ **do**
  for all $x \in R_l$ **do**
    Collect labels of this task for $\frac{2}{\varepsilon_l^2}\log(\frac{3m}{\delta_l})$ times.
    Let $\hat{q}_i^l$ be the empirical labeling quality of the labeling task x.
  end for
  Find $R'_l \subset R_l$ such that $|R'_l| = \max\left(\left\lceil \frac{|R_l|}{2} \right\rceil, m\right)$, and $\forall i \in R_l \forall j \in (R_l - R'_l) : (\hat{q}_i \leq \hat{q}_j)$.
  $R_{l+1} \leftarrow R'_l$.
  $\varepsilon_{l+1} \leftarrow \frac{3}{4}\varepsilon_l$; $\delta_{l+1} \leftarrow \frac{1}{2}\delta_l$.
end for
Output $R_{\lceil \log(\frac{n}{m}) \rceil} + 1$.

and the union bound we obtain:

$$P(E_i) \leq m \exp\left\{ -\frac{\varepsilon_l^2}{2} \lceil \frac{2}{\varepsilon_l^2} \log(\frac{3m}{\delta_l}) \rceil \right\} \leq \frac{\delta_l}{3}. \tag{5.16}$$

Let $B_l = \{ j \in R_l, q_j < q_m^l - \varepsilon_l \}$: B_l is the set of labeling tasks that are not (ε_l, m)-optimal in R_l. We call a labeling task b in B_l "bad" if its empirical average equals or exceeds that of some labeling task in A_l. If E_1 does not occur, note that b can be bad only if $\hat{q}_b \geq q_b + \frac{\varepsilon_l}{2}$; a similar application of Hoeffding's inequality shows that the probability of the latter event is at most $\delta_l/(3m)$. Let the number of bad labeling tasks in B_l be Bad; counting bad labeling tasks as Boolean results of $|B_l|$ coin tosses, we obtain that $\mathbb{E}[Bad|\neg E_1]$ is at most $|B_l|\frac{\delta_l}{3m}$.

E_2 denotes the event that $Bad \geq |R_{l+1}| - m + 1$. A moment's reflection informs us that if E_1 does not occur, a mistake can be made on round l only if E_2 occurs. Markov's inequality establishes that

$$\begin{aligned} P(E_2|\neg E_1) &= P(Bad \geq (|R_{l+1}| - m + 1)|\neg E_1) \\ &\leq \frac{E[Bad|\neg E_1]}{|R_{l+1}| - m + 1} \leq \frac{|B_l|}{|R_{l+1}| - m + 1}\left(\frac{\delta_l}{3m}\right) \\ &\leq \frac{|R_l| - m}{|R_{l+1}| - m + 1}\left(\frac{\delta_l}{3m}\right) \leq \frac{2}{3}\delta_l. \end{aligned} \tag{5.17}$$

Now we have shown that for a mistake to occur on the lth phase, at least one of two events, E_1 and E_2, must occur; however, $P(E_1) + P(E_2|\neg E_1) \leq \delta_l$.

The last step comes from the observation that $|R_l| \leq 2|R_l + 1|$ (lines 8 and 9). According to (5.16) and (5.17), the total number of labels across all rounds (line 5) is $\sum_{l=1}^{\lceil \log(\frac{n}{m}) \rceil} \lceil \frac{2|R_l|}{\log(\frac{3m}{\delta_l})} \rceil$. We complete the proof. □

Therefore, the **Improved Best** m **Labeling** achieves the lower label complexity bound $O(\frac{n}{\varepsilon^2} \log(\frac{m}{\delta}))$ comparing with the **Best** m **Labeling** , while to launch Naive Repeated Labeling algorithm and the **Best** m **Labeling** are more convenient for real-world applications due to simplicity.

5.5 Experiments

In this section, we empirically evaluate the effectiveness of the proposed algorithms. We present experiments on 9 real-world data sets [89, 144].

5.5.1 Data sets

We consider in this study 9 data sets from different domains: bmg, kr-vs-kp, mushroom, sick, spambase, splice, thyroid, tic-tac-toe and waveform. These data sets are simplified for classification problems with a moderate number of examples. Then they allow the development of learning curves based on a large numbers of individual experiments. The data sets are described in Table 5.1.

Each data set was transformed as the setting of the binary classification. We converted the target to binary according to the data types. 30% of a data set are held out, in every run, as the test set, from which we calculate generalization performance. The rest is the train set from which we acquire unlabeled and labeled examples. To simulate noisy label acquisition, we first hide the labels of all examples for each data set. In an experiment, when a label is acquired, we assign the example's original label with probability $1 - v$ and the opposite value with probability v, where v is a parameter simulating the noise for labeling. We used different noises for different example labeling tasks.

We collected the results of performance on test set. After obtaining the labels, we added them to the training set to induce a classifier. We used a popular classification model, named J48, which is the implementation of C4.5 in WEKA [109, 138]. The classifier is evaluated on the test set (with the true labels). Each experiment is repeated 100 times with a random data partition, and we reported the average results.

Data Set	# Attributes	# Examples	Positive	Negative
bmg	41	2,417	547	1,840
kr-vs-kp	37	3,196	1,669	1,527
mushroom	22	8,124	4,208	3,916
sick	30	3,772	231	3,541
spambase	58	4,601	1,813	2,788
splice	61	3,190	1,535	1,655
thyroid	30	3,772	291	3,481
tic-tac-toe	10	958	332	626
waveform	41	5,000	1,692	3,308

Table 5.1 *The 9 data sets used in the experiments, including the numbers of attributes, and examples in each, and the split into positive and negative examples.*

5.5.2 Labeling strategies

In our experimental setup, we examined a set of basic design choices that can be varied to create different repeated labeling strategies. We mainly verified our proposed algorithms, the **Best** m **Labeling** and the **Improved Best** m **Labeling** . Specifically, all the designed selection strategies that we explored are as follows:

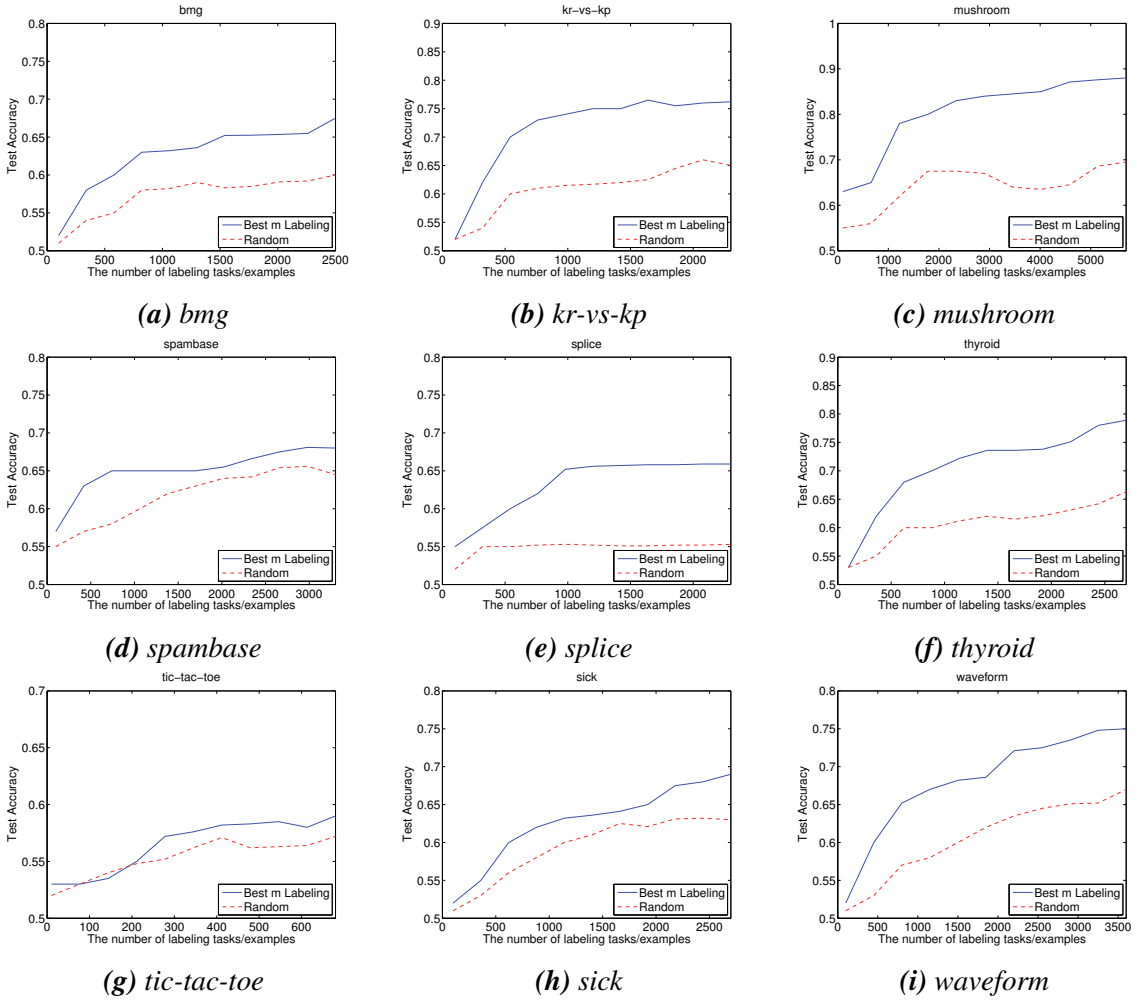
- **Best** m **Labeling** : Our Algorithm 5.3.
- **Improved Best** m **Labeling** : Our Algorithm 5.4.
- Random: It randomly selects m labeling tasks with their labels.
- Single Labeling: It selects the examples with a single label each.

5.5.3 Integration methods

In the repeated labeling setting, each example has multiple repeated labels. In order to investigate the labeling quality for each labeling task, we consider the case where for induction each repeated labeling task is assigned a single "integrated" label $\hat{y}_i$, inferred from the individual y_{ij}'s by majority voting or average ratings. In our experiments, we used the majority voting to generate the integration label for the sake of binary classification.

5.5.4 Results of the selective repeated labeling strategies

We first study different selection strategies for a large number of repeated labeling tasks on the performance of classification. We compared two algorithms: the **Best** m **Labeling** and



(a) bmg *(b) kr-vs-kp* *(c) mushroom*

(d) spambase *(e) splice* *(f) thyroid*

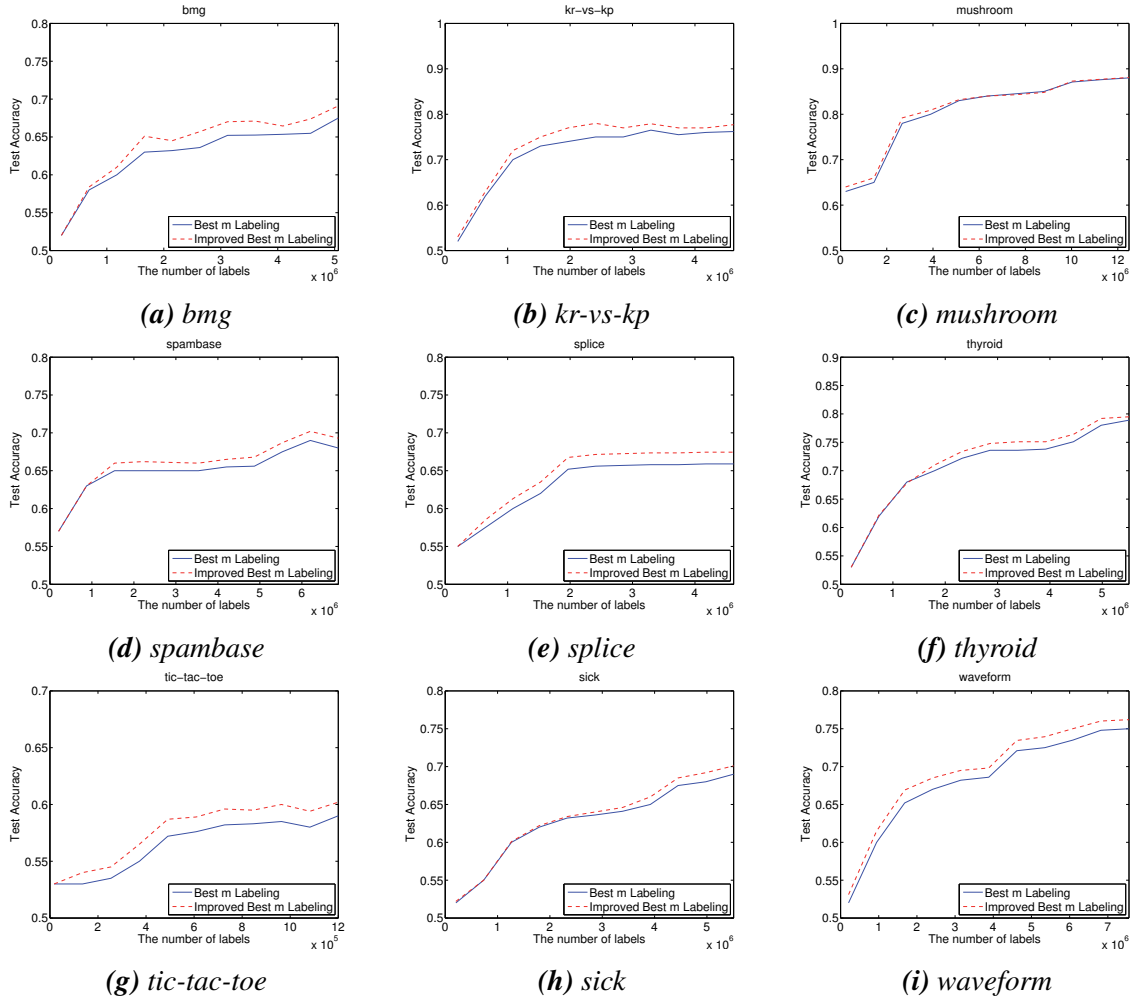
(g) tic-tac-toe *(h) sick* *(i) waveform*

