Budget efficient online active learning and its applications

Hao, Shuji

2017

Hao, S. (2017). Budget efficient online active learning and its applications. Doctoral thesis, Nanyang Technological University, Singapore.

<http://hdl.handle.net/10356/69462>

<https://doi.org/10.32657/10356/69462>

NANYANG TECHNOLOGICAL UNIVERSITY

Budget Efficient Online Active Learning and Its Applications

Hao Shuji

Interdisciplinary Graduate School

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Doctor of Philosophy

January 23, 2017

Abstract

Online Active Learning (OAL) has been an important research area in machine learning, which aims to minimize the number of labeled instances and maximize the predictive performance meanwhile. OAL has both the efficiency and effectiveness of online learning and the labeling frugality of active learning. Due to these advantages, OAL has been widely used in real-world large-scale applications, such as information retrieval, data mining, recommendation system, and so on. However, there are still several problems existing in current OAL designs.

First, in the online learning with expert advice setting, most of the exiting OAL algorithms assume that all the experts are comparably reliable, which is usually not true in reality. For example, noisy workers are quite common in the crowdsourcing platforms. To correct this weak assumption, this study proposes two robust online active learning algorithms, which not only consider the predictions of experts on current instance, but also consider the cumulative performance of experts on past instances. To validate the proposed algorithms, a series of experiments are conducted, in which the results show that the proposed algorithms greatly outperform the state-of-the-art existing algorithms and can achieve robust performance both in the normal and noisy scenarios.

Second, to obtain reliable labels in crowdsourcing, most of the algorithms either require a set of golden questions to filter out the noisy workers, or require several labels for each instance. These types of requirements are labeling costly both in terms of money and time. To save costs, a framework of Active Crowdsourcing for Annotation (ACA) is proposed based on the online learning with expert advice. The proposed framework consists of two main components: “Who to label” and “When to query”. The first component actively allocates the instance to reliable workers to gain labels, and the second component actively decides which instance is worthy to be a golden question. The empirical studies both on

simulated and real-world crowdsourcing datasets show that the proposed framework can robustly learn the reliability of each worker and wisely allocate the task to more reliable workers.

Third, in the typical online learning setting, most of the OAL algorithms adopt the margin-based query strategies, which usually assume that the model is well trained and the margin value is accurate. However, this assumption is often not true in reality, such as in the early training phrase. To alleviate this assumption, a second-order based online active learning algorithm is proposed, which considers not only the margin value of current instance, but also the confidence value of the current model. To validate the efficacy of the proposed algorithm, a theoretical mistake bound is provided and a set of empirical studies are conducted on real-world datasets. Both the theoretical and empirical studies show that our proposed second-order based algorithm can achieve the best performance in terms of accuracy.

Last, for the online relative similarity learning problem, most of the studies assume that there are large-scale labeled triplets. However, labeling datasets are usually costly and time consuming, especially for the large-scale similarity learning problems. To reduce the high computation cost, this study proposes two online active relative similarity learning algorithms: (i) first-order based Passive-Aggressive Active Similarity learning (PAAS); (ii) second-order based Confidence-Weighted Active Similarity learning (CWAS). In order to validate the effectiveness of our algorithms, the proposed algorithms are firstly theoretically analyzed, and then empirically evaluated on several real-world applications. The experiments show that the proposed PAAS and CWAS algorithms can greatly reduce the labeling cost in the relative similarity learning process.

In sum, to tackle the critical challenges of existing OAL algorithms, this study proposes four main OAL algorithms, most of them are theoretically sound algorithms. And all of the proposed algorithms are carefully evaluated on a large number of large-scale real-world applications and achieved promising results. Although promising results have been generated from this study, the proposed OAL algorithms are far from perfect. In future, there are several directions to study, such as OAL for concept drifting problems, distributed OAL algorithms, OAL for crowdsourcing and so on.

Acknowledgments

First, I would send my great gratitude to my thesis committee members Dr. Miao Chunyan, Dr. Steven C.H. Hoi and Dr. Shen Zhiqi.

Second, I also learned a lot from my collaborator Dr. Peilin Zhao. He not only helps me a lot in a weekly discussion, but also doesn't make my trivial questions stupid.

Third, I am very grateful to make a lot of friends, Xiaogang Han, Jianpin Cai, Di Wang, Shan Gao, Peng Cheng, Yundong Cai, Zhengjin Guo, Huiguo Zhang and so on.

Fourth, my gratitude also goes to the Interdisciplinary Graduate School of Nanyang Technological University. With their financial support, I could have the chance to study in beautiful Singapore and attend conferences abroad. In addition, I would also thanks all the staff from IGS, GSO and other related offices for their help.

What's more, I would send my sincere thanks to my parents, three elder sisters, my lovely nephews and nieces of my sisters. I hope they can keep learning when possible.

Lastly, I present this thesis to my wife Dr. Zhang Han, and my lovely daughter Hao Yiran (Qiqi). With them, I know that there is more important thing to enjoy in life than a piece of paper.

Contents

Abstract	i
Acknowledgments	iii
List of Figures	viii
List of Tables	xi
1 Introduction	1
1.1 Background and Motivation	1
1.2 Research Problems and Methodology	4
1.3 Summary of Contributions	6
1.4 Organization of the report	7
2 Literature Review	9
2.1 Online Learning	9
2.1.1 First-order Based Online Learning	11
2.1.2 Second-order Based Online Learning	12
2.1.3 Online Similarity Learning	13
2.2 Active Learning	14
2.3 Machine Learning for Crowdsourcing	17
2.4 Summary	19
3 Robust Active Learning with Expert Advice	21
3.1 Problem Setting and Background	22
3.2 Robust Active Learning with Expert Advice	23
3.2.1 Robust Active Exponentially Weighted Average Forecaster (RAE-WAF)	25
3.2.2 Robust Active Greedy Forecaster (RAGF).	25
3.3 Experimental Results	27
3.3.1 Experts and Compared Algorithms	28
3.3.2 Experimental Testbed and Setup	29
3.3.3 Evaluation on Normal Setting	30
3.3.4 Evaluation on Noise-added Setting	32
3.4 Conclusion	41

4	Active Crowdsourcing for Annotation	43
4.1	A Learning Framework for Active Crowdsourcing	44
4.1.1	Problem Setting	44
4.1.2	Overview of Framework	45
4.1.3	Problem Formulation	47
4.1.4	EW and GF Forecasters	48
4.1.5	Who to Label	49
4.1.6	When to Query	49
4.2	Experimental Results	52
4.2.1	Baseline Algorithms	52
4.2.2	Experimental Testbed and Setup	52
4.2.3	Experiments on Different Noisy Scenarios	54
4.2.4	Experiments on Who to Label	55
4.2.5	Experiments on When to Query	58
4.2.6	Experiments on the Framework	60
4.3	Summary	62
5	Second-order Online Active Learning	63
5.1	Second-order Online Active Learning	64
5.1.1	Problem Formulation	64
5.1.2	SOAL Algorithm	65
5.1.3	Theoretical Analysis	68
5.2	Experiments	73
5.2.1	Compared Algorithms and Experimental Testbed	73
5.2.2	Evaluation of Fixed Query Ratio	74
5.2.3	Evaluation of Varied Query Ratios	76
5.2.4	Evaluation of Parameter Sensitivity	79
5.2.5	Evaluation of Scalability and Efficiency	79
5.2.6	Application to Large-Scale Web Data Classification	81
5.3	Summmary	82
6	Online Active Learning for Relative Similarity Learning	83
6.1	Online Active Learning of Relative Similarity from Data Streams	84
6.1.1	Problem Formulation	84
6.1.2	PAAS: Passive-Aggressive Active Similarity learning	86
6.1.3	CWAS: Confidence-Weighted Active Similarity learning	87
6.1.4	Theoretical Analysis	89
6.2	Experiments	94
6.2.1	Datasets	95
6.2.2	Comparison Schemes	96
6.2.3	Evaluation of Fixed Query Ratio	97
6.2.4	Evaluation of Varied Query Ratios	98

6.2.5	Evaluation of Parameter Sensitivity	101
6.2.6	Evaluation of Efficiency and Scalability	103
6.3	Summary	105
7	Conclusions and Future Researches	107
7.1	Conclusions	107
7.2	Contributions	109
7.3	Future Directions	110
7.3.1	Online Active Learning for Concept Drifting Problems	110
7.3.2	Distributed Online Active Learning	111
7.3.3	Online Active Learning with Crowdsourcing	111
7.4	Summary	112
A	List of Publications	113
	List of Publications	113
	References	115

List of Figures

3.1	Per-round regret V.S. Query Ratio in Normal setting, where all the experts are comparably reliable.	33
3.2	Per-round regret V.S. Query Ratio in the normal setting, where all the experts are reliable.	34
3.3	Per-round regret V.S. Query Ratio in Noisy setting, where 2 noisy experts are added.	36
3.4	Per-round regret V.S. Query Ratio in the noisy setting, where 2 noisy experts are added.	37
3.5	Per-round regret V.S. Query Ratio in the noisy setting, where different number of noisy experts are added.	39
3.6	Per-round regret V.S. Query Ratio in the noisy setting, where a varied number of noisy experts are added.	40
4.1	A Framework of Budget Active Learning with Expert Advice (BALEA) .	46
4.2	Accuracy V.S. Number of Group-cheating Workers.	54
4.3	Accuracy V.S. Budget Ratio in Different Group-cheating Scenarios on <i>fashion</i> Dataset.	56
4.4	Accuracy V.S. Budget Ratio with 10 Extra Noisy Workers Added.	57
4.5	Accuracy V.S. Budget Ratio with 0 Extra Noisy Workers Added Based on EW Forecaster.	58
4.6	Accuracy V.S. Budget Ratio with 0 Extra Noisy Workers Added Based on GF Forecaster.	59
4.7	Accuracy of Framework 4.1 V.S. Budget Ratio with 10 Extra Noisy Workers Added.	61
5.1	Evaluation of accuracy against varied query ratio.	77
5.2	Evaluation of accuracy against varied query ratio.	78
5.3	Evaluation of accuracy with respect to different settings of parameters η and γ in Algorithm 5.	80
5.4	Evaluation of time cost w.r.t. varied query ratios.	80
5.5	Evaluation of accuracy against the varied query ratios for large-scale web data classification tasks.	82

6.1	AP V.S. varied query ratios. AP@10 denotes Average Precision at 10. .	99
6.2	mAP V.S. varied query ratios. AP@10 denotes Average Precision at 10.	100
6.3	Sensitivity of the parameters	102
6.4	Performance on the large-scale datasets	103
6.5	Time cost corresponding to varied querying ratio on large-scale datasets .	104

List of Tables

3.1	Datasets used in the experiments	29
3.2	Evaluation of on the Normal setting, where all experts are comparably reliable.	31
3.3	Evaluation of on the noisy setting, where 2 noisy experts are added. . .	35
4.1	Datasets used in the following experiments.	53
5.1	Summary of datasets in the experiments.	74
5.2	Evaluation of SOAL against other baselines with fixed query ratio about 20%.	75
6.1	Datasets used in the following experiments	95
6.2	Performance of algorithms on different datasets with the query ratio fixed to about 20%	97

Chapter 1

Introduction

This chapter first briefly introduces the background of Online Active Learning (OAL) and several challenges of existing OAL works in literature. After that, four solutions provided in this thesis are presented to address several challenges of existing OAL. At last, the contributions and organization of this thesis is presented.

1.1 Background and Motivation

Online active learning is an integrated research area between online learning and active learning in machine learning. Online learning, as an effective and efficient learning scheme, has been well studied in machine learning field [BLO62, CDKSs06, SSS07, CKD13]. In contrast to the offline learning in which the predictor is trained on the whole training data at once, the predictor in online learning is trained in a sequential manner. Taking the malicious URL detection for example, when receiving a URL (which is comprised by a set of features), the online predictor will make a prediction to decide whether the URL is malicious or not. After making the prediction, the learner will query an oracle to receive the true label of this URL. And after receiving this true label, the learner will suffer a loss depending on its prediction and the true label. Finally, the learner will update the model to make sure it will make better predictions on the following URLs. Here, the pair of question (URL) and answer (true label) is defined as an “example”, while the question and answer are denoted as “instance” and “label”, respectively.

The above example follows one of the online learning settings where the online learner directly makes prediction based on a URL. Another slight different setting of online

learning is named as online learning with expert advice [CBL06], where the online learner doesn't directly make prediction based on a URL, but makes prediction based on advices provided by a pool of experts (these experts could be workers in crowdsourcing platforms, or a set of trained models). Specifically, in the malicious URL detection problem, when a URL comes, a pool of experts makes predictions (also known as advices) of this URL. Based on these predictions made by experts, the online learner makes a final prediction of this URL. After making the prediction, both the learner and the experts will be told the true label (whether the URL is malicious or not) of this URL. After obtaining the true label, both the learner and each expert will suffer a loss depending on its prediction and the true label, respectively. Finally, the online learner will update the reliability of each expert based on his cumulative loss suffered and go on to process the next URL. The goal of this learner in the learning with expert advice setting is to make predictions as well as the best expert among all of the experts.

The malicious URL detection discussed above is a binary classification problem, in which a URL is classified as either a malicious URL or a normal one. However, online learning has also been very popular in other problems, such as multi-class classification [CDK09], regression [SHL14], metric learning [CSSB10a, HZHM15], multi-task learning [DLS06] and online convex optimization [SS11] and so on.

As it only processes one instance each time, online learning is very time-efficient for large-scale applications. One major limitation of the online learning is that it requires the true label of each instance. However, labeling the instances is usually costly and time-consuming, especially for large-scale applications. To alleviate this limitation, researchers have been interested in designing active learning approaches for online learning [HZHM15, LZH16, Aga13, CBGZ06]. Active learning, as a type of machine learning, has been widely studied [GFT15, SHX15, LZH16, CDP13, LM13, FZL12], in which a learning algorithm is able to actively decide which instance to query for the true label. The aim of active learning is to minimize the total number of queried instances and maintain comparable performance as the learning algorithm which queries all of the instances. The algorithms combining online learning and active learning are commonly named as Online Active Learning (OAL) [BEWB05, CZL⁺11, GZFX11, LZH16], and the active learning designed for online learning is typically known as selective sampling [Aga13, CCBG10, CBGZ06, FSST97, GALH13, MMK00, OCB11].

Different from online learning, OAL has an extra query strategy to decide whether to query the true label of current instances. With this query strategy, the online learner in OAL can make prediction as well as fully supervised online learner by only querying a subset of instances. Taking the malicious URL detection for example, after the online learner makes a prediction, the query strategy in OAL will decide whether to query the true label of current URL. If the query strategy recommends to query, the online learner will receive the true label by query an oracle (domain experts or other information sources which are believed to be high reliable to give accurate labels), and proceed as the online learner in the following steps. If the query strategy doesn't recommend to query, the learner will keep unchanged and process the next URL. By designing smart query strategy, the online active learner will only query a small subset of informative URLs, and achieve comparable performance as the online learner which queries the true labels of all instances.

It can be observed that OAL not only enjoys the efficiency and effectiveness of online learning, but also enjoys the low labeling cost of active learning. These attributes make the algorithms very suitable to real-world large-scale applications. For example, in the online social media, data often arrives sequentially with very high speed and high volume. With OAL, the data can be efficiently processed as the learner in OAL only processes one instance each time. Besides, with the active learning in OAL, it also can greatly reduce the labeling cost.

Although online active learning has been successfully adopted in many real-world applications, there are several critical limitations of the existing OAL. First, most OAL algorithms in the learning with expert advice setting either adopt the totally random query strategy or assume all of the experts are comparably reliable, which is usually unrealistic in practice. For example, in the crowdsourcing platforms, noisy workers commonly exist in real crowdsourcing platforms [CCLM15, YSL⁺15, BKMW14, TTHR⁺14, CLZ13]. Second, the performance of existing OAL algorithms is usually limited as most of them only consider the first-order information of current instance, such as margin value, which usually is not accurate enough to decide whether to query the true label or not. Finally, most of the exiting OAL algorithms mainly focus on the classification tasks, and few of them explore the other applications of OAL, such as in image retrieval, annotation task in crowdsourcing and so on.

1.2 Research Problems and Methodology

This thesis mainly focuses on designing new algorithms to address the limitations mentioned in the last section, and applying the new algorithms to real-world applications.

First, in the framework of active learning with expert advices, two robust OAL algorithms are proposed based on learning with expert advice setting. The proposed robust OAL algorithms can consistently achieve best performance both in the normal and noisy settings. Traditionally, learning with expert advice methods usually assumes a learner is always receiving the true label of every incoming training instance at the end of each trial. However, in real applications, acquiring the true labels of all instances from oracle can be costly or time consuming, especially for large-scale problems. Such as in the crowdsourcing task, in order to know the reliability of each work, the task organizers usually have to prepare a set of control questions whose answers are labeled by organizers or domain experts. In this scenario, designing active learning approaches to reduce the number of control questions would be attractive. In literature, several active learning with expert advice algorithms have been proposed [CBLS05, ZHZ13]. However, the proposed active learning approaches either adopt the totally random query strategy or assume the experts share comparable reliability. To tackle the limitations, two robust active learning with expert advice algorithms are proposed, named Robust Active Exponentially Weighted Average Forecaster (RAEWAF) and Robust Active Greedy Forecaster (RAGF). The proposed query strategies not only consider the prediction of each expert on the current instance, but also the historical performance of experts. Empirically, the proposed algorithms are validated by an extensive set of experiments both in the normal scenario and the noisy scenarios. The experimental results show that the proposed algorithms can robustly identify the valuable instances to query, and thus largely reduce the labeling cost.

Second, for the annotation task in crowdsourcing, most of the existing approaches allocate the tasks to workers without considering the different reliability of workers, and these allocation approaches may possibly waste money and time on noisy workers. Meanwhile, when aggregating the noisy annotations received from workers, most of existing approaches adopt the joint statistical approaches by assuming that all the workers share comparable reliability. These aggregation methods perform well when the assumption

is met, however, the methods also suffer most when noisy workers dominate the pool of workers. In order to wisely use the money on annotation task and robustly aggregate the noisy labels, it's necessary to explicitly measure the reliability of workers. To tackle this limitation, learning with expert (also known as worker in crowdsourcing) advice framework is adopted. Specifically, in order to wisely use the money, a reliability-based allocation approach is proposed, in which a reliable worker has more chances to receive an annotation task than a noisy worker. In order to explicitly measure the reliability of each worker, a typical method is scoring the workers based on the correctness of the workers on control questions (which have true labels given by reliable information sources). However, preparing too many control questions will leave us less resources on the annotation tasks which are our direct interest. In order to reduce the number of control questions, two robust OAL algorithms are proposed, which can robustly score the workers' reliability with limited control questions. The proposed algorithms are evaluated both on simulated and real crowdsourcing tasks. And the experiment results show that the proposed algorithms can achieve robust and promising performance both in the normal and noisy scenarios with limited budget.

Third, traditionally, OAL mainly focuses on first-order based query strategies [Aga13, GALH13, Dek12, CBGZ06], where margin-value of current instance is used. However, this first-order based query strategy may fail in the scenarios where the prediction model is not well trained thus the margin-value is not accurate. To overcome this, a second-order based OAL algorithm is proposed, where not only the margin-value is considered, but also the confidence of the predictive model is used. Specifically, a new framework of Second-order Online Active Learning (SOAL) is proposed, which fully exploits both first-order and second-order information to achieve high learning accuracy with low labeling cost. Both theoretical analysis and empirical studies are conducted to evaluate the proposed SOAL algorithm, which is further applied to two large-scale web data classification tasks. The encouraging results show clear advantages of the proposed algorithm over a family of state-of-the-art OAL algorithms.

Fourth, while all the above problems have focused on the online binary classification problems, the last problem is to study the online active learning algorithms for metric learning. Specifically, I study the online active algorithms for relative similarity learning,

which is an important learning scheme for information retrieval, and aims to learn a bi-linear similarity function from a collection of labeled instance-pairs. The learned function would assign a high similarity value for a similar instance-pair and a low value for a dissimilar pair. The existing algorithms usually assume the labels of all the pairs in data streams are always made available for learning. However, this is not always realistic in practice since the number of possible pairs is quadratic to the number of instances in the database, and manually labeling the pairs could be very costly and time consuming. To overcome the limitation, I propose a novel framework of active online similarity learning. Specifically, two new algorithms are proposed: (i) PAAS: Passive-Aggressive Active Similarity learning; (ii) CWAS: Confidence-Weighted Active Similarity learning. To validate the efficacy, the mistake bounds of these two algorithms are first proved, and followed by an extensive set of experiments on a variety of real-world data sets, in which the encouraging results validate the empirical effectiveness of the proposed algorithms.

In summary, the objectives of this thesis are mainly two-folds. In technique, we investigate the second-order information in online active learning to tackle the challenges met in existing algorithms, such as importing the historical performance of expert to improve the robustness of query strategy, and considering the confidence of the model to adaptively place query in order to quickly explore the distribution of the tasks, which can consistently improve the performance of the model in the early stage. In application, we successfully applied the robust active learning with expert advice algorithms into the crowdsourcing tasks, which can greatly reduce the number of questions and adaptively allocate the annotation tasks to the reliable workers, and we also successfully applied the online active learning algorithm into the similarity learning tasks, in which fully-supervised learning algorithm is highly costly and time-consuming.

1.3 Summary of Contributions

The contributions of this thesis are four folds as follows.

First, in the online learning with expert advice setting, two novel robust active learning algorithms are proposed, which can achieve consistent best performance both in the normal and noisy setting.

Second, based on the algorithms proposed in Chapter 1, an active annotation for crowdsourcing framework is proposed, which not only can robustly learn the reliability of each worker with limited number of control questions, but also adaptively allocate the tasks based on the learned reliability of each worker.

Third, to further investigate the second-order active learning on traditional online learning setting, a new second-order based online active learning algorithm is proposed, which can achieve the best performance by considering both the margin information and the confidence value of model.

Fourth, to reduce the labeling cost of learning algorithms in information retrieval tasks, two new OAL algorithms for the relative similarity learning tasks are proposed. The proposed algorithms can greatly reduce the labeling cost and speed up the learning phrase.

1.4 Organization of the report

In chapter 2, the literature on online learning, active learning, machine learning for crowdsourcing are reviewed.

In chapter 3, two robust OAL with expert advice algorithms (REWAF and RAGF) are presented followed by an extensive set of experiments both in the normal and noisy scenarios.

In chapter 4, based on proposed algorithms in chapter 3, a framework of online active annotation for crowdsourcing is proposed, and the proposed framework consists of two main components: (a) who to label the next task; (b) when to query the oracle for true label. The proposed framework is validated both in the real crowdsourcing datasets and simulated datasets.

In chapter 5, a Second-order based OAL (SOAL) algorithm is first proposed for typical online learning problem where the online learner directly access the instances. After that, the proposed algorithm is theoretically analyzed. At last, a series of experiments are conducted to evaluate the performance of the algorithm.

In chapter 6, the problem of OAL for similarity learning is first formulated, and then two Online Active Relative Similarity Learning algorithms are proposed. At last, the

proposed algorithms are evaluated both from the theoretical perspective and empirical perspective.

Finally, in chapter 7, this thesis is summarized and concluded, and some future directions are discussed.

Chapter 2

Literature Review

This chapter reviews literature on online learning, active learning and machine learning for crowdsourcing. For online learning, the basic procedure of online learning is firstly introduced, and then the online learning works are reviewed by summarizing them into two main categories: first-order based online learning, and second-order based online learning. Lastly, we review the literature of online learning for relative similarity learning. For active learning, we first summarize the active learning into four main categories, and then carefully review the works on online active learning, which is one special case of active learning. For the crowdsourcing, we mainly review the related works on machine learning for task allocation and annotation aggregation.

2.1 Online Learning

Online learning [BLO62, HWZ14, KZB⁺14] is a method of machine learning in which the online learner makes predictions and update its predictor on sequential data, which is opposite to batch machine learning (also known as offline machine learning) which updates the predictor by learning on the entire training data at once. Consider online binary classification as an example, where the learner aims to train a classifier $f_t(\mathbf{x}_t)$ with a sequence of instances $\mathbf{x}_t$, where $t \in \{1, 2, \dots, T\}$. At time t , the learner receives an instance $\mathbf{x}_t$ from the environment, and then makes a prediction of its class label $\hat{y}_t = \text{sign}(f_t(\mathbf{x}_t))$, where $\text{sign}(z)$ is a function that extracts the *sign* of real number z . After that, it often assumes the learner will receive the true label $y_t \in \{+1, -1\}$ from an

Algorithm 1 The Online Learning Procedure.

Input: $\mathbf{w}_0 \in \mathbb{R}^d$;**for** $t = 1, 2, \dots, T$ **do**

1. Receive instance: $\mathbf{x}_t \in \mathbb{R}^d$;
2. Predict $\hat{y}_t = \text{sign}(f_t(\mathbf{x}_t) = \mathbf{w}_t^\top \mathbf{x}_t)$ as the label of $\mathbf{x}_t$;
3. Receive true label $y_t \in \{-1, +1\}$;
4. Suffer loss ℓ_t , which depends on the difference between $\hat{y}_t$ and y_t ;
5. Update the $\mathbf{w}_t$ to $\mathbf{w}_{t+1}$ in f_t .

end for**Output:** $\mathbf{w}_{T+1}$.

oracle, and then be updated whenever necessary (e.g., $\hat{y}_t \neq y_t$ or other criteria). This online binary classification problem is formally summarized in Algorithm 1.

In Algorithm 1, the performance of the learner $f_t(\mathbf{x}_t)$ is measured by the cumulative loss $\sum_{t=1}^T \ell_t$ suffered. It should be noted that there are several loss functions that can be adopted in practice. Here are the most popular ones:

- 0-1 loss: $\ell_t = I(\hat{y}_t \neq y_t)$,
- Hinge loss: $\ell_t = \max(0, 1 - f_t(\mathbf{x}_t) \cdot y_t)$,
- Squared loss: $\ell_t = (f_t(\mathbf{x}_t) - y_t)^2$,

where I is an indicator function, *0-1 loss* and *Hinge loss* are usually adopted by classification problems, and *Squared loss* is usually adopted by regression problems.

According to the number of information maintained by the online learner, the existing online learning algorithms can be summarized in to two main categories: first-order based online learning [BLO62, CDKSs06, LL02, CS03], and second-order based online learning [CBCG05, DCP08, CKD13]. For the first-order based online learning, the online learner only maintains one category of information: $\mathbf{w}_t$. For the second-order based online learning, the online learner not only maintains $\mathbf{w}_t$, but also maintains second-order information [CKD13, CDP08], such as covariance matrix Σ_t of the weight vector $\mathbf{w}_t$. These two categories of works in literature are reviewed respectively as follows.

2.1.1 First-order Based Online Learning

Among the first-order based online learning algorithms, Perceptron [BLO62] is the most well-known and simplest one. This algorithm maintains a weight vector $\mathbf{w}$ which is initialized as zero, and then make a prediction $\hat{y}_t = \text{sign}(f_t(\mathbf{x}_t))$, where $f_t(\mathbf{x}_t) = \mathbf{w}_t^\top \mathbf{x}_t$ is a linear function. After that, the true label y_t of this instance $\mathbf{x}_t$ is revealed, and a 0-1 loss is suffered. The weight vector $\mathbf{w}_t$ is updated by an additive form $\mathbf{w}_{t+1} = \mathbf{w}_t + \mathbb{1}_{\hat{y}_t \neq y_t} y_t \mathbf{x}_t$, where $\mathbb{1}_z$ is an indicator function, which outputs 1 if z is valid, otherwise 0. Similar as Perceptron, a large margin Perceptron algorithm was also proposed [FSP99] by using a combination of the weighted vectors learned during the whole learning process. Shai Shalev-Shwartz and Yoram Singer [SSS05] also proposed Ballseptron, which updates the weight vector whenever a margin error suffered (when $y_t(\mathbf{w}_t^\top \mathbf{x}_t)/\|\mathbf{w}_t\|^2$ is less or equal one, the weight vector will be updated).