Fig. 5.2 *A comparison of test accuracy between the* ***Best*** *m* ***Labeling*** *and the Random on different data sets.*

Random, on each data set. We set $\varepsilon = 0.1$ and $\delta = 0.1$ for **Best** m **Labeling** . Each noise v for a labeling task is drawn from $\{0.2, 0.3, 0.4\}$ equally. We collected average results of accuracy for different data sets. Different data sets have different numbers of labeling tasks, i.e different values of n. Thus the numbers of labeling required for different data sets are different, defined as follows:

$$\frac{2}{\varepsilon^2} \log\left(\frac{n}{\delta}\right). \tag{5.18}$$

The results of accuracy on different data sets are shown in Figure 5.2. We can see that **Best** m **Labeling** outperforms Random significantly on every data set. This is because the labeling tasks returned by Random strategy contained more examples with high noisy labels than the **Best** m **Labeling** . However, **Best** m **Labeling** addresses this problem and actively selects the examples with low noisy labels. Therefore carefully selecting the labeling tasks can improve the performance of classification.



(a) *bmg* ***(b)*** *kr-vs-kp* ***(c)*** *mushroom*

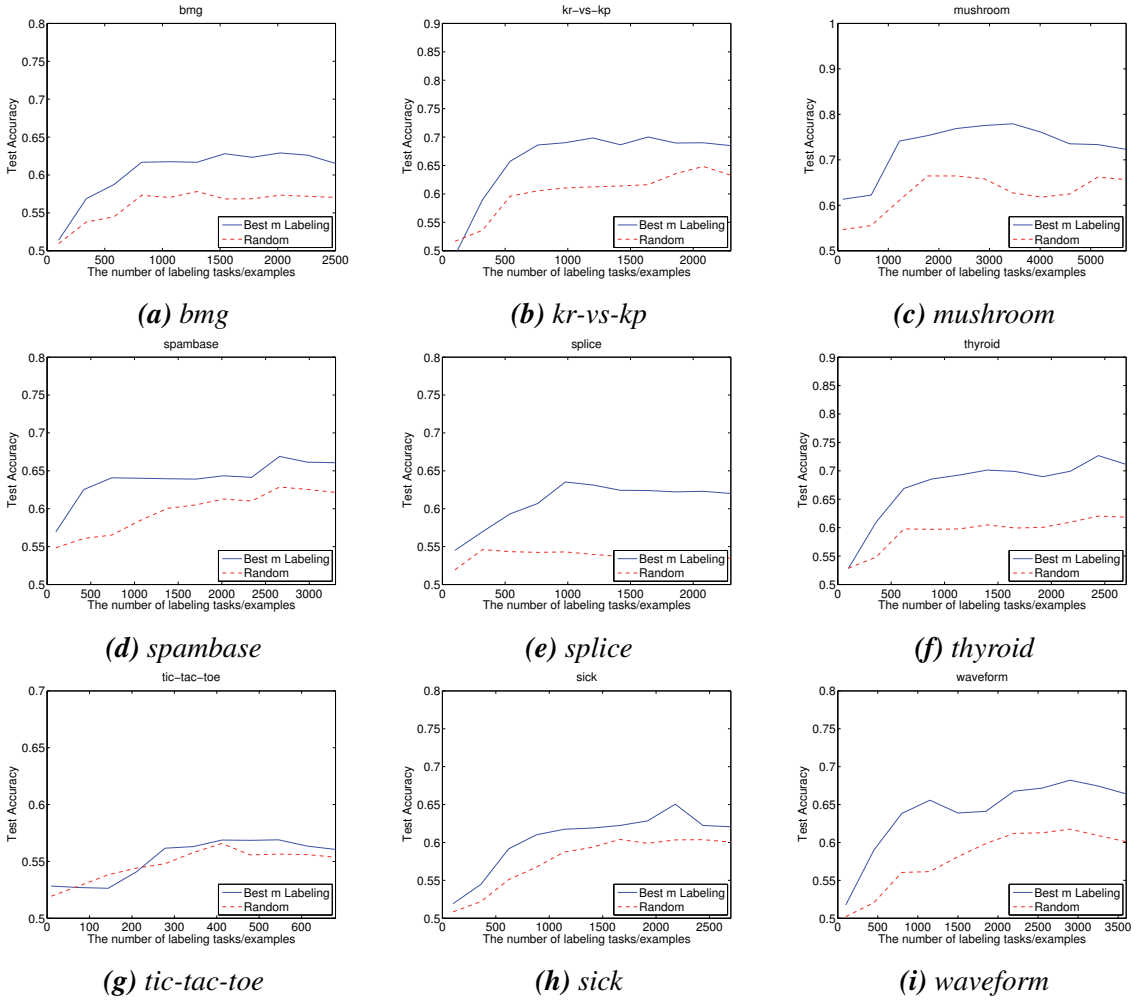
(d) *spambase* ***(e)*** *splice* ***(f)*** *thyroid*

(g) *tic-tac-toe* ***(h)*** *sick* ***(i)*** *waveform*

Fig. 5.3 *A comparison of test accuracy between the* ***Best*** *m* ***Labeling*** *and the* ***Improved Best*** *m* ***Labeling*** *on different data sets.*

5.5.5 Comparison between the selective repeated labeling and the single labeling

We then verify the effectiveness of the selective repeated labeling comparing to the single labeling. The selective repeated labeling often makes queries for each example more than once. Thus the number of labels in the selective repeated labeling is larger than the number of labels in the single labeling. In the selective repeated labeling, for each data set, we used the **Best** *m* **Labeling** with a fixed m, which is the half size of training data set. Thus we used half of training set. We set $\varepsilon = 0.1$ and $\delta = 0.1$ and the noise v for a labeling task was drawn from $\{0.2, 0.3, 0.4\}$ equally. In the single labeling, for each data set, we used all the training data set. A single label for each example is generated by a noise, drawn from $\{0.2, 0.3, 0.4\}$ equally.



(a) *bmg* ***(b)*** *kr-vs-kp* ***(c)*** *mushroom*

(d) *spambase* ***(e)*** *splice* ***(f)*** *thyroid*

(g) *tic-tac-toe* ***(h)*** *sick* ***(i)*** *waveform*

Fig. 5.4 *The test accuracy of the* ***Best*** *m* ***Labeling*** *and the Random on different data sets.*

In Table 5.2, it is shown that the performance of the **Best** *m* **Labeling** is better than the Single Labeling for most of data sets. It verified that the selective repeated labeling works better than single labeling in this noise setting. Because the single labeling contains more examples with high noisy labels than the selective repeated labeling.

5.5.6 Comparison between the Best *m* Labeling and the Improved Best *m* Labeling

We compared the Best *m* Labeling with Improved Best *m* Labeling. In reality, the Best *m* Labeling algorithm is more general than Improved Best *m* Labeling algorithm because its implementation is much easier. For each data set, we set $\varepsilon = 0.1$ and $\delta = 0.1$. We evaluated the performance based on different numbers of labels. The results are shown in Figure 5.3. We observed that the **Improved Best** *m* **Labeling** outperforms the **Best** *m* **Labeling** overall,

Data Set	**Best *m* Labeling**	Single Labeling
bmg	0.65	0.61
kr-vs-kp	0.76	0.65
mushroom	0.83	0.64
sick	0.64	0.63
spambase	0.65	0.64
splice	0.66	0.55
thyroid	0.73	0.66
tic-tac-toe	0.57	0.57
waveform	0.69	0.67

Table 5.2 *The test accuracy of the* ***Best*** *m* ***Labeling*** *and the Single Labeling.*

which is consistent with our previous analysis about this two methods: the **Improved Best** *m* **Labeling** achieves better performance than **Best** *m* **Labeling** using less labels. Under the same number of labelers, the **Improved Best** *m* **Labeling** may select more examples with high confident labels than **Best** *m* **Labeling** .

5.5.7 Study on the size of selected labeling tasks

In addition, we studied the performance for different values of m. In order to focus on the m, we used the same noise for all the labeling tasks and we set the noise drawn from $\{0.4, 0.45\}$ equally. We used the **Best** *m* **Labeling** and set $\varepsilon = 0.1$ and $\delta = 0.1$. The results is shown in Figure 5.4. It is shown that examples with high noisy labels decreases the performance of the selective repeated algorithms. At the beginning, the **Best** *m* **Labeling** selected the examples with good quality. However, when the value of m grew, it became easy that the examples with high noisy labels were collected, which diminished performance.

5.6 Conclusion

In this work, we formalized the repeated labeling as a bandit model. We studied the problem that how to select a proportion of labeling tasks from a pool of data, in which the labels of each example are repeatedly acquired from multiple labelers. We presented a simple repeated labeling algorithm and proposed two selective repeated labeling algorithms: **Best** *m* **Labeling** and **Improved Best** *m* **Labeling** . We demonstrated their effectiveness and efficiency both in the theoretical results and experiments.

Chapter 6

Active learning via bandits

In this work, we introduce a new perspective about active learning by formalizing it in the bandit framework. Active learning aims to learn a classifier by actively acquiring the data points, whose labels are hidden initially and incur querying cost. The multi-armed bandit problem is a framework that can adapt the decision in sequence based on rewards that have been observed so far. Inspired by the multi-armed bandits, we consider active learning as to identify the best hypothesis in an optimal candidate set of hypotheses by involving querying the labels of points as less as possible. Our algorithms are proposed to maintain the candidate set of hypotheses using the error or the corresponding general lower and upper error bounds to help select or eliminate hypotheses. To maintain the candidate set of hypotheses, in the realizable PAC setting, we directly use the error. In the agnostic setting, we use the lower and upper error bounds of the hypotheses. To label the data points, we use the uncertainty strategy based on the candidate set of hypotheses. We show that the proposed algorithms for active learning via bandits work efficiently over the stand sample complexity of supervised learning. We also analyze the correctness and label complexities of our algorithms.

6.1 Introduction

Active learning is popular in machine learning. There is always a pool of data and we do not use all of them for training a learner because there is a budget for labeling or labeling all data is impossible. The active learner is to selectively pay for the label of any example in the pool in order to obtain a good classifier with significantly fewer labels by making the query. It is significantly interesting to design strategies to minimize the number of labeled examples in the setting where the unlabeled data are easy to obtain but labeling is expensive.

The theory of active learning has been well studied. Previously, Cohn et al. [38] proposed the CAL algorithm. This selective sampling algorithm is based on the uncertainty sampling

in the noise-free case, named as the realizable PAC setting. In this setting, Dasgupta [41] studied a greedy strategy for active learning. Then Dasgupta [42] used the splitting index to characterize the label complexity of active learning algorithms. The first algorithm designed for the agnostic setting is A^2 [17]. Then Dasgupta et al. [43] developed an alternative of A^2. Later Hanneke [69, 70] theoretically analyzed the A^2 algorithm regarding the disagreement coefficient. These algorithms can achieve an exponential improvement for the usual sample complexity of supervised learning [18].

In this work, we study active learning from a bandit perspective. Active learning often involves interaction with an expert or oracle and decides which example to label in sequence. This process can be casted as a multi-armed bandit problem, which is a sequential decision problem. In multi-armed bandit, there are a number of alternative arms, each of which is associated with a stochastic reward and the corresponding probability distribution is initially unknown. At each time step, the player pulls one out of K given stochastic arms and obtains a reward drawn at random according to the distribution of the chosen arm. We try these arms in some order, which may depend on the sequence of rewards that has been observed so far. A common objective of exploration in this context is to find a near-optimal arm whose expected reward is with a small error ε from the expected reward of the best arm with a high probability [92]. With this regard, active learning can be understood in the bandit framework, i.e. active learning is a process that we select the hypothesis depending on the sequential data and labeling. We treat a set of hypotheses as a set of arms and consider the error of a hypothesis as the reward of an arm. Similar to the problem of exploration in stochastic multi-armed bandits, our objective is to select the best hypothesis which has the lowest expected error.

The exploration problem for multi-armed bandits has received much attention recently in the stochastic setting [12, 28, 46, 63]. Even-Dar et al. [46] proposed the Successive Elimination strategy for ε-optimal arm identification. Audibert et al. [12] addressed the fixed budget setting and presented the algorithms UCB-E and Successive Rejects for identifying the best-arm. Recently, some extensions of exploration of multi-armed bandits to subset selection have been studied [29, 83, 84].

Inspired by the bandit studies, we propose three active learning algorithms for the realizable PAC and agnostic settings. The active learning process is re-explained as that there are a set of hypotheses and the objective is to return the best hypothesis by making the queries as less as possible. The proposed algorithms are guaranteed to find a hypothesis with the lowest error with a high probability. We resolve two issues. One issue is how to select the best hypothesis. Regarding the realizable PAC setting, which assumes there always exists at least one correct hypothesis that can correctly classify all the examples, we exploit the error

of this hypothesis. For the agnostic setting, which assumes even the target hypothesis cannot correctly classify all the data points, we consider the error bounds of different hypotheses. We maintain a candidate set of hypotheses by selecting good hypotheses having low errors and eliminating bad hypotheses having high errors. Another issue is which data point we should label. We query the labels of points according to the disagreement for the candidate set of hypotheses.

Our main contributions are as follows. Firstly, we formalize active learning as a bandit problem. Secondly, we introduce two algorithms ALB-1 and ALB-2 for active learning in the realizable PAC setting. The difference between ALB-1 and ALB-2 is their uncertainty strategies. Thirdly, we propose ALB-LUB for active learning in the agnostic setting. Fourthly, we theoretically show the correctness of our algorithms and analyze their label complexities.

This work is organized as follows. Section 6.2 introduces the related work about active learning and bandits. Algorithms for the realizable PAC and the agnostic settings are presented in Section 6.3. The theoretical analyses including correctness and label complexities are provided in Section 6.4. The experimental results are shown in Section 6.5. Section 6.6 provides the proof. Section 6.7 concludes this work.

6.2 Related work

Active learning has been comprehensively studied from theories, algorithms to applications. Existing traditional active learning algorithms usually are grouped into three categories: (1) uncertainty sampling [90, 129], which focuses on selecting the instance that a classifier is most uncertain about; (2) query by committee [61, 96], in which the most informative instance is the one that has the largest disagreement on a committee of classifiers; and (3) expected error reduction [113], which aims to query instance that minimizes the expected error of the classifier. Most existing studies have focused on a single domain and have assumed that an omniscient oracle always exists for providing an accurate label for each query.

For the theory of active learning, we focus on two settings: the noise-free and noise settings. The early work Query by Committee is based on the assumption that there is a perfect classifier in the known set [60]. The CAL algorithm [38] is a selective sampling scheme based on the uncertainty sampling. Both methods are based on the noise-free setting and query the label of points about which the learner is least sure. In this setting, active learning was further studied in [41, 42]. Dasgupta [41] analyzed the generalization properties of active learning algorithms and provided the upper and lower bounds on the number of labels needed. He considered an average-case model for the distribution over target

hypotheses, in which the average-case model is often approximated by Bayesian methods, and showed that a target hypothesis chosen uniformly from $\mathcal{H}$ can query only $O(\log n)$ labels in expectation. His work started the theoretical study of the generalization properties of active learning algorithms. Then Dasgupta [42] studied the label complexity of active learning and exploited the splitting index to measure this complexity, in which the splitting index captures the relevant local geometry of $\mathcal{H}$ in the surrounding region of the target hypothesis h^*, at scale ε. This complexity coarsely characterizes the sample complexity of active learning and shows a general active learner has a label complexity $O((d/\rho)\log^2(1/\varepsilon))$.

In the noisy setting, Balcan et al. [17] proposed the A^2 algorithm. A^2 contains an important subroutine for discarding at least half of the current region of uncertainty by rejection sampling. Kaariainen [79] developed an active learner and analyzed a lower bound on the number of label required in order to achieve a particular generalization error involving the inherent noise rate, which matched the lower bound of A^2. Hanneke [69]. provided a general bound on the number of label requests for A^2, which is applicable to any concept space and distribution in terms of the disagreement coefficient. The disagreement coefficient was further studied by [62, 135]. The disagreement coefficient was generalized to general loss functions and was used in the analysis of label complexity [22].

Traditional multi-armed bandit algorithms select the optimal arm in sequence for maximizing the cumulative payoffs [31]. In the stochastic setting, the payoff of each arm is i.i.d. drawn from an unknown distribution. The upper confidence bound (UCB) strategy has been used to balance the exploration-exploitation trade-off [11, 14], in which an upper bound estimate is constructed on the mean of each arm at a fixed level. However, unlike maximizing the cumulative payoffs, the exploration problem in stochastic bandits is to find a near-optimal arm whose expected reward is with ε from the expected reward of best arm with a high probability [12, 28, 46, 63]. Even-Dar et al. [46] proposed the Successive Elimination strategy for ε-optimal arm identification problems. The inferior arm is eliminated when its expected upper estimate is not good enough. Audibert et al. [12] addressed the fixed budget setting and presented the algorithms UCB-E and Successive Rejects for the best-arm identification. Recently, some extensions of exploration of multi-armed bandits to subset selection have been studied [29, 83, 84]. Bubeck et al. [29] proposed a variant of Successive Rejects algorithm that outputs a set of m-best arms under a fixed budget of pulls with high probability. Kaufmann and Kalyanakrishnan [83] presented an explore-m algorithm under the PAC setting. Kaufmann and Kalyanakrishnan [84] drove a new bounds using KL-divergence-based confidence intervals under the PAC and the fixed-budget setting.

To date, there is few work considering the active learning in the bandit framework. Recently, there is one work considering the trade-off between exploration-exploitation trade-

off by improving the confidence bound [64]. They proposed a heuristic method using confidence bound and their method was designed for specific settings and cannot be applied to the general setting widely used in validating active learning algorithms. In addition, no theoretical bound is available for them to guarantee their generalizations.

6.3 Methodology

Let X be a point space and $Y = \{-1, 1\}$ be the set of possible labels. Let $\mathcal{H}$ be the hypothesis class, i.e. a set of functions mapping from X to Y. Let $\mathcal{D}$ be the distribution over points in X, and that the data points are labeled by a possibly randomized oracle $\mathcal{O}$. The label y of x is obtained after querying. Active learner decides which point to query and returns a hypothesis according to the collected data.

We study active learning in the bandit framework. In this framework, we define a hypothesis as an arm and the error of a hypothesis as the reward of an arm. The error of a hypothesis $h : X \to Y$ is

$$\text{err}(h) := \Pr(h(x) \neq y). \tag{6.1}$$

Then the empirical error of h with respect to a labeled sample L is defined as $\text{err}_L(h) = \frac{1}{|L|}\sum_{(x,y)\in L} \mathbb{1}[h(x) \neq y]$, representing the fraction of points in L on which h makes mistakes.

Let $h^* = \arg\min\{\text{err}(h) : h \in \mathcal{H}\}$ be the hypothesis associated with the minimum error in $\mathcal{H}$. Simply we assume that the minimum error always exists. The goal of the learner is to obtain a hypothesis $h \in \mathcal{H}$ with error $\text{err}(h)$ not much more than $\text{err}(h^*)$.

For a particular hypothesis h and the corresponding sequence $L = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), \cdots\}$, the reward of h is defined as

$$g = 1 - \text{err}_L(h). \tag{6.2}$$

We can simply consider the loss $l = \text{err}_L(h)$ in the rest part of this work.

To consider the active learning process, we have a set of hypotheses at the beginning and then return the optimal hypothesis. From the bandit perspective, the process of active learning can be considered as that we have a set of arms and optimally select the best arm. Thus active learning via bandits algorithm is designed to eliminate or select hypotheses through querying the labels of points. In particular, given a set of hypotheses $\mathcal{H}$, we need to select a subset of hypotheses $\mathcal{H}'$, where for any $h \in \mathcal{H}'$ its $\text{err}(h)$ is not much more than $\text{err}(h^*)$.

Querying the label of a point depends on the existing of the disagreement within the set of hypotheses. On one hand, if there is no disagreement in the candidate set of hypotheses,

then the label of the point cannot be used to distinguish between any hypothesis in this candidate set. On the other hand, if hypotheses in the candidate set are not good enough, then the labeling of the point cannot work efficiently for distinguishing the hypotheses. Thus carefully maintaining the candidate set of hypotheses is very important and thus the optimal hypothesis h^* should be always included in.

To carefully maintain the candidate set of hypotheses, we need to resolve two issues: (1) how to decide which point to label, and (2) how to decide which hypothesis to eliminate or select in the candidate set of hypotheses.

We introduce three algorithms in the bandit framework. We follow two settings: the first is the realizable PAC setting [38], where it is assumed that there exists a perfect hypothesis $h^* \in \mathcal{H}$ that correctly labels any point in the data set, i.e. $\Pr(h(x) = y) = 1$; and the second is the agnostic setting [17], where it is not assumed that there exists a perfect hypothesis in $\mathcal{H}$ that correctly labels any points in the data set. The agnostic setting is more practical than the realizable PAC setting [69].

6.3.1 Realizable PAC setting

Inspired by the uncertainty sampling, the decision of whether or not to query the label of x is made based on the disagreement among hypotheses in the candidate set. After querying the label of a point, the candidate set of hypotheses will be updated according to error, i.e., empirical $\mathrm{err}_L(h)$ when receiving (x,y).

According to the assumption of this setting, the selected hypotheses should at least contain h^*. That means given any labeled point x we have $h^*(x) = y$ and $h^* \in \mathcal{H}$. The strategy for selecting hypotheses is to maintain the hypotheses whose errors are not larger than $\mathrm{err}(h^*) = 0$. Our first simple algorithm is shown in Algorithm 6.1.

ALB-1 can be regarded as a stream-based active learning. Its main drawback is that it cannot eliminate incorrect hypotheses as many as possible. To address this problem, we develop an improved algorithm ALB-2 that can be used in the pool-based active learning, shown in Algorithm 6.2. Considering that we have a pool of data, it is natural to query a data point that can eliminate the incorrect hypotheses mostly. Thus we actively query the data point which can incur that half of hypotheses in the candidate set disagree another half. In addition, the label of a point is synthesized when there is an agreement among the hypotheses in the candidate set.

ALB-2 is a pool-based active learning. Both simple algorithms are slight variants of CAL [38]. The difference between these two bandit active learning algorithms and CAL is we assume that $\mathcal{H}$ is finite to simplify the subsequent analysis. This can be extended to infinite class using covering numbers.

Algorithm 6.1. ALB-1

Input: $\mathcal{H}_0 = \mathcal{H}$.
for $t = 1, 2, \cdots, n$ **do**
  Receive an unlabeled point x.
  if there exist $h_i(x) \neq h_j(x)$ for any hypothesis $h \in \mathcal{H}_{t-1}$ **then**
    Query y and set $Z_t := Z_{t-1} \cup \{(x, y)\}$.
  else
    Get the synthetic label $\bar{y}$ and set $Z_t := Z_{t-1} \cup \{(x, \bar{y})\}$.
  end if
  if Queried **then**
    For any h in $\mathcal{H}_{t-1}$ calculate $\mathrm{err}_{Z_t}(h)$ and sort the values.
    Set $\mathcal{H}_t = \{h \in \mathcal{H}_{t-1} : \mathrm{err}_{Z_t}(h) = 0\}$.
  end if
end for
return any $h \in \mathcal{H}_t$.