Although Perceptron algorithm is extremely simple and also enjoys theoretical promise for linearly separable datasets, the performance of this algorithm is limited due to the passive updating rule (the correctly predicted instances are dropped). To tackle this issue, several aggressive online learning algorithms are proposed [CDKS06, LL02, CS03], where the weight vector is updated not only on the wrongly predicted instance but also on the correctly predicted instance whenever the margin value $|\mathbf{w}_t^\top \mathbf{x}_t|$ is small. By using hinge loss and different regularizations, Crammer et.al. [CDKS06] proposed three variant Passive-Aggressive based online learning algorithms (PA, PA-I and PA-II) with different updating rules. With the similar idea, ROMMA [LL02] and MIRA [CS03] are another two first-order based online learning algorithms.

The updating rule adopted in the above algorithms is known as additive rule, where the weight vector is updated with an additive form. It should be noted that besides the additive learning algorithms discussed above, there are also several algorithms which adopt the multiplicative learning scheme, such as Weighted Majority Voting (WMV) [LW89, CBFH⁺97, HWB98], Winnow [Lit88], Exponentiated Gradient [KW97] and so on. Here we mainly discuss the Weighted Majority Voting algorithm as it is closest to this thesis. Different from the Algorithm 1, in step 1, WMV algorithm receives advices or predictions from a pool of maintained experts, who could be real people or any well trained learning algorithms. After receiving these advices, WMV algorithm outputs the

final prediction by considering these advices and the weight of each expert (the weight can be thought as the reliability of each expert). Once the true label is revealed, WMV suffers some loss and updates the weight of each expert. It should be noted that the weight vector in WMV corresponds to the pool of experts, and the weight vector in Algorithm 1 corresponds to the instance $\mathbf{x}_t$.

It can be observed that the first-order based online learning algorithms are simple and efficient, however, the performance of these algorithms is usually limited by the amount of information available. To improve the performance, recently, second-order based online learning algorithms are studied in literature.

2.1.2 Second-order Based Online Learning

Second-order based online learning is a group of online learning algorithms which maintain not only the first-order information but also the second-order information [CBCG05, DCP08, CKD13]. Among those algorithms, Second-order Perceptron (SOP) [CBCG05] is first proposed based on the first-order version of Perceptron [BLO62] algorithm, which can be viewed as a variant of the whitened Perceptron algorithm, where the authors explore the online correlation matrices of the previously seen instances to achieve the whitened effect. By modelling the weight vector with a Gaussian distribution, Confidence Weighted (CW) [DCP08] algorithm incorporates the second-order information: covariance matrix of the weight vector. By doing this, the weight with a large variance value will be updated more aggressively, and the weight with a small variance value will be updated more passively. In another words, the covariance matrix could be treated as an adaptive learning rate. This updating rule make the CW algorithm outperform most of the first-order based algorithms. Although CW enjoys formal guarantees in the mistake-bound model, it can overfit in certain scenarios due to its aggressive update rules based on a separable data assumption. To overcome this limitation, Crammer et. al. [CKD13] proposed the AROW algorithm with the similar idea as CW by considering two adaptive regularization terms for each training example, which consist a Kullback-Leibler divergence between the Gaussian distributed weight vectors and a confidence penalty of vectors. By doing this, AROW can handle both the noisy and non-separable cases. The directly added loss and confidence regularization make AROW lose an important property of CW algorithm,

i.e., Adaptive Margin property [CC12]. Following the similar idea of soft margin support vector machines, Soft Confidence-Weight algorithm (SCW) [WZH12] is proposed to assign adaptive margins for different samples, which enables SCW to gain extra efficiency and effectiveness. Similar second-order online active learning algorithms also include Normal HERD [CL10], NAROW [OC10] and so on. In general, the second order algorithms are more accurate, converge faster.

2.1.3 Online Similarity Learning

In the above subsections, the online learning works for classification tasks are reviewed in literature. However, it should be noted that there are a large number online learning works for the other tasks, such as online learning for multi-class classification [CS03, FSSSU06], online learning for ranking problem [CS02, CS05, SS07] and so on. Among these tasks, online similarity learning [SSSN04, DKJ⁺07, JKDG09], also known as online metric learning, is one of the most challenging tasks.

As a method of supervised machine learning, similarity learning, also known as metric learning, aims to learn from labeled examples a similarity function that can measure how similar two objects are. Similarity learning is closely related to information retrieval, in which the retrieved results can be ranked by the learned similarity function. Most existing similarity learning algorithms have been focused on Distance Metric Learning (DML) [YJ06] for learning a Mahalanobis distance through a Positive Semi-Definite (PSD) matrix [JWZ09, SJ03, SSSN04]. Weinberger et al. [WBS05] proposed a large margin nearest-neighbour classifier to address the DML problem in the context of ranking. Globerson and Roweis [GR05] proposed to address the positive constraints in a supervised setting. However, most of these algorithms are batch-based similarity learning algorithms, in which the similarity function has to be updated with the entire data whenever new examples come. This updating rule makes those batch-based similarity learning algorithms inefficient, especially for large-scale applications, such as a search engine system.

To reduce the computation cost, Davis et al. [DKJ⁺07] proposed to learn the Mahalanobis distance function with an online learning scheme based on LogDet-regularization with different loss functions. Although the proposed algorithm is fast and scalable, it still

requires to impose the PSD constraint on the matrix A which parameterizes the Mahalanobis distance. As it's well known that PSD constraint usually has high computation cost, and this makes the algorithm inappropriate for large-scale problems. To overcome this limitation, researchers have been focusing on the bilinear form similarity functions [CSSB10b, CDKSs06], where the matrix A is not required to be PSD. Along this direction, Chechik et al. [CSSB10b] proposed the first-order relative similarity learning algorithm “OASIS” based on the first-order based online learning algorithms [CDKSs06, ZJYH11]. Recently, Crammer and Chechik [CC12] proposed a second-order relative similarity learning algorithm named as “AROMA” based on second-order based online learning algorithm [CKD09, ZHJ11].

Although both the first-order and second-order based online learning algorithms are efficient and effective for large-scale problems, all of them assume that all of the instances are well labeled. However, to label all of the instances is usually time-consuming and costly, especially for large-scale problems, which is very common in the big data era. To tackle this issue, researchers have been studying various methods to reduce the labeling cost. Typically, there are two main methods: active learning [Ols09, Set10a, FZL12] and crowdsourcing [Bra13, GWKP11], in which related works are reviewed in the following sections, respectively.

2.2 Active Learning

Active learning is a family of machine learning techniques in which the aim of a learner is to reduce the total number of labeled instances without sacrificing its performance too much by actively query the informative instances. Traditionally, active learning has been extensively studied in the batch or offline machine learning settings [Ols09, Set10a, FZL12], where it assumes that all of the training data are available to be selected by active learning at each iteration. The active learning algorithms in the offline setting are commonly named as batch-based active learning as it usually selects several instances to query at each iteration. Existing batch-based active learning techniques could be generally grouped into four categories: (a) uncertainty-based [LC94, TK02, BDL09, SYTC11, GSS13] active learning in which instances are ordered and selected by their

uncertainty values (such as the margin values made by the predictor); (b) searching through the hypothesis space [FM97, Bri03, Han12, HLYG13], and the instances on which the hypothesis mostly disagrees with each other are selected; (c) selecting instances which can minimize the expected error and variance on the pool of unlabeled instances [RM01, GS08, BDL09, GZHD12]; (d) exploiting the structure information among the instances [XYT⁺03, SPI08, SHV⁺10, ESYS13, NWHD13], such as selecting the centroids of each cluster to query the label.

The batch-based active learning assumes that all the unlabeled data are available at each iteration, and the active learning algorithm can select several instances to query. However, these requirements are hard to be met in the online setting, in which only one instance is available at each iteration. And there is no way to know the entire unlabeled instances before the learning process. What's more importantly, the received instances would not be stored, which makes the information that can be accessed limited. These characteristics of online learning make most of the batch-based active learning algorithms failed, and researchers have been studying the new active learning algorithms for online learning, named as Online Active Learning (OAL), also known as selective sampling [CBCG03, DGS10, Aga13].

Algorithm 2 shows the procedure of OAL. It can be observed that it typically follows the procedure of Algorithm 1. The difference is that in Algorithm 2 there is an extra step 3, which decides whether to query the true label of instance $\mathbf{x}_t$. In the online learning procedure shown in Algorithm 1, the label of each instance will be always queried no matter how useless the instance is. However, in the OAL procedure in Algorithm 2, a query strategy in step 3 is added to decide when it is worth to query the true label of the instance in order to save the labeling cost. If the OAL algorithm decides to skip the current instance, it will keep the current model f_t unchanged and continue to proceed the next instance.

In the literature of OAL, uncertainty-based, also known as margin-based query strategy, is the most popular one [CBGZ06, ZH13, LZH16]. Cesa-Bianchi et al. [CBGZ06, CBGZ06] first proposed margin-based active learning based on Perceptron algorithm [BLO62, CBCG05]. Specifically, at each iteration, the algorithms first compute a margin value p_t of the current instance $\mathbf{x}_t$ with the formula $p_t = \mathbf{w}_t^\top \mathbf{x}_t$, and then computes a Bernoulli

Algorithm 2 The Online Active Learning (OAL) Procedure.

Input: $\mathbf{w}_0 \in \mathbb{R}^d$;**for** $t = 1, 2, \dots, T$ **do**1. Receive instance: $\mathbf{x}_t \in \mathbb{R}^d$;2. Predict $\hat{y} = \text{sign}(f_t(\mathbf{x}_t) = \mathbf{w}_t^\top \mathbf{x}_t)$ as the label of $\mathbf{x}_t$;

3. Decide whether to query the true label;

if Yes **then**4. Query the true label $y_t \in \{-1, +1\}$;5. Suffer loss ℓ_t , which depends on the difference between $\hat{y}_t$ and y_t ;6. Update the prediction function f_t to f_{t+1} ;**else**7. Keep f_t unchanged;**end if****end for****Output:** $\mathbf{w}_{T+1}$.

random variable $Z_t \in \{0, 1\}$ with probability $\delta/(\delta + |p_t|)$, where δ is a input parameter to control the number of instances to be queried, in practice, this parameter can be tuned based on our budget. If $Z_t = 1$, the algorithm will query the true label of instance $\mathbf{x}_t$ and skip the instance otherwise. Here, p_t could be treated as the confidence of the current model on the prediction. If p_t is small, it means that the current model is less confident on the prediction, and the probability of instance $\mathbf{x}_t$ to be queried will be high. Based on the similar idea, Zhao and Hoi [ZH13] proposed online active learning algorithm for the malicious URL detection, Lu et al. proposed the online passive-aggressive active learning based on PA algorithms [CDKSs06], and Krammer [Cra14] proposed a more aggressive algorithm DAGGER based on the second-order Perceptron [CBCG05].

Based on Regularized Least Squares (RLS), Cesa-Bianchi et al. [CBGO09, OCB11] proposed the Bound on Bias Query (BBQ) strategy which trades off between the cumulative regret and the number of queried labels. With the similar idea, Gu et al. [GALH13] proposed the SSLGC algorithm for the graph classification. Dekel et al. [DGS10, Dek12] proposed a robust selective sampling strategy based on Ridge-Regression-like algorithm both for the single teacher and multiple teachers settings.

In the learning with expert advice framework [Vov95, CBL06, ZHZ13], Cesa-Bianchi and Lugosi [CBL05] first proposed a uniform random query strategy with parameter a manually defined parameter, which controls the number of labeled instances. Later,

Zhao et al. [ZHZ13] proposed another new query strategy by considering the difference of advices on the current instance.

It also should be noted that there is a series of active learning work on agnostic setting [BBL09, DHM07, Han07, BHLZ10]. However, these works are usually computationally intractable and infeasible to large-scale big datasets. These algorithms will be discussed in detail in the following section if used.

Although these proposed OAL algorithms have been proved effective in reducing labeling cost, they have several limitations. First, most of the algorithms are inferior due to the underline learner models adopted. Second, all of these algorithms adopt the margin-based query strategies, which may not be robust as the margin-value is not accurate when the learner is not well trained, such as in the early learning period. Third, most of the proposed online active learning algorithms are focused on the classification tasks, and few of them explore the application of OAL on the other tasks, such as information retrieval, crowdsourcing. These major limitations motivate this study to design new OAL algorithms and study the possibilities of OAL on new applications.

2.3 Machine Learning for Crowdsourcing

Crowdsourcing, as an approach to collect large-scale and cheap labeled data, has attracted great attention in recent years [Bra13, DRH11, KOS14, LYZ13, VGK⁺14, ZCZJ14]. There are two main challenges in crowdsourcing in order to receive high quality labeled data. The first one is how to wisely allocate the tasks to reliable workers thus receive high quality labels at the first place. The second question is how to aggregate the labels received from worker in order to obtain accurate finalized labels of all data.

For the first challenge, Karger et.al. [KOS11b, KOS14] formalized the tasks allocation as a problem to designing a bipartite graph $G(\{\mathbf{X}\} \cup J, E)$, where $\mathbf{X}$ is the tasks, J is a pool of workers, and E are a set of edges. Each edge corresponds to a task-worker assignment. In the first step, the authors proposed to adopt the scheme known as *configuration model* [Bol98, RU08] to allocate the tasks. In the second-step, to infer the target label for each task, the authors proposed an interactive algorithm inspired by the standard Belief Propagation (BP) algorithm. Karger et.al. [KOS11a] also proposed

an Low-rank Approximation for the annotation matrix L for the above second step. Inspired by Multi-Armed Bandit (MAB) problem, Abraham et. al. [AAKS13] proposed an adaptive bandit survey model for task allocation.

For the second challenge, the most common method to infer the ground label is majority voting [SPI08]. However, this approach would suffer when the number of noisy annotators is larger than the number of reliable annotators, which is common in the crowdsourcing platforms. The idea of majority voting approach has taken one step further by looking at the consistency between annotators. Most of recent work follows this line by modelling the true label, workers' abilities, and the task's difficulty as the latent variables within graph models, by which the problem is transferred to inferring latent variables with EM-style algorithms. Dawid and Skene [DS79] modelled the reliabilities of workers using a confusing matrix. Whitehill et.al. [WRW⁺09] further modelled both worker reliability and image difficulty. However, the algorithm would be worse than the majority voting when the variance of reliabilities between workers is high [WP10]. To tackle this issue, Welinder et al. [WP10] further introduced a high-dimensional concept of image difficulty and annotator bias. These aggregation algorithms could achieve promising performance without explicitly evaluating workers' reliability by querying golden label of sample instances. However, these algorithms would suffer in noisy scenarios, such as group of people cheating together to give the same labels [EdV13].

It should be noted that most of existing methods are proposed for either task allocation problem or aggregation problem. Few of them combined together. In this thesis, based on the learning with expert (also known as worker in crowdsourcing) advice framework, an active annotation for crowdsourcing framework is proposed to systematically tackle both the task allocation and aggregation problems. Specifically, in order to wisely use the money, a reliability-based allocation approach is proposed, in which a reliable worker has high chance to receive an annotation task than a noisy worker. The proposed method is intuitive and easy to be used in real crowdsourcing platform. In order to explicitly measure the reliability of each worker, a typical method used is scoring the workers based on the correctness of the workers on control questions (which is commonly adopted by exiting crowdsourcing platforms, such as Amazon Mechanical Turk, Baidu Test). However, preparing too many control questions will leave us less resources on the annotation tasks

which are our direct interest. In order to reduce the number of control questions, two robust OAL algorithms are proposed, which can robustly score the workers' reliability with limited control questions.

2.4 Summary

In this chapter, three categories of works in literature are reviewed, online learning, active learning and machine learning for crowdsourcing.

In the online learning section, the typical procedure of online learning is firstly discussed, and then the existing online learning algorithms are summarized into two groups to be reviewed: first-order based online learning and second-order based online learning. At last, the application of online learning in similarity learning is discussed.

For the active learning, batch-based active learning, which is summarized into four categories, is first discussed, and then one particular of active learning, OAL in literature is reviewed.

In the section of machine learning for crowdsourcing, two main problems in which the machine learning researchers are interested are discussed: task allocation and label aggregation.

Chapter 3

Robust Active Learning with Expert Advice

In literature, learning with expert advice methods usually assume a learner is always receiving the true label of every incoming training instance at the end of each trial. However, in real applications, acquiring the true labels of all instances from oracle can be costly or time consuming, especially for large-scale problems. In this chapter, this problem is addressed with active learning with expert advice, where the outcome of an instance is disclosed only when it is requested by the online learner. The goal is to minimize the number of queried instances while training an online learning model without sacrificing the performance. To address this challenge, two robust active learning with expert advice algorithms, named Robust Active Exponentially Weighted Average Forecaster (RAEWAF) and Robust Active Greedy Forecaster (RAGF) are proposed. The efficacy of the proposed algorithms is first validated and followed by an extensive set of experiments both in the normal and noisy scenarios.

The rest of this chapter is organized as follows. Section 1 introduces the problem setting of learning with expert advice and two well-known forecasters. Section 2 presents two robust online active learning with expert advice algorithms, RAEWAF and RAGF. Section 3 shows our empirical results and application in crowdsourcing and Section 4 concludes this work.

3.1 Problem Setting and Background

Online classification has been extensively studied in machine learning in the past few years [BLO62, CDKSs06, ZHJ11, HWZ14] for different problems, like sentiment detection [LHCJ10], cost-sensitive classification [WZH14], feature selection [WHZL13] and etc. A typical learning with expert advice task is trained as follows. Consider an unknown sequence of instances $\mathbf{x}_1, \dots, \mathbf{x}_T \in \mathbb{R}^d$, a decision maker termed as “forecaster” aims to predict the outcomes (e.g., class labels) of every incoming instance $\mathbf{x}_t$. The forecaster sequentially computes its predictions based on the advices from a set of N reference named as “experts”. Specifically, at the t -th round, after receiving an instance $\mathbf{x}_t$, the forecaster first accesses the predictions made by a set of experts $\{f_{i,t} : \mathbb{R}^d \rightarrow [0, 1] | i = 1, \dots, N\}$, and then based on which the forecaster computes its own prediction $p_t \in [0, 1]$. After p_t is computed, the true outcome $y_t \in \{0, 1\}$ is disclosed.

With the true outcome revealed from the environment or an oracle, the performance of the forecaster and experts can be scored by some nonnegative loss function between their predictions $p_t, f_{i,t}$ and the true label y_t , e.g., the absolute loss that is defined as $\ell(p_t, y_t) = |p_t - y_t|$. The cumulative loss suffered by each expert and the forecaster respectively can be further calculated as follows:

$$L_{i,t} = \sum_{j=1}^t \ell(f_i(\mathbf{x}_j), y_j), \quad L_t = \sum_{j=1}^t \ell(p_j, y_j)$$

The loss difference between the forecaster and the expert is known as the “regret”, i.e.,

$$R_{i,t} = L_t - L_{i,t}, i = 1, \dots, N \quad (3.1)$$

The goal of learning the forecaster is to make the regret with respect to each expert as small as possible, which is equivalent to minimizing the overall regret R_T , i.e.,

$$R_T = \max_{1 \leq i \leq N} R_{i,T} = L_T - \min_{1 \leq i \leq N} L_{i,T} \quad (3.2)$$

In general, an ideal forecaster should achieve vanishing per-round regret, i.e.,

$$R_T = o(T) \iff \lim_{T \rightarrow \infty} \frac{1}{T} \left(L_T - \min_{1 \leq i \leq N} L_{i,T} \right) = 0$$

The above property is known as the Hannan consistency [Han57]. A forecaster satisfying this property is called a Hannan-consistent forecaster [Han57, CBL06].

To solve the above task of learning with expert advice, a natural framework is based on the weighted average prediction strategy. Specifically, at time t , the forecaster makes its own prediction as:

$$p_t = \frac{\sum_{i=1}^N w_{i,t-1} f_i(\mathbf{x}_t)}{\sum_{i=1}^N w_{i,t-1}} \quad (3.3)$$

where $w_{i,t-1}$ is the performance weight assigned to the i -th expert at time $t-1$. The intuitive idea of learning the performance weights is to assign large weights for those experts of low regrets/loss and small weights for those of high regrets/loss.

A special case of learning with expert advice is called ‘‘Exponentially Weighted Average Forecaster’’ (EWA). In particular, by choosing $w_{i,t-1} = \exp(-\eta L_{i,t-1}) / \sum_{j=1}^N \exp(-\eta L_{j,t-1})$, the EWA strategy makes the following prediction:

$$p_t = \frac{\sum_{i=1}^N \exp(-\eta L_{i,t-1}) f_i(\mathbf{x}_t)}{\sum_{i=1}^N \exp(-\eta L_{i,t-1})}, \quad (3.4)$$

where $\eta > 0$ is the learning rate.

In addition to the weighted average forecaster, another kind of forecaster, known as the ‘‘Greedy Forecaster’’ (GF), is also considered in this work. GF makes the following prediction:

$$p_t = \pi_{[0,1]} \left(\frac{1}{2} + \frac{1}{2\eta} \ln \frac{\sum_{i=1}^N \exp(-\eta L_{i,t-1} - \eta \ell(f_i(\mathbf{x}_t), 1))}{\sum_{i=1}^N \exp(-\eta L_{i,t-1} - \eta \ell(f_i(\mathbf{x}_t), 0))} \right),$$

where $\pi_{[0,1]}(\cdot) = \max(0, \min(1, \cdot))$.

3.2 Robust Active Learning with Expert Advice

In this section, two robust active learning with expert advice algorithms are proposed for online active learning problems. Unlike the above regular learning with expert advice task where the outcome of every incoming instance is *always* revealed to the forecaster, in an active learning with expert advice task, the outcome of an incoming instance is *only* revealed whenever the learner has made a request for acquiring the label from the

environment or oracle. In this section, a framework of *active forecasters* to tackle the challenging task of active learning with expert advice is developed.

In the following sections, $z_s \in \{0, 1\}$, $s = 1, \dots, t$ denotes if an active forecaster has decided to request the class label of an incoming instance received at s -th trial. $\widehat{L}_{i,t}$ denotes the loss function experienced by the active learner w.r.t. the i -th expert, i.e., $\widehat{L}_{i,t} = \sum_{s=1}^t \ell(f_i(\mathbf{x}_s), y_s) z_s$.

Hence, the class label for the t -th example predicted by the active forecaster, denoted by $\widehat{p}_t$, is computed as $\widehat{p}_t = \pi_{[0,1]}(\bar{p}_t)$, where $\bar{p}_t$ is computed by different approaches for different forecasters:

$$\begin{aligned} \bar{p}_t &= \frac{\sum_{i=1}^N \exp(-\eta \widehat{L}_{i,t-1}) f_i(\mathbf{x}_t)}{\sum_{i=1}^N \exp(-\eta \widehat{L}_{i,t-1})} \text{ (EWAF);} \\ \bar{p}_t &= \frac{1}{2} + \frac{1}{2\eta} \ln \frac{\sum_{i=1}^N \exp[\eta(-\widehat{L}_{i,t-1} - \ell(f_i(\mathbf{x}_t), 1))]}{\sum_{i=1}^N \exp[\eta(-\widehat{L}_{i,t-1} - \ell(f_i(\mathbf{x}_t), 0))]} \text{ (GF).} \end{aligned} \quad (3.5)$$

In the above formula of EWAF, since $\bar{p}_t \in [0, 1]$, $\widehat{p}_t = \pi_{[0,1]}(\bar{p}_t) = \bar{p}_t$ always exists.

The key challenge for active learning with expert advice is to decide when the active forecaster should or should not make a request to acquire the true class label w.r.t. an incoming instance from the environment or an oracle. A naive solution is to consider a random sampling approach, which however may not be effective enough (this will be considered as a baseline for comparison in our empirical study). To tackle this challenge, Zhao et. al. [ZH13] proposed AEWAF and AGF algorithms by considering the difference of advice from experts. The basic idea is to make sure a small difference between $\widehat{p}_t$ based on active learner and p_t based on passive learner (EWAF or GF).

The proposed algorithms AEWAF and AGF are found promising in reducing labeling cost while achieving comparable empirical performance in a normal scenario where all the experts generally perform comparable well. However, the proposed query strategies, which treat every expert equally, implicitly rely on a strong assumption, i.e., the experts must have competing performance. This may not be always realistic for many real-world applications, where some experts may perform well and some may perform badly or even are completely noisy. For example, in crowdsourcing, some workers may know very well about an annotation task, and some workers may not have background knowledge with respect to the annotation task. Even worse, some workers are totally noisy [KHH12].

To tackle this issue, a novel query strategy, which attempt to weigh the good experts' advices more importantly than that of poor or noisy experts, is proposed. More specifically, the cumulative performance of each expert are imported into the query strategy, so as to give good experts' advice higher weights than the poor or noisy ones.

3.2.1 Robust Active Exponentially Weighted Average Forecaster (RAEWAF)

The theorem below based on EWAFF guarantees a small difference between $\hat{p}_t$ and p_t .

Theorem 1 *For a small constant $\delta > 0$, if the following condition is satisfied, then*

$$|p_t - \hat{p}_t| \leq \delta$$

$$\left| \frac{\sum_{i=1}^N \sum_{j=1}^N \gamma_{i,j,t-1} (f_i(\mathbf{x}_t) - f_j(\mathbf{x}_t))}{\sum_{i=1}^N \sum_{j=1}^N \gamma_{i,j,t-1}} \right| \leq \delta, \quad (3.6)$$

where $\gamma_{i,j,t-1} = \exp \left(-\eta \left[\hat{L}_{i,t-1} + \hat{H}_{i,t-1} + \hat{L}_{j,t-1} \right] \right)$.

Proof: The difference between p_t and $\hat{p}_t$ is computed as follows:

$$\begin{aligned} & |p_t - \hat{p}_t| \\ &= \left| \frac{\sum_{i=1}^N \exp(-\eta L_{i,t-1}) f_i(\mathbf{x}_t)}{\sum_{i=1}^N \exp(-\eta L_{i,t-1})} - \frac{\sum_{i=1}^N \exp(-\eta \hat{L}_{i,t-1}) f_i(\mathbf{x}_t)}{\sum_{i=1}^N \exp(-\eta \hat{L}_{i,t-1})} \right| \\ &= \left| \frac{\sum_{i=1}^N \exp(-\eta (\hat{L}_{i,t-1} + \hat{H}_{i,t-1})) f_i(\mathbf{x}_t)}{\sum_{i=1}^N \exp(-\eta \hat{L}_{i,t-1}) \exp(-\eta \hat{H}_{i,t-1})} - \frac{\sum_{i=1}^N \exp(-\eta \hat{L}_{i,t-1}) f_i(\mathbf{x}_t)}{\sum_{i=1}^N \exp(-\eta \hat{L}_{i,t-1})} \right| \\ &= \left| \frac{\sum_{i=1}^N \sum_{j=1}^N \gamma_{i,j,t-1} (f_i(\mathbf{x}_t) - f_j(\mathbf{x}_t))}{\sum_{i=1}^N \sum_{j=1}^N \gamma_{i,j,t-1}} \right| \leq \delta. \end{aligned}$$

3.2.2 Robust Active Greedy Forecaster (RAGF).

The theorem below gives the confidence condition for RAGF to ensure a small difference between $\hat{p}_t$ and p_t .

Theorem 2 For a small constant $\delta > 0$, if the following condition is satisfied, then $|p_t - \hat{p}_t| \leq \delta$.

$$\left| \frac{1}{2\eta} \ln \frac{\sum_{i=1}^N \mu_{i,t} \exp(2\eta(f_i(\mathbf{x}_t) - \bar{p}_t))}{\sum_{i=1}^N \mu_{i,t}} \right| \leq \delta, \quad (3.7)$$

where $\mu_{i,t} = \exp\left(-\eta[\hat{L}_{i,t-1} + \ell(f_i(\mathbf{x}_t), 0) + \hat{H}_{i,t-1}]\right)$, $i \in [N]$.