Algorithm 6.2. ALB-2

Input: $\mathcal{H}_0 = \mathcal{H}$.
for $t = 1, 2, \cdots, n$ **do**
  Observe a pool of data points $\{x_{1,t}, x_{2,t}, \cdots, x_{n_t,t}\}$.
  Calculate the disagreement value for each point based on $\mathcal{H}_t$ and sort the values.
  Select the point x where disagreement value is $1/2$, i.e. $|\{h \in \mathcal{H}_{t-1} : h(x) = 1\}| = |\{h \in \mathcal{H}_{t-1} : h(x) = -1\}|$.
  then Query y and set $Z_t := Z_{t-1} \cup \{(x, y)\}$.
  if Queried **then**
    For any h in $\mathcal{H}_{t-1}$ calculate $\mathrm{err}_{Z_t}(h)$ and sort the values.
    Set $\mathcal{H}_t = \{h \in \mathcal{H}_{t-1} : \mathrm{err}_{Z_t}(h) = 0\}$.
  end if
end for
return any $h \in \mathcal{H}_t$.

6.3.2 Agnostic setting

In this setting, the choice of whether or not to label x is made based on whether there is disagreement about how to label x among hypotheses in the candidate set of hypotheses. However, we cannot select or eliminate the hypotheses from the candidate set using $\mathrm{err}_L(h) = 0$. Because there is no perfect hypothesis in the candidate set and even the best hypothesis h^* cannot correctly classify all the points. Therefore the objective of active learning is to find the ε-optimal hypothesis which has an error at most ε with querying the labels as less as possible.

Given the candidate set of hypotheses and the collected data points, the error of each hypothesis is completely determined by the points and hypothesis. We assume an indexing

of the hypotheses such that

$$\text{err}(h_1) \leq \text{err}(h_2) \leq \cdots \leq \text{err}(h_K).$$

We assume that there exists a subset of m ε-optimal hypotheses. The assumption that there exists only one best hypothesis can be viewed as a special case where $m = 1$. It is natural that we need to find an optimal set of hypotheses which are ε-optimal at the end of learning. Specially, for some fixed tolerance $\varepsilon \in (0,1)$, let $S^*_{m,\varepsilon}$ be the set of all (ε, m)-optimal hypotheses. That is for any hypothesis h in the set such that $\forall \varepsilon \in (0,1)$ and $\forall 1 \leq m < K$, $\text{err}(h) \leq \text{err}(h_m) + \varepsilon$. Thus we need to find a set of m ε-optimal hypotheses $S^*_m = \{h_1, h_2, \cdots, h_m\}$ and this set is necessarily a subset of $S^*_{m,\varepsilon}$.

Let S denote the set of selected hypotheses, and let R denote the candidate set of hypotheses. Thus any hypothesis h in S is (ε, m)-optimal and there are at least m (ε, m)-optimal hypotheses existing in $\mathcal{H}$. Generally we assume that the size of the set of all (ε, m)-optimal hypotheses is larger than m.

As querying the labels of the points, each hypothesis corresponds to an error. Different hypotheses have different error estimates based on the collected data points. Given a hypothesis h_i and a sample of points L drawn i.i.d. from $\mathcal{D}$, with probability at least $1 - \delta$ the following bounds hold:

$$LB(L, h_i, \delta) \leq \text{err}(h_i) \leq UB(L, h_i, \delta), \tag{6.3}$$

where $LB(L, h_i, \delta)$ is the lower bound on the error of hypothesis h_i and $UB(L, h_i, \delta)$ is the upper bound on the error of hypothesis h_i. These bounds can generally indicate VC bound [131] and the Occam Razor bound [25].

At each round t, we use generic confidence intervals for each hypothesis as follows: $[LB_h, UB_h]$, where LB_h and UB_h are the lower and upper bounds on the error of hypothesis h respectively according to (6.3). As querying the label of the point, we can consider using a hypothesis as pulling an arm. That is when a point is labeled, each hypothesis in the candidate set is pulled once for calculating the error. Then we let $B(t)$ be the set of m hypotheses with the highest empirical errors from the candidate set and $B^c(t)$ be the set of the rest hypotheses. We focus on two critical hypothesis on current candidate set $B(t) \cup B^c(t)$, that is

$$LB_t = \arg\min_{i \in B^c(t)} LB_i \text{ and } UB_t = \arg\min_{j \in B(t)} UB_j. \tag{6.4}$$

Inspired by the arm selection that involves the confidence bounds in bandit problem [31], the decision is that selects the empirical best hypothesis if its upper bound (UB) is smaller

Algorithm 6.3. ALB-LUB

Input: $\varepsilon \geq 0$, m, $R = \mathcal{H}$.
for $t = 1, 2, \cdots, n$ **do**
  Receive an unlabeled point x.
  if there exist $h_i(x) \neq h_j(x)$ for any hypothesis $h \in R$ **then**
    Query y and set $Z_t := Z_{t-1} \cup \{(x, y)\}$.
  else
    Get the synthetic label $\bar{y}$ and set $Z_t := Z_{t-1} \cup \{(x, \bar{y})\}$.
  end if
  if Queried **then**
    Update LB, UB and empirical error for each hypothesis $h \in R$.
    Calculate the empirical best $m - |S|$ hypotheses $B(t)$ and $B^c(t) = R \setminus B(t)$.
    Calculate LB_t, UB_t, $h_b \leftarrow$ empirical best in R, $h_w \leftarrow$ empirical worst in R.
    if $\Delta(h_b, h_{LB_t}) < \varepsilon$ or $\Delta(h_{UB_t}, h_w) < \varepsilon$ **then**
      $h = \arg\max_{h_b, h_w} (\Delta(h_b, h_{LB_t}) \mathbb{1}(\Delta(h_b, h_{LB_t}) < \varepsilon); \Delta(h_{UB_t}, h_w) \mathbb{1}(\Delta(h_{UB_t}, h_w) < \varepsilon))$
      Remove the hypothesis h from R: $R \leftarrow R \setminus \{h\}$
      If $h = h_b$ then select this h_b: $S \leftarrow S \cup \{h_b\}$
    end if
  end if
end for
return any $h \in S$.

than the lower bounds (LBs) of all hypotheses in $B^c(t)$ or eliminates the empirical worst hypothesis if its lower bound is larger than the upper bounds of all hypotheses in $B(t)$. We define that

$$\Delta(h_a, h_b) = UB_a - LB_b, \tag{6.5}$$

which indicates the separation between the upper error bound of hypothesis h_a and the lower error bound of hypothesis h_b. Then the conditions of decision criterion are rewritten as

$$\Delta(h_b, h_{LB_t}) < \varepsilon \text{ or } \Delta(h_{UB_t}, h_w) < \varepsilon. \tag{6.6}$$

The first condition is for selecting the empirical best hypothesis and the second condition is for eliminating the empirical worst hypothesis. We show our algorithm in Algorithm 6.3.

6.4 Theoretical analysis

We present the theoretical details about correctness and label complexity for our three algorithms by using the disagreement coefficient. The label complexity is defined as the number of labels required by an active learner to achieve an error no more than ε. We

follow some basic concepts in [69, 71] and introduce the refined notions of the region of disagreement and the disagreement coefficient.

Definition 6.1. *For a random observation $x \in X$, the disagreement metric ρ on $\mathcal{H}$ is defined by*

$$\rho(h,h') := \Pr(h(x) \neq h'(x)).$$

Let $\tilde{B}(h,r) := \{h' \in \mathcal{H} : \rho(h,h') \leq r\}$ denote the ball centered at $h \in \mathcal{H}$ with radius $r \geq 0$.

Definition 6.2. *The region of disagreement $\tilde{R}(h,r)$ of radius r around a hypothesis $h \in \mathcal{H}$ in the disagreement metric space $(\mathcal{H},\rho)$ is*

$$\tilde{R}(h,r) := \{x \in X : \exists h' \in \tilde{B}(h,r) \text{ such that } h(x) \neq h'(x)\}.$$

It is a set of unlabeled examples x for which there exists a hypothesis h' at distance at most r from h (under ρ) that disagrees with h on x.

Definition 6.3. *The disagreement coefficient $\theta(h,\mathcal{H},\mathcal{D})$ for a hypothesis $h \in \mathcal{H}$ in the disagreement metric space $(\mathcal{H},\rho)$ is*

$$\theta(h,\mathcal{H},\mathcal{D}) := \sup\left\{\frac{\Pr(x \in \tilde{R}(h,r))}{r} : r > 0\right\}.$$

We simply write θ to mean $\theta(h^*,\mathcal{H},\mathcal{D})$.

We firstly provide the correctness and label complexity analysis for ALB-1 and ALB-2.

Theorem 6.1. *Assume there exists h^* for all the x satisfying $h^*(x) = y$. The algorithm ALB-1 receives the label y or synthetic $\bar{y}$ under its different conditions respectively. The synthetic $\bar{y}$ generated by ALB-1 is always the true label.*

According to the standard PAC learning, given a random sample of n examples Z_n, then for any $h \in \mathcal{H}$ consistent with Z_n, with the probability at least $1-\delta$, has error at most $O(1/n(\log|\mathcal{H}| + \log(1/\delta)))$ [97].

Theorem 6.2. *Let $\mathcal{H}$ be a finite set of functions mapping from X to Y. For $\delta \in (0,1)$, with probability at least $1-\delta$, the label complexity of ALB-1 is*

$$O(\theta \cdot (\log|\mathcal{H}| + \log(1/\delta))\log n). \tag{6.7}$$

Theorem 6.2 establishes the label complexity in terms of the disagreement coefficient and hypothesis class $\mathcal{H}$. If we substitute n for which the error in PAC bound is at most ε, we have the similar results in [69].

Similar to ALB-1, we analyze the correctness and the label complexity of ALB-2.

Theorem 6.3. *Assume there exists h^* for all the x satisfying $h^*(x) = y$. The algorithm ALB-2 queries the label y when the disagreement value is $1/2$ or synthesize $\bar{y}$ when there is an agreement. The synthetic $\bar{y}$ generated by ALB-2 is always the true label.*

Theorem 6.4. *Let $\mathcal{H}$ be a finite set of functions mapping from X to Y. For $\delta \in (0,1)$, with probability at least $1-\delta$, the label complexity of ALB-2 is*

$$O(\theta \cdot (\log\log|\mathcal{H}| + \log(1/\delta))\log n). \tag{6.8}$$

Theorem 6.4 has a similar result with Theorem 6.2. However, the label complexity of ALB-2 is improved in the pool-based setting by replacing $\log\log|\mathcal{H}|$ with $\log|\mathcal{H}|$.

We now provide the correctness and the label complexity analysis about ALB-LUB. We define the sample complexity $m_0(\varepsilon,\delta,V)$ as the minimum number of samples m_0 such that for all L, we have $|UB(h) - LB(h)| \leq \varepsilon$ for all $h \in \mathcal{H}$.

Theorem 6.5. *[7] Let $\mathcal{H}$ be a set of functions from X to $\{-1,1\}$ with finite VC-dimension $V \geq 1$. Let $\mathcal{D}$ be an arbitrary, but fixed probability distribution over $X \times \{-1,1\}$. For any $\varepsilon,\delta > 0$, if we draw a sample from $\mathcal{D}$ of size*

$$m_0(\varepsilon,\delta,V) = \frac{64}{\varepsilon^2}\left(2V\log\left(\frac{12}{\varepsilon}\right) + \log\left(\frac{4}{\delta}\right)\right), \tag{6.9}$$

then with probability at least $1-\delta$, we have $|\mathrm{err}(h) - \mathrm{err}_L(h)| \leq \varepsilon$ for all $h \in \mathcal{H}$.

In the following, just for simplicity we rewrite $\mathrm{err}(h_a)$ as p_a, $LB_{h_a}(t)$ as $LB_a(t)$, and $UB_{h_a}(t)$ as $UB_a(t)$.

Theorem 6.6. *The ALB-LUB is correct on the event*

$$W = \bigcap_{t\in N}\bigcap_{a\in S_m^*}(LB_a(t) < p_a) \bigcap_{b\in(S_m^*)^c}(UB_b(t) > p_b), \tag{6.10}$$

where UB and LB denote the genetic upper and lower error bounds respectively used by ALB-LUB algorithm.

Theorem 6.7. *Let $\mathcal{H}$ be a finite set of functions mapping from X to Y. For any $\varepsilon \in (0,1)$ and $\delta \in (0,1)$, with probability at least $1-\delta$, the label complexity of ALB-LUB is*

$$O\left(\theta \cdot \frac{|\mathcal{H}|}{\varepsilon}\left(\log|\mathcal{H}|\log\left(\frac{1}{\varepsilon}\right)+\log\left(\frac{1}{\delta}\right)\right)\log n\right). \tag{6.11}$$

The bound in Theorem 6.7 implies the label complexity of ALB-LUB is determined by the disagreement coefficient and hypothesis class $\mathcal{H}$. Theorem 6.7 gives a good estimate of the number of queries made by ALB-LUB. The label complexity in Theorem 6.7 validates the fact that the number of labels required increases as the size of training sample becomes large. It also increases as the cardinality of the hypothesis class increases. However, it decreases as the error ε increases.

6.5 Experiments

We provide the experimental analysis based on a few simple cases to show the efficiency and the label complexity improvements of our proposed algorithms. In the following, we firstly show the experimental results of ALB-1 and ALB-2 because they are simple and straightforward. Then we provide the experimental results of ALB-LUB.

6.5.1 Experimental results of realizable setting

We used the same data set and the same target hypothesis with the experiment of ALB-LUB. However, in this experiment, it is constructed as the noise-free setting because of the realizable PAC assumption.

The results of labeled data points rate are shown in Figure 6.1. It shows that the required labels of our proposed algorithms are much smaller than the total number of points. Especially for the ALB-2, its number of queries does not increase after querying 20 times.

The results of test errors for classification are shown in Figure 6.2. It is shown that our proposed algorithms are both much better than passive learning. Especially for the ALB-2, it has the best performance. That is because ALB-2 reached the target hypothesis very fast, as $O(\log n)$.

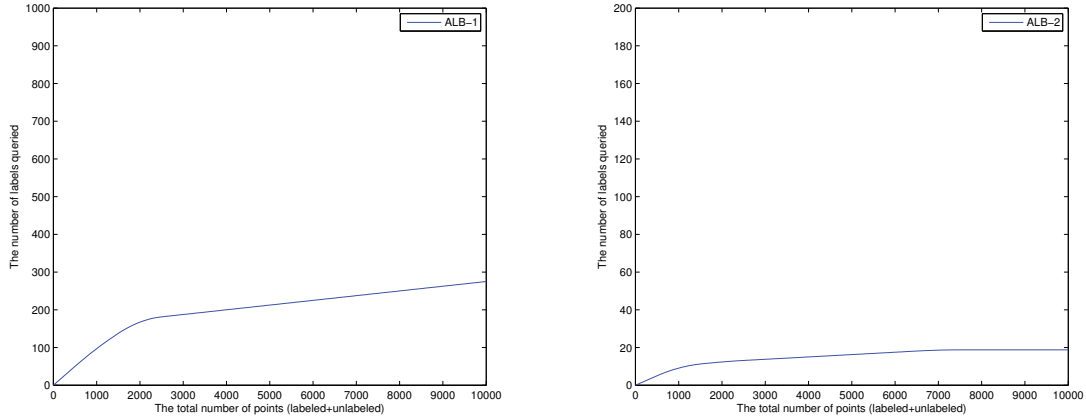
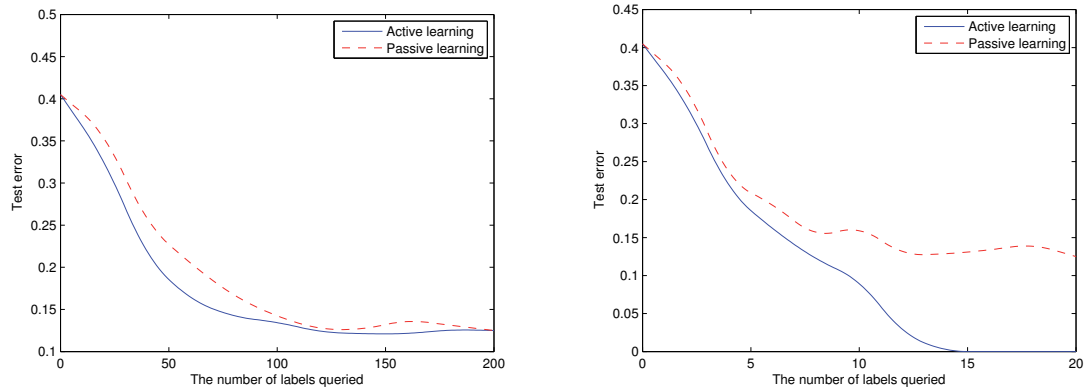


Fig. 6.1 *Labeled data points rate.*



(a) *ALB-1* ***(b)*** *ALB-2*

Fig. 6.2 *Test error rates for the classification experiments.*

6.5.2 Experimental results of agnostic setting

We constructed the data set as follows: in each case, the data was distributed uniformly over $[0,1]$; the stream length was $m = 10000$, and each experiment was repeated 20 times with different random seeds.

We studied the linear thresholds on the line $[0,1]$. The target hypothesis was fixed to be $h^*(x) = sign(x-0.5)$. We constructed two settings: noise-free setting and noise setting. In the noise-free setting, it is simple that the data was correctly labeled. In the noise setting, for this hypothesis class, we used random misclassification as the noise model. The misclassification model is as follows: for each point $x \sim \mathcal{D}_x$, we independently labeled it as true label with probability $1-v$ and false label with probability v. We looked at two particular runs of the algorithm: the first was with $\mathcal{H} = thresholds$, $m = 10000$, and $v = 0.1$; the second was with $\mathcal{H} = thresholds$, $m = 10000$, and $v = 0.3$.

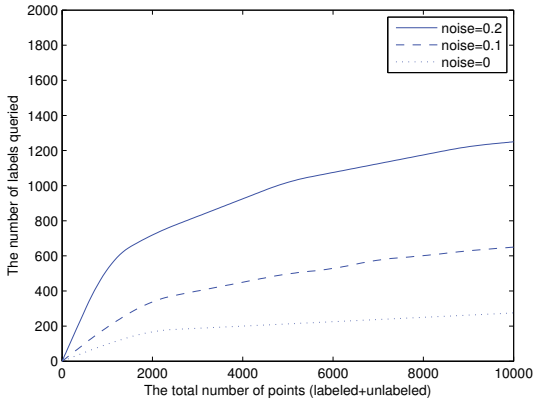


Fig. 6.3 *Labeled data points rate.*

The labeled data points rates are plotted in Figure 6.3. It is shown that the required labels of our active learner are significantly smaller than the total number of points.

For a reasonable check, we examined the locations of data for which our algorithm requested a label. The locations of queried points are shown in Figure 6.4. The results in Figure 6.4 show that at beginning the querying is a little bit varied. However, as the active learner processed, the queries for labeling became concentrated near the boundary of the target hypothesis.

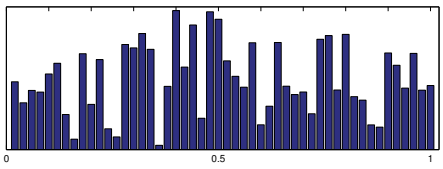
In addition, we showed the results about the error comparison for the classification. We used all the three cases and compared our method with the passive learning. We randomly constructed 1000 data by the same way with the same noise for testing. The test errors for the two learners are plotted in Figure 6.5. The results in Figure 6.5 verify that the active learner dramatically improves over the passive learning on the data sets. Comparing different cases, it also shows that the benefits of active learning for low levels of noise are more apparent than for high levels of noise.

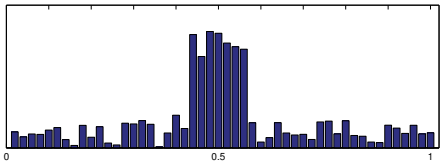
6.6 Proofs

6.6.1 Proof of Theorem 6.1

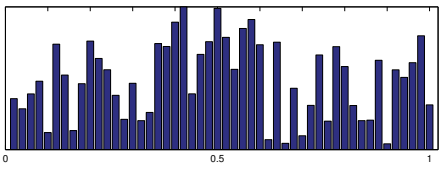
Proof. At round $t = 1$, it is correct that h^* is in $\mathcal{H}_0$. Given x, if y is synthesized by $\mathcal{H}_0$, then we have

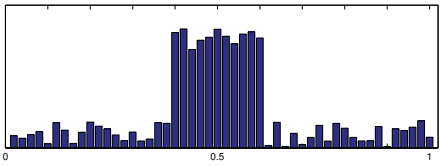
$$h_1(x) = h_2(x) = \cdots = h_m(x), h_i \in \mathcal{H}_0 \tag{6.12}$$



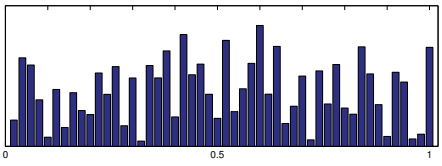


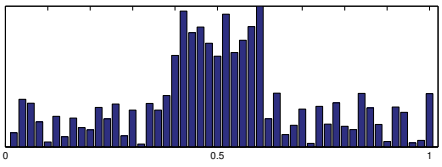
(a) *Noise=0, Top: 100/ Botton: 200*





(b) *Noise=0.1, Top: 300/ Bottom: 600*





(c) *Noise=0.3, Top: 500/ Bottom:1000*

Fig. 6.4 *The locations of label queries. The x-axis is the unit interval and the y-axis is the rate of numbers in the corresponding interval. The top histogram shows the locations of label requests at the early stage; the bottom histogram is for all label queries.*

and h^* is in this hypothesis set. It means

$$\bar{y} = h^*(x) = y, \tag{6.13}$$

If y is queried, y is the ground truth.

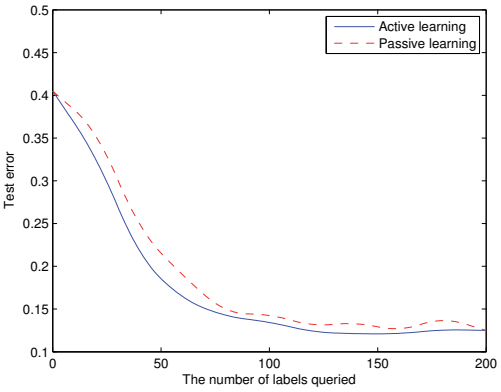
We assume the correctness assumption also holds at round t . That means h^* is in $\mathcal{H}_t$.

Then when $t \leftarrow t+1$, we have if y is synthesized that is

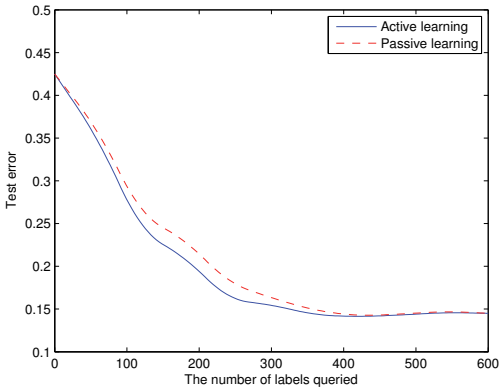
$$\bar{y} = h_1(x) = \cdots = h_m(x), h_i \in \mathcal{H}_t \tag{6.14}$$

and h^* is in these hypotheses. Thus $\bar{y} = h^*(x) = y$.

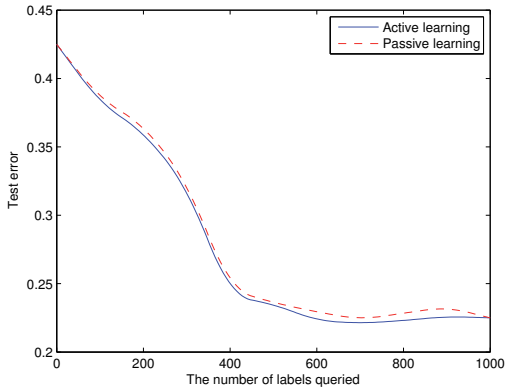
If y is queried, y is the ground truth and we have $h^*(x) = y$ for $\mathcal{H}_{t+1}$. Thus at round $t+1$, $\mathcal{H}_{t+1}$ still contains h^*. □



(a) Noise=0



(b) Noise=0.1



(c) Noise=0.3

Fig. 6.5 *Test error rates for the classification experiments.*

6.6.2 Proof of Theorem 6.2

Proof. We simply rewrite the PAC bound as $O(1/n(\log|\mathcal{H}| + \log(1/\delta)))$ for every t by replacing δ with $\delta/((t^2+t))$ and using the union bound over all t.