Proof:

p_t can be bounded from above as follows:

$$\begin{aligned} p_t &= \pi_{[0,1]} \left(\frac{1}{2} + \frac{1}{2\eta} \ln \frac{\sum_{i=1}^N \exp(-\eta L_{i,t-1} - \eta \ell(f_i(\mathbf{x}_t), 1))}{\sum_{i=1}^N \exp(-\eta L_{i,t-1} - \eta \ell(f_i(\mathbf{x}_t), 0))} \right) \\ &= \pi_{[0,1]} \left(\frac{1}{2} + \frac{1}{2\eta} \ln \frac{\sum_{i=1}^N \exp(-\eta(\hat{L}_{i,t-1} + \hat{H}_{i,t-1}) - \eta \ell(f_i(\mathbf{x}_t), 1))}{\sum_{i=1}^N \exp(-\eta(\hat{L}_{i,t-1} + \hat{H}_{i,t-1}) - \eta \ell(f_i(\mathbf{x}_t), 0))} \right) \\ &\leq \pi_{[0,1]} \left(\frac{1}{2} + \frac{1}{2\eta} \ln \frac{\sum_{i=1}^N \exp(-\eta \hat{L}_{i,t-1} - \eta \ell(f_i(\mathbf{x}_t), 1))}{\sum_{i=1}^N \exp(-\eta \hat{L}_{i,t-1} - \eta \ell(f_i(\mathbf{x}_t), 0))} \right) \\ &\quad + \pi_{[0,1]} \left(\frac{1}{2\eta} \ln \frac{[\sum_{i=1}^N \alpha_{i,t} \exp(-\eta \hat{H}_{i,t-1})]}{[\sum_{i=1}^N \beta_{i,t} \exp(-\eta \hat{H}_{i,t-1})]} \right) \\ &\leq \hat{p}_t + \pi_{[0,1]} \left(\frac{1}{2\eta} \ln \frac{[\sum_{i=1}^N \alpha_{i,t} \exp(-\eta \hat{H}_{i,t-1})]}{[\sum_{i=1}^N \beta_{i,t} \exp(-\eta \hat{H}_{i,t-1})]} \right) \\ &\leq \hat{p}_t + \frac{1}{2\eta} \ln \frac{\sum_{i=1}^N \beta_{i,t} \exp(2\eta(f_i(\mathbf{x}_t) - \bar{p}_t) - \eta \hat{H}_{i,t-1})}{\sum_{i=1}^N \beta_{i,t} \exp(-\eta \hat{H}_{i,t-1})} \\ &= \hat{p}_t + \frac{1}{2\eta} \ln \frac{\sum_{i=1}^N \mu_{i,t} \exp(2\eta(f_i(\mathbf{x}_t) - \bar{p}_t))}{\sum_{i=1}^N \mu_{i,t}} \end{aligned}$$

where

$$\begin{aligned} \alpha_{i,t} &= \frac{\exp\left(-\eta\left[\hat{L}_{i,t-1} + \ell(f_i(\mathbf{x}_t), 1)\right]\right)}{\sum_{j=1}^N \exp\left(-\eta\left[\hat{L}_{j,t-1} + \ell(f_j(\mathbf{x}_t), 1)\right]\right)} \quad \forall i \in [N], \\ \beta_{i,t} &= \frac{\exp\left(-\eta\left[\hat{L}_{i,t-1} + \ell(f_i(\mathbf{x}_t), 0)\right]\right)}{\sum_{j=1}^N \exp\left(-\eta\left[\hat{L}_{j,t-1} + \ell(f_j(\mathbf{x}_t), 0)\right]\right)} \quad \forall i \in [N], \end{aligned}$$

and $\alpha_{i,t} = \beta_{i,t} \exp(2\eta(f_i(\mathbf{x}_t) - \bar{p}_t))$.

It's clear that if

$$\left| \frac{1}{2\eta} \ln \frac{\sum_{i=1}^N \mu_{i,t} \exp(2\eta(f_i(\mathbf{x}_t) - \bar{p}_t))}{\sum_{i=1}^N \mu_{i,t}} \right| \leq \delta \quad (3.8)$$

$|p_t - \hat{p}_t| \leq \delta$ exists.

From Theorem 1 and 2, It can be observed that, at the early stage, there is no much difference of $\gamma_{i,j,t-1}$ or $\mu_{i,t}$ among the experts. At this stage, the strategy is mainly dominated by $|f_i(\mathbf{x}_t) - f_j(\mathbf{x}_t)|$ or $|f_i(\mathbf{x}_t) - \bar{p}_t|$. Along with the learning process, loss $\hat{L}_{i,t-1}$ suffered by good experts would be much smaller than the one suffered by bad experts. So the values $\gamma_{i,j,t-1}$ or $\mu_{i,t}$ of good experts are much larger than those of bad experts. This characteristic makes the strategies pay more attention to the difference of advices obtained from the good experts, and thus makes the strategies robust in the setting where noisy experts exist.

Although these strategies can be robust in noise-added setting, $\gamma_{i,j,t-1}$ and $\mu_{i,t}$ depends on unknown $\hat{H}_{i,t-1}$. To solve this issue, it can be assumed that all the examples are independently and identically distributed from some unknown distribution and the performance of one expert will not change over time. Given these two assumptions, it can be verified that the cumulative loss of one expert on the unlabeled examples is propositional to that on the labeled examples. Formally, if there are m labeled examples and n unlabeled examples until time t , then $\hat{H}_{i,t-1} = \frac{n}{m} \hat{L}_{i,t-1}$ exists.

Based on the above analysis of the confidence conditions, the general framework of robust OAL forecasters for binary classification tasks is presented in Algorithm 3.

As shown in Algorithm 1, at each round, after receiving an input instance $\mathbf{x}_t$, the prediction made by each expert is obtained by the learner, i.e., $f_i(\mathbf{x}_t)$. Then, the equation (3.6) or (3.7) is measured. If one of the equations is satisfied, the true label of this input instance will not be queried; otherwise, the learner will request the true label for this instance from the environment, and make updates to the losses of the experts.

3.3 Experimental Results

In this section, the empirical performance of the proposed robust active forecasters RAEWAF and RAGF are evaluated for online binary classification tasks.

Algorithm 3 A Framework of Robust Active Forecaster

Input: a pool of experts f_i , $i = 1, \dots, N$.
Initialize tolerance threshold δ and $\hat{L}_{i,t} = 0$, $i \in [N]$.
for $t = 1, \dots, T$ **do**
 receive $\mathbf{x}_t$ and compute $f_i(\mathbf{x}_t)$, $i \in [N]$;
 compute $\bar{p}_t$ according to Equation (3.5) and set $\hat{p}_t = \pi_{[0,1]}(\bar{p}_t)$;
 if Equation (3.6) or (3.7) is satisfied **then**
 skip the label request for instance $\mathbf{x}_t$
 else
 request label y_t ;
 update $\hat{L}_{i,t} = \hat{L}_{i,t-1} + \ell(f_i(\mathbf{x}_t), y_t)$, $i \in [N]$;
 end if
end for

3.3.1 Experts and Compared Algorithms

To construct experts for an online sequential prediction task, the pool of experts is chosen by adopting five well-known online learning algorithms [DCP08, OC10], which include: (implemented as in [HWZ14])

- PERCEPTRON: the classical Perceptron algorithm [BLO62];
- ROMMA: the Relaxed Online Maximum Margin Algorithm [LL02];
- $\text{ALMA}_p(\alpha)$: the Approximate Maximal Margin Algorithm [Gen02];
- PA: the Passive-Aggressive online learning algorithm [CDKSs06];
- AROW: the Adaptive Regularization Of Weights algorithm [CKD13].

The proposed robust active learning algorithms (RAEWAF and RAGF) are compared with two state-of-the-art active learning algorithms (AEWAF and AGF), two regular forecasters (EWAF and GF) and their random variants as well. These algorithms are listed as follows:

- EWAF: the Exponentially Weighted Forecaster [CBL06], which queries all of the instances;
- GF: the Greedy Forecaster algorithm [CBL06], which also queries all of the instances;

- REWAF: the Random Exponentially Weighted Forecaster, a variant of EWAF, which will randomly select the instances to query according to a uniform distribution;
- RGF: the Random Greedy Forecaster algorithm, a variant of GF, which will randomly select the instances to query according to a uniform distribution;
- AEWAF: the Active Exponentially Weighted Forecaster algorithm [ZHZ13];
- AGF: the Active Greedy Forecaster algorithm shown in Theorem [ZHZ13];
- **RAEWAF**: the proposed Robust Active Exponentially Weighted Average Forecaster shown in Theorem 1;
- **RAGF**: the proposed Robust Active Greedy Forecaster algorithm shown in Theorem 2.

3.3.2 Experimental Testbed and Setup

To evaluate the performance, several experiments are conducted on a variety of benchmark datasets from web machine learning repositories. Table 3.1 shows the details of nine datasets used in our experiments. All of them can be downloaded from LIBSVM website ¹ and UCI machine learning repository ².

Table 3.1: Datasets used in the experiments

Dataset	# Class	# Instances	# Features	Sources
a8a	2	32561	123	LIBSVM
magic04	2	19020	10	UCI
mushrooms	2	8124	112	UCI
spambase	2	4601	57	UCI
svmguide1	2	7089	4	LIBSVM
w8a	2	64700	300	LIBSVM

All the expert algorithms learn a linear classifier for a binary classification task. The parameter p and α in $\text{ALMA}_p(\alpha)$ are set to be 2 and 0.9 respectively. The parameter C in PA is set to 5, and the parameter γ is set to 1 for AROW. To make fair comparisons, all

¹<http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/>

²<http://www.ics.uci.edu/~mllearn/MLRepository.html>

the compared forecasters adopt the same setup. The learning rate η is set to $\sqrt{8 \ln N/T}$, for all the datasets and forecasters. The sampling ratio for requesting labeled data by the two random algorithms (REWAF and RGF) are set in the range of $[0, 1]$.

Each dataset is randomly divided into two subsets: a training set consisting of 20% of the entire data for training the experts; and a test set consisting of the remaining data for learning the forecasters. The five experts algorithms are applied on the training set to simulate the five reliable expert functions $\mathbf{u}_i \in \mathbb{R}^d$, $i \in [5]$, where d is the dimension of the instance. To satisfy the assumptions, $f_i(\mathbf{x}) = \pi_{[0,1]}(\mathbf{u}_i^\top \mathbf{x} + 0.5)$ is adopted as the expert functions. Then the forecasters are evaluated on the test set. All the test experiments were conducted over 20 runs of different random permutations and reported by averaging these 20 runs. For performance metric, the regret rate and ratio of requested labeled data are adopted.

3.3.3 Evaluation on Normal Setting

In this section, the performance of the proposed algorithms is evaluated in the normal setting, where the experts pool are only consisted by the five reliable online learners. The algorithms are first evaluated in the setting of fixed query ratio, and then they are evaluated in the setting of varied query ratio.

3.3.3.1 Fixed Query Ratio

Table 3.2 summarizes the average performance of the algorithms in the normal setting on datasets *a8a* and *mushrooms*, where the query ratios of the algorithms whose baseline forecasters are same are kept similar. From the experimental results, several observations can be drawn as follows.

First, compared with the random algorithms REWAF and RGF, both the exiting active learning algorithms (AEWAF and AGF) and the proposed algorithms in this article (RAEWAF and RAGF) can achieve lower regret with similar query ratio. This proves that the proposed querying strategies in these algorithms (AEWAF, AGF, RAEWAF, RAGF) are capable to query informative instances by considering the difference of advices provided by reliable experts. More importantly, the proposed algorithms RAEWAF and

RAGF can achieve the lowest regret, this indicates the importance of considering the cumulative performance of each expert when considering their advices.

Second, compared to the fully supervised learning algorithm EWAF and GF, the proposed algorithms RAEWAF and RAGF can achieve similar regret value with less query ratio (less than 20%). This further indicates that the effectiveness of the proposed query strategies in this article in saving labeling cost.

Finally, our proposed algorithms cost more time when comparing the other baselines due to the computation of query strategies. Compared to AEWAF and AGF, the robust algorithms RAEWAF and RAGF cost more time due to the extra terms ($\gamma_{t-1}^{i,j}$ and μ_{t-1}^i) imported in Theorem 1 and 2. However, the extra time cost could be ignored considering the efficiency of online learning scheme.

Table 3.2: Evaluation of on the **Normal** setting, where all experts are comparably reliable.

Data	Algorithm	Query (%)	Regret (%)	Time (s)
a8a	EWAF	100.00 $\pm$ 0.001	28.350 $\pm$ 0.001	0.430
	REWAF	17.181 $\pm$ 0.002	135.210 $\pm$ 0.090	0.277
	AEWAF	16.999 $\pm$ 0.001	31.450 $\pm$ 0.016	0.501
	RAEWAF	16.772 $\pm$ 0.011	28.970 $\pm$ 0.001	1.323
	GF	100.00 $\pm$ 0.001	28.350 $\pm$ 0.001	0.749
	RGF	11.542 $\pm$ 0.002	161.590 $\pm$ 0.083	0.599
	AGF	11.305 $\pm$ 0.002	68.580 $\pm$ 0.052	0.823
	RAGF	11.729 $\pm$ 0.010	31.020 $\pm$ 0.001	0.924
mushrooms	EWAF	100.00 $\pm$ 0.001	39.120 $\pm$ 0.001	0.108
	REWAF	12.795 $\pm$ 0.004	119.030 $\pm$ 0.074	0.067
	AEWAF	12.861 $\pm$ 0.001	46.410 $\pm$ 0.007	0.122
	RAEWAF	12.679 $\pm$ 0.002	39.110 $\pm$ 0.001	0.327
	GF	100.00 $\pm$ 0.001	39.120 $\pm$ 0.001	0.189
	RGF	11.957 $\pm$ 0.004	120.560 $\pm$ 0.069	0.152
	AGF	12.512 $\pm$ 0.001	49.570 $\pm$ 0.012	0.207
	RAGF	12.057 $\pm$ 0.002	39.120 $\pm$ 0.001	0.231

3.3.3.2 Varied Query Ratio

In this section, the performance of proposed algorithms with varied query ratio in the normal setting is evaluated, where the trained experts generally share competing performance. Figure 3.1 and 3.2 present the per-round regret with respect to varied query

ratio on all of the datasets. For each figure, left figures show the results of algorithms which are based on EWAF forecaster, and right figures show the results of algorithms which are based on GF forecaster. Based on the results, several observations could be made.

First, for all of the active learning algorithms, they converge to the fully queried forecaster (EWAF and GF) faster than the randomly queried forecasters (REWAF and RGF), which validates that all the query strategies are generally effective to select informative instances in the normal setting.

Second, by comparing the proposed algorithms (RAEWAF and RAGF) with the existing algorithms (AEWAF and AGF) respectively, It is found that RAEWAF and RAGF converge to the minimal regret achieved by the fully queried forecasters (EWAF and GF) faster than the existing active learning algorithms (AEWAF and AGF), respectively. This indicates that the proposed active learning algorithms in this article can select much more informative instances to query by comparing with the exiting algorithms (AEWAF and AGF).

The above encouraging result validates the importance of considering the cumulative performance of each expert when designing query strategies, and the proposed active query strategies can outperform the existing algorithms even in the normal setting where there is no obviously noisy expert.

3.3.4 Evaluation on Noise-added Setting

In this subsection, the performance of the proposed algorithms in the noisy setting is examined, where **two** extra random noisy experts are added to the pool of experts.

3.3.4.1 Evaluation on Fixed Ratios of Queries

In this section, the proposed algorithms are evaluated by fixing the query ratio. As it is difficult to keep all the algorithms with same query ratio, the query ratio of all algorithms (whose baseline forecasters are same) is only kept roughly similar. The experimental result is shown in Table 3.3, based on table, several observations can be made.

Figure 3.1: Per-round regret V.S. Query Ratio in **Normal** setting, where all the experts are comparably reliable.

Figure 3.2: Per-round regret V.S. Query Ratio in the **normal** setting, where all the experts are reliable.

First, the existing algorithms AEWAF and AGF fail in this noisy scenario. Although AEWAF can outperform the random algorithm REWAF, but its performance decreases compared to the one in normal scenario shown in Figure 3.1 and 3.2. This observation would be clearer in the following subsections. The proposed algorithm AGF even performs worse than the random algorithm RGF. These observations confirm our analysis that the query strategies only considering experts advices would be easily misled by noisy experts.

Second, the proposed robust active learning algorithms RAEWAF and RAGF still can achieve the best performance. Both RAEWAF and RAGF could statistically achieve the smallest regrets with similar query ratio as AEWAF and AGF. What's more, with around 20% query ratio, RAEWAF and RAGF could achieve similar performance as the regular algorithms EWAF and GF, respectively. These observations first confirm the importance of considering the cumulative performance of each expert, and second validate the effectiveness of the proposed robust active learning algorithms in reducing label cost while achieving comparable performance as the ones which query all of the instances.

Table 3.3: Evaluation of on the **noisy** setting, where 2 noisy experts are added.

Data	Algorithm	Query (%)	Regret (%)	Time (s)
a8a	EWAF	100.000 $\pm$ 0.001	31.150 $\pm$ 0.001	0.45
	REWAF	22.239 $\pm$ 0.003	125.300 $\pm$ 0.065	0.308
	AEWAF	22.612 $\pm$ 0.001	99.330 $\pm$ 0.067	0.553
	RAEWAF	22.582 $\pm$ 0.015	31.340 $\pm$ 0.001	2.166
	GF	100.000 $\pm$ 0.001	31.150 $\pm$ 0.001	0.774
	RGF	12.159 $\pm$ 0.002	201.570 $\pm$ 0.085	0.625
	AGF	11.936 $\pm$ 0.014	580.610 $\pm$ 0.594	0.849
	RAGF	11.943 $\pm$ 0.011	35.170 $\pm$ 0.004	1.023
mushrooms	EWAF	100.000 $\pm$ 0.001	46.390 $\pm$ 0.001	0.113
	REWAF	8.017 $\pm$ 0.003	246.120 $\pm$ 0.185	0.070
	AEWAF	7.989 $\pm$ 0.003	176.510 $\pm$ 0.161	0.127
	RAEWAF	7.962 $\pm$ 0.002	48.370 $\pm$ 0.002	0.531
	GF	100.000 $\pm$ 0.001	46.390 $\pm$ 0.001	0.194
	RGF	13.861 $\pm$ 0.005	163.870 $\pm$ 0.087	0.158
	AGF	13.055 $\pm$ 0.054	840.100 $\pm$ 2.735	0.213
	RAGF	13.081 $\pm$ 0.004	46.410 $\pm$ 0.001	0.259

Figure 3.3: Per-round regret V.S. Query Ratio in **Noisy** setting, where **2** noisy experts are added.

Figure 3.4: Per-round regret V.S. Query Ratio in the **noisy** setting, where **2** noisy experts are added.

3.3.4.2 Evaluation on Varied Ratios of Queries

Figure 3.3 and 3.4 show the performance of different algorithms on all of the datasets in the noisy scenario, where two random noisy workers are added.

Figure 3.3 could illustrate several meaningful results. The first one is that both the existing algorithms AEWAF and AGF fail to outperform random variants in this scenario. What's worse, the random algorithms REWAF and RGF could outperform them on nearly all of the datasets respectively. Especially for the AGF, the aggressive characteristic makes it even perform worse than random algorithm RGF. These results are consistent with our analysis in Section 3 and confirm the importance of considering the experts performance when designing the query strategies.

The second one is that the proposed two robust algorithms RAEWAF and RAGF consistently outperform the existing algorithms (AEWAF and AGF) and the random algorithms (REWAF and RGF). This suggests that the proposed algorithms are more likely to query the labels of instances on which good experts disagree. These queried instances are informative to improve the forecasters. Thus, the proposed strategies could robustly use less number of queried instances to obtain the smaller regret.

The above observation shows that the proposed algorithms are more attractive and practical for many real-world applications, in which the aim is to find the best experts with minimal labeling cost while under challenging environment with noisy experts, e.g., annotation tasks in crowdsourcing platforms.

3.3.4.3 Evaluation on Varied Number of Noisy Experts

In this section, the algorithms are evaluated in different noisy scenarios. Figure 3.5 shows the results on *a8a* and *mushrooms* datasets, where the numbers (0, 2, 5, 7) at the end of each legend corresponds to the number of noisy experts added. As there are totally 5 reliable experts, these numbers of noisy experts could simulate all the noisy scenarios. Figure 3.6 shows the results on the other datasets.

First, as the number of noisy experts increases, it becomes challenging to learn the tasks. On all of the datasets, the regret all of the algorithms gets to the highest when 7 noisy experts are added. The performance of random algorithms (REWAF and RGF) and active learning algorithms (AEWAF and AGF) decreases fast as the number of noisy

Figure 3.5: Per-round regret V.S. Query Ratio in the noisy setting, where different number of noisy experts are added.

spambase

svmguide1

w8a

Figure 3.6: Per-round regret V.S. Query Ratio in the noisy setting, where a varied number of noisy experts are added.

experts increases. Even for the regular algorithms (EWAF and GF) which query 100% of all the instances, the performance also decreases. These observations indicate that the noisy scenarios usually make the learning tasks difficult than the normal scenarios.

Second, the proposed robust active learning algorithms RAEWAF and RAGF consistently achieve the best performance on all noisy scenarios. On all of the datasets, it can be observed that RAEWAF and RAGF could converge fast with similar query ratio no matter how many noisy experts are added. On the contrary, the performance of AEWAF and AGF decreases fast, and these two algorithms also require more queried instances to converge to the lowest regret.

3.4 Conclusion

This chapter addressed a new problem of active learning with expert advice for online sequential prediction tasks. A new framework of online active learning with expert advice is proposed. Specifically, two robust OAL algorithms by incorporating the cumulative performance of expert are presented. Two theorems are proposed which can theoretically guarantee the distance between predication of active learner and the prediction of passive learner is small. To evaluate the efficacy of the algorithms, an extensive set of experiments both in normal and noisy setting are conducted. Promising empirical results validate the effectiveness and robustness of the proposed active learning algorithms.

Despite the encouraging results, some limitations and open challenges of the current work remain. One issue is about the settings of the learning rate η and tolerance parameter δ , which were fixed manually in our experiments. It would be more attractive if one is able to design a self-tuned strategy for the active learning task. For the crowdsourcing application, it's interesting to investigate the impact of noisy workers on multi-class annotation tasks, and design new robust algorithms.

Chapter 4

Active Crowdsourcing for Annotation

Crowdsourcing has shown great potential in obtaining large-scale and cheap labels for different tasks. However, obtaining reliable labels is challenging due to several reasons, such as noisy annotators, limited budget and so on. The state-of-the-art approaches, either suffer in some noisy scenarios, or rely on unlimited resources to acquire reliable labels. In this chapter, a novel framework of active crowdsourcing for annotation is proposed based on the learning with expert (also known as worker in crowdsourcing) advice to robustly infer accurate labels by considering the reliability of each worker. However, in order to accurately predict the reliability of each worker, traditional learning with expert advice will consult with external oracles (AKA domain experts) on the true label of each instance. To reduce the cost of consultation, two active learning approaches are proposed, margin-based and weighted difference of advices based. Meanwhile, to address the problem of limited annotation budget, a reliability-based assigning approach is proposed to actively decide who to annotate the next instance based on each worker's cumulative performance. The experimental results both on real and simulated datasets show that our algorithms can achieve robust and promising performance both in the normal and noisy scenarios with limited budget.

In this chapter, Section 1 presents the proposed framework and algorithms. Section 2 discusses our empirical study and Section 3 summarize this chapter.

4.1 A Learning Framework for Active Crowdsourcing

4.1.1 Problem Setting

For most of the supervised machine learning algorithms, a large-scale of training data with golden labels are usually required. However, labeling the data is usually costly and time-consuming. To speed up the labeling process and obtaining reliable labels, crowdsourcing platforms, such as Amazon Mechanical Turk ¹, CrowdFlower ² and Baidu Test ³ and so on, are developed to easily obtain large-scale and reliable labels. However, obtaining reliable labels from crowdsourcing faces two main challenges. First, the annotations are usually noisy due to different knowledge bases of workers, also known as annotators, varied difficulty of each task and so on [WBBP10]. This gives rise to the question how to combine the noisy annotations to get finalized accurate labels. Second, the budget size is usually limited, and obtaining unlimited annotations for each task is costly and impractical.

To tackle the problem of noisy annotations, several approaches in literature are proposed. One promising approach is repeated labeling [SPI08]. The basic idea is to score the workers by their agreement with other workers. Most of the existing work followed this approach by treating the true label and workers' reliability as latent variables, and inferred these variables by building joint statistical models [WP10, WRW⁺09, YFRD11]. These algorithms are attractive because of the promising performances they have shown in real crowdsourcing datasets without referring to gold standard questions. However, these algorithms are heavily relying on the assumption that the majority of workers are correct, which may not be always true [LIS13]. These unsupervised algorithms would perform as badly as the majority voting (MV) algorithm when the assumption could not be met, such as in group-cheating scenarios, where a group of people cheat together by giving the same and noisy label for each instance. To tackle the budget issue, Welinder et al. [WP10] proposed to actively decide who to label by building a reliable worker list and

¹<https://www.mturk.com/mturk/>

²<http://www.crowdflower.com/>

³<http://test.baidu.com/crowdtest/>

a noisy worker list. However, this algorithm also suffers the same issue of requiring most of the workers to be reliable.

Considering the group-cheating and limited budget issues, it is necessary to build some gold standard questions with known answers, based on which workers' reliability can be accurately estimated and thus workers' annotations can be properly aggregated to get finalized accurate labels even in extreme noisy scenarios. Meanwhile, the usage of gold standard questions should be limited considering their high prices and the limited budget. Another argument made is that it is unsuitable to uniformly allocate tasks to workers considering workers' varied reliabilities.

In this chapter, the learning with expert (corresponding to worker in crowdsourcing) advice framework is adopted to tackle the issues of noisy workers and limited budget on crowdsourcing tasks. However, traditional learning with expert advice will consult with external oracles on the true labels of each instance. To reduce the cost of consultation, two active learning approaches are proposed, margin-based and weighted difference of advices based. To further address the limited budget issue, a reliability-based allocation approach is proposed to actively decide who to annotate the given instance based on each worker's cumulative performance.

4.1.2 Overview of Framework

Consider a real world online binary annotation task, such as image annotation (the workers are asked to label whether an image contains a bird or not), where images (AKA instances) arrive sequentially. Our goal is to construct a predictive model which can accurately infer the true label of the image given several noisy labels annotated by a pool of noisy workers. In general, this can be formulated as the framework of learning with expert advice for binary classification, where each expert corresponds to a noisy worker, and a piece of advice corresponds to a label given by a noisy worker. Based on the labels given by the workers, the goal of learning with expert advice is to train a forecaster which can correctly combine the noisy annotations to predict a correct label of an image. In particular, on each learning round, all the workers first receive a new coming image, and then give their annotations, based on which the forecaster predicts the label of the image. After that, the forecaster receives the ground-truth of the image from external

oracles (who are usually domain experts), and both the workers and forecaster suffer some positive loss based on the ground-truth and their predicted labels. At last, the forecaster updates the reliability of each worker in order to make better prediction in future. It

Figure 4.1: A Framework of Budget Active Learning with Expert Advice (BALEA)

is natural to apply the learning with expert advice forecasters, such as “Exponentially Weighted Average Forecaster” (EW) [CBL06] and “Greedy Forecaster” (GF) [CBL06], to the annotation tasks in the crowdsourcing platforms. However, it is impractical to directly apply the EW and GF forecasters to the problem due to several reasons. First, for the regular forecasters, each instance in the annotation task would need all workers to annotate. However, for a real annotation task, the budget is usually limited and noisy workers commonly exist, which make it costly and unnecessary to obtain all workers’ annotations for each instance. Second, for the traditional forecasters, they require that all instances at last receive the ground-truth from the external oracles, such as domain experts. However, obtaining the ground-truth from the domain experts is much more costly than obtaining the annotations from noisy workers, so it is impractical to obtain a large set of standard questions with ground-truth in order to accurately predict the

reliability of each worker. To address these issues, a Budget Active Learning with Expert Advice (BALEA) framework is proposed, as shown in Figure 4.1.

In general, the proposed BALEA framework attempts to conquer the following challenges with a systematic and synergistic way: (i) in the “Who to Label” module, the challenge is how to actively decide which workers to label the incoming instance in order to reduce cost and get reliable annotations; and (ii) in the “When to Query” module, the challenge is how to actively decide whether to query domain experts in order to update the workers’ reliabilities. To conquer the first challenge, our basic idea is considering the cumulative performance of each worker when allocating the incoming instances. For the second challenge, two active learning strategies are proposed to reduce the cost in query domain experts.

4.1.3 Problem Formulation

In this section, the problem of learning the reliabilities (AKA competences or weights) of workers is defined. For simplicity, the binary annotation task is considered in this chapter.