At round t, we condition on the event that the above property holds before. Then active learner acquires the label of x_t if and only if there exists any $h \in \mathcal{H}_t$ such that (1) h is consistent with Z_{t-1} and (2) h disagrees with h^* on x_t.

The condition (1) means that the error of such a hypothesis $h \in \mathcal{H}_t$ can be bounded by PAC bound. Let r_t denote the value of this bound.

At the same time, the condition (2) means that x_t is in the region of disagreement for the subset of hypotheses at most r_t away from h^*, that is $x_t \in \tilde{R}(h^*, r_t)$. Following the definition of the disagreement coefficient θ, we have $\Pr(x_t \in \tilde{R}(h^*, r_t)) \leq \theta \cdot r_t$.

Let $Q_t \in \{0,1\}$ be the random variable that indicates if the label y_t is queried. Then the expected number of queries after n rounds is

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{n} Q_t\right] = \sum_{t=1}^{n} \mathbb{E}[\mathbb{E}[Q_t | Z_{t-1}, x_t]] \\
&\leq \sum_{t=1}^{n} \mathbb{E}[\Pr(\exists h \in \mathcal{H}_t, \text{ there exists disagree } | Z_{t-1}, x_t)] \\
&\leq \sum_{t=1}^{n} \mathbb{E}[\Pr(\exists h \in \tilde{B}(h^*, r_t), \text{ s.t. } h(x_t) \neq h^*(x_t)] \\
&= \sum_{t=1}^{n} \Pr(x_t \in \tilde{R}(h^*, r_t)) \\
&\leq \sum_{t=1}^{n} \theta \cdot O\left(\frac{1}{t}\left(\log|\mathcal{H}| + \log\left(\frac{t(t+1)}{\delta}\right)\right)\right) \\
&= O\left(\theta \cdot \left(\log|\mathcal{H}| + \log\left(\frac{1}{\delta}\right)\right)\log n\right).
\end{aligned}$$

□

6.6.3 Proof of Theorem 6.3

Proof. The following is similar to proof of Theorem 6.1.

At round $t = 1$, it is correct that h^* is in $\mathcal{H}_0$. Given x_i, if y_i is synthesized by $\mathcal{H}_0$, then we have

$$h_1(x_i) = h_2(x_i) = \cdots = h_m(x_i), h_i \in \mathcal{H}_0 \tag{6.15}$$

and h^* is in this hypothesis set. It means

$$\bar{y}_i = h^*(x_i) = y_i, \tag{6.16}$$

If y_i is queried, y_i is the ground truth.

We assume the correctness assumption also holds at round t. That means h^* is in $\mathcal{H}_t$.

Then when $t \leftarrow t+1$, if y_i is synthesized that is

$$\bar{y} = h_1(x_i) = \cdots = h_m(x_i), h_i \in \mathcal{H}_t \tag{6.17}$$

and h^* is in these hypotheses. Thus $\bar{y} = h^*(x) = y$.

If y is queried, y is the ground truth and we get $h^*(x) = y$. Thus at this time, $\mathcal{H}_{t+1}$ still contains h^*. □

6.6.4 Proof of Theorem 6.4

Proof. It has the same conditions with Theorem 6.2 for querying the label of a point.

Let $Q_t \in \{0,1\}$ be the random variable that indicates if the label y_t is queried. Then the expected number of queries after n rounds is

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{n} Q_t\right] = \sum_{t=1}^{n} \mathbb{E}[\mathbb{E}[Q_t|Z_{t-1}, x_t]] \\
&\leq \sum_{t=1}^{n} \mathbb{E}[\Pr(\exists h \in \mathcal{H}_t, \text{ there exists disagree } |Z_{t-1}, x_t)] \\
&\leq \sum_{t=1}^{n} \mathbb{E}[\Pr(\exists h \in \tilde{B}(h^*, r_t), \text{ s.t. } h(x_t) \neq h^*(x_t)] \\
&= \sum_{t=1}^{n} \Pr(x_t \in \tilde{R}(h^*, r_t)) \\
&\leq \sum_{t=1}^{n} \theta \cdot O\left(\frac{1}{t}\left(\log|\mathcal{H}| + \log\left(\frac{t(t+1)}{\delta}\right)\right)\right) \\
&= O\left(\theta \cdot \left(\log n \log(\log|\mathcal{H}|) + \log\left(\frac{1}{\delta}\right)\log n\right)\right) \\
&= O\left(\theta \cdot \left(\log(\log|\mathcal{H}|) + \log\left(\frac{1}{\delta}\right)\right)\log n\right).
\end{aligned}$$

□

6.6.5 Proof of Theorem 6.6

Proof. If the ALB-LUB is not correct, it happens that either a hypothesis in $(S^*_{m,\varepsilon})^c$ is selected (first situation) or a hypothesis in S^*_m is dismissed (second situation) at some first round t. Before t, all the hypotheses in the set of selected hypotheses S are in $S^*_{m,\varepsilon}$, and all the hypotheses in set of discarded hypotheses are in $(S^*)^c$.

In the first situation, at round t, let h_b be the hypothesis in $(S^*_{m,\varepsilon})^c$ selected: for all hypotheses h_a in $B^c(t)$, one has $\Delta(h_b, h_a) < \varepsilon$. Among these hypotheses, at least one must be in S^*_m. So there exists $h_a \in S^*_m$ and $h_b \in (S^*_{m,\varepsilon})^c$ such that $\Delta(h_b, h_a) < \varepsilon$. The second situation

leads to the same conclusion. Hence if the algorithm fails, the following event holds:

$$\begin{aligned}
&\bigcup_{t\in N}(\exists a \in S_m^*, \exists b \in (S_{m,\varepsilon}^*)^c : UB_b(t) - LB_a(t) < \varepsilon) \\
&\subset \bigcup_{t\in N}(\exists a \in S_m^*, \exists b \in (S_{m,\varepsilon}^*)^c : (LB_a(t) > p_a) \cup (UB_b(t) > \varepsilon + p_a > p_b)) \\
&\subset \bigcup_{t\in N}\bigcup_{a\in S_m^*}(LB_a(t) > p_a) \bigcup_{b\in (S_{m,\varepsilon}^*)^c}(UB_b(t) < p_b) \subset W^c.
\end{aligned}$$

□

6.6.6 Proof of Theorem 6.7

Proof. Recall that the following event holds:

$$W = \bigcap_{t\in N}\bigcap_{a\in S_m^*}(LB_a(t) < p_a) \bigcap_{b\in (S_m^*)^c}(UB_b(t) > p_b).$$

It means that any hypothesis h in S_m^* is (ε, m)-optimal and any hypothesis h in $(S_m^*)^c$ is not (ε, m)-optimal. In other words, we can relate two sets S_m^* and $(S_m^*)^c$ as follows:

$$\forall i \in S_m^* \forall j \in (S_m^*)^c : (p_j - p_i > \varepsilon). \tag{6.18}$$

Since $|S| = m$, an hypothesis j in $(S_m^*)^c$ can occur in S only if there is some arm i in S_m^* such that $\hat{p}_i \geq \hat{p}_j$. According to (6.18), it implies that the latter event can only occur if

$$\hat{p}_i \geq p_i + \frac{\varepsilon}{2} \text{ or } \hat{p}_j \leq p_j - \frac{\varepsilon}{2}. \tag{6.19}$$

Adapting probabilities, according to the union bound and Hoeffding's inequality we have

$$\begin{aligned}
&P(\exists j \in (S_m^*)^c : (j \in S)) \\
&\leq P(\exists i \in S_m^* : \hat{p}_i \geq p_i + \frac{\varepsilon}{2}) + P(\exists j \in (S_m^*)^c : \hat{p}_j \leq p_j - \frac{\varepsilon}{2}) \\
&\leq \sum_{i\in S_m^*} P(\hat{p}_i \geq p_i + \frac{\varepsilon}{2}) + \sum_{j\in (S_m^*)^c} P(\hat{p}_j \leq p_j - \frac{\varepsilon}{2}).
\end{aligned}$$

If the above two events happen, according to Theorem 6.4 we have

$$\begin{aligned} N &\geq m \cdot m_0(\varepsilon/2, \delta, V) + (|\mathcal{H}| - m) \cdot m_0(\varepsilon/2, \delta, V) \\ &= \frac{128|\mathcal{H}|}{\varepsilon^2}\left(2V \log\left(\frac{24}{\varepsilon}\right) + \log\left(\frac{4}{\delta}\right)\right). \end{aligned}$$

Above result is the at least number of use of hypotheses. In the worst case, each point updates one hypothesis once, then we have the querying times of points at least $\frac{128|\mathcal{H}|}{\varepsilon^2}\left(2V\log\left(\frac{24}{\varepsilon}\right)+\log\left(\frac{4}{\delta}\right)\right)$ with probability $1-\delta$. Then we have the following generalization bound: for any $\varepsilon \in (0,1)$ and $\delta \in (0,1)$, with probability $1-\delta$, has error $\mathrm{err}(h)$ at most $\frac{128|\mathcal{H}|}{n\varepsilon}\left(2V\log\left(\frac{24}{\varepsilon}\right)+\log\left(\frac{4}{\delta}\right)\right)$.

Similar to Theorem 6.2, then active learner makes the query if and only if there exists any $h \in \mathcal{H}_t$ such that h is consistent with Z_{t-1} and h disagrees with h^* on x_t. Following the definition of the disagreement coefficient θ, we have $\Pr(x_t \in \tilde{R}(h^*, r_t)) \leq \theta \cdot r_t$.

Let $Q_t \in \{0,1\}$ be the random variable that indicates if the label y_t is queried. Then the expected number of queries after n rounds is

$$\begin{aligned} &\mathbb{E}\left[\sum_{t=1}^{n} Q_t\right] = \sum_{t=1}^{n} \mathbb{E}[\mathbb{E}[Q_t | Z_{t-1}, x_t]] \\ &\leq \sum_{t=1}^{n} \mathbb{E}[\Pr(\exists h \in \mathcal{H}_t, \text{ there exists disagree } |Z_{t-1}, x_t)] \\ &\leq \sum_{t=1}^{n} \mathbb{E}[\Pr(\exists h \in \tilde{B}(h^*, r_t), \text{ s.t. } h(x_t) \neq h^*(x_t)] \\ &= \sum_{t=1}^{n} \Pr(x_t \in \tilde{R}(h^*, r_t)) \\ &\leq \sum_{t=1}^{n} \theta \cdot O\left(\frac{|\mathcal{H}|}{t\varepsilon}\left(V \log\left(\frac{1}{\varepsilon}\right) + \log\left(\frac{t(t+1)}{\delta}\right)\right)\right) \\ &\leq \sum_{t=1}^{n} \theta \cdot O\left(\frac{|\mathcal{H}|}{t\varepsilon}\left(\log|\mathcal{H}| \log\left(\frac{1}{\varepsilon}\right) + \log\left(\frac{t(t+1)}{\delta}\right)\right)\right) \\ &= O\left(\theta \cdot \frac{|\mathcal{H}|}{\varepsilon}\left(\log|\mathcal{H}| \log\left(\frac{1}{\varepsilon}\right) + \log\left(\frac{1}{\delta}\right)\right) \log n\right). \end{aligned}$$

Thus the label complexity is as follows:

$$O\left(\theta \cdot \frac{|\mathcal{H}|}{\varepsilon}\left(\log|\mathcal{H}| \log\left(\frac{1}{\varepsilon}\right) + \log\left(\frac{1}{\delta}\right)\right) \log n\right).$$

□

6.7 Conclusion

We have carefully casted the active learning into bandit framework and proposed three active learning algorithms. The active learning procedure is transformed as an exploration problem of bandits. We considered the hypothesis selection as an arm selection problem and optimally maintain the candidate set of hypotheses according to the error differences when querying the labels of the points. We provided two kinds of algorithms to handle different settings. In the realizable PAC setting, our algorithms ALB-1 and ALB-2 rely on the empirical error and select the hypothesis according to this quality. In the agnostic setting, our algorithm ALB-LUB relies on the lower and upper bounds on the error and selects or eliminates the hypotheses according to the differences between these bounds. We provided the analysis about theoretic guarantee and experimental results. Our results show that active learning via bandits strategy can work efficiently.

Chapter 7

Conclusion

In the final chapter, we reflect on the finding of the previous chapters. We review the main topics covered in this dissertation.

Multi-armed bandit has been used for sequential decision making problems. The exploration and exploitation is the key issue for solving the sequential decision problems. The multi-armed bandit framework provides a way for developing learning algorithms to address the trade-off between exploration and exploitation.

In the first chapter of this thesis, we introduced the framework of traditional multi-armed bandits. Then we briefly demonstrated that bandit learning can be extended to some real scenarios, such as social networks, multi-view problem, mult-task problem, repeated labeling and active learning.

In Chapter 2, we introduced "networked bandits", a novel framework that considers the correlation between arms. Based on this new framework, we studied the problem of how to select the arm with multiple payoffs in networked bandits by considering a multi-armed bandit of interconnected arms, one of which can invoke other related arms at each round. After selecting an arm, we can obtain payoffs from this arm and its relations. We considered this approach in the contextual bandit setting and assumed disjoint linear payoffs for arms. We proposed a new networked bandit algorithm NetBandits that considers the uncertainty of the payoffs using integrated confidence sets. We also provide a regret bound for our solution. Our experiments show that it is better to consider both the network topology and the payoffs of arms, and we observe that our approach performs well in this setting.

In Chapter 3, we defined and examined the problem of view subset selection for multi-view learning in the sequential environment. Unlike a traditional bandit problem, the forecaster aims to select a subset $m(m \geq 2)$ from n views after observing the view information generated from different views to make the final prediction and then obtain payoff. We formalized this problem as a multi-view bandit subset selection in the steam-based multi-

view setting. Few previous studies have examined multiple view selection in multi-view learning, and there have been no studies on subset selection in the multi-view bandit setting. We first introduced the multi-view bandit and its corresponding multi-view simple regret. Incorporating the Rademacher complexity, which plugs into the generalization bound, we proposed the multi-view bandit algorithm by considering this generalization bound, which can theoretically bound the multi-view simple regret. We provided an upper bound of expected multi-view simple regret using the Rademacher complexity and proved an upper bound of order $1/\sqrt{t}$. We also showed that the consistency of predictions based on different views can improve the simple regret bound. Our experiments verified that our view subset selection method is efficient.

In Chapter 4, we proposed a new active learning algorithm for multi-task learning, named active multi-task learning via bandits, which is a general active learning framework for multi-task learning problem. We cast the selection procedure as a bandit framework. We considered the hypothesis as an arm and select the instances to filter out the hypotheses which have the better performance than others. The active learning strategy balances the trade-off between minimizing the risk and improving the confidence bounds for the hypothesis. In our algorithm, at each round, we maintain a sampling distribution on data for selection, query the label of an instance according to the distribution and update the distribution based on the performance of newly trained multi-task learner. We also provided an implementation of our approach based on multi-task learning with trace-norm regularization method. Moreover, experimental results demonstrate the effectiveness and efficiency of our algorithm.

In Chapter 5, we formalized the repeated labeling as a bandit model. We studied the problem of how to select a proportion of labeling tasks from a pool of data, in which the labels of each example are repeatedly acquired from multiple labelers. We presented a simple repeated labeling algorithm and proposed two selective repeated labeling algorithms: Best m Labeling algorithm and Improved Best m Labeling. We demonstrated their effectiveness and efficiency both in the theoretical results and experiments.

In Chapter 6, we carefully cast the active learning into bandit framework and proposed three active learning algorithms. The active learning procedure is transformed as an exploration problem of bandits. We considered the hypothesis selection as an arm selection problem and we optimally maintained the candidate set of hypotheses according to the error differences when querying the labels of the points. We provided two kinds of algorithms to handle different settings. In the realizable PAC setting, our algorithms ALB-1 and ALB-2 rely on the empirical error and select the hypothesis according to this quality. In the agnostic setting, our algorithm ALB-LUB relies on the lower and upper bounds on the error and selects or eliminates the hypotheses according to the differences between these bounds. We

provided the analysis about theoretic guarantee and experimental results. Our results show that active learning via bandits strategy can work efficiently.

References

[1] Yasin Abbasi-Yadkori, Csaba Szepesvári, and David Tax. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, 2011.

[2] Jacob Abernethy, Francis Bach, Theodoros Evgeniou, and Jean-Philippe Vert. A new approach to collaborative filtering: Operator estimation with spectral regularization. *The Journal of Machine Learning Research*, 10:803–826, 2009.

[3] Ayan Acharya, Raymond J Mooney, and Joydeep Ghosh. Active multitask learning using both latent and supervised shared topics. In *SIAM International Conference on Data Mining*, 2014.

[4] Shotaro Akaho. A kernel method for canonical correlation analysis. *arXiv preprint*, 2006.

[5] Aris Anagnostopoulos, Ravi Kumar, and Mohammad Mahdian. Influence and correlation in social networks. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008.

[6] Rie Kubota Ando and Tong Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *The Journal of Machine Learning Research*, 6:1817–1853, 2005.

[7] Martin Anthony and Peter L Bartlett. *Neural network learning: Theoretical foundations*. Cambridge university press, 2009.

[8] Andreas Argyriou, Andreas Maurer, and Massimiliano Pontil. An algorithm for transfer learning in a heterogeneous environment. In *Machine Learning and Knowledge Discovery in Databases*, pages 71–85. 2008.

[9] Jean-Yves Audibert and Sébastien Bubeck. Minimax policies for adversarial and stochastic bandits. In *Annual Conference on Learning Theory*, 2009.

[10] Jean-Yves Audibert and Sébastien Bubeck. Regret bounds and minimax policies under partial monitoring. *Journal of Machine Learning Research*, 11:2785–2836, 2010.

[11] Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári. Exploration-exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19):1876–1902, 2009.

[12] Jean-Yves Audibert, Sébastien Bubeck, and Remi Munos. Best arm identification in multi-armed bandits. In *Annual Conference on Learning Theory*, 2010.

[13] Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3:397–422, 2003.

[14] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002.

[15] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.

[16] Bart Bakker and Tom Heskes. Task clustering and gating for bayesian multitask learning. *The Journal of Machine Learning Research*, 4:83–99, 2003.

[17] Maria-Florina Balcan, Alina Beygelzimer, and John Langford. Agnostic active learning. In *Proceedings of the 23rd International Conference on Machine Learning*, 2006.

[18] Maria-Florina Balcan, Steve Hanneke, and Jennifer Wortman Vaughan. The true sample complexity of active learning. *Machine Learning*, 80(2-3):111–139, 2010.

[19] Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *The Journal of Machine Learning Research*, 3:463–482, 2003.

[20] Jonathan Baxter. Theoretical models of learning to learn. In *Learning to learn*, pages 71–94. 1998.

[21] Alain Berlinet and Christine Thomas-Agnan. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer, 2004.

[22] Alina Beygelzimer, Sanjoy Dasgupta, and John Langford. Importance weighted active learning. In *Proceedings of the 26th International Conference on Machine Learning*, 2009.

[23] Steffen Bickel, Jasmina Bogojeska, Thomas Lengauer, and Tobias Scheffer. Multitask learning for hiv therapy screening. In *Proceedings of the 25th International Conference on Machine Learning*, 2008.

[24] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Annual Conference on Learning Theory*, 1998.

[25] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. Occam's razor. *Information processing letters*, 24(6):377–380, 1987.

[26] Zahy Bnaya, Rami Puzis, Roni Stern, and Ariel Felner. Social network search as a volatile multi-armed bandit problem. *HUMAN*, 2(2):pp–84, 2013.

[27] Ulf Brefeld, Thomas Gärtner, Tobias Scheffer, and Stefan Wrobel. Efficient coregularised least squares regression. In *Proceedings of the 23rd International Conference on Machine Learning*, 2006.

[28] Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure exploration in multi-armed bandits problems. In *International Conference on Algorithmic Learning Theory*, 2009.

[29] Sébastien Bubeck, Tengyao Wang, and Nitin Viswanathan. Multiple identifications in multi-armed bandits. In *Proceedings of the 30th International Conference on Machine Learning*, 2013.

[30] Sébastien Bubeck, Tengyao Wang, and Nitin Viswanathan. Multiple identifications in multi-armed bandits. In *Proceedings of the 30th International Conference on Machine Learning*, 2013.

[31] Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.

[32] Swapna Buccapatnam, Atilla Eryilmaz, and Ness B Shroff. Multi-armed bandits in the presence of side observations in social networks. *OSU, Technical Report*, 2013.

[33] Nicolo Cesa-Bianchi, Gábor Lugosi, et al. *Prediction, learning, and games*. Cambridge University Press Cambridge, 2006.

[34] Nicolò Cesa-Bianchi, Claudio Gentile, and Giovanni Zappella. A gang of bandits. In *Advances in Neural Information Processing Systems*, 2013.

[35] Olivier Chapelle and Alexander Zien. Semi-supervised classification by low density separation. 2004.

[36] Kamalika Chaudhuri, Sham M Kakade, Karen Livescu, and Karthik Sridharan. Multi-view clustering via canonical correlation analysis. In *Proceedings of the 26th International Conference on Machine Learning*, 2009.

[37] Wei Chu, Lihong Li, Lev Reyzin, and Robert E Schapire. Contextual bandits with linear payoff functions. In *International Conference on Artificial Intelligence and Statistics*, 2011.

[38] David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.

[39] David Crandall, Dan Cosley, Daniel Huttenlocher, Jon Kleinberg, and Siddharth Suri. Feedback effects between similarity and social influence in online communities. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008.

[40] Varsha Dani, Thomas P Hayes, and Sham M Kakade. Stochastic linear optimization under bandit feedback. In *Annual Conference on Learning Theory*, 2008.

[41] Sanjoy Dasgupta. Analysis of a greedy active learning strategy. In *Advances in Neural Information Processing Systems*, 2004.

[42] Sanjoy Dasgupta. Coarse sample complexity bounds for active learning. In *Advances in Neural Information Processing Systems*, 2005.

[43] Sanjoy Dasgupta, Claire Monteleoni, and Daniel J Hsu. A general agnostic active learning algorithm. In *Advances in Neural Information Processing Systems*, 2007.

[44] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[45] Pinar Donmez, Jaime G Carbonell, and Jeff G Schneider. A probabilistic framework to learn from multiple annotators with time-varying accuracy. In *SIAM International Conference on Data Mining*, 2010.

[46] Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *The Journal of Machine Learning Research*, 7:1079–1105, 2006.

[47] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[48] Theodoros Evgeniou, Charles A Micchelli, and Massimiliano Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6:615–637, 2005.

[49] Meng Fang and Dacheng Tao. Networked bandits with disjoint linear payoffs. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2014.

[50] Meng Fang and Dacheng Tao. Active multi-task learning via bandits. In *SIAM International Conference on Data Mining*, 2015.

[51] Meng Fang and Xingquan Zhu. Active learning with uncertain labeling knowledge. *Pattern Recognition Letters*, 43:98–108, 2014.

[52] Meng Fang, Xingquan Zhu, Bin Li, Wei Ding, and Xindong Wu. Self-taught active learning from crowds. In *IEEE International Conference on Data Mining*, 2012.

[53] Meng Fang, Jie Yin, and Xingquan Zhu. Active exploration: simultaneous sampling and labeling for large graphs. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, 2013.

[54] Meng Fang, Jie Yin, and Xingquan Zhu. Knowledge transfer for multi-labeler active learning. In *Machine Learning and Knowledge Discovery in Databases*, pages 273–288. 2013.

[55] Meng Fang, Jie Yin, and Xingquan Zhu. Transfer learning across networks for collective classification. In *IEEE International Conference on Data Mining*, 2013.

[56] Meng Fang, Jie Yin, Xingquan Zhu, and Chengqi Zhang. Active class discovery and learning for networked data. In *SIAM International Conference on Data Mining*, 2013.

[57] Meng Fang, Jie Yin, and Dacheng Tao. Active learning for crowdsourcing using knowledge transfer. In *The AAAI Conference on Artificial Intelligence*, 2014.

[58] Meng Fang, Jie Yin, and Xingquan Zhu. Active exploration for large graphs. *Data Mining and Knowledge Discovery*, pages 1–39, 2015.

[59] Meng Fang, Jie Yin, Xingquan Zhu, and Chengqi Zhang. Trgraph: Cross-network transfer learning via common signature subgraphs. *IEEE Transactions on Knowledge and Data Engineering*, 27(9):2536–2549, 2015.

[60] Yoav Freund, H Sebastian Seung, Eli Shamir, and Naftali Tishby. Information, prediction, and query by committee. In *Advances in Neural Information Processing Systems*, 1993.