For a fixed pool of N annotators, the goal is to train an online learner (AKA forecaster, which describes the way of how to combine the annotations) from a sequence of instances $\{\mathbf{x}_1, \dots, \mathbf{x}_T\}$ where $\mathbf{x}_t$ is an instance in the annotation task. After receiving $\mathbf{x}_t$, N annotators give their annotations $\{f_i(\mathbf{x}_t) : \mathbb{R}^d \rightarrow \{0, 1\} | i = 1, \dots, N\}$, based on which the learner makes the final prediction $p_t \in [0, 1]$, p_t can be treated as the weighted majority voting or margin of forecaster on $\mathbf{x}_t$. After making the prediction, the learner is revealed with the ground-truth $y_t \in \{0, 1\}$ from an external oracle. At last, the performance of each annotator according to some non-negative loss functions between each worker’s label and the ground-truth is evaluated.

the problem above can be formulated as the problem of learning with expert advice. The label given by each worker corresponds to the advice of each expert. Here, the absolute loss function is adopted to score each worker’s performance. For example, the i -th worker’s loss on $\mathbf{x}_t$ is defined as $\ell(f_i(\mathbf{x}_t), y_t) = |f_i(\mathbf{x}_t) - y_t|$, and the forecaster’s loss is defined as $\ell(p_t, y_t) = |p_t - y_t|$. The cumulative losses suffered by the i -th expert and the forecaster are computed respectively as follows:

$$L_T^i = \sum_{t=1}^T \ell(f_i(\mathbf{x}_t), y_t), \quad L_T = \sum_{t=1}^T \ell(p_t, y_t).$$

The regret of i -th expert is defined as the loss difference between the i -th expert and the forecaster:

$$R_T^i = L_T^i - L_T, i = 1, \dots, N. \quad (4.1)$$

4.1.4 EW and GF Forecasters

To aggregate the advices of experts, a natural strategy of combining advices is weighted average prediction strategy. More specifically, the forecaster makes the prediction of $\mathbf{x}_t$ as follows:

$$p_t = \frac{\sum_{i=1}^N w_{i,t-1} f_i(\mathbf{x}_t)}{\sum_{i=1}^N w_{i,t-1}}, \quad (4.2)$$

where $w_{i,t-1}$ is the weight computed at time $t-1$ for the i -th expert. The intuitive idea of learning the weight is to assign large weights for those experts with low loss and small weights for those with high loss.

One special case of learning with expert advice is known as “Exponentially Weighted average forecaster” (EW) [CBL06]. In particular, by choosing

$$w_{i,t-1} = \frac{\exp(-\eta L_{i,t-1})}{\sum_{j=1}^N \exp(-\eta L_{j,t-1})},$$

where $\eta > 0$ is a parameter to control the learning rate, the EW forecaster makes the following prediction:

$$p_t = \sum_{i=1}^N w_{i,t-1} f_i(\mathbf{x}_t). \quad (4.3)$$

In addition to the weighted average forecaster, another kind of forecaster, known as the “Greedy Forecaster” (GF) [CBL06], is also considered, and GF makes the following prediction:

$$p_t = \pi_{[0,1]} \left(\frac{1}{2} + \frac{1}{2\eta} \ln \frac{E(i, 1)}{E(i, 0)} \right), \quad (4.4)$$

where $\pi_{[0,1]}(\cdot) = \max(0, \min(1, \cdot))$, $E(i, 1) = \sum_{i=1}^N \exp(-\eta L_{i,t-1} - \eta \ell(f_i(\mathbf{x}_t), 1))$ and $E(i, 0) = \sum_{i=1}^N \exp(-\eta L_{i,t-1} - \eta \ell(f_i(\mathbf{x}_t), 0))$.

4.1.5 Who to Label

For the “Who to Label” module in Figure 4.1, the basic idea is to give more chance to the reliable workers. Here, the metric to evaluate the reliability of each worker is the cumulative loss suffered. The higher the loss $L_{i,t}$ is, the less reliable of the i -th worker is.

Specifically, the **unreliability** of i -th worker on t -th round is defined as follows:

$$u_{i,t} = \frac{\exp(L_{i,t-1})}{\sum_{i=1}^N \exp(L_{i,t-1})}. \quad (4.5)$$

It’s clear that $u_{i,t} \in (0, 1)$ and $\sum_i u_{i,t} = 1$. The larger $u_{i,t}$ is, the more **unreliable** the i -th worker is.

For the incoming $\mathbf{x}_t$, a Bernoulli random variable $S_t^i \in \{0, 1\}$ is drawn for i -th worker with probability

$$a_{i,t} = \frac{\sigma}{\sigma + u_{i,t}}, \quad (4.6)$$

where $\sigma > 0$ is a parameter to trade off the budget size and the number of annotations the t -th instance could get. If $S_t^i = 1$, the i -th worker would get the chance to annotate $\mathbf{x}_t$. If $S_t^i = 0$, the algorithm will skip the i -th worker. From the equation, it can be observed that the more **unreliable** the i -th work is, the more chance the work would **not** get the t -th annotation task.

4.1.6 When to Query

For the traditional learning with expert advice, the ground-truth is always obtained from the domain experts for each instance. However, consulting with domain experts is highly costly, so the key challenge is how to design effective *Active learner* for the “When to Query” module in Figure 4.1. By observing the Equation (4.3) and (4.4), p_t can be treated as the confidence of forecaster on instance $\mathbf{x}_t$, based on which a simple yet effective active learning scheme is adopted to decide when $\mathbf{x}_t$ should be queried.

Specifically, for both the EW and GF forecasters, at the t -th round, the algorithm decides when $\mathbf{x}_t$ should be queried according to a Bernoulli random variable $Z_t \in \{0, 1\}$ with probability

$$q_t = \frac{\delta}{\delta + |p_t|}, \quad (4.7)$$

where p_t is computed based on Equation (4.3) or (4.4), and $\delta > 0$ is a sampling parameter to control the query ratio to trade off the budget size and the number of consultation with domain experts. This strategy is similar to the margin-based active learning [Set10b], and has been used in online classification problem [CBGZ06]. If $Z_t = 1$, the true label y_t provided by domain experts is revealed to the forecaster, and the algorithm will update workers' reliabilities. If $Z_t = 0$, the true label will not be queried and the reliabilities of workers will not be updated.

The active learning strategies proposed in Chapter 3 is also adopted in this chapter to decide when to query the true label. To keep consistence, the query strategies proposed in previous chapter are also presented here. In the related works [ZHZ13], the authors assumed the advice given by each expert is a float point in the range of $[0, 1]$. However, for the annotation tasks in the crowdsourcing platforms, the label given by each worker is in $\{0, 1\}$, which makes the proposed strategies infeasible. Besides, the proposed strategies [ZHZ13] are only based on the annotations $f_i(\mathbf{x}_t)$ on current instance $\mathbf{x}_t$, and the cumulative performance of each worker shown in Equation (4.5) is not considered, which makes the strategies suffer when noisy workers exist. To overcome this limitation, two robust active learning strategies are proposed in previous chapter by considering both workers' annotations and their cumulative performances.

For an instance $\mathbf{x}_t$, $\hat{p}_t$ denotes the prediction of forecaster which is trained based on queried instances until t -th round, and p_t as the prediction of forecaster which is trained based on all instance until t -th round. The basic idea is to design querying strategies which guarantee a small difference between $\hat{p}_t$ and p_t . Formally, for the EW forecaster, if the following theorem is satisfied, the ground-truth of $\mathbf{x}_t$ is **unnecessary** to be revealed ($Z_t = 0$), vice versa.

Theorem 3 *For a small constant $\beta > 0$, if the following condition is satisfied, then $|p_t - \hat{p}_t| \leq \beta$*

$$\left| \frac{\sum_{i=1}^N \sum_{j=1}^N \gamma_{i,j,t-1} (f_i(\mathbf{x}_t) - f_j(\mathbf{x}_t))}{\sum_{i=1}^N \sum_{j=1}^N \gamma_{i,j,t-1}^{i,j}} \right| \leq \beta, \quad (4.8)$$

where $\gamma_{i,j,t-1} = \exp\left(-\eta \left[\hat{L}_{i,t-1} + \hat{H}_{i,t-1} + \hat{L}_{j,t-1}\right]\right)$, $\hat{L}_{i,t-1}$ and $\hat{H}_{i,t-1}$ are the losses suffered of i -th worker until t -th round on queried and un-queried instances, respectively. $\beta > 0$ is a small constant to control the query ratio.

Similarly, for the GF forecaster, if the following theorem is satisfied, the ground-truth of $\mathbf{x}_t$ is **unnecessary** to be revealed ($Z_t = 0$), vice versa.

Theorem 4 *For a small constant $\beta > 0$, if the following condition is satisfied, then $|p_t - \hat{p}_t| \leq \beta$.*

$$\frac{1}{2\eta} \ln \frac{\sum_{i=1}^N \mu_{i,t} \exp(2\eta(f_i(\mathbf{x}_t) - p_t))}{\sum_{i=1}^N \mu_{i,t}} \leq \beta, \quad (4.9)$$

where $\mu_{i,t} = \exp\left(-\eta[\widehat{L}_{i,t-1} + \ell(f_i(\mathbf{x}_t), 0) + \widehat{H}_{i,t-1}]\right)$, $i \in [N]$.

Remark. Detailed proof of these two theorems is in Chapter 3. $\widehat{H}_{i,t-1}$ is an unknown loss suffered on the un-queried instances. If all the instances are independently and identically distributed from some unknown distribution and the performance of one worker also does not change over time. It is easy to verify that the cumulative loss of one worker on the unlabeled examples is propositional to that on the labeled examples. Formally, if there are m labeled examples and n unlabeled examples until time t , then $\mathbb{E}[\widehat{H}_{i,t-1}] = \frac{m}{n} \widehat{L}_{i,t-1}$ exists.

Algorithm 4 summarizes the procedure of the proposed framework.

Algorithm 4 Budget Active Learning with Expert Advice

Input: a pool of experts f^i , $i = 1, \dots, N$.
Output: cumulative loss $\widehat{L}^i$, $i = 1, \dots, N$.
Initialize tolerance threshold $\delta > 0, \beta > 0$ and $\widehat{L}_t^i = 0$, $i \in [N]$.
for $t = 1, \dots, T_1$ **do**
 receive $\mathbf{x}_t$;
 decide **who** to label according to Equation (4.6);
 get annotations $f_i(\mathbf{x}_t)$, $i \in [N]$;
 decide **when** to query $\mathbf{x}_t$ based on Equation (4.7) or Theorem 3 and 4;
 if $Z_t = 1$ **then**
 request label y_t ;
 update $\widehat{L}_t^i = \widehat{L}_{i,t-1} + \ell(f_i(\mathbf{x}_t), y_t)$, $i \in [N]$;
 else
 skip the label request for instance $\mathbf{x}_t$
 end if
end for

4.2 Experimental Results

4.2.1 Baseline Algorithms

All the algorithms used in the following experiments are listed as follows:

- MV: the Majority Voting algorithm which takes the majority label as the predicted label;
- SIGNAL: the classical Maximum Likelihood Estimation algorithm [DS79];
- BIAS: the state-of-the-art Multidimensional Wisdom of Crowds algorithm [WP10];
- EW: the regular Exponentially Weighted average forecaster algorithm [CBL06];
- MAEW/WAEW: the Margin-based/Weighted difference of advices based Active EW algorithm based on Equation (4.7) or Theorem 3, and REW is the random version;
- EW-Who: the EW algorithm with strategy of Who to label based on Equation (4.6), and REW-Who is the random version;
- EW-All: the EW algorithm combining MAEW and EW-Who, and REW-All is the random version;
- GF: the regular Greedy Forecaster algorithm [CBL06];
- MAGF/WAGF: the Margin-based/Weighted difference of annotations Active GF algorithm based on Equation (4.7) or Theorem 4, and RGF is the random version;
- GF-Who: the GF algorithm with strategy of Who to label based on Equation (4.6), and RGF-Who is the random version;
- GF-All: the GF algorithm combining MAGF and GF-Who, and RGF-All is the random version;

4.2.2 Experimental Testbed and Setup

Table 4.1 shows the datasets used in our experiments. *fashion*, *fashion-cloth* are two real crowdsourcing datasets downloaded from UMassTrace ⁴ repository, and the other datasets

⁴<http://traces.cs.umass.edu/index.php/Mmsys/Mmsys>

Table 4.1: Datasets used in the following experiments.

Dataset	# Instances	# Train	# Test	# Workers	Type
fashion	4005	400	3605	6	real
fashion-cloth	4005	400	3605	6	real
a8a	32561	2604	23444	5	simulated
spambase	4601	368	3312	5	simulated

are simulated datasets downloaded from LIBSVM ⁵ repository. To simulate the reliable workers in the simulated datasets, the following online learning algorithms are adopted.

- PA: the Passive-Aggressive algorithm [CDKSs06];
- AROW: the Adaptive Regularization Of Weights algorithm [CKD09];
- $\text{ALMA}_p(\alpha)$: the Approximate Maximal Margin Algorithm [Gen02];
- ROMMA: the Relaxed Online Maximum Margin Algorithm [LL02];
- PERCEPTRON: the classical Perceptron algorithm [BLO62].

The parameter C of the PA algorithm is set to 5, and the parameter γ of the AROW algorithm is set to 1. For the $\text{ALMA}_p(\alpha)$ algorithm, p and α are set to 2 and 0.9, respectively. The learning rate η for Equation (4.3) and (4.4) is set to $\sqrt{8 \ln N/T}$ on all datasets, where N denotes the number of workers and $T = \# \text{ Train}$ in Table 4.1.

To investigate the performance of the algorithms in the scenarios where noisy workers exist, $\{0, 3, 5, 10\}$ number of noisy workers are manually added to each dataset. These four different number of noisy workers represent the scenarios where the number of noisy workers is much less than, less than, the same as and larger than the number of reliable workers, respectively. These four scenarios could almost simulate the situations happening in the real crowdsourcing platforms. Here the noise considered is the group-cheating noise, where all the noisy workers added would give the same and random label for all the instances.

For each simulated dataset, 20% of the instances are used to train the five reliable workers. The rest of instances are divided into 10 parts. Each part of these 10 parts is

⁵<http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/>

used as standard questions to train the proposed algorithms, and the other 9 parts are used as test sets. The accuracy (Y-axis) showed in the following figures is the average accuracy on the test sets among the 10 folds. Please notice, on the 9 test parts, true labels are not queried from domain experts and only their annotations and workers' reliabilities learned on the 1 part are used.

4.2.3 Experiments on Different Noisy Scenarios

Figure 4.2: Accuracy V.S. Number of Group-cheating Workers.

In this section, influence of noisy workers on baseline algorithms is investigated. Figure 4.2 shows the accuracy of different algorithms with different number of noisy

workers. The performance on the other datasets is shown in supplementary material.

As expected, both the EW and GF algorithms are stable and outperform the state-of-the-art algorithms in all noisy levels. All of the other algorithms could perform as well as EW and GF in the noise-free scenario (0 noisy workers added) without any ground-truth labels requirement. This makes SIGNAL and BIAS algorithms attractive in large-scale annotation tasks, however, as the group-cheating noisy workers are added, all of the baseline algorithms greatly suffered as expected. Especially for the MV algorithm, as the number of group-cheating workers increases, the annotations from noisy workers rapidly dominant the vote. This makes the MV algorithm infeasible. BIAS algorithm considers each workers bias based on the SIGNAL algorithm, this feature makes it outperform the other two baseline algorithms in the noise-free or random noisy setting [WP10], however, its performance also suffers most as group-cheating noisy workers are added. These observations motivate to consult domain experts to robustly learn workers' reliabilities in the group-cheating scenarios.

Although EW and GF algorithms could robustly achieve better performance, task allocation doesn't consider workers' reliabilities which could waste money on allocating task to noisy workers. What's worse, both EW and GF require all the golden labels on the training set, which is costly considering the high price to querying the domain experts. In the following subsections, the proposed algorithms are evaluated on tackling these two limitations, respectively.

4.2.4 Experiments on Who to Label

In this section, the performance of the proposed active allocation strategy on "Who to Label" module in Figure 4.1 is evaluated. To simplify the evaluation, it is assumed that each worker will spend 1 resource to label one instance, so the budget size equals the multiplication of # Train and # Workers shown in Table 4.1. Figure 4.3 shows the performance with varied budget ratio under different noisy scenarios. Based on the results, several observations could be made as following.

First, on all noisy settings, the proposed active allocation algorithms, EW-Who and GF-Who, could consistently infer much more accurate labels than the random versions REW-Who and GF-Who, respectively. Specifically, In Figure 4.3 (a), where there is no

extra noisy workers added, random algorithms REW-Who and RGF-Who could achieve similar performance as the active allocation algorithms EW-Who and GF-Who since the reliabilities of all workers are comparable, and there is not much difference of deciding who to label the given instance. However, as the number of noisy workers increases, shown in Figure 4.3 (b), (c), (d), the proposed reliability-based allocation algorithms based on Equation (4.6) greatly outperform their random versions. This confirms the effectiveness of considering the cumulative performance of each worker when assigning the annotation tasks.

Figure 4.3: Accuracy V.S. Budget Ratio in Different Group-cheating Scenarios on *fashion* Dataset.

Second, the proposed strategies could reduce the annotation budget to get comparable performance with baseline algorithms when noisy workers exist. For example, in Figure 4.3 (d), with less than 20% budget, GF-who could outperform all the other algorithms in terms of accuracy. This feature makes it much more attractive as the number of noisy workers increases. For example, in Figure 4.3 (c), the GF-Who algorithm could outperform the other algorithms with %20 of the budget size, the EW-Who also could achieve similar performance with %50 of the budget size.

Figure 4.4: Accuracy V.S. Budget Ratio with **10** Extra Noisy Workers Added.

Figure 4.4 shows the performance of proposed algorithms on the other datasets, where 10 extra noisy workers are added. Similar observations could be made. It should be noted that *BIAS* algorithm performs as bad as *MV* algorithm. Although the proposed

active allocation algorithms could consistently achieve promising results, they are built on the workers' reliabilities which are learned by querying domain experts, who usually charge much higher than the noisy workers in the crowdsourcing platforms. The following experiments investigate the performance of the proposed algorithms in reducing the number of queries of domain experts.

4.2.5 Experiments on When to Query

Figure 4.5: Accuracy V.S. Budget Ratio with 0 Extra Noisy Workers Added Based on EW Forecaster.

In this experiment, it is assumed that the budgets equal “# Train” shown in Table 4.1,

and querying one instance for true label would cost one budget. The accuracy on test set with respect to the querying ratio on training set is investigated, as shown in Figure 4.5 and 4.6. The performance on the other datasets is shown in supplementary material.

Figure 4.6: Accuracy V.S. Budget Ratio with 0 Extra Noisy Workers Added Based on **GF** Forecaster.

First, it can be observed that the proposed algorithms (MAEW, MAGF, WAEW and WAGF) greatly outperform their random versions REW and RGF. Specially, the proposed WAEW and WAGF outperform the margin-based algorithms MAEW and MAGF once querying ratio is larger than a certain point. Besides, the MAEW and MAGF algorithms could achieve similar performance as EW and GF with limited querying budget, such as 40% of querying budget in Figure 4.5 (a) (b). More importantly, with little extra cost in

querying domain experts, the proposed algorithms are robust in accurately aggregating workers’ annotations on different noisy scenarios.

Second, all active learning algorithms and their random versions could outperform the MV algorithm with less than 10% of querying budget (when the ratio of querying budget equals 0, the active learning algorithms are down to MV algorithm, where all workers are assigned with the same weight). These findings verify the effectiveness of the proposed active learning algorithms in reducing the cost of querying domain experts and achieving promising performance meanwhile.

4.2.6 Experiments on the Framework

In this section, the overall performance of the proposed Framework 4.1 is investigated by combining the strategies of who to label (EW-Who and GF-Who) and strategies of when to query (MAEW and MAGF) together. To make a fair comparison with baseline algorithms, it is assumed the *total budget* as the number of annotations collected on “# Train” and “# Test” in Table 4.1, and querying one instance with domain experts will cost m resources, where m equals the average number of annotations received from noisy workers for each instance. Ratio of budget equals the resources used on consulting domain experts with MAEW or MAGF algorithms and annotating with EW-Who or GF-Who algorithms over *total budget* defined.

Figure 4.7 shows the performance of the proposed framework with varied budget ratio on four datasets when 10 noisy workers are added. The performance on the other datasets is shown in supplementary material. Firstly, similar observations as in Figure 4.3 could be made. EW based algorithms achieve comparable performance as the one based on GF.

Secondly, by combining the “Who to Label” and “When to Query” strategies, the proposed framework could robustly infer much more accurate labels with limited budget. Although it needs domain experts to query the true labels, only 1% of the instances are queried. Besides, the learned reliabilities of workers could be directly and smoothly adopted in the allocation phrase. Considering all these advantages, the framework shows great potential to be applied into real annotation tasks, especially for the group-cheating scenarios.

Figure 4.7: Accuracy of Framework 4.1 V.S. Budget Ratio with 10 Extra Noisy Workers Added.

4.3 Summary

In this chapter, a framework, budget active learning with expert advice, is proposed to infer accurate labels from noisy annotations. Specifically, two active learning strategies are proposed to decide when to query the ground-truth (which is usually given by the domain experts) of an instance. Relying on the reliability of each worker learned, an active allocation strategy is presented to decide who to label for an instance. To evaluate the algorithms, several experiments are conducted both on real and simulated datasets in different scenarios. The empirical studies show the proposed active learning strategies could greatly alleviate intervention of domain experts, and the proposed active allocation strategy could effectively choose reliable workers to label. In summary, the proposed framework could achieve comparable results with the baseline algorithms in the normal setting, and robustly outperform the baselines in the group-cheating scenarios.

Chapter 5

Second-order Online Active Learning

Online learning represents a family of efficient and scalable machine learning algorithms which are promising for large-scale learning tasks from big data streams [HWZ14, KZB⁺14]. Online learning typically works in a sequential manner. Consider online binary classification as an example, where the learner aims to train a classifier $f(x_t)$ via online learning. At time t , the learner receives an instance $\mathbf{x}_t$ from the environment, and then makes a prediction of its class label $\hat{y}_t = \text{sign}(f(x_t))$. After making the prediction, it often assumes the learner will be revealed with the true label $y_t \in \{+1, -1\}$ from the environment, and then make an update of the classifier whenever necessary (e.g., wrongly classified $\hat{y}_t \neq y_t$ or other criteria). In contrast to traditional batch learning that often suffers from expensive re-training cost when new training data comes, online learning avoids re-training and learns incrementally from data streams in an efficient and scalable way.

Although various online learning algorithms have been proposed over the past decades [HWZ14], conventional supervised online learning methods often assume the feedback (e.g., the class label in a classification task) is always revealed to the learner at the end of each iteration. However, this is not always true for many real applications where data streams could be unlabeled and human manual labeling cost could be quite expensive in many scenarios. This has raised an open challenge, that is, how to learn iteratively from unlabeled data streams by maximizing the learning efficacy while minimizing human labeling cost, a challenging problem known as “online active learning”.

Some existing studies have attempted to attack this challenge. A pioneering study is the “Perceptron-based active learning” [CBGZ06], where the learner decides when to

query by drawing a Bernoulli sampling of random variable $Z_t \in \{0, 1\}$ with parameter $\delta/(\delta + |p_t|)$, where $|p_t|$ is a form of prediction margin and δ is a sampling parameter. If $Z_t = 1$, the learner will then place a query. The similar approach has also been used by the online Passive Aggressive (PA) active learning in another recent study [LZH16]. Despite their simplicity, these algorithms often suffer some critical limitations. First, they often adopt first-order online learning algorithms for training. Second, as the margin $|p_t|$ only depends on the weight vector $\mathbf{w}_t$, making the query strategy sub-optimal when the weight vector $\mathbf{w}_t$ is not precise, particularly in the early rounds of online learning.

To overcome these limitations, a new online active learning method is proposed, named as “Second-order Online Active Learning” (SOAL), which explores second-order online learning techniques for both training the classifiers and forming the online query strategy. the proposed SOAL algorithm is theoretically analyzed, and further empirically validated via extensive studies.

In this chapter, Section 1 presents the SOAL algorithm. Section 2 analyzes the mistake bound of the proposed algorithm. Section 3 discusses the results of our empirical studies, and finally Section 4 concludes this work.

5.1 Second-order Online Active Learning

In general, there are two open challenges with an online active learning task: (i) how to devise an effective query strategy that queries the most informative unlabeled example for training; and (ii) how to update the classifier more effectively once a query has been placed and the feedback is revealed to the learner. In the following, a new framework of Second-order Online Active Learning is presented.

5.1.1 Problem Formulation

Without loss of generality, online active learning for binary classification tasks is considered. Consider a typical online binary classification task, at time t , a learner iteratively learns from a sequence of training instances $\{(\mathbf{x}_t, y_t) \mid t = 1, \dots, T\}$, where each instance $\mathbf{x}_t \in \mathbb{R}^d$ has $y_t \in \{-1, +1\}$ as its true class label. The goal of online binary classification is to learn a linear classifier sequentially from arriving data $f(\mathbf{x}_t; \mathbf{w}_t) = \text{sign}(\mathbf{w}_t \cdot \mathbf{x}_t)$, where $\mathbf{w}_t \in \mathbb{R}^d$ is the weight vector at the t -th round.

Unlike regular supervised online learning, when receiving an unlabeled instance $\mathbf{x}_t$, an online active learner needs to first decide whether to query its class label y_t or not. If the true label class is queried, the learner can then adopt some regular online learning algorithm to update the model $\mathbf{w}_t$; otherwise, the learner will simply skip the instance and process the next instance. In this way, online active learning is allowed to query the class labels for a *small* fraction of informative unlabeled instances, but aims to achieve a competitive accuracy with a regular online learning algorithm that simply queries the label of *every* incoming instance.

Following the similar idea of AROW [CKD09], it's assumed the model follows a Gaussian distribution, i.e., $\mathbf{w} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and predicts the label $\hat{y}_t = \text{sgn}(p_t)$, where $p_t = \mathbf{w}^\top \mathbf{x}$. In practice, however, it is often easier to simply use the average weight vector $\boldsymbol{\mu} = \mathbb{E}[\mathbf{w}]$ to make predictions. The values $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_{i,i}$ encode the model's knowledge and confidence in the weight of i -th dimensional variable w_i : the smaller the value of $\boldsymbol{\Sigma}_{i,i}$ is, the more confident the learner has in the mean weight value $\boldsymbol{\mu}_i$. The covariance term $\boldsymbol{\Sigma}_{i,j}$ captures interactions between w_i and w_j .

5.1.2 SOAL Algorithm

The proposed algorithm mainly consists of two parts: (i) how to online update the model when the true label is obtained, and (ii) when is the right time to query an unlabeled instance. I will discuss each of these issues in detail below.

5.1.2.1 How to Update.

At the t -th round, if the label y_t of $\mathbf{x}_t$ is queried, the distribution can be updated by minimizing the objective function:

$$f_t(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = D_{KL}(\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \| \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)) + \eta \mathbf{g}_t^\top \boldsymbol{\mu} + \frac{1}{2\gamma} \mathbf{x}_t^\top \boldsymbol{\Sigma} \mathbf{x}_t,$$

where

$$D_{KL}(\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \| \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)) = \frac{1}{2} \log \left(\frac{\det \boldsymbol{\Sigma}_t}{\det \boldsymbol{\Sigma}} \right) + \frac{1}{2} \text{Tr}(\boldsymbol{\Sigma}_t^{-1} \boldsymbol{\Sigma}) + \frac{1}{2} \|\boldsymbol{\mu}_t - \boldsymbol{\mu}\|_{\boldsymbol{\Sigma}_t^{-1}}^2 - \frac{d}{2},$$

$\mathbf{g}_t = \partial \ell_t(\boldsymbol{\mu}_t) = -y_t \mathbf{x}_t$, and $\ell_t(\boldsymbol{\mu}_t) = \max(0, 1 - y_t \boldsymbol{\mu}_t^\top \mathbf{x})$ is the hinge loss function adopted, and η, γ are positive parameters. This objective function consists of three terms: the first

term is to avoid the new model deviated too much from the previous model, the second term is to minimize the (linearized) loss of the new model on the current example, and the final term is to optimize the confidence of the model.