[61] Yoav Freund, H Sebastian Seung, Eli Shamir, and Naftali Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2-3):133–168, 1997.

[62] Eric Friedman. Active learning for smooth problems. In *Annual Conference on Learning Theory*, 2009.

[63] 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*, 2012.

[64] Ravi Ganti and Alexander G Gray. Building bridges: Viewing active learning from the multi-armed bandit lens. *arXiv preprint*, 2013.

[65] Jinyang Gao, Xuan Liu, Beng Chin Ooi, Haixun Wang, and Gang Chen. An online cost sensitive decision-making method in crowdsourcing systems. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, 2013.

[66] Aurélien Garivier and Olivier Cappé. The kl-ucb algorithm for bounded stochastic bandits and beyond. *arXiv preprint*, 2011.

[67] Yong Ge, Hui Xiong, Alexander Tuzhilin, Keli Xiao, Marco Gruteser, and Michael Pazzani. An energy-efficient mobile recommender system. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2010.

[68] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007.

[69] Steve Hanneke. A bound on the label complexity of agnostic active learning. In *Proceedings of the 24th International Conference on Machine Learning*, 2007.

[70] Steve Hanneke. Adaptive rates of convergence in active learning. In *Annual Conference on Learning Theory*, 2009.

[71] Steve Hanneke. Theory of disagreement-based active learning. *Foundations and Trends in Machine Learning*, 2014.

[72] Abhay Harpale and Yiming Yang. Active learning for multi-task adaptive filtering. In *Proceedings of the 27th International Conference on Machine Learning*, 2010.

[73] Junya Honda and Akimichi Takemura. An asymptotically optimal bandit algorithm for bounded support models. In *Annual Conference on Learning Theory*, 2010.

[74] Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28(3-4): 321–377, 1936.

[75] Panagiotis G Ipeirotis and Praveen K Paritosh. Managing crowdsourced human computation: a tutorial. In *Proceedings of the 20th International Conference on World Wide Web*, 2011.

[76] Tomoharu Iwata, Amar Shah, and Zoubin Ghahramani. Discovering latent influence in online social activities via shared cascade poisson processes. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2013.

[77] Laurent Jacob, Jean-philippe Vert, and Francis R Bach. Clustered multi-task learning: A convex formulation. In *Advances in Neural Information Processing Systems*, 2009.

[78] Zhaoyin Jia, Yao-Jen Chang, and Tsuhan Chen. Active view selection for object and pose recognition. In *IEEE International Conference on Computer Vision Workshops*, 2009.

[79] Matti Kääriäinen. Active learning in the non-realizable case. In *International Conference on Algorithmic Learning Theory*, 2006.

[80] Sham M Kakade and Dean P Foster. Multi-view regression via canonical correlation analysis. In *Learning Theory*, pages 82–96. 2007.

[81] Sham M Kakade, Karthik Sridharan, and Ambuj Tewari. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In *Advances in Neural Information Processing Systems*, 2009.

[82] Sham M Kakade, Shai Shalev-Shwartz, and Ambuj Tewari. Regularization techniques for learning with matrices. *The Journal of Machine Learning Research*, 13(1):1865–1890, 2012.

[83] Shivaram Kalyanakrishnan and Peter Stone. Efficient selection of multiple bandit arms: Theory and practice. In *Proceedings of the 27th International Conference on Machine Learning*, 2010.

[84] Emilie Kaufmann and Shivaram Kalyanakrishnan. Information complexity in bandit subset selection. In *Annual Conference on Learning Theory*, 2013.

[85] Marius Kloft and Gilles Blanchard. The local rademacher complexity of lp-norm multiple kernel learning. In *Advances in Neural Information Processing Systems*, 2011.

[86] Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Probability*, 6(1):4–22, 1985.

[87] Gert RG Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I Jordan. Learning the kernel matrix with semidefinite programming. *The Journal of Machine Learning Research*, 5:27–72, 2004.

[88] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, 2010.

[89] M. Lichman. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml.

[90] D.J.C. MacKay. Information-based objective functions for active data selection. *Neural computation*, 4(4):590–604, 1992.

[91] Odalric-Ambrym Maillard, Rémi Munos, and Gilles Stoltz. A finite-time analysis of multi-armed bandits problems with kullback-leibler divergences. *arXiv preprint*, 2011.

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

[93] Andreas Maurer. Bounds for linear multi-task learning. *The Journal of Machine Learning Research*, 7:117–139, 2006.

[94] Andreas Maurer and Massimiliano Pontil. Excess risk bounds for multitask learning with trace norm regularization. *arXiv preprint*, 2012.

[95] P. Melville and R. Mooney. Diverse ensembles for active learning. In *Proceedings of the 21st International Conference on Machine Learning*, 2004.

[96] Prem Melville and Raymond J Mooney. Diverse ensembles for active learning. In *Proceedings of the 21st International Conference on Machine Learning*, 2004.

[97] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2012.

[98] Ion Muslea, Steven Minton, and Craig A Knoblock. Active+ semi-supervised learning= robust multi-view learning. In *Proceedings of the 19th International Conference on Machine Learning*, 2002.

[99] Ion Muslea, Steven Minton, and Craig A Knoblock. Active learning with strong and weak views: A case study on wrapper induction. In *International Joint Conference on Artificial Intelligence*, 2003.

[100] Ion Muslea, Steven Minton, and Craig A Knoblock. Active learning with multiple views. *Journal of Artificial Intelligence Research*, 27:203–233, 2006.

[101] Seth A Myers, Chenguang Zhu, and Jure Leskovec. Information diffusion and external influence in networks. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2012.

[102] Kamal Nigam and Rayid Ghani. Analyzing the effectiveness and applicability of co-training. In *Proceedings of the 9th International Conference on Information and Knowledge Management*, 2000.

[103] Guillaume Obozinski, Ben Taskar, and Michael Jordan. Multi-task feature selection. *Statistics Department, UC Berkeley, Technical Report*, 2006.

[104] Lucas Paletta and Axel Pinz. Active object recognition by view integration and reinforcement learning. *Robotics and Autonomous Systems*, 31(1):71–86, 2000.

[105] Victor H Peña, Tze Leung Lai, and Qi-Man Shao. *Self-normalized processes: Limit theory and Statistical Applications*. Springer, 2008.

[106] Ting Kei Pong, Paul Tseng, Shuiwang Ji, and Jieping Ye. Trace norm regularization: reformulations, algorithms, and multi-task learning. *SIAM Journal on Optimization*, 20(6):3465–3489, 2010.

[107] G-J Qi, Xian-Sheng Hua, Yong Rui, Jinhui Tang, and Hong-Jiang Zhang. Two-dimensional active learning for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[108] Novi Quadrianto and Christoph H Lampert. Learning multi-view neighborhood preserving projections. In *Proceedings of the 28th International Conference on Machine Learning*, 2011.

[109] J Ross Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.

[110] Alain Rakotomamonjy, Francis Bach, Stéphane Canu, and Yves Grandvalet. More efficiency in multiple kernel learning. In *Proceedings of the 24th International Conference on Machine Learning*, 2007.

[111] Vikas C Raykar, Shipeng Yu, Linda H Zhao, Anna Jerebko, Charles Florin, Gerardo Hermosillo Valadez, Luca Bogoni, and Linda Moy. Supervised learning from multiple experts: whom to trust when everyone lies a bit. In *Proceedings of the 26th International Conference on Machine Learning*, 2009.

[112] Vikas C Raykar, Shipeng Yu, Linda H Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *The Journal of Machine Learning Research*, 11:1297–1322, 2010.

[113] Roi Reichart, Katrin Tomanek, Udo Hahn, and Ari Rappoport. Multi-task active learning for linguistic annotations. In *Annual Meeting of the Association for Computational Linguistics*, 2008.

[114] Herbert Robbins. Some aspects of the sequential design of experiments. In *Herbert Robbins Selected Papers*, pages 169–177. 1985.

[115] David S Rosenberg and Peter L Bartlett. The rademacher complexity of co-regularized kernel classes. In *International Conference on Artificial Intelligence and Statistics*, 2007.

[116] Nicholas Roy and Andrew McCallum. Toward optimal active learning through monte carlo estimation of error reduction. In *Proceedings of the 18th International Conference on Machine Learning*, 2001.

[117] Paat Rusmevichientong and John N Tsitsiklis. Linearly parameterized bandits. *Mathematics of Operations Research*, 35(2):395–411, 2010.

[118] Mathieu Salzmann, Carl H Ek, Raquel Urtasun, and Trevor Darrell. Factorized orthogonal latent spaces. In *International Conference on Artificial Intelligence and Statistics*, 2010.

[119] Burr Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52: 55–66, 2010.

[120] Victor S Sheng, Foster Provost, and Panagiotis G Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008.

[121] Roland Siegwart, Illah Reza Nourbakhsh, and Davide Scaramuzza. *Introduction to autonomous mobile robots*. MIT press, 2011.

[122] Vikas Sindhwani and S Sathiya Keerthi. Large scale semi-supervised linear svms. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 2006.

[123] Vikas Sindhwani and David S Rosenberg. An rkhs for multi-view learning and manifold co-regularization. In *Proceedings of the 25th International Conference on Machine Learning*, 2008.

[124] Vikas Sindhwani, Partha Niyogi, and Mikhail Belkin. A co-regularization approach to semi-supervised learning with multiple views. In *Proceedings of ICML Workshop on Learning with Multiple Views*, 2005.

[125] Sören Sonnenburg, Gunnar Rätsch, and Christin Schäfer. A general and efficient multiple kernel learning algorithm. In *Advances in Neural Information Processing Systems*, 2005.

[126] Sören Sonnenburg, Gunnar Rätsch, Christin Schäfer, and Bernhard Schölkopf. Large scale multiple kernel learning. *The Journal of Machine Learning Research*, 7:1531–1565, 2006.

[127] Karthik Sridharan and Sham M Kakade. An information theoretic framework for multi-view learning. In *Annual Conference on Learning Theory*, 2008.

[128] Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang. Social influence analysis in large-scale networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009.

[129] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2:45–66, 2002.

[130] Antonio Torralba, Kevin P Murphy, and William T Freeman. Sharing features: efficient boosting procedures for multiclass object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[131] Vladimir N Vapnik and A Ya Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16(2):264–280, 1971.

[132] Blanca Vargas-Govea, Gabriel González-Serna, and Rafael Ponce-Medellın. Effects of relevant contextual features in the performance of a restaurant recommender system. In *ACM Conference on Recommender Systems*, 2011.

[133] Paolo Viappiani, Sandra Zilles, Howard J Hamilton, and Craig Boutilier. Learning complex concepts using crowdsourcing: a bayesian approach. In *Algorithmic Decision Theory*, pages 277–291. 2011.

[134] Chih-Chun Wang, Sanjeev R Kulkarni, and H Vincent Poor. Bandit problems with arbitrary side observations. In *IEEE Conference on Decision and Control*, 2003.

[135] Liwei Wang. Sufficient conditions for agnostic active learnable. In *Advances in Neural Information Processing Systems*, 2009.

[136] Ting Wang, Dashun Wang, and Fei Wang. Quantifying herding effects in crowd wisdom. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2014.

[137] Gary M Weiss and Foster Provost. Learning when training data are costly: the effect of class distribution on tree induction. *Journal of Artificial Intelligence Research*, 19: 315–354, 2003.

[138] Ian H Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[139] Chang Xu, Dacheng Tao, and Chao Xu. A survey on multi-view learning. *arXiv preprint*, 2013.

[140] Yan Yan, Glenn M Fung, Rómer Rosales, and Jennifer G Dy. Active learning from crowds. In *Proceedings of the 28th International Conference on Machine Learning*, 2011.

[141] Shipeng Yu, Balaji Krishnapuram, Rómer Rosales, and R Bharat Rao. Bayesian co-training. *The Journal of Machine Learning Research*, 12:2649–2680, 2011.

[142] Deming Zhai, Hong Chang, Shiguang Shan, Xilin Chen, and Wen Gao. Multiview metric learning with global consistency and local smoothness. *ACM Transactions on Intelligent Systems and Technology*, 3(3):53, 2012.

[143] Yi Zhang. Multi-task active learning with output constraints. In *The AAAI Conference on Artificial Intelligence*, 2010.

[144] Zhiqiang Zheng and Balaji Padmanabhan. Selectively acquiring customer information: A new data acquisition problem and an active learning-based solution. *Management Science*, 52(5):697–712, 2006.