Following AROW [CKD09], the above minimization can be solved by the following two steps:

- Update the mean parameters:

$$\boldsymbol{\mu}_{t+1} = \arg \min_{\boldsymbol{\mu}} f_t(\boldsymbol{\mu}, \boldsymbol{\Sigma});$$

- If $\ell_t(\boldsymbol{\mu}_t) \neq 0$, update the confidence matrix parameters:

$$\boldsymbol{\Sigma}_{t+1} = \arg \min_{\boldsymbol{\Sigma}} f_t(\boldsymbol{\mu}, \boldsymbol{\Sigma});$$

For the first step, by setting the derivative to zero, i.e., $\partial_{\boldsymbol{\mu}} f_t(\boldsymbol{\mu}_{t+1}, \boldsymbol{\Sigma}) = 0$, a closed-form update can be obtained:

$$\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t - \eta \boldsymbol{\Sigma}_t \mathbf{g}_t, \quad (5.1)$$

and for the second step, by setting the derivative to zero, i.e., $\partial_{\boldsymbol{\Sigma}} f_t(\boldsymbol{\mu}, \boldsymbol{\Sigma}_{t+1}) = 0$, a closed-form update can also be computed:

$$\boldsymbol{\Sigma}_{t+1} = \boldsymbol{\Sigma}_t - \frac{\boldsymbol{\Sigma}_t \mathbf{x}_t \mathbf{x}_t^\top \boldsymbol{\Sigma}_t}{\gamma + \mathbf{x}_t^\top \boldsymbol{\Sigma}_t \mathbf{x}_t}, \quad (5.2)$$

where the Woodbury identity is used. Furthermore, since the update of the mean relies on the confidence parameter, it would be better to update the mean based on the updated covariance matrix $\boldsymbol{\Sigma}_{t+1}$, i.e.,

$$\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t - \eta \boldsymbol{\Sigma}_{t+1} \mathbf{g}_t, \quad (5.3)$$

which should be more accurate than the update in Equation (5.1). To intuitively explain the above updating Equation (5.3), let us assume $\boldsymbol{\Sigma}_{t+1}$ is a diagonal matrix. Then, this update rule actually assigns different feature dimension with different learning rate, so that the less confident weights will be updated more aggressively. In order to handle high-dimensional data, the full matrix update can be reduce to the diagonal version, and the updating rules in Equation (5.2) and (5.3) becomes

$$\boldsymbol{\Sigma}_{t+1} = \boldsymbol{\Sigma}_t - \frac{\boldsymbol{\Sigma}_t \odot \mathbf{x}_t \odot \mathbf{x}_t \odot \boldsymbol{\Sigma}_t}{\gamma + \sum_{i,j} (\mathbf{x}_t \odot \boldsymbol{\Sigma}_t \odot \mathbf{x}_t)_{ij}}, \quad (5.4)$$

$$\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t - \eta \boldsymbol{\Sigma}_{t+1} \odot \mathbf{g}_t, \quad (5.5)$$

where $\odot$ denotes the element-wise matrix multiplication.

5.1.2.2 When to Query.

For each incoming unlabeled instance $\mathbf{x}_t$, two kinds of information are exploited when designing a query strategy for active learning. The first factor is the margin value $|p_t|$ of predicting the instance $\mathbf{x}_t$, which represents how far the instance is away from the decision boundary of the current classifier. The larger the value $|p_t|$ is, the more confident the classifier is about the prediction on the current instance. The second factor is the confidence of the learner on the classifier (i.e., the trained weight vector). Intuitively, in the early training phase, the weight vector is less precise than the ones in the later phases. Thus, it is necessary to place more queries in the early phases. Formally, the proposed query strategy is shown as follows:

$$q(\mathbf{x}_t) = \frac{\delta}{\delta + \rho_t}, \text{ where } \rho_t = \max\left(|p_t| - \frac{\eta\gamma v_t}{2(\gamma + v_t)}, 0\right), \quad (5.6)$$

where $|p_t| = |\boldsymbol{\mu}_t^\top \mathbf{x}_t|$ denotes the absolute margin value of $\mathbf{x}_t$, $v_t = \mathbf{x}_t^\top \boldsymbol{\Sigma}_t \mathbf{x}_t$ denotes how well $\boldsymbol{\mu}_t$ is trained, and $\delta > 0$ is a smoothing constant to balance the query budget amount. The larger v_t is, the less well the weight vector is trained, and vice versa. Formally, the query strategy runs as follows:

- Draw a Bernoulli random variable $Z_t \in \{0, 1\}$ with parameter $q(\mathbf{x}_t) = \frac{\delta}{\delta + \rho_t}$;
- If $Z_t = 1$, query y_t ; Otherwise, skips the query on y_t .

To better understand the proposed query strategy, ρ_t is elaborated with the following two cases.

If $\rho_t = 0$, which means either $|p_t| = 0$ or $|p_t| \leq \frac{\eta\gamma v_t}{2(\gamma + v_t)}$ (this implies that margin value p_t is too small), the algorithm will **always** place query on the unlabeled instance even if the value $|p_t|$ is large. Our strategy is more reasonable in contrast to traditional query strategies [CBGZ06, LZH16], where a large value $|p_t|$ always results in a **very small** probability by which the instance would be *seldom* queried no matter how unreliable the classifier is.

If $\rho_t > 0$, it means $|p_t|$ is large and $\frac{\eta\gamma v_t}{2(\gamma + v_t)}$ is small. This implies that the classifier is well trained and can be trusted, thus the margin value $|p_t|$ based on the current classifier is informative enough for shaping the query strategy. In this situation, a Bernoulli random

variable $Z_t \in \{0, 1\}$ is drawn with parameter $q(\mathbf{x}_t) = \frac{\delta}{\delta + \rho_t}$. Here, ρ_t considers both the first-order p_t and second-order information v_t , which is thus more reliable than the margin value p_t alone.

In summary, our query strategy balances the trade-off between the uncertainty of instance prediction and the confidence of the classification model with the weight vector $\boldsymbol{\mu}$. Finally, Algorithm 5 summarizes the proposed algorithm.

Algorithm 5 SOAL: Second-order Online Active Learning.

Input: learning rate η ; regularization parameter γ ; budget parameter δ
Initialize: $\boldsymbol{\mu}_1 = 0$, $\boldsymbol{\Sigma}_1 = I$.
for $t = 1, \dots, T$ **do**
 Receive $\mathbf{x}_t \in \mathbb{R}^d$;
 Compute $p_t = \boldsymbol{\mu}_t^\top \mathbf{x}_t$, $v_t = \mathbf{x}_t^\top \boldsymbol{\Sigma}_t \mathbf{x}_t$;
 Compute $\rho_t = \max(|p_t| - \eta\gamma v_t / (2\gamma + 2v_t), 0)$;
 Draw Bernoulli random variable $Z_t \in \{0, 1\}$ with parameter $\delta / (\delta + \rho_t)$
 if $Z_t = 1$ **then**
 Query $y_t \in \{-1, +1\}$;
 Compute $\ell_t(\boldsymbol{\mu}_t) = [1 - y_t \mathbf{x}_t^\top \boldsymbol{\mu}_t]_+$;
 if $\ell_t > 0$ **then**
 Update model based on Equation (5.2) and (5.3) for full $\boldsymbol{\Sigma}$
 Or update model based on Equation (5.4) and (5.5) for diagonal $\boldsymbol{\Sigma}$
 end if
 end if
end for

5.1.3 Theoretical Analysis

To be concise, two notations are introduced:

$$M_t = \mathbb{I}(\hat{y}_t \neq y_t), L_t = \mathbb{I}(\ell_t(\boldsymbol{\mu}_t) > 0, \hat{y}_t = y_t).$$

Next the performance of the proposed algorithm are analyzed in terms of expected mistake bound $\mathbb{E}[\sum_{t=1}^T M_t]$. The detailed proof is given in the supplementary file.

Theorem 5 *Let $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_T, y_T)$ be a sequence of input examples, where $\mathbf{x}_t \in \mathbb{R}^d$ and $y_t \in \{-1, +1\}$ for all t . If the SOAL algorithm is run on this sequence of examples, then*

the expected number of prediction mistakes made is bounded from above by the following inequality, for any vector $\boldsymbol{\mu} \in \mathbb{R}^d$,

$$\begin{aligned} \mathbb{E} \left[\sum_{t=1}^T M_t \right] &\leq \mathbb{E} \sum_{t=1}^T Z_t \ell_t(\boldsymbol{\mu}) + \frac{D_{\boldsymbol{\mu}} + (1-\delta)^2 \|\boldsymbol{\mu}\|^2}{\eta \delta} \text{Tr}(\boldsymbol{\Sigma}_{T+1}^{-1}) \\ &\quad + \frac{1}{\delta} \mathbb{E} \left[\sum_{\rho_t \leq 0} \frac{\eta \gamma v_t}{(\gamma + v_t)} \right] + \frac{2}{\delta} \mathbb{E} \left[\sum_{\rho_t > 0} L_t \right] - \mathbb{E} \left[\sum_{t=1}^T L_t \right] \end{aligned} \quad (5.7)$$

where $\delta > 0$, $D_{\boldsymbol{\mu}} = \max_{t \leq T} \|\boldsymbol{\mu}_t - \boldsymbol{\mu}\|^2$.

Proof: When $Z_t = 0$, it is easy to verify the inequality in the theorem, using the fact $\ell_t \geq 0$.

When $Z_t = 1$, it is easy to observe that

$$\boldsymbol{\mu}_{t+1} = \arg \min_{\boldsymbol{\mu}} h_t(\boldsymbol{\mu}), \quad h_t(\boldsymbol{\mu}) = \frac{1}{2} \|\boldsymbol{\mu}_t - \boldsymbol{\mu}\|_{\boldsymbol{\Sigma}_{t+1}^{-1}}^2 + \eta \mathbf{g}_t^\top \boldsymbol{\mu}.$$

Because h_t is convex, the following inequality $\forall \boldsymbol{\mu}$ exists,

$$0 \leq \partial h_t(\boldsymbol{\mu}_{t+1})^\top (\boldsymbol{\mu} - \boldsymbol{\mu}_{t+1}) = [\boldsymbol{\Sigma}_{t+1}^{-1}(\boldsymbol{\mu}_{t+1} - \boldsymbol{\mu}_t) + \eta \mathbf{g}_t]^\top (\boldsymbol{\mu} - \boldsymbol{\mu}_{t+1}).$$

Re-arranging the above inequality will result in

$$\begin{aligned} &\eta \mathbf{g}_t^\top (\boldsymbol{\mu}_{t+1} - \boldsymbol{\mu}) \\ &\leq (\boldsymbol{\mu}_{t+1} - \boldsymbol{\mu}_t)^\top \boldsymbol{\Sigma}_{t+1}^{-1} (\boldsymbol{\mu} - \boldsymbol{\mu}_{t+1}) \\ &= \frac{1}{2} \left[\|\boldsymbol{\mu}_t - \boldsymbol{\mu}\|_{\boldsymbol{\Sigma}_{t+1}^{-1}}^2 - \|\boldsymbol{\mu}_{t+1} - \boldsymbol{\mu}\|_{\boldsymbol{\Sigma}_{t+1}^{-1}}^2 - \|\boldsymbol{\mu}_t - \boldsymbol{\mu}_{t+1}\|_{\boldsymbol{\Sigma}_{t+1}^{-1}}^2 \right] \end{aligned} \quad (5.8)$$

Now, a lower bound for $\mathbf{g}_t^\top (\boldsymbol{\mu}_{t+1} - \boldsymbol{\mu})$ becomes,

$$\begin{aligned} &\mathbf{g}_t^\top (\boldsymbol{\mu}_{t+1} - \boldsymbol{\mu}) \\ &= \mathbf{g}_t^\top (\boldsymbol{\mu}_t - \boldsymbol{\mu}) + \mathbf{g}_t^\top (\boldsymbol{\mu}_{t+1} - \boldsymbol{\mu}_t) \\ &= (L_t + M_t)(-y_t \mathbf{x}_t^\top \boldsymbol{\mu}_t) + (L_t + M_t)y_t \mathbf{x}_t^\top \boldsymbol{\mu} - \frac{1}{\eta} \|\boldsymbol{\mu}_{t+1} - \boldsymbol{\mu}_t\|_{\boldsymbol{\Sigma}_{t+1}^{-1}}^2, \end{aligned} \quad (5.9)$$

where the second inequality used the facts $\mathbf{g}_t = (L_t + M_t)(-y_t \mathbf{x}_t)$ and $\partial h_t(\boldsymbol{\mu}_{t+1}) = 0$ i.e.,

$$\boldsymbol{\Sigma}_{t+1}^{-1}(\boldsymbol{\mu}_{t+1} - \boldsymbol{\mu}_t) + \eta \mathbf{g}_t = 0. \quad (5.10)$$

Combining the above equality (5.9) with the facts

$$\begin{aligned} M_t(-y_t \mathbf{x}_t^\top \boldsymbol{\mu}_t) &= M_t|y_t \mathbf{x}_t^\top \boldsymbol{\mu}_t| = M_t|p_t| \\ L_t(-y_t \mathbf{x}_t^\top \boldsymbol{\mu}_t) &= L_t(-|y_t \mathbf{x}_t^\top \boldsymbol{\mu}_t|) = -L_t|p_t| \\ y_t \mathbf{x}_t^\top \boldsymbol{\mu}_t + \delta \ell_t(\boldsymbol{\mu}/\delta) &\geq y_t p_t + \delta(1 - y_t p_t/\delta) = \delta, \end{aligned}$$

The bound for $\mathbf{g}_t^\top(\boldsymbol{\mu}_{t+1} - \boldsymbol{\mu})$ is shown as follows,

$$\begin{aligned} &\mathbf{g}_t^\top(\boldsymbol{\mu}_{t+1} - \boldsymbol{\mu}) \\ &\geq (M_t|p_t| - L_t|p_t|) + (L_t + M_t)[\delta - \delta \ell_t(\boldsymbol{\mu}/\delta)] - \frac{1}{\eta} \|\boldsymbol{\mu}_{t+1} - \boldsymbol{\mu}_t\|_{\boldsymbol{\Sigma}_{t+1}^{-1}}^2 \\ &= [M_t(\delta + |p_t|) + L_t(\delta - |p_t|)] - (L_t + M_t)\delta \ell_t(\boldsymbol{\mu}/\delta) - \frac{1}{\eta} \|\boldsymbol{\mu}_{t+1} - \boldsymbol{\mu}_t\|_{\boldsymbol{\Sigma}_{t+1}^{-1}}^2 \end{aligned} \quad (5.11)$$

Combining the above two inequalities (5.8) and (5.11), will give the following important inequality

$$\begin{aligned} &[M_t(\delta + |p_t|) + L_t(\delta - |p_t|)] \\ &\leq \frac{1}{2\eta} \left[\|\boldsymbol{\mu}_t - \boldsymbol{\mu}\|_{\boldsymbol{\Sigma}_{t+1}^{-1}}^2 - \|\boldsymbol{\mu}_{t+1} - \boldsymbol{\mu}\|_{\boldsymbol{\Sigma}_{t+1}^{-1}}^2 - \|\boldsymbol{\mu}_t - \boldsymbol{\mu}_{t+1}\|_{\boldsymbol{\Sigma}_{t+1}^{-1}}^2 \right] + \frac{1}{\eta} \|\boldsymbol{\mu}_{t+1} - \boldsymbol{\mu}_t\|_{\boldsymbol{\Sigma}_{t+1}^{-1}}^2 + \delta \ell_t(\boldsymbol{\mu}/\delta) \\ &= \frac{1}{2\eta} \left[\|\boldsymbol{\mu}_t - \boldsymbol{\mu}\|_{\boldsymbol{\Sigma}_{t+1}^{-1}}^2 - \|\boldsymbol{\mu}_{t+1} - \boldsymbol{\mu}\|_{\boldsymbol{\Sigma}_{t+1}^{-1}}^2 + \|\boldsymbol{\mu}_t - \boldsymbol{\mu}_{t+1}\|_{\boldsymbol{\Sigma}_{t+1}^{-1}}^2 \right] + \delta \ell_t(\boldsymbol{\mu}/\delta) \end{aligned}$$

Replacing $\boldsymbol{\mu}$ with $\delta \boldsymbol{\mu}$ concludes the proof.

Theorem 6 *Let $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_T, y_T)$ be a sequence of input examples, where $\mathbf{x}_t \in \mathbb{R}^d$ and $y_t \in \{-1, +1\}$ for all t . If the SOAL algorithm is run on this sequence of examples, then the expected number of prediction mistakes made is bounded from above by the following inequality, for any vector $\boldsymbol{\mu} \in \mathbb{R}^d$,*

$$\begin{aligned} \mathbb{E} \left[\sum_{t=1}^T M_t \right] &\leq \mathbb{E} \left[\sum_{t=1}^T Z_t \ell_t(\boldsymbol{\mu}) \right] + \frac{D_{\boldsymbol{\mu}} + (1 - \delta)^2 \|\boldsymbol{\mu}\|^2}{\eta \delta} \text{Tr}(\boldsymbol{\Sigma}_{T+1}^{-1}) \\ &\quad + \frac{1}{\delta} \mathbb{E} \left[\sum_{\rho_t < 0} \frac{\eta \gamma v_t}{(\gamma + v_t)} \right] + \frac{2}{\delta} \mathbb{E} \left[\sum_{\rho_t > 0} L_t \right] - \mathbb{E} \left[\sum_{t=1}^T L_t \right] \end{aligned} \quad (5.12)$$

where $\delta > 0$.

Proof: First, according to the equality (5.10), it's clear that

$$\begin{aligned}
 & \|\boldsymbol{\mu}_t - \boldsymbol{\mu}_{t+1}\|_{\boldsymbol{\Sigma}_{t+1}^{-1}}^2 \\
 &= \eta^2 \mathbf{g}_t^\top \boldsymbol{\Sigma}_{t+1} \mathbf{g}_t \\
 &= \eta^2 (M_t + L_t) \mathbf{x}_t^\top \boldsymbol{\Sigma}_{t+1} \mathbf{x}_t \\
 &= \eta^2 (M_t + L_t) \left(\mathbf{x}_t^\top \boldsymbol{\Sigma}_t \mathbf{x}_t - \frac{\mathbf{x}_t^\top \boldsymbol{\Sigma}_t \mathbf{x}_t \mathbf{x}_t^\top \boldsymbol{\Sigma}_t \mathbf{x}_t}{\gamma + \mathbf{x}_t^\top \boldsymbol{\Sigma}_t \mathbf{x}_t} \right) \\
 &= \eta^2 (M_t + L_t) \frac{\gamma v_t}{\gamma + v_t},
 \end{aligned} \tag{5.13}$$

where the updating rule of $\boldsymbol{\Sigma}$ is used. Plugging it into the inequality in the Theorem 6 and re-arranging it will give

$$\begin{aligned}
 & Z_t \left[M_t \left(\delta + |p_t| - \frac{\eta \gamma v_t}{2(\gamma + v_t)} \right) + L_t \left(\delta - |p_t| - \frac{\eta \gamma v_t}{2(\gamma + v_t)} \right) \right] \\
 & \leq \frac{1}{2\eta} Z_t \left[\|\boldsymbol{\mu}_t - \delta \boldsymbol{\mu}\|_{\boldsymbol{\Sigma}_{t+1}^{-1}}^2 - \|\boldsymbol{\mu}_{t+1} - \delta \boldsymbol{\mu}\|_{\boldsymbol{\Sigma}_{t+1}^{-1}}^2 \right] + \delta \ell_t(\boldsymbol{\mu})
 \end{aligned} \tag{5.14}$$

Summing the above inequality over $t = 1, 2, \dots, T$ and using the definition of ρ_t can give

$$\begin{aligned}
 & \sum_{t=1}^T Z_t \left[M_t(\delta + \rho_t) + L_t(\delta + \rho_t - 2|p_t|) \right] \\
 & \leq \delta \sum_{t=1}^T Z_t \ell_t(\boldsymbol{\mu}) + \frac{1}{2\eta} \sum_{t=1}^T Z_t \left[\|\boldsymbol{\mu}_t - \delta \boldsymbol{\mu}\|_{\boldsymbol{\Sigma}_{t+1}^{-1}}^2 - \|\boldsymbol{\mu}_{t+1} - \delta \boldsymbol{\mu}\|_{\boldsymbol{\Sigma}_{t+1}^{-1}}^2 \right]
 \end{aligned} \tag{5.15}$$

Now, the right hand side of the above inequality can be bounded as follows.

First, the first term is bounded as

$$\begin{aligned}
 & \sum_{t=1}^T Z_t \left[\|\boldsymbol{\mu}_t - \delta \boldsymbol{\mu}\|_{\boldsymbol{\Sigma}_{t+1}^{-1}}^2 - \|\boldsymbol{\mu}_{t+1} - \delta \boldsymbol{\mu}\|_{\boldsymbol{\Sigma}_{t+1}^{-1}}^2 \right] \\
 & \leq \|\boldsymbol{\mu}_1 - \delta \boldsymbol{\mu}\|_{\boldsymbol{\Sigma}_2^{-1}}^2 + \sum_{t=2}^T \left[\|\boldsymbol{\mu}_t - \delta \boldsymbol{\mu}\|_{\boldsymbol{\Sigma}_{t+1}^{-1}}^2 - \|\boldsymbol{\mu}_t - \delta \boldsymbol{\mu}\|_{\boldsymbol{\Sigma}_t^{-1}}^2 \right] \\
 & \leq \|\boldsymbol{\mu}_1 - \delta \boldsymbol{\mu}\|^2 \text{Tr}(\boldsymbol{\Sigma}_2^{-1}) + \sum_{t=2}^T \|\boldsymbol{\mu}_t - \delta \boldsymbol{\mu}\|^2 \text{Tr}(\boldsymbol{\Sigma}_{t+1}^{-1} - \boldsymbol{\Sigma}_t^{-1}) \\
 & = \max_{t \leq T} \|\boldsymbol{\mu}_t - \delta \boldsymbol{\mu}\|^2 \text{Tr}(\boldsymbol{\Sigma}_{T+1}^{-1}) \\
 & \leq 2(D_{\boldsymbol{\mu}} + (1 - \delta)^2 \|\boldsymbol{\mu}\|^2) \text{Tr}(\boldsymbol{\Sigma}_{T+1}^{-1})
 \end{aligned} \tag{5.16}$$

where $D_{\boldsymbol{\mu}} = \max_{t \leq T} \|\boldsymbol{\mu}_t - \boldsymbol{\mu}\|^2$. Plugging the above upper bound (5.16) into the inequality (5.15), it's easy to get

$$\begin{aligned} & \sum_{t=1}^T Z_t [M_t(\delta + \rho_t) + L_t(\delta + \rho_t - 2|p_t|)] \\ & \leq \frac{1}{\eta} (D_{\boldsymbol{\mu}} + (1 - \delta)^2 \|\boldsymbol{\mu}\|^2) \text{Tr}(\boldsymbol{\Sigma}_{T+1}^{-1}) + \delta \sum_{t=1}^T Z_t \ell_t(\boldsymbol{\mu}) \end{aligned} \quad (5.17)$$

When $\rho_t > 0$, using $\mathbb{E}_t Z_t = \delta/(\delta + \rho_t)$, we have

$$\begin{aligned} & \mathbb{E} \left\{ Z_t [M_t(\delta + \rho_t) + L_t(\delta + \rho_t - 2|p_t|)] \right\} \\ & = \delta \mathbb{E} M_t + \delta \mathbb{E} L_t [1 - 2|p_t|/(\delta + \rho_t)] \geq \delta \mathbb{E} M_t + (\delta - 2) \mathbb{E} L_t. \end{aligned} \quad (5.18)$$

When $\rho_t \leq 0$, i.e., $|p_t| \leq \frac{\eta\gamma v_t}{2(\gamma + v_t)}$, using $\mathbb{E}_t Z_t = 1$, we have

$$\begin{aligned} & \mathbb{E} \left\{ Z_t [M_t(\delta + \rho_t) + L_t(\delta + \rho_t - 2|p_t|)] \right\} \\ & \geq \mathbb{E} M_t \left(\delta - \frac{\eta\gamma v_t}{2(\gamma + v_t)} \right) + \mathbb{E} L_t \left(\delta - \frac{\eta\gamma v_t}{\gamma + v_t} \right) \end{aligned} \quad (5.19)$$

To summarize,

$$\begin{aligned} & \sum_{t=1}^T \mathbb{E} \left\{ Z_t [M_t(\delta + \rho_t) + L_t(\delta + \rho_t - 2|p_t|)] \right\} \\ & \geq \delta \mathbb{E} \left[\sum_{t=1}^T M_t \right] + \delta \mathbb{E} \left[\sum_{t=1}^T L_t \right] - 2 \mathbb{E} \left[\sum_{\rho_t > 0} L_t \right] - \mathbb{E} \left[\sum_{\rho_t < 0} M_t \frac{\eta\gamma v_t}{2(\gamma + v_t)} \right] - \mathbb{E} \left[\sum_{\rho_t < 0} L_t \frac{\eta\gamma v_t}{\gamma + v_t} \right] \end{aligned} \quad (5.20)$$

Taking expectation of the inequality (5.17) and combining with the above inequality conclude the proof.

Remark: First, when $\gamma = 1$, $\mathbb{E} \sum_{\rho_t < 0} \frac{\gamma v_t}{(\gamma + v_t)} \leq \sum_{i=1}^d \ln(1 + \lambda_i)$ (λ_i is the i -th eigenvalue of the correlation matrix $\sum_{t=1}^T M_t X_t \mathbf{x}_t \mathbf{x}_t^\top$), where the right hand side is used in the Theorem 3 of [CBGZ06], which implies our term is better.

Secondly, when the third term $\frac{\eta\gamma v_t}{(\gamma v_t)}$ in the above mistake bound is too big, it is important to query the true label and make the mistake bound small in the following instances, thus make sure the expectation of this term is small. This supports our query strategy when $\rho_t \leq 0$.

Thirdly, the fourth term $\frac{2}{\delta}\mathbb{E}[\sum_{\rho_t > 0} L_t]$ roughly corresponds to the condition: $\rho_t > 0$. As L_t is defined as the loss suffered on the instances $\hat{y}_t = y_t$ (see the beginning of section 3.3), it's not worth to always query the true labels, thus this motivates us to draw a Bernoulli random variable based on ρ_t , which could be treated as an adjusted margin on the t -th instance. It also should be noted that when L_t is big, it indicates p_t is small, which usually corresponds to a small ρ_t , thus the instance has more chance to be queried based on our query strategy to make L_t small.

Lastly, since $\mathbb{E} \sum_{\rho_t < 0} \frac{\gamma v_t}{(\gamma + v_t)} \leq \gamma \mathbb{E} \ln(|\Sigma_{T+1}^{-1}|)$, if η is set as $\sqrt{\frac{(D_{\mu} + (1-\delta)^2 \|\mu\|^2) \text{Tr}(\Sigma_{T+1}^{-1})}{\gamma \ln|\Sigma_{T+1}^{-1}|}}$, the following minimum mistake bound can be obtained,

$$\begin{aligned} \mathbb{E} \left[\sum_{t=1}^T M_t \right] &\leq \mathbb{E} \sum_{t=1}^T Z_t \ell_t(\mu) + \frac{2}{\delta} \mathbb{E} \left[\sum_{\rho_t > 0} L_t \right] - \mathbb{E} \left[\sum_{t=1}^T L_t \right] \\ &\quad + \frac{2}{\delta} \sqrt{D_{\mu} + (1-\delta)^2 \|\mu\|^2} \sqrt{\gamma \text{Tr}(\Sigma_{T+1}^{-1}) \ln |\Sigma_{T+1}^{-1}|}. \end{aligned} \quad (5.21)$$

5.2 Experiments

5.2.1 Compared Algorithms and Experimental Testbed

The proposed SOAL algorithm is compared with the state-of-the-art algorithms, which are listed as follows:

- “APE”: the Active PErceptron algorithm [CBGZ06];
- “APAI”: the state-of-the-art Active Passive-Aggressive algorithm [LZH16];
- “ASOP”: the state-of-the-art Second-Order Active Perceptron algorithm [CBGZ06];
- “SOL”: the passive version of SOAL algorithm which queries all of the instances;
- “SORL”: the random version of SOAL algorithm with random query strategy;
- “SOAL”: the proposed Second-order Online Active Learning in Algorithm 5.

To examine the performance of proposed algorithm, extensive experiments are conducted on a variety of benchmark datasets from machine learning repositories. Table 5.1 shows the details of datasets used in the following experiments. All of these datasets

can be downloaded from LIBSVM website ¹ and UCI machine learning repository ². These datasets are chosen fairly randomly in order to cover various aspects of dataset characteristics.

Table 5.1: Summary of datasets in the experiments.

Dataset	# Instances	# Features	Source
a8a	32,561	123	LIBSVM
aloi	108,000	128	LIBSVM
clean	7,074	167	UCI
covtype	116,405	54	LIBSVM
ijcnn1	191,681	22	LIBSVM
kddcup99	494,012	41	UCI
letter	20,000	16	LIBSVM
magic04	19,002	10	LIBSVM
optdigits	5,620	64	UCI
satimage	6,435	36	LIBSVM
spambase	4,601	57	LIBSVM
w8a	64,700	300	LIBSVM
URL	1,603,985	3,231,961	LIBSVM
webspam	350,000	254	LIBSVM

All the compared algorithms learn a linear classifier for the binary classification tasks. All of the parameters are searched from $10^{[-5:5]}$ through cross validation for all the algorithms and datasets. The smoothing parameter δ is set as $2^{[-10:10]}$ in order to examine varied querying ratios. All the experiments were conducted over 20 runs of different random permutations for each dataset. All the results were reported by averaging over these 20 runs. The online classification accuracy is adopted as the predictive performance metric.

5.2.2 Evaluation of Fixed Query Ratio

In this section, the performance of the proposed SOAL algorithm with fixed query ratios is evaluated. Table 5.2 shows the results on all of the datasets. From the results, several observations can be made.

¹<http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/>

²<http://www.ics.uci.edu/~mllearn/MLRepository.html>

Table 5.2: Evaluation of SOAL against other baselines with fixed query ratio about 20%.

a8a				aloi		
Algorithm	Query	Accuracy	Time (s)	Query	Accuracy	Time (s)
APE	20.462±0.431	0.810±0.002	0.015±0.001	20.100±0.216	0.538±0.002	0.067±0.009
APAI	20.000±0.319	0.836±0.001	0.019±0.006	20.400±0.281	0.549±0.002	0.074±0.010
ASOP	20.400±0.745	0.815±0.002	1.190±0.066	20.199±0.665	0.558±0.002	4.987±0.047
SORL	19.448±0.491	0.837±0.002	0.079±0.005	19.843±0.261	0.587±0.002	0.776±0.016
SOAL	19.457±0.363	0.843±0.001	0.133±0.008	19.816±0.274	0.597±0.002	0.884±0.017
clean				covtype		
Algorithm	Query	Accuracy	Time (s)	Query	Accuracy	Time (s)
APE	20.618±1.072	0.617±0.006	0.006±0.001	20.482±0.089	0.650±0.001	0.237±0.012
APAI	20.535±1.230	0.663±0.005	0.006±0.001	20.194±0.127	0.685±0.000	0.273±0.013
ASOP	20.645±0.734	0.680±0.007	0.574±0.005	20.749±1.372	0.754±0.002	3.509±0.145
SORL	20.277±1.845	0.655±0.006	0.062±0.006	20.389±0.148	0.744±0.000	0.626±0.022
SOAL	20.247±1.714	0.700±0.005	0.130±0.006	20.412±0.121	0.772±0.000	0.833±0.028
ijcnn1				kddcup09		
Algorithm	Query	Accuracy	Time (s)	Query	Accuracy	Time (s)
APE	20.978±0.264	0.917±0.001	0.070±0.008	20.388±1.275	0.991±0.000	0.206±0.016
APAI	20.516±0.131	0.931±0.000	0.080±0.002	20.067±0.321	0.993±0.000	0.234±0.013
ASOP	20.766±1.901	0.925±0.001	0.310±0.007	20.670±4.438	0.993±0.000	1.772±0.033
SORL	20.235±0.321	0.929±0.001	0.083±0.007	19.410±3.284	0.993±0.000	0.229±0.014
SOAL	20.229±0.298	0.942±0.000	0.139±0.010	19.394±3.270	0.994±0.000	0.406±0.022
letter				magic04		
Algorithm	Query	Accuracy	Time (s)	Query	Accuracy	Time (s)
APE	20.171±0.935	0.977±0.001	0.006±0.001	20.139±1.113	0.680±0.006	0.004±0.007
APAI	20.474±0.244	0.984±0.000	0.007±0.001	20.475±2.303	0.695±0.009	0.003±0.006
ASOP	20.366±0.903	0.986±0.001	0.024±0.008	20.306±12.866	0.683±0.010	0.013±0.006
SORL	20.026±0.543	0.988±0.001	0.006±0.001	20.093±0.613	0.705±0.012	0.007±0.008
SOAL	20.097±0.401	0.991±0.000	0.011±0.001	19.889±0.799	0.744±0.001	0.006±0.008
optdigits				satimage		
Algorithm	Query	Accuracy	Time (s)	Query	Accuracy	Time (s)
APE	20.786±0.379	0.858±0.005	0.003±0.001	20.489±0.956	0.828±0.002	0.004±0.007
APAI	20.439±0.390	0.889±0.002	0.003±0.001	20.735±0.592	0.878±0.003	0.002±0.005
ASOP	20.357±0.454	0.900±0.004	0.053±0.001	20.641±1.459	0.921±0.015	0.019±0.007
SORL	20.508±0.806	0.907±0.003	0.005±0.000	19.957±0.369	0.968±0.003	0.002±0.006
SOAL	20.388±0.583	0.923±0.002	0.012±0.001	20.096±0.296	0.977±0.001	0.006±0.008
spambase				w8a		
Algorithm	Query	Accuracy	Time (s)	Query	Accuracy	Time (s)
APE	20.954±0.630	0.586±0.002	0.001±0.000	19.771±0.545	0.974±0.000	0.034±0.005
APAI	20.526±0.606	0.600±0.004	0.002±0.001	20.846±0.277	0.981±0.000	0.039±0.004
ASOP	20.467±2.903	0.629±0.031	0.034±0.001	19.951±0.599	0.977±0.000	12.732±0.030
SORL	20.371±1.612	0.718±0.016	0.006±0.001	20.190±0.434	0.979±0.000	0.176±0.006
SOAL	20.115±1.248	0.765±0.012	0.008±0.001	20.141±0.363	0.984±0.000	0.373±0.012

First of all, the proposed SOAL algorithm consistently achieves the best classification accuracy on all the datasets. This confirms the effectiveness of proposed algorithm in querying informative labels by considering both the margin value and the second-order information. Moreover, the random query variant algorithm SORL also outperforms the first-order based algorithms (APE and APAII) on most of the datasets, which is consistent with the previous study [CBGZ06], and further confirms the effectiveness of exploiting second-order information.

Second, by conducting unpaired t -tests, it's found the difference between our proposed SOAL algorithm with the other algorithms is statistically extremely significant. This confirms that the proposed algorithm is able to significantly outperform the other existing online active learning algorithms.

Third, by examining the time cost, the proposed SOAL algorithm spent slightly more time than the other baselines except ASOP. Compared with the first-order based algorithms (APE and APAII), the extra time cost of SOAL is due to the computation of second-order information. Compared with its random variant SORL, the extra small time cost of SOAL is mainly due to the computation of the query strategy. However, the extra time cost can almost be ignored considering the high efficiency of the online learning scheme. In practice, for the datasets with high dimensionality, the diagonal-variant algorithm can be adopted with Equation (5.4) and (5.5) to further reduce the running time cost, as shown in subsequent experiments.

5.2.3 Evaluation of Varied Query Ratios

In this experiment, the performance of proposed algorithm SOAL with varied query ratio is evaluated by setting different δ . Figure 5.1 and 5.2 summarizes the average performance on all of the datasets. Based on the results, several observations could be made.

First of all, similar observations could be made as the ones in Table 5.2. With same query ratio, the proposed SOAL algorithm could consistently achieve the best performance. This observation further confirms the effectiveness and efficiency of proposed second-order online active learning algorithm.

Second, on all of the datasets, it can be observed that the accuracy of SOAL algorithm usually increases along with the increase of the query ratio at the beginning, and saturates

(a) a8a

(b) aloi

(c) clean

(d) covtype

(e) ijcn1

(f) kddcup09

Figure 5.1: Evaluation of accuracy against varied query ratio.

(a) letter

(b) magic04

(c) optdigits

(d) satimage

(e) spambase

(f) w8a

Figure 5.2: Evaluation of accuracy against varied query ratio.

quickly after the query ratio exceeds some value (usually around 25%). This result indicates that the proposed query strategy can effectively explore those most informative instances for updating the classifiers in a rather effective and efficient way. Besides, with query ratio around 25%, the proposed SOAL algorithm could achieve similar even better accuracy than its passive algorithm SOL which query 100% of the instances. This observation further confirms the effectiveness of the proposed query strategy in reducing labeling cost while achieving comparable performance.

Finally, on some datasets (covtype and ijcnn1 etc.), the accuracy achieved by SOAL could get the highest value with few query ratio (around 10%), and decreases when the query ratio increases further. This observation is interesting as it's expected to see increasing performance when the query ratio increases. It may be caused by the over-fitting issue on the noisy datasets.

5.2.4 Evaluation of Parameter Sensitivity

In the previous section, the parameters η and γ are searched from $10^{[-5:5]}$. In this section, the sensitivity of these parameters in Algorithm 5 is investigated. Figure 5.3 shows the experimental results on both **clean** and **covtype** datasets. On each of the datasets, x-axis and y-axis correspond to parameters η and γ , respectively, and different colors correspond to different results in terms of accuracy. Overall, it can be observed that Algorithm 5 is a little sensitive to the values of parameter η and γ .

5.2.5 Evaluation of Scalability and Efficiency

In this section, the scalability of the proposed SOAL algorithm is evaluated, and Figure 5.4 shows the time cost corresponding to varied query ratios on two datasets. As expected, the first-order based algorithms APE and APAII are the most efficient ones among all the algorithms, and the second-order based algorithms (ASOP, SOL, SORL, SOAL) cost more time due to the computation of the second-order information. Moreover, the proposed SOAL algorithm costs more time than its random query version SORL due to the cost of the query strategy. Although the proposed SOAL algorithm took more time than its random version SORL and the first-order based algorithms as well, the extra time cost almost can be ignored considering the high efficiency of the online learning scheme.

Figure 5.3: Evaluation of accuracy with respect to different settings of parameters η and γ in Algorithm 5.

Figure 5.4: Evaluation of time cost w.r.t. varied query ratios.

5.2.6 Application to Large-Scale Web Data Classification

In this section, the proposed algorithm is applied to two large-scale web data classification tasks:

- (i) Malicious URL detection [MSSV09] which contains 1,603,985 URLs with 3,231,961 features, and the task is to classify an URL into normal vs. malicious;
- (ii) Webspam classification [WIP12], which has a total of 350,000 instances with 254 features.

These two datasets can be downloaded from LIBSVM website ³. For the URL dataset, the dimensionality is too high to maintain the full matrix Σ in memory for the second-order based algorithms when running the experiments on a PC. So the diagonal updating rules (Equation (5.4) and (5.5)) are adopted in Algorithm 5. The diagonal versions of SOAL and SORL algorithms are named as SOAL-diag and SORL-diag, respectively. Similarly, the diagonal version of ASOP algorithm is also adopted, which is named as ASOP-diag.

Figure 5.5 shows the experimental results on two large-scale web classification tasks. Similar observations can be drawn from the results, in which the proposed SOAL algorithm significantly outperforms the other competing baselines particularly when the query ratio is small. Moreover, from the results of Figure 5.5 (b), it also can be observed the diagonal version of the proposed algorithm, i.e., SOAL-diag, still considerably outperforms the baselines from the beginning of the query ratio. These encouraging results again validate the effectiveness of the proposed second order online active learning technique. Finally, by examining the time cost, most of these algorithms are computationally efficient even for such large-scale classification tasks. For example, on the dataset of *URL*, SOAL-diag took only around 3 seconds when querying 20% ratio of instances, which though is slightly slower than the first-order algorithm APAIL that took around 1.5 seconds, but is still very efficient for such a large-scale classification task.

³<http://www.csie.ntu.edu.tw/~cjlin/libsvmtools>

Figure 5.5: Evaluation of accuracy against the varied query ratios for large-scale web data classification tasks.

5.3 Summary

In this chapter, a new framework of online active learning is proposed, named as Second-order Online Active Learning (SOAL). The proposed SOAL algorithm is theoretically analyzed and empirically evaluated with a set of extensive experiments. Finally, the proposed SOAL algorithm is applied into large-scale web data classification tasks and encouraging results are found. For future work, it's interesting to explore some self-tuned learning strategy for automatically adjusting the parameters γ and η in the online learning process. Besides, it's also interesting to investigate how the proposed SOAL algorithm may be able to handle some challenging online learning tasks with concept drifts [KJ00].

Chapter 6

Online Active Learning for Relative Similarity Learning

Similarity learning, also known as distance learning, has been widely studied in machine learning and data mining communities. It plays a critical role in a wide range of real-world applications, such as ranking[CSSB10b], recommendation system[AT05], image retrieval [RHC99, SJHL06, CYL13, WLX13, Lew91] and text information retrieval [FBY92], etc. In previous research, a variety of machine learning approaches have been proposed for learning distance or similarity functions from training data [YJ06, WJH⁺09, HLLM06]. One of the most notable schemes is the Distance Metric Learning (DML) approach, which aims to learn a Mahalanobis distance [YJ06] and requires the distance metric satisfying the Positive Semi-Definiteness (PSD) property. However, imposing the PSD requirement often makes the DML task computationally challenging, particularly when handling large-scale training data in high dimensional spaces, although various approximate techniques have been proposed to improve computational efficiency. Another strong assumption of DML approaches is that the exact distances of all pairs of instances should be obtained, which are usually costly and time consuming.

Instead of learning the Mahalanobis distance, another technique is to explore the relative similarity learning [CSSB10b], which aims to learn a bilinear similarity model for measuring the similarity of any instance pair (two instances) by eliminating the PSD requirement, which is thus much more computationally efficient and scalable. More importantly, only relative similarities among any two pairs of instances are needed, which greatly alleviates the requirement of exact distances among all instances. In literature,

several online learning algorithms have been proposed for learning relative similarity from data streams. Most existing studies generally assume that the data streams of instance pairs are fully labeled. However, this may not be a realistic setting since the number of instance pairs is quadratic with respect to the number of instances, and labeling such a massive pool of instance pairs could be extremely costly and time consuming for large-scale applications, especially when the labeling task has to explicitly involve human in the loop in an early stage of deploying a new system.

Unlike the existing relative similarity learning approaches, in this chapter, a novel scheme of active online learning of relative similarity is investigated. Specifically, two efficient and scalable online learning algorithms are proposed to tackle the new problem: (i) PAAS: Passive-Aggressive Active Similarity learning and (ii) CWAS: Confidence-Weighted Active Similarity learning. The theoretical bounds of the proposed online learning algorithms are investigated, and further their empirical performances are studied via an extensive set of experiments on various real datasets. The encouraging results show that the proposed online active relative similarity learning algorithms can achieve highly competitive performance while significantly reducing the amount of labeling costs.

In this chapter, Section 1 present the formulation of the proposed two algorithms as well as their theoretical analysis. Section 2 empirically validate the proposed algorithms in terms of effectiveness and efficiency, and finally section 3 concludes this chapter.

6.1 Online Active Learning of Relative Similarity from Data Streams

In this section, the problem setting for active online learning of relative similarity is first formulated, and then the details of the proposed Active Online Similarity Learning algorithms are presented.

6.1.1 Problem Formulation

Following [CSSB10b], this study focuses on the problem of online learning a relative similarity function $S(\mathbf{x}, \mathbf{x}')$, $\forall \mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$, which measures the similarity between $\mathbf{x}$ and $\mathbf{x}'$. Formally, let

$$\{(\mathbf{x}_t, \mathbf{x}_t^1, \mathbf{x}_t^2; y_t) \in \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^d \times \{-1, +1\} | t \in [T]\}$$

be a sequence of triplets (where $[T] = \{1, \dots, T\}$). For the t -th triplet, $y_t = 1$ indicates $\mathbf{x}_t$ instance is more similar with $\mathbf{x}_t^1$ than $\mathbf{x}_t^2$, while $y_t = -1$ implies $\mathbf{x}_t$ is less similar with $\mathbf{x}_t^1$ than $\mathbf{x}_t^2$. Our goal is to learn a similarity function $S(\mathbf{x}, \mathbf{x}')$ that assigns a higher similarity score to the similar pair and a lower similarity score to the non-similar pair. Formally, given a triplet $(\mathbf{x}_t, \mathbf{x}_t^1, \mathbf{x}_t^2; y_t)$, the learned similar function should satisfy:

$$y_t[S(\mathbf{x}_t, \mathbf{x}_t^1) - S(\mathbf{x}_t, \mathbf{x}_t^2)] \geq 0, \quad \forall t \in [T].$$

For the similarity function, a linear similarity function is adopted that has a bi-linear form,

$$S(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \mathbf{M} \mathbf{x}', \quad (6.1)$$

where $\mathbf{M} \in \mathbb{R}^{d \times d}$. To note, it is possible to learn a $\mathbf{M} \in \mathbb{R}^{d \times d'}$, where $d \neq d'$, as similarity function between two different spaces, however it is assumed $d = d'$ for simplicity.

To learn the parameter $\mathbf{M}$, some loss functions should be defined to measure its performance on the t -th triplet, for example, the hinge loss is defined as

$$\ell(M; (\mathbf{x}_t, \mathbf{x}_t^1, \mathbf{x}_t^2; y_t)) = \left[1 - y_t[S(\mathbf{x}_t, \mathbf{x}_t^1) - S(\mathbf{x}_t, \mathbf{x}_t^2)] \right]_+, \quad (6.2)$$

where the hinge loss $[\cdot]_+ = \max(0, \cdot)$ encourages $y_t[S(\mathbf{x}_t, \mathbf{x}_t^1) - S(\mathbf{x}_t, \mathbf{x}_t^2)] \geq 1$. It is also possible to define other loss functions, e.g. logistic loss. In this article, the hinge loss is adopted in this study for simplicity.

To solve this online relative similarity learning task, several existing methods can be adopted, including Perceptron, Online Gradient Descent, Online Passive Aggressive, etc. Although these online learning algorithms can achieve a cumulative loss comparable with the one attained by any fixed hypothesis, one major limitation is that they require all the labels of the instances while labeling instances might be very expensive or time consuming. To solve this problem, active online similarity learning is studied, which will try to only query the labels of a few informative instances so that the number of queried labels is reduced while the performance does not degrade too much.

6.1.2 PAAS: Passive-Aggressive Active Similarity learning

To solve Active Online Similarity Learning, the selective sampling technique used in [CBGZ06] is adopted in this study. Specifically, a stochastic sampling scheme is adopted to decide whether it is necessary to query the label of the current instance. This scheme maintains a Bernoulli random variable $Z_t \in \{0, 1\}$, where $Z_t = 1$ indicates the label should be queried at the t -th step. More specifically, this scheme employs the following sampling probability

$$\Pr(Z_t = 1) = \frac{\delta}{\delta + |p_t|}, \quad (6.3)$$

where $\delta > 0$ is a parameter to tune the number of queried labels and

$$p_t = \mathbf{x}_t^\top \mathbf{M}_t (\mathbf{x}_t^1 - \mathbf{x}_t^2) \quad (6.4)$$

is the margin of similarities between $\mathbf{x}_t^\top \mathbf{M}_t \mathbf{x}_t^1$ and $\mathbf{x}_t^\top \mathbf{M}_t \mathbf{x}_t^2$. Intuitively, the smaller the value of $|p_t|$, the lower the confidence of the model on the current prediction; as a consequence, the label with a higher probability of $\Pr(Z_t = 1)$ would be queried. Once the label is queried, the algorithm will update the model using the online Passive Aggressive strategy, i.e.,

$$\mathbf{M}_{t+1} = \arg \min_{\mathbf{M}} \left\{ \frac{1}{2} \|\mathbf{M} - \mathbf{M}_t\|_F^2 + C \ell_t(M) \right\},$$

where $\ell_t(M) = \ell(M; (\mathbf{x}_t, \mathbf{x}_t^1, \mathbf{x}_t^2; y_t))$ and $C > 0$ is a trade-off between minimizing the adjustment of the model and minimizing the loss of the new model on the current example. This will produce a first-order update method [CSSB10b, CDKSs06], i.e.,

$$\mathbf{M}_{t+1} = \mathbf{M}_t + \tau_t y_t \mathbf{X}_t,$$

where

$$\tau_t = \min \left\{ C, \frac{\ell_t(\mathbf{M}_t)}{\|\mathbf{X}_t\|_F^2} \right\}, \quad \mathbf{X}_t = \mathbf{x}_t (\mathbf{x}_t^1 - \mathbf{x}_t^2)^\top$$

Finally, the proposed Passive-Aggressive Active Similarity learning (PAAS) algorithm is summarized in Algorithm 6.

Algorithm 6 PAAS: The proposed algorithm of Passive-Aggressive Active Similarity learning

Input: smooth parameters $\delta > 0, C > 0$
Initialize: $\mathbf{M}_1 = I$
for $t = 1, 2, \dots, T$ **do**
 Receive $(\mathbf{x}_t, \mathbf{x}_t^1, \mathbf{x}_t^2)$ and compute $\mathbf{X}_t = \mathbf{x}_t(\mathbf{x}_t^1 - \mathbf{x}_t^2)^\top$;
 Compute $p_t = \text{Tr}(\mathbf{M}_t \mathbf{X}_t^\top)$ and $\hat{y}_t = \text{sgn}(p_t)$;
 Sample $Z_t \in \{0, 1\}$ with $\Pr(Z_t = 1) = \frac{\delta}{\delta + |p_t|}$;
 if $Z_t = 1$ **then**
 Query y_t and compute $\ell_t(\mathbf{M}_t) = \max(0, 1 - y_t p_t)$;
 Update: $\mathbf{M}_{t+1} = \mathbf{M}_t + \tau_t y_t \mathbf{X}_t$;
 where $\tau_t = \min \left\{ C, \frac{\ell_t(\mathbf{M}_t)}{\|\mathbf{x}_t\|_F^2} \right\}$.
 else
 $\mathbf{M}_{t+1} = \mathbf{M}_t$;
 end if
end for
Output: $\mathbf{M}_{T+1}$

6.1.3 CWAS: Confidence-Weighted Active Similarity learning

To improve the learning performance, a second order similarity learning algorithm AROMA is proposed [CC12], which does not only use the first order information but also the second-order information, i.e., the covariance matrix for all the features, to update the model. Compared to the first-order algorithm OASIS, the effectiveness of AROMA has been theoretically and empirically verified. However, directly combining AROMA with query strategy in Equation (6.3) makes the theoretical analysis challenging.

Following the similar idea of AROMA, it is assumed the model maintains a Gaussian distribution, $\mathcal{N}(\mathbf{M}, \mathbf{\Sigma})$, where $\mathbf{M} \in \mathbb{R}^{d \times d}$ and $\mathbf{\Sigma} \in \mathbb{R}^{d^2 \times d^2}$ encode the model's knowledge of and confidence of the model. At the t -th round, when receiving $(\mathbf{x}_t, \mathbf{x}_t^1, \mathbf{x}_t^2)$, the first step is to decide whether it is necessary to query the true label based on Equation (6.3). If $Z_t = 1$, the label of this triplet is queried and the distribution is updated by minimizing the following objective function

$$f_t(\mathbf{M}, \mathbf{\Sigma}) = D_{KL}(\mathcal{N}(\text{vec}(\mathbf{M}), \mathbf{\Sigma}) \| \mathcal{N}(\text{vec}(\mathbf{M}_t), \mathbf{\Sigma}_t)) + \eta \text{Tr}(\mathbf{G}_t^\top \mathbf{M}) + \frac{1}{2\gamma} \text{vec}(\mathbf{X}_t)^\top \mathbf{\Sigma} \text{vec}(\mathbf{X}_t),$$

where

$$D_{KL}(\mathcal{N}(\mu, \Sigma) \| \mathcal{N}(\mu_t, \Sigma_t)) = \frac{1}{2} \left[\ln \left(\frac{|\Sigma_t|}{|\Sigma|} \right) + \text{Tr}(\Sigma_t^{-1} \Sigma) + (\mu_t - \mu)^\top \Sigma_t^{-1} (\mu_t - \mu) - d \right],$$

is the Kullback-Leibler divergence of two distributions, $\mathbf{G}_t = \partial \ell_t(\mathbf{M}_t) = -y_t \mathbf{X}_t$, where $\mathbf{X}_t = \mathbf{x}_t(\mathbf{x}_t^1 - \mathbf{x}_t^2)^\top$ and $\text{vec}(\mathbf{X}) = [\mathbf{X}_{11}, \dots, \mathbf{X}_{1d}, \dots, \mathbf{X}_{d1}, \dots, \mathbf{X}_{dd}]^\top$. The first term is to keep the new distribution not far away from the old one. The second term is a first order approximation of the current loss function at the current model, which is used to minimize the loss of the new model on the current example. The final term is to update the confidence of the model, since a new triplet is observed. η and γ are two positive parameters to trade off these objectives.

The optimization problem can be solved by setting the derivatives $\partial_{\mathbf{M}} f_t(\mathbf{M}_{t+1}, \mathbf{\Sigma})$ and $\partial_{\mathbf{\Sigma}} f_t(\mu, \mathbf{\Sigma}_{t+1})$ as zeros, respectively. The updating rules are as follows:

$$\begin{aligned} \mathbf{M}_{t+1} &= \text{mat} \left[\text{vec}(\mathbf{M}_t) - \eta \mathbf{\Sigma}_t \text{vec}(\mathbf{G}_t) \right], \\ \mathbf{\Sigma}_{t+1} &= \mathbf{\Sigma}_t - \frac{\mathbf{\Sigma}_t \text{vec}(\mathbf{X}_t) \text{vec}(\mathbf{X}_t)^\top \mathbf{\Sigma}_t}{\gamma + \text{vec}(\mathbf{X}_t)^\top \mathbf{\Sigma}_t \text{vec}(\mathbf{X}_t)}, \end{aligned} \quad (6.5)$$

where $\text{mat}(\cdot)$ is the inverse function of $\text{vec}(\cdot)$.

Finally, the proposed CWAS algorithm is summarized as shown in Algorithm 7.

Algorithm 7 CWAS: The proposed algorithm of Confidence-Weighted Active Similarity learning

Input: learning rate η ; regularization parameter γ
Initialize: $\mathbf{\Sigma}_1 = I^{d^2 \times d^2}$, $\mathbf{M}_1 = I^{d \times d}$
for $t = 1, 2, \dots, T$ **do**
 Receive $(\mathbf{x}_t, \mathbf{x}_t^1, \mathbf{x}_t^2)$ and compute $\mathbf{X}_t = \mathbf{x}_t(\mathbf{x}_t^1 - \mathbf{x}_t^2)^\top$;
 $p_t = \text{Tr}(\mathbf{M}_t^\top \mathbf{X}_t)$, and $\hat{y}_t = \text{sgn}(p_t)$;
 Sample $Z_t \in \{0, 1\}$ with $\Pr(Z_t = 1) = \frac{\delta}{\delta + |p_t|}$;
if $Z_t = 1$ **then**
 Query y_t , and compute $\ell_t(\mathbf{M}_t) = [1 - y_t p_t]_+$;
if $\ell_t(\mathbf{M}_t) > 0$ **then**
 Update $\mathbf{M}, \mathbf{\Sigma}$ with Eq. (6.5);
else
 $\mathbf{\Sigma}_{t+1} = \mathbf{\Sigma}_t$, $\mathbf{M}_{t+1} = \mathbf{M}_t$;
end if
else
 $\mathbf{\Sigma}_{t+1} = \mathbf{\Sigma}_t$, $\mathbf{M}_{t+1} = \mathbf{M}_t$;
end if
end for
Output: $\mathbf{M}_{T+1}$

Remark One key drawback of this algorithm is that the dimension of Σ_t is $d^2 \times d^2$, which will result in very high memory and computational complexities. In practice, the diagonal Σ_t can be used to reduce these complexities. Adopting the diagonal version of matrices changes the Equations (6.5) into

$$\begin{aligned} \mathbf{M}_{t+1} &= \mathbf{M}_t - \eta \Sigma_t \odot \mathbf{G}_t, \\ \Sigma_{t+1} &= \Sigma_t - \frac{\Sigma_t \odot \mathbf{X}_t \odot \mathbf{X}_t \odot \Sigma_t}{\gamma + \sum_{ij} (\mathbf{X}_t \odot \Sigma_t \odot \mathbf{X}_t)_{ij}}, \end{aligned} \quad (6.6)$$

where $\odot$ is element-wise product.

6.1.4 Theoretical Analysis

Denote $m_t = \mathbb{I}(y_t \neq \hat{y}_t)$, the performance of the proposed two algorithms can be analyzed to be the expected mistake bounds $\mathbb{E}[\sum_{t=1}^T m_t]$.

Firstly, The mistake bound of the first-order algorithm PAAS is shown in the following theorem.

Theorem 7 *If the PAAS algorithm is run with a sequence of triplets $(\mathbf{x}_t, \mathbf{x}_t^1, \mathbf{x}_t^2; y_t)$, $t \in [T]$, with $D_X = \max \|X_t\|_F$, then for any $T > 0$, and $\mathbf{M} \in \mathbb{R}^{d \times d}$, the mistake bound of PAAS algorithm becomes*

$$\mathbb{E}[\sum_{t=1}^T m_t] \leq \frac{\beta}{\delta} \left\{ \left(\frac{\delta+1}{2} \right)^2 \|\mathbf{M}\|_F^2 + (\delta+1) CL_T(\mathbf{M}) \right\},$$

where $\beta = \max(1/C, D_X^2)$, and $L_T(\mathbf{M}) = \sum_{t=1}^T \ell_t(\mathbf{M})$. In addition, the expected number of labels queried equals to $\sum_{t=1}^T \mathbb{E}[\frac{\delta}{\delta+|p_t|}]$.

Remark: The above bound depends on the choice of parameter δ . Generally, δ could be viewed as a parameter to rule the extent to which the learning model fits the present data [CBGZ06]. Minimizing the right hand side over δ , it can be observed that setting $\delta = \sqrt{1 + 4CL_T(\mathbf{M})/\|\mathbf{M}\|_F^2}$ in the above theorem gives the following upper bound for $\mathbb{E}[\sum_{t=1}^T m_t]$

$$\beta \left\{ \frac{1}{2} \|\mathbf{M}\|_F^2 + CL_T(\mathbf{M}) + \frac{1}{2} \|\mathbf{M}\|_F \sqrt{\|\mathbf{M}\|_F^2 + 4CL_T(\mathbf{M})} \right\}.$$

To prove the Theorem 7, the following lemma is needed.

Lemma 1 Let $(\mathbf{x}_t, \mathbf{x}_t^1, \mathbf{x}_t^2; y_t)$, $t \in [T]$ be a sequence of input triplets, where $\mathbf{x}_t, \mathbf{x}_t^1, \mathbf{x}_t^2 \in \mathbb{R}^d$ and $y_t \in \{-1, +1\}$ for all t . Let τ_t be the step size parameter for PAAS as given in the algorithm. Then the following bound holds for any $\mathbf{M} \in \mathbb{R}^{d \times d}$:

$$\sum_{t=1}^T 2\tau_t Z_t [l_t(\alpha - |p_t|) + m_t(\alpha + |p_t|)] \leq \alpha^2 \|\mathbf{M}\|_F^2 + \sum_{t=1}^T \tau_t^2 \|\mathbf{X}_t\|_F^2 + \sum_{t=1}^T 2\alpha \tau_t \ell_t(\mathbf{M}),$$

where $l_t = \mathbb{I}(\ell_t(\mathbf{M}_t) > 0)$ and $\text{sgn}(p_t) = y_t$, $m_t = \mathbb{I}(\text{sgn}(p_t) \neq y_t)$, $\mathbb{I}$ is the indicator function, $\alpha > 0$ and $\mathbf{X}_t = \mathbf{x}_t(\mathbf{x}_t^1 - \mathbf{x}_t^2)^\top$.

Lemma 1 can be proved using similar techniques in [CBGZ06]. Given Lemma 1, Theorem 1 can be proven as follows: **Proof:** According to Lemma 1, the following inequality can be obtained.

$$\begin{aligned} & \alpha^2 \|\mathbf{M}\|_F^2 + \sum_{t=1}^T 2\alpha \tau_t \ell_t(\mathbf{M}_t) \\ & \geq \sum_{t=1}^T 2\tau_t Z_t [l_t(\alpha - |p_t|) + m_t(\alpha + |p_t|)] - \sum_{t=1}^T \tau_t^2 \|\mathbf{X}_t\|_F^2 \\ & = \sum_{t=1}^T 2\tau_t Z_t [l_t(\alpha - |p_t| - \frac{\tau_t}{2} \|\mathbf{X}_t\|_F^2) + m_t(\alpha + |p_t| - \frac{\tau_t}{2} \|\mathbf{X}_t\|_F^2)] \\ & \geq \sum_{t=1}^T 2\tau_t Z_t [l_t(\alpha - |p_t| - \frac{\ell_t(\mathbf{M}_t)}{2}) + m_t(\alpha + |p_t| - \frac{\ell_t(\mathbf{M}_t)}{2})] \\ & = \sum_{t=1}^T l_t Z_t 2\tau_t (\alpha - \frac{1 + |p_t|}{2}) + \sum_{t=1}^T m_t Z_t 2\tau_t (\alpha - \frac{1 - |p_t|}{2}). \end{aligned}$$

Plugging $\alpha = \frac{\delta+1}{2}$, $\delta \geq 1$ into the above inequality will result in

$$(\frac{\delta+1}{2})^2 \|\mathbf{M}\|_F^2 + \sum_{t=1}^T (\delta+1) \tau_t \ell_t(M) \geq \sum_{t=1}^T m_t Z_t \tau_t (\delta + |p_t|).$$

Since $\tau_t \geq \min(C, 1/D_X^2)$, the above inequality implies:

$$(\frac{\delta+1}{2})^2 \|\mathbf{M}\|_F^2 + \sum_{t=1}^T (\delta+1) \tau_t \ell_t(M) \geq \min(C, 1/D_X^2) \sum_{t=1}^T m_t Z_t (\delta + |p_t|).$$

Taking expectation with the above inequality, plugging the equality $\mathbb{E}Z_t = \delta/(\delta + |p_t|)$ and re-arranging the result conclude this theorem.

Under the same assumptions as in the above theorem, the following theoretical guarantee for the second-order algorithm CWAS can be obtained.

Theorem 8 *If the algorithm CWAS is run with a sequence of examples $(\mathbf{x}_t, \mathbf{x}_t^1, \mathbf{x}_t^2; y_t)$, $t \in [T]$, then for any $T > 0$, and $\mathbf{M} \in \mathbb{R}^{d \times d}$,*

$$\mathbb{E} \sum_{t=1}^T [m_t] \leq \sum_{t=1}^T Z_t \ell_t(\mathbf{M}) + \frac{1}{\eta \delta} (D_M + |1 - \delta| \|\mathbf{M}\|)^2 \text{Tr}(\Sigma_{T+1}^{-1}) + \frac{\eta \gamma}{2\delta} \ln(|\Sigma_{T+1}^{-1}|).$$

Setting $\eta = (D_M + |1 - \delta| \|\mathbf{M}\|) \sqrt{\frac{2}{\gamma} \text{Tr}(\Sigma_{T+1}^{-1}) / \ln(|\Sigma_{T+1}^{-1}|)}$, the following bound holds for CWAS

$$\mathbb{E} \left[\sum_{t=1}^T m_t \right] \leq L_T(\mathbf{M}) + \frac{|1 - \delta| \|\mathbf{M}\|_F^2 + D_M}{\delta} \sqrt{2\gamma \text{Tr}(\Sigma_{T+1}^{-1}) \ln|\Sigma_{T+1}^{-1}|},$$

where $D_M = \max_t \|\mathbf{M}_t - \mathbf{M}\|_F^2$.

Remark By optimizing over δ , and setting $\delta = 1$, the mistake bound becomes

$$\mathbb{E} \left[\sum_{t=1}^T m_t \right] \leq L_T(\mathbf{M}) + D_M \sqrt{2\gamma \text{Tr}(\Sigma_{T+1}^{-1}) \ln|\Sigma_{T+1}^{-1}|}.$$

In this subsection, $\mathbf{M}_t$, G_t , $\mathbf{X}_t$ denote $\text{vec}(M_t)$, $\text{vec}(\mathbf{G}_t)$, and $\text{vec}(X_t)$, respectively. To prove Theorem 8, the following lemma are needed.

Lemma 2 *Let $(\mathbf{x}_t, \mathbf{x}_t^1, \mathbf{x}_t^2; y_t)$, $t \in [T]$ be a sequence of input triplets, where $\mathbf{x}_t, \mathbf{x}_t^1, \mathbf{x}_t^2 \in \mathbb{R}^d$ and $y_t \in \{-1, +1\}$ for all t . If the CWAS is run on this sequence of triplets, then the following bound holds for any $\mathbf{M} \in \mathbb{R}^{d \times d}$,*

$$\begin{aligned} & Z_t [m_t(\delta + |p_t|) + l_t(\delta - |p_t|)] \\ & \leq \frac{Z_t}{2\eta} \left[\|\mathbf{M}_t - \delta \mathbf{M}\|_{\Sigma_{t+1}^{-1}}^2 - \|\mathbf{M}_{t+1} - \delta \mathbf{M}\|_{\Sigma_{t+1}^{-1}}^2 + \|\mathbf{M}_t - \mathbf{M}_{t+1}\|_{\Sigma_{t+1}^{-1}}^2 \right] + \delta Z_t \ell_t(\mathbf{M}_t), \end{aligned}$$

where $l_t = \mathbb{I}(\ell_t(\mathbf{M}_t) > 0)$ and $\text{sgn}(p_t) = y_t$, $m_t = \mathbb{I}(\text{sgn}(p_t) \neq y_t)$, $\mathbb{I}$ is the indicator function, $\delta > 0$ and $\|\mathbf{M}_t - \delta \mathbf{M}\|_{\Sigma_{t+1}^{-1}}^2$ actually denotes $\|\text{vec}(\mathbf{M}_t) - \text{vec}(\delta \mathbf{M})\|_{\Sigma_{t+1}^{-1}}^2$.

Proof: When $Z_t = 0$, it is easy to verify the inequality in the theorem.

When $Z_t = 1$, it is easy to observe that

$$\mathbf{M}_{t+1} = \arg \min_{\mathbf{M}} f_t(\mathbf{M}),$$

where

$$f_t(\mathbf{M}) = \frac{1}{2} \|\mathbf{M}_t - \mathbf{M}\|_{\Sigma_{t+1}^{-1}}^2 + \eta \mathbf{G}_t^\top \mathbf{M}.$$

Because f_t is convex, the following inequality exists $\forall \mathbf{M}$,

$$\begin{aligned} 0 &\leq \partial f_t(\mathbf{M}_{t+1})^\top (\mathbf{M} - \mathbf{M}_{t+1}) \\ &= [\boldsymbol{\Sigma}_{t+1}^{-1}(\mathbf{M}_{t+1} - \mathbf{M}_t) + \eta \mathbf{G}_t]^\top (\mathbf{M} - \mathbf{M}_{t+1}). \end{aligned}$$

Re-arranging the above inequality will result in

$$\begin{aligned} &\eta \mathbf{G}_t^\top (\mathbf{M}_{t+1} - \mathbf{M}) \\ &\leq (\mathbf{M}_{t+1} - \mathbf{M}_t)^\top \boldsymbol{\Sigma}_{t+1}^{-1} (\mathbf{M} - \mathbf{M}_{t+1}) \\ &= \frac{1}{2} [\|\mathbf{M}_t - \mathbf{M}\|_{\boldsymbol{\Sigma}_{t+1}^{-1}}^2 - \|\mathbf{M}_{t+1} - \mathbf{M}\|_{\boldsymbol{\Sigma}_{t+1}^{-1}}^2 - \|\mathbf{M}_t - \mathbf{M}_{t+1}\|_{\boldsymbol{\Sigma}_{t+1}^{-1}}^2]. \end{aligned}$$

Now, a lower bound becomes for $\mathbf{G}_t^\top (\mathbf{M}_{t+1} - \mathbf{M})$,

$$\begin{aligned} &\mathbf{G}_t^\top (\mathbf{M}_{t+1} - \mathbf{M}) - \mathbf{G}_t^\top (\mathbf{M}_t - \mathbf{M}) + \mathbf{G}_t^\top (\mathbf{M}_{t+1} - \mathbf{M}_t) \\ &= (l_t + m_t)(-y_t X_t^\top \mathbf{M}_t) + (l_t + m_t)y_t X_t^\top \mathbf{M} - \frac{1}{\eta} \|\mathbf{M}_t - \mathbf{M}_{t+1}\|_{\boldsymbol{\Sigma}_{t+1}^{-1}}^2, \end{aligned}$$

where the second inequality used the facts $\mathbf{G}_t = (l_t + m_t)(-y_t \mathbf{X}_t)$ and $\partial f_t(\mathbf{M}_{t+1}) = 0$, i.e.,

$$\boldsymbol{\Sigma}_{t+1}^{-1}(\boldsymbol{\mu}_{t+1} - \boldsymbol{\mu}_t) + \eta \mathbf{G}_t = 0.$$

Combining the above equality with the facts

$$m_t(-y_t \mathbf{X}_t^\top \mathbf{M}_t) = M_t |p_t|, l_t(-y_t \mathbf{X}_t^\top \mathbf{M}_t) = -l_t |p_t|$$

and

$$y_t \mathbf{X}_t^\top \mathbf{M}_t + \delta \ell_t(\mathbf{M}/\delta) \geq y_t p_t + \delta(1 - y_t p_t/\delta) = \delta,$$

The bound for $\mathbf{G}_t^\top (\mathbf{M}_{t+1} - \mathbf{M})$ becomes,

$$\begin{aligned} &\mathbf{G}_t^\top (\mathbf{M}_{t+1} - \mathbf{M}) \\ &\geq (m_t |p_t| - l_t |p_t|) + (l_t + m_t)[\delta - \delta \ell_t(\mathbf{M}/\delta)] - \frac{1}{\eta} \|\mathbf{M}_t - \mathbf{M}_{t+1}\|_{\boldsymbol{\Sigma}_{t+1}^{-1}}^2 \\ &= [m_t(\delta + |p_t|) + l_t(\delta - |p_t|)] - (l_t + m_t)\delta \ell_t(\mathbf{M}/\delta) - \frac{1}{\eta} \|\mathbf{M}_t - \mathbf{M}_{t+1}\|_{\boldsymbol{\Sigma}_{t+1}^{-1}}^2. \end{aligned}$$

Combining the previous inequalities, will give the following important inequality

$$\begin{aligned}
 & [m_t(\delta + |p_t|) + l_t(\delta - |p_t|)] \\
 & \leq \frac{1}{2\eta} [\|\mathbf{M}_t - \delta\mathbf{M}\|_{\Sigma_{t+1}^{-1}}^2 - \|\mathbf{M}_{t+1} - \delta\mathbf{M}\|_{\Sigma_{t+1}^{-1}}^2 \\
 & \quad - \|\mathbf{M}_t - \mathbf{M}_{t+1}\|_{\Sigma_{t+1}^{-1}}^2] + \frac{1}{\eta} \|\mathbf{M}_t - \mathbf{M}_{t+1}\|_{\Sigma_{t+1}^{-1}}^2 + \delta\ell_t(\mathbf{M}/\delta) \\
 & = \frac{1}{2\eta} [\|\mathbf{M}_t - \delta\mathbf{M}\|_{\Sigma_{t+1}^{-1}}^2 - \|\mathbf{M}_{t+1} - \delta\mathbf{M}\|_{\Sigma_{t+1}^{-1}}^2 \\
 & \quad + \|\mathbf{M}_t - \mathbf{M}_{t+1}\|_{\Sigma_{t+1}^{-1}}^2] + \delta\ell_t(\mathbf{M}/\delta).
 \end{aligned}$$

Replacing $\mathbf{M}$ with $\delta\mathbf{M}$ concludes the proof.

Given Lemma 2, the theorem 2 can be proven as follows:

Proof: Firstly, according to the update rule

$$\mathbf{M}_{t+1} = \text{mat}[\text{vec}(\mathbf{M}_t) - \eta\Sigma_t\text{vec}(G_t)],$$

The following equality can be derived.

$$\|\mathbf{M}_t - \mathbf{M}_{t+1}\|_{\Sigma_{t+1}^{-1}}^2 = \eta^2 \mathbf{G}_t^\top \Sigma_{t+1} \mathbf{G}_t = \eta^2 (m_t + l_t) \mathbf{X}_t^\top \Sigma_{t+1} \mathbf{X}_t = \eta^2 \gamma (1 - \frac{|\Sigma_t^{-1}|}{|\Sigma_{t+1}^{-1}|}),$$

where it is a fact that $A = B + \mathbf{x}\mathbf{x}^\top$ implies $\mathbf{x}^\top A^{-1} \mathbf{x} = 1 - \frac{|B|}{|A|}$. Plugging the above equality into the inequality in the Lemma 2, and re-arranging it will gives

$$\begin{aligned}
 & Z_t[m_t(\delta + |p_t|) + l_t(\delta - |p_t|)] \\
 & \leq \frac{Z_t}{2\eta} [\|\mathbf{M}_t - \delta\mathbf{M}\|_{\Sigma_{t+1}^{-1}}^2 - \|\mathbf{M}_{t+1} - \delta\mathbf{M}\|_{\Sigma_{t+1}^{-1}}^2] + \frac{Z_t\eta\gamma}{2} (1 - \frac{|\Sigma_t^{-1}|}{|\Sigma_{t+1}^{-1}|}) + \delta Z_t \ell_t(\mathbf{M}_t).
 \end{aligned}$$

Summing the above inequality over $t = 1, 2, \dots, T$ can give

$$\begin{aligned}
 & \sum_{t=1}^T Z_t[m_t(\delta + |p_t|) + l_t(\delta - |p_t|)] \\
 & \leq \sum_{t=1}^T \delta Z_t \ell_t(\mathbf{M}_t) + \sum_{t=1}^T \frac{Z_t}{2\eta} [\|\mathbf{M}_t - \delta\mathbf{M}\|_{\Sigma_{t+1}^{-1}}^2 - \|\mathbf{M}_{t+1} - \delta\mathbf{M}\|_{\Sigma_{t+1}^{-1}}^2] \\
 & \quad + \sum_{t=1}^T \frac{Z_t\eta\gamma}{2} (1 - \frac{|\Sigma_t^{-1}|}{|\Sigma_{t+1}^{-1}|}).
 \end{aligned}$$

Now, the right hand side of the above inequality can be bounded. First, the bound of the first term becomes

$$\begin{aligned}
 & \sum_{t=1}^T Z_t \left[\|\mathbf{M}_t - \delta \mathbf{M}\|_{\Sigma_{t+1}^{-1}}^2 - \|\mathbf{M}_{t+1} - \delta \mathbf{M}\|_{\Sigma_{t+1}^{-1}}^2 \right] \\
 & \leq \|\mathbf{M}_1 - \delta \mathbf{M}\|_{\Sigma_2^{-1}}^2 + \sum_{t=2}^T [\|\mathbf{M}_t - \delta \mathbf{M}\|_{\Sigma_{t+1}^{-1}}^2 - \|\mathbf{M}_t - \delta \mathbf{M}\|_{\Sigma_t^{-1}}^2] \\
 & \leq \|\mathbf{M}_1 - \delta \mathbf{M}\|^2 \text{Tr}(\Sigma_2^{-1}) + \sum_{t=2}^T \|\mathbf{M}_t - \delta \mathbf{M}\|^2 \text{Tr}(\Sigma_{t+1}^{-1} - \Sigma_t^{-1}) \\
 & = \max_{t \leq T} \|\mathbf{M}_t - \delta \mathbf{M}\|^2 \text{Tr}(\Sigma_{T+1}^{-1}) \\
 & \leq 2(D_M + |1 - \delta| \|\mathbf{M}\|)^2 \text{Tr}(\Sigma_{T+1}^{-1}),
 \end{aligned}$$

where $D_M = \max_{t \leq T} \|\mathbf{M}_t - \mathbf{M}\|^2$. Combining the above two inequalities, result in

$$\begin{aligned}
 & \sum_{t=1}^T Z_t [m_t(\delta + |p_t|)] \\
 & \leq \delta \sum_{t=1}^T Z_t \ell_t(\mathbf{M}) + \frac{1}{\eta} (D_M + (1 - \delta)^2 \|\mathbf{M}\|^2) \text{Tr}(\Sigma_{T+1}^{-1}) + \sum_{t=1}^T \frac{Z_t \eta \gamma}{2} \left(1 - \frac{|\Sigma_t^{-1}|}{|\Sigma_{t+1}^{-1}|}\right) \\
 & \leq \delta \sum_{t=1}^T Z_t \ell_t(\mathbf{M}) + \frac{1}{\eta} (D_M + |1 - \delta| \|\mathbf{M}\|)^2 \text{Tr}(\Sigma_{T+1}^{-1}) + \frac{\eta \gamma}{2} \ln(|\Sigma_{T+1}^{-1}|),
 \end{aligned}$$

where the fact $(1 - \frac{|A|}{|B|}) \leq \ln \frac{|B|}{|A|}$ is used, since $1 - x \leq -\ln x$ for all $x > 0$.

Taking expectation with the above inequality and using the fact $\mathbb{E}[Z_t] = \frac{\delta}{\delta + |p_t|}$ gives

$$\delta \mathbb{E} \sum_{t=1}^T [m_t] \leq \delta \sum_{t=1}^T Z_t \ell_t(\mathbf{M}) + \frac{1}{\eta} (D_M + |1 - \delta| \|\mathbf{M}\|)^2 \text{Tr}(\Sigma_{T+1}^{-1}) + \frac{\eta \gamma}{2} \ln(|\Sigma_{T+1}^{-1}|).$$

Dividing the above inequality with δ concludes the proof.

6.2 Experiments

In this section, several experiments are conducted to evaluate the efficacy of the proposed algorithms for online similarity learning from data streams on several benchmark datasets.

6.2.1 Datasets

To examine the performance, the experiments are conducted on a variety of benchmark datasets from web machine learning repositories. Table 6.1 shows the details of five datasets used in the following experiments. Caltech256 [GHP07] is a standard dataset for the image classification and ranking problem ¹. the same classes as [CSSB10b] are used to generate the *caltech50* and *caltech249*. The other datasets are standard machine learning datasets publicly available at LIBSVM ² and UCI Machine Learning Repository ³.

Table 6.1: Datasets used in the following experiments

Dataset	# Classes	# Instances	# Features	# Triplets
caltech249	249	16185	1000	1200000
caltech50	50	3250	1000	250000
covtype	7	455	54	10000
letter	26	1690	16	10140
pendigits	10	650	16	16848
satimage	6	390	36	18000
segment	7	455	19	25000
shuttle	7	458	9	4000

The performance of all algorithms is evaluated in terms of precision-at-top-k, a standard ranking precision measure based on nearest neighbors. For each query instance in the test set, all other test instances were ranked according to their similarity to the query instance based on Equation (6.1), and the number of same-class instances among the top k instances (the k nearest neighbors) is computed, and then averaged across test instances. This measure is short named as AP. Mean Average Precision (mAP), a measure that is widely used in the information retrieval community, is also considered in the following experiments. For both of the AP and mAP measures, k is set to 10.

On all datasets, the standard 5-fold cross validation is used and both the AP and the mAP on test set are reported. In Table 6.1, $\# Triplets$ represents the number of triplets $(\mathbf{x}_t, \mathbf{x}_t^1, \mathbf{x}_t^2)$ constructed based on the training set in each fold.

¹http://www.vision.caltech.edu/Image_Datasets/Caltech256/

²<http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/>

³<https://archive.ics.uci.edu/ml/datasets.html>

6.2.2 Comparison Schemes

To examine the efficacy, the proposed algorithms are compared with the state-of-the-art relative similarity learning algorithms, where they have to query every triplet's label during training stage.

- **OASIS**: the state-of-the-art first-order online similarity learning algorithm [CSSB10b]. It should be noted that this algorithm is also the passive version of the proposed PAAS algorithm;
- **AROMA**: the state-of-the-art second-order online similarity learning algorithm [CC12], and **AROMA-d** represents its diagonal version.

To our knowledge, there is no work on exploring active learning on the relative similarity learning, so the proposed algorithms are compared with their random versions, where they randomly decide whether to query the label of t -th triplet.

- **PAAS**: the proposed first-order Passive-Aggressive Active online Similarity Learning as shown in Algorithm 6;
- **PARS**: a variant of the proposed algorithm PAAS, where a uniform distribution is used to randomly query labels and keep the query ratio same as PAAS;
- **CWS**: the passive version of the algorithm CWAS, where each triplet in the training phrase is queried.
- **CWAS**: the proposed Confidence-Weighted Active online Similarity Learning as shown in Algorithm 7. **CWAS-d** is the diagonal version with Equation (6.6) as updating rules.
- **CWRS**: a variant of the algorithm CWAS, where a uniform distribution is used to randomly query labels and keep the query ratio same as CWAS. **CWRS-d** is the diagonal version with Equation (6.6) as updating rules;

For all algorithms, the parameters are searched within the space $10^{[-5:1:5]}$ using cross validation. For both PAAS and CWAS algorithms, their performances is evaluated on 10

Table 6.2: Performance of algorithms on different datasets with the query ratio fixed to about 20%

Dataset	Algorithm	Query(%)	AP@10	mAP@10	Time(s)
covtype	PARS	20.620 $\pm$ 0.955	0.446 $\pm$ 0.036	0.334 $\pm$ 0.035	0.006 $\pm$ 0.008
	PAAS	20.710 $\pm$ 0.336	0.512 $\pm$ 0.035	0.409 $\pm$ 0.027	0.006 $\pm$ 0.008
	CWRS	20.960 $\pm$ 0.670	0.585 $\pm$ 0.027	0.476 $\pm$ 0.040	11.860 $\pm$ 1.741
	CWAS	20.890 $\pm$ 0.677	0.624 $\pm$ 0.038	0.515 $\pm$ 0.043	22.390 $\pm$ 1.448
letter	PARS	20.6 $\pm$ 0.827	0.198 $\pm$ 0.024	0.127 $\pm$ 0.025	0.007 $\pm$ 0.000
	PAAS	20.6 $\pm$ 0.667	0.233 $\pm$ 0.020	0.158 $\pm$ 0.023	0.007 $\pm$ 0.000
	CWRS	19.4 $\pm$ 1.078	0.362 $\pm$ 0.019	0.276 $\pm$ 0.021	0.104 $\pm$ 0.005
	CWAS	19.4 $\pm$ 0.507	0.385 $\pm$ 0.012	0.298 $\pm$ 0.012	0.134 $\pm$ 0.005
pendigits	PARS	22.2 $\pm$ 0.378	0.527 $\pm$ 0.056	0.418 $\pm$ 0.063	0.010 $\pm$ 0.001
	PAAS	22.2 $\pm$ 0.526	0.554 $\pm$ 0.027	0.452 $\pm$ 0.024	0.010 $\pm$ 0.001
	CWRS	19.3 $\pm$ 1.068	0.673 $\pm$ 0.020	0.588 $\pm$ 0.032	0.181 $\pm$ 0.016
	CWAS	19.3 $\pm$ 0.925	0.700 $\pm$ 0.010	0.619 $\pm$ 0.018	0.287 $\pm$ 0.021
satimage	PARS	20.4 $\pm$ 0.387	0.493 $\pm$ 0.044	0.408 $\pm$ 0.040	0.046 $\pm$ 0.001
	PAAS	20.5 $\pm$ 0.274	0.509 $\pm$ 0.031	0.423 $\pm$ 0.022	0.050 $\pm$ 0.001
	CWRS	19.6 $\pm$ 0.860	0.643 $\pm$ 0.010	0.560 $\pm$ 0.013	4.476 $\pm$ 0.257
	CWAS	19.5 $\pm$ 0.517	0.657 $\pm$ 0.008	0.574 $\pm$ 0.011	9.390 $\pm$ 0.446
segment	PARS	21.2 $\pm$ 0.366	0.361 $\pm$ 0.034	0.253 $\pm$ 0.041	0.020 $\pm$ 0.001
	PAAS	21.2 $\pm$ 0.607	0.436 $\pm$ 0.029	0.316 $\pm$ 0.031	0.021 $\pm$ 0.000
	CWRS	19.2 $\pm$ 0.817	0.690 $\pm$ 0.074	0.575 $\pm$ 0.103	0.297 $\pm$ 0.024
	CWAS	19.2 $\pm$ 0.781	0.787 $\pm$ 0.019	0.729 $\pm$ 0.039	0.573 $\pm$ 0.036
shuttle	PARS	21.07 $\pm$ 0.898	0.435 $\pm$ 0.058	0.313 $\pm$ 0.054	0.001 $\pm$ 0.000
	PAAS	20.64 $\pm$ 0.807	0.513 $\pm$ 0.027	0.403 $\pm$ 0.042	0.001 $\pm$ 0.000
	CWRS	20.25 $\pm$ 1.117	0.502 $\pm$ 0.037	0.417 $\pm$ 0.059	0.006 $\pm$ 0.001
	CWAS	20.84 $\pm$ 0.873	0.561 $\pm$ 0.024	0.493 $\pm$ 0.017	0.009 $\pm$ 0.001

different query ratios which are achieved by setting the sampling threshold $\delta = 2^{[-10:2:10]}$. The query ratio represents the ratio of queried labels over the total number of triplets in the data stream. PARS and CWRS are also evaluated on 10 query ratios by setting the random sampling parameters according to the query ratios in PAAS and CWAS, respectively.

6.2.3 Evaluation of Fixed Query Ratio

In this experiment, the proposed algorithms are evaluated on the setting where the query ratio is fixed. Table 6.2 shows the experimental results on different datasets.

First, with the same query ratio, the proposed active similarity learning algorithms (CWAS

and PAAS) consistently outperform their random versions (CWRS and PARS), respectively. Besides, CWAS greatly outperforms PAAS over all the datasets. Moreover, both CWAS and PAAS also achieve smaller variances as compared to their random algorithms (PARS and CWRS), respectively. These observations further confirm the effectiveness and robustness of the proposed algorithms.

Second, on all the datasets, the proposed second-order active online similarity learning algorithm CWAS always achieves the best performance and the smallest variance. This is consistent with the previous results as shown in Figure 6.1, 6.2 and 6.4, which further confirms the effectiveness of exploiting the second-order information.

Third, by examining the running time cost, PAAS and CWAS spent slightly more time cost as compared to their random variants, respectively, mainly due to the cost of computing the query strategies.

In addition, CWAS in general spends more time than PAAS due to the computation of the second order information. However, the extra time cost could almost be ignored considering the high efficiency of the online learning scheme. In practice, for the higher dimensional datasets, the diagonal version algorithm “CWAS-d” is adopted to further reduce the running time cost, as shown in subsequent experiments.

6.2.4 Evaluation of Varied Query Ratios

In this section, the proposed algorithms are evaluated in the setting where query ratio ranges from 0 to 1. Figure 6.1 and 6.2 show the results in terms of AP and mAP, respectively. Based on the results, several observations could be drawn from the results.

First, the proposed active online learning algorithms can consistently outperform their random versions, respectively. This observation is consistent with the one in Table 6.2 and confirms the effectiveness of the proposed algorithms on selecting informative instances. More importantly, with around 30% query ratio, the proposed algorithm CWAS could achieve the similar performance as CWS which queries all. This means that the proposed second-order active algorithm CWAS could save us around 70% effort compared to the traditional passive algorithms. For the proposed first-order active algorithm, PAAS could save us around 50% effort compared to the passive algorithm OASIS.

Figure 6.1: AP V.S. varied query ratios. AP@10 denotes Average Precision at 10.

Figure 6.2: mAP V.S. varied query ratios. AP@10 denotes Average Precision at 10.

Second, the proposed second-order algorithm CWAS can outperform the first-order algorithm PAAS. From Figure 6.1, 6.2, it can be observed that CWAS could outperform PAAS on most of the datasets from the very small query ratio. The same result also could be observed on the CWRS compared to the PARS. These findings confirmed the effectiveness of introducing the second-order information. From the figure, it also can be observed that the baseline second-order algorithm AROMA could achieve a little higher performance than the proposed algorithm CWS, on some datasets. One possible reason may be that the learning rate η is fixed in the Equation (6.5), while it is adaptive in the algorithm AROMA [CKD09]. However, CWS in general is comparable with AROMA, while its active version is much easier to be theoretically analyzed.

6.2.5 Evaluation of Parameter Sensitivity

In this section, the sensitivity of the parameters both in Algorithm 6 and 7 are evaluated, respectively. However, it is difficult to evaluate the algorithms with active query strategies. Thus, their passive versions OASIS and CWS are evaluated, respectively. Figure 6.3 shows the varied performances corresponding to different parameters on four of the datasets. For each dataset, the left figure shows the performance (Y-axis) corresponding to the parameter C (X-axis) in OASIS, and the right figure shows the performance (different color) corresponding to parameter γ (X-axis) and η (Y-axis) in CWS, where bright color corresponds to higher performance in terms of Average Precision at 10 than the dark color.

First, it can be observed that C greatly affects the results of OASIS, on all of the datasets, which could be explained using the mistake bound in Theorem 7. Specifically, the mistake bound could be divided into two terms, and the parameter C divides the first term and multiplies the second term. When C is too small, the first term will dominate the mistake bound, and the second term will dominate the mistake bound when C is too large. This is consistent with the results shown in Figure 6.3. For most of the datasets, $C = 1$ would present a promising result.

Second, for algorithm CWS, the relationship between the performance and the parameters is relatively complex. From Figure 6.3, it can be observed that the learning rate parameter η should be not too small in general. And the regularization parameter γ should be search around 1.

Figure 6.3: Sensitivity of the parameters

6.2.6 Evaluation of Efficiency and Scalability

Figure 6.4: Performance on the large-scale datasets

Although the second-order algorithm CWAS greatly outperforms the first-order algorithm PAAS, it is costly to maintain the covariance matrix when the number of features is large. To reduce the complexity, their diagonal versions can be adopted with Equations (6.6). To test the diagonal algorithm CWAS-d, two experiments on large-scale datasets *caltech50* and *caltech249* are conducted, and the results are shown in Figure 6.4, in which similar results can be observed as the one in Figure 6.1, 6.2. In addition, the increase of performance of CWAS-d over PAAS is not as large as CWAS in Figure 6.1, 6.2 as expected, since only part of the second-order information is used.

Figure 6.5: Time cost corresponding to varied querying ratio on large-scale datasets

Figure 6.5 shows the relationship between the querying ratio and the time-cost of the proposed algorithms. The figures show that the time cost of the proposed algorithms is linearly increasing with respect to the increasing querying ratio. Besides, CWAS-d cost a little extra time than its random version CWRS-d due to the computation of the query strategy, and PAAS costs almost the same as its random version PARS. What's more, the second-order algorithm CWAS-d cost more than the first-order algorithm PAAS due to the computation of the second-order information, which is consistent with the finding in Table 6.2. More importantly, in Figure 6.4, it can be seen that with around 30% query ratio, the proposed algorithm PAAS and CWAS-d could obtain similar performances as their passive versions OASIS and CWS-d, respectively. Meanwhile, when the query ratio is around 30%, in Figure 6.5, PAAS and CWAS-d only spends around 30% of the time spent by OASIS and CWS-d, respectively. These observations illustrate that the proposed algorithms can save a lot effort in labeling and training the model without sacrificing performance compared to the passive algorithms which query the labels of all the triplets.

6.3 Summary

To overcome the critical limitation of traditional passive online similarity learning from data streams, in this chapter, a novel framework of active online learning for relative similarity learning is proposed. Specifically, two active online similarity learning algorithms are proposed for reducing the number of queried labels in the learning process. The proposed algorithms are theoretically analyzed and an extensive set of experiments are conducted to examine the effectiveness of their empirical performance. The encouraging empirical results validate the effectiveness and efficiency of the proposed algorithms.

However, there are several aspects that are valuable to explore in future. Firstly, it would be more interesting to design an automatic method to assign the parameter δ to control the query ratio, currently, it is manually assigned. Secondly, for the proposed second-order based algorithm CWAS, it would be more effective to consider the second-order information when designing the query strategy, such as the covariance information contained in Σ . Thirdly, it is also interest to design a real online active relative similarity learning where the labels of instance come from several noisy online workers, such as from the crowdsourcing platforms.

Chapter 7

Conclusions and Future Researches

7.1 Conclusions

Online active learning has been an increasingly important research area in machine learning. The scalability and labeling efficiency have made it more suitable for large-scale and labeling-costly applications, such as web-scale classification, regression, ranking and similarity learning. However, there still are several challenges. The first challenge is to improve the robustness of active learning algorithm in the learning with expert advice setting. The second one is to solve the aggregation and allocation problem in crowdsourcing with the proposed active learning with expert advice algorithms. The third one is to improve the efficiency of query strategy in the tradition online learning setting. The last one is to reduce the labeling cost in the metric learning applications.

For the first challenge, based on the learning with expert advice framework, two robust OAL algorithms are proposed based on two forecasters, respectively. The proposed query strategies not only consider the difference of predictions made by experts, but also consider the cumulative performance of experts. By considering these two types of information, the query strategies can robustly query the instances on which reliable experts do not agree thus achieve better performance than the exiting algorithms. To validate the proposed OAL algorithms, an extensive set of experiments are conducted both on the normal setting (where all experts achieve comparable performance) and noisy setting (where noisy experts exist), and achieve promising results which validate the robustness of our proposed query strategy. This work is submitted to ACM Transactions on Intelligent Systems and Technology (ACM TIST).

For the second challenge, a framework of active crowdsourcing for annotation is presented based on the learning with expert advice. Specifically, First, a module of “Who to label” is presented, which is an adaptive allocation strategy based the reliability of workers. Second, in order to reduce the golden questions, a module of “When to query” is presented, which includes two effective query strategies, one margin-based query strategy and one which considers both the cumulative performance of workers and their annotations on current instance. By combining the “Who to label” and “When to query”, the proposed framework can systemically gain reliable labels with a limited budget and a small set of golden questions. A set of experiments are conducted both in simulated and real datasets, and the promising results validate the effectiveness of the proposed framework. This work is accepted by IEEE/WIC/ACM Web Intelligence 2015, IEEE/WIC/ACM Intelligent Agent Technology 2015 (WI’15/IAT’15).

For the third challenge, a second-order based active learning algorithm is proposed, which not only considers the margin value of the instance, but also considers the confidence of algorithm. The proposed query strategy will always query the instances on which the model is not well trained, and draw a Bernoulli random variable with the adjusted margin value if the model is well trained. To validate the effectiveness of the proposed algorithm, a theoretical analysis is presented, and then a serial of empirical experiments are conducted on several large-scale datasets. The promising results show that the proposed algorithm is more effective in reducing labeling cost and achieving better accuracy meanwhile. This work is accepted by The IEEE International Conference on Data Mining (ICDM 2016).

For the last challenge, two OAL algorithms are presented for relative similarity learning problems. Specifically, an online active passive-aggressive learning algorithm is firstly presented, and then a second-order based OAL algorithm is presented in order to improve the performance. To further reduce the time cost of our second-order based OAL algorithm, the diagonal version of the algorithm is investigated. the mistake bounds are provided for both of the algorithms, and conducted several empirical studies. The promising results validate the effectiveness of the proposed OAL algorithms in relative similarity learning problems. This work is accepted by The 24th ACM International Conference on Information and Knowledge Management (CIKM 2015).

Although the proposed algorithms show promising performance on real dataset, there are several limitations. First, for the robust active learning with expert advice algorithms,

we don't how the algorithms perform in theory. And it's also interesting to investigate relationship between the performance of the algorithms and the reliability of each worker. Second, for the ACA framework in chapter 4, we only investigate the group-cheating noisy scenarios, and it is more meaningful to evaluate the algorithm in random noisy scenarios. Third, for the proposed SOAL algorithm in chapter 5, we importing two hyper-parameters, and the hyper-parameters are manually fixed via a grid search algorithm on the validation dataset, this would be a major issue when the number of hyper-parameters increases. Lastly, the proposed online similarity active learning algorithms only consider the margin-based query strategy, it would be more interesting to consider the other state-of-the-art query strategy, such as the second-order based query strategy proposed in chapter 6.

7.2 Contributions

In sum, the contributions of this thesis mainly falls into two categories: theory and application.

For theory, this thesis has proposed several novel theoretically sound OAL algorithms. First, to tackle the limitations of existing OAL with expert advice on noisy setting, this study proposed two robust OAL with expert advices algorithms, which make sure the predictions made by the proposed OAL algorithms are not far away from the prediction made by the passive online learning with expert advice. Second, to overcome the exiting OAL in classification problems, a second-order OAL algorithm is proposed in this study. It is theoretically proved that the mistake made by the proposed algorithm is not far away from a best model in hindsight. Third, for the online similarity learning problems, the proposed two OAL algorithms for relative similarity learning are both guaranteed that the mistakes made on the triplets are not far away from the the ones made by the best predictors in hindsight.

For application, for all of the proposed algorithms, a large set of empirical studies are conducted with the datasets from several domains, such as image classification, text classification, malicious URL detection, noisy workers detection in crowdsourcing. Particularly, a online active crowdsourcing for annotation framework is proposed for collecting reliable labels with limited budget. This proposed framework can learn the

reliability of each work with limited number of control questions, and the learned reliability can be used for task allocation in crowdsourcing. What's more, the application of OAL on image retrieval is also studied in this thesis. Specifically, two novel OAL algorithms are proposed for relative similarity learning. And the proposed algorithms can greatly reduce the labeling cost when training an image retrieval system.

In sum, this thesis proposed a series of robust OAL learning algorithms, which can robustly find the most informative instances to query the true label thus greatly reduce the labeling cost. The purpose of the proposed algorithms are mostly general to several classification tasks. For example, for the proposed robust active learning with expert advices algorithms, the tasks could be image classification, text classification, malicious URL detection, sentiment analysis and so on. It's reasonable to believe that the labeling cost would be major issue in the era of big data, and this thesis has explored several new approaches to tackle this issue.

7.3 Future Directions

Although several challenges are tackled for OAL problems, there are still several potential problems for future research

7.3.1 Online Active Learning for Concept Drifting Problems

Concept drift is very common in real world applications [KJ00, Kli04, Fan04]. Take users' interest in social media for example, a user's interest is usually changing along with seasons, career development and so on. For these kinds of datasets, training a fixed model is usually unreasonable.

Although most of the OAL algorithms do not make any assumption on the distribution of the data stream, their theoretical analysis is usually compared with the offline algorithms, for which the data distribution is assumed to be fixed. From this perspective, most existing OAL algorithms are not suitable for data stream from a varied distribution.

What's more importantly, most of the exiting query strategies usually depend on the margin value of the instances; however, the margin value will fail to represent the true label information as the old model has not been able to model the new concept. In order

to be adaptively query instances along with the concept drift, it would be interesting to develop new OAL algorithm for concept drift datasets, which not only can capture the margin value, but also can capture the concept drift.

7.3.2 Distributed Online Active Learning

Although OAL has been very efficient both in terms of training time and labeling effort, the exiting algorithms are still hard to handle the large-scale and high velocity data streams in real-world applications, such as social media, e-commercial platforms and so on.

In order to update the models and make predictions in time, distributed machine learning algorithms have been proposed in literature [Boy10, TCS13]. By adopting the distributed scheme, datasets can be processed parallel, and the predictions also can be made timely in order to immediately response users' requests.

However, most of the exiting distributed machine learning algorithms assume that all the coming instances are labeled ahead, which is nearly impossible considering how large and fast the data stream is. Thus, it is very interesting to investigate new distributed OAL algorithms in order to greatly reduce the number of labeled instances [HFRS10, ABDL13].

7.3.3 Online Active Learning with Crowdsourcing

For most of the OAL algorithms, they usually assume that there exists one oracle who can provide reliable labels. However, in reality, the situation is that there are several noisy oracles that can provide labels, such as in crowdsourcing platforms. In other words, different from the traditional OAL, in crowdsourcing, several noisy labels can be obtained when making a query.

In order to train a model, the first step is to prepare a set of labeled instances. When the dataset is small, it is still possible to hire one domain expert to label the data. However, it becomes impossible even unreasonable when dataset become large. Thanks for the crowdsourcing platforms, workers can be hired from worldwide to label the large-scale datasets [DDS⁺09].

As the main purpose of labeling dataset is for training a model, it would be interesting to investigate new algorithms which directly learn with the crowdsourcing [KTK12].

Meanwhile, the labels given by workers in crowdsourcing are usually noisy, so it is necessary to propose new algorithms to aggregate the noisy labels.

7.4 Summary

In this chapter, I firstly conclude this thesis by summarizing the four works proposed, and then present three potential future directions of OAL. It should be noted that there are still several challenges of existing OAL algorithms, and there are also many directions worth to explore. I hope that this thesis can attract more researchers to join this research area and more practitioners to apply the OAL algorithms to more real-world applications.

Appendix A

List of Publications

Conferences:

1. Shuji Hao, Peilin Zhao, Steven Hoi, Chunyan Miao. “SOAL: Second Order Online Active Learning”. The IEEE International Conference on Data Mining (**ICDM 2016**). Barcelona, Spain,
2. Chi Zhang, Peilin Zhao, Shuji Hao, Yeng Chai Soh, Bu Sung Lee. “ROM: A Robust Online Multi-task Classification Approach”. The IEEE International Conference on Data Mining (**ICDM 2016**). Barcelona, Spain,
3. Shuji Hao, Peilin Zhao, Steven C. H. Hoi, Chunyan Miao, “Learning Relative Similarity from Data Streams: Active Online Learning Approaches”. The 24th ACM International Conference on Information and Knowledge Management (**CIKM 2015**). Melbourne, Australia: 1181-1190,
4. Shuji Hao, Steven C. H. Hoi, Chunyan Miao, Peilin Zhao, “Active Crowdsourcing for Annotation”. IEEE/WIC/ACM Web Intelligence 2015, IEEE/WIC/ACM Intelligent Agent Technology 2015 (**WI’15/IAT’15**)(2) 2015: 1-8, Singapore,
5. Chi Zhang, Shuji Hao, Peilin Zhao, Yengchai Soh, Bu Sung Lee and Steven Hoi, “Distributed Multi-task Classification: A Decentralized Online Learning Approach”, IJCAI 2017, (Under submission),
6. Shuji Hao, Peilin Zhao, Steven C.H. Hoi, Chunyan Miao, “Online Multi-task Relative Similarity Learning”, IJCAI 2017, (Under submission)

Journals:

1. Shuji Hao, Peilin Zhao, Steven C.H. Hoi, Chunyan Miao, “Robust Active Learning with Expert Advice and Its Application in Crowdsourcing”. *ACM Transactions on Intelligent Systems and Technology (ACM TIST)*, (Under review),
2. Shuji Hao, Peilin Zhao, Steven C.H. Hoi, Chunyan Miao, “Second-order Online Active Learning Algorithms and Its Applications”. *IEEE Transactions on Knowledge and Data Engineering*, (Under review),

Bibliography

- [AAKS13] Ittai Abraham, Omar Alonso, Vasilis Kandyas, and Aleksandrs Slivkins. Adaptive crowdsourcing algorithms for the bandit survey problem. In *COLT 2013 - The 26th Annual Conference on Learning Theory, June 12-14, 2013, Princeton University, NJ, USA*, pages 882–910, 2013.
- [ABDL13] Alekh Agarwal, Leon Bottou, Miroslav Dudik, and John Langford. Para-active learning. *arXiv preprint arXiv:1310.8243*, 2013.
- [Aga13] Alekh Agarwal. Selective sampling algorithms for cost-sensitive multiclass prediction. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, pages 1220–1228, 2013.
- [AT05] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):734–749, 2005.
- [BBL09] Maria-Florina Balcan, Alina Beygelzimer, and John Langford. Agnostic active learning. *J. Comput. Syst. Sci.*, 75(1):78–89, 2009.
- [BDL09] Alina Beygelzimer, Sanjoy Dasgupta, and John Langford. Importance weighted active learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 49–56. ACM, 2009.
- [BEWB05] Antoine Bordes, Seyda Ertekin, Jason Weston, and Léon Bottou. Fast kernel classifiers with online and active learning. *The Journal of Machine Learning Research*, 6:1579–1619, December 2005.

- [BHLZ10] Alina Beygelzimer, Daniel J. Hsu, John Langford, and Tong Zhang. Agnostic active learning without constraints. In *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada.*, pages 199–207, 2010.
- [BKMW14] Jonathan Bragg, Andrey Kolobov, Mausam, and Daniel S. Weld. Parallel task routing for crowdsourcing. In *Proceedings of the Seconf AAAI Conference on Human Computation and Crowdsourcing, HCOMP 2014, November 2-4, 2014, Pittsburgh, Pennsylvania, USA*, 2014.
- [BLO62] HD BLOCK. The perceptron: A model for brain functioning. *Reviews of Modern Physics*, 35:123–135, 1962.
- [Bol98] Béla Bollobás. *Random graphs*. Springer, 1998.
- [Boy10] Stephen Boyd. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends? in Machine Learning*, 3(1):1–122, 2010.
- [Bra13] Daren C Brabham. *Crowdsourcing*. Mit Press, 2013.
- [Bri03] Klaus Brinker. Incorporating diversity in active learning with support vector machines. In *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA*, pages 59–66, 2003.
- [CBCG03] Nicolo Cesa-Bianchi, Alex Conconi, and Claudio Gentile. Learning probabilistic linear-threshold classifiers via selective sampling. In *Learning Theory and Kernel Machines*, pages 373–387. Springer, 2003.
- [CBCG05] Nicolò Cesa-Bianchi, Alex Conconi, and Claudio Gentile. A second-order perceptron algorithm. *SIAM Journal on Computing*, 34(3):640–668, January 2005.

- [CBFH⁺97] Nicolò Cesa-Bianchi, Yoav Freund, David Haussler, David P. Helmbold, Robert E. Schapire, and Manfred K. Warmuth. How to use expert advice. *Journal of the ACM*, 44(3):427–485, May 1997.
- [CBGO09] Nicolo Cesa-Bianchi, Claudio Gentile, and Francesco Orabona. Robust bounds for classification via selective sampling. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 121–128. ACM, 2009.
- [CBGZ06] Nicolò Cesa-Bianchi, Claudio Gentile, and Luca Zaniboni. Worst-case analysis of selective sampling for linear classification. *The Journal of Machine Learning Research*, 7:1205–1230, December 2006.
- [CBL06] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.
- [CBLS05] N. Cesa-Bianchi, G. Lugosi, and G. Stoltz. Minimizing regret with label efficient prediction. *IEEE Transactions on Information Theory*, 51(6):2152–2162, June 2005.
- [CC12] Koby Crammer and Gal Chechik. Adaptive regularization for weight matrices. In John Langford and Joelle Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, ICML ’12, pages 425–432, New York, NY, USA, July 2012. Omnipress.
- [CCBG10] Giovanni Cavallanti, Nicolò Cesa-Bianchi, and Claudio Gentile. Learning noisy linear classifiers via adaptive and selective sampling. *Machine Learning*, 83(1):71–102, May 2010.
- [CCLM15] Cen Chen, Shih-Fen Cheng, Hoong Chuin Lau, and Archan Misra. Towards city-scale mobile crowdsourcing: Task recommendations under trajectory uncertainties. In Qiang Yang and Michael Wooldridge, editors, *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 1113–1119. AAAI Press, 2015.

- [CDK09] Koby Crammer, Mark Dredze, and Alex Kulesza. Multi-class confidence weighted algorithms. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009, 6-7 August 2009, Singapore, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 496–504, 2009.
- [CDKSs06] K Crammer, O Dekel, Joseph Keshet, and Shai Shalev-shwartz. Online passive-aggressive algorithms. *The Journal of Machine Learning Research*, 7:551–585, 2006.
- [CDP08] Koby Crammer, Mark Dredze, and Fernando Pereira. Exact convex confidence-weighted learning. In *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, pages 345–352, 2008.
- [CDP13] Robert Capo, Karl B. Dyer, and Robi Polikar. Active learning in nonstationary environments. In *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, August 2013.
- [CKD09] Koby Crammer, Alex Kulesza, and Mark Dredze. Adaptive regularization of weight vectors. In *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada.*, pages 414–422, 2009.
- [CKD13] Koby Crammer, Alex Kulesza, and Mark Dredze. Adaptive regularization of weight vectors. *Machine Learning*, 91(2):155–187, March 2013.
- [CL10] Koby Crammer and DD Daniel D Lee. Learning via gaussian herding. In *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada.*, pages 1–9, 2010.

- [CLZ13] Xi Chen, Qihang Lin, and Dengyong Zhou. Optimistic knowledge gradient policy for optimal budget allocation in crowdsourcing. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 64–72, 2013.
- [Cra14] Koby Crammer. Doubly aggressive selective sampling algorithms for classification. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics, AISTATS 2014, Reykjavik, Iceland, April 22-25, 2014*, volume 33 of *The Journal of Machine Learning Research Proceedings*, pages 140–148. JMLR.org, 2014.
- [CS02] Koby Crammer and Yoram Singer. A new family of online algorithms for category ranking. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 151–158. ACM, 2002.
- [CS03] Koby Crammer and Y Singer. Ultraconservative online algorithms for multiclass problems. *The Journal of Machine Learning Research*, 3:951–991, 2003.
- [CS05] Koby Crammer and Yoram Singer. Loss bounds for online category ranking. In *Learning Theory*, pages 48–62. Springer, 2005.
- [CSSB10a] Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. Large scale online learning of image similarity through ranking. *The Journal of Machine Learning Research*, 11:1109–1135, 2010.
- [CSSB10b] Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research*, 11:1109–1135, 2010.
- [CYL13] Qiong Cao, Yiming Ying, and Peng Li. Similarity metric learning for face recognition. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2408–2415. IEEE, 2013.

- [CZL⁺11] Wei Chu, Martin Zinkevich, Lihong Li, Achint Thomas, and Belle Tseng. Unbiased online active learning in data streams. *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '11*, page 195, 2011.
- [DCP08] Mark Dredze, Koby Crammer, and Fernando Pereira. Confidence-weighted linear classification. pages 264–271, 2008.
- [DDS⁺09] J. Deng, W. Dong, R. Socher, L. J. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255, June 2009.
- [Dek12] Ofer Dekel. Selective sampling and active learning from single and multiple teachers. *Journal of Machine Learning Research*, 13:2655–2697, 2012.
- [DGS10] Ofer Dekel, Claudio Gentile, and Karthik Sridharan. Robust selective sampling from single and multiple teachers. In *COLT 2010 - The 23rd Conference on Learning Theory, Haifa, Israel, June 27-29, 2010*, pages 346–358, 2010.
- [DHM07] Sanjoy Dasgupta, Daniel J. Hsu, and Claire Monteleoni. A general agnostic active learning algorithm. In *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, pages 353–360, 2007.
- [DKJ⁺07] Jason V Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pages 209–216. ACM, 2007.
- [DLS06] Ofer Dekel, Philip M Long, and Yoram Singer. Online multitask learning. In *Learning Theory*, pages 453–467. Springer, 2006.

- [DRH11] Anhui Doan, Raghu Ramakrishnan, and Alon Y Halevy. Crowdsourcing systems on the world-wide web. *Communications of the ACM*, 54(4):86–96, 2011.
- [DS79] Alexander Philip Dawid and Allan M Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied statistics*, pages 20–28, 1979.
- [EdV13] Carsten Eickhoff and Arjen P de Vries. Increasing cheat robustness of crowdsourcing tasks. *Information retrieval*, 16(2):121–137, 2013.
- [ESYS13] Ehsan Elhamifar, Guillermo Sapiro, Allen Y. Yang, and S. Shankar Sastry. A convex optimization framework for active learning. In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*, pages 209–216, 2013.
- [Fan04] Wei Fan. Systematic data selection to mine concept-drifting data streams. *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '04*, page 128, 2004.
- [FBY92] William B. Frakes and Ricardo Baeza-Yates, editors. *Information Retrieval: Data Structures and Algorithms*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1992.
- [FM97] Yoav Freund and Yishay Mansour. Learning under persistent drift. In *Computational Learning Theory*, pages 109–118, 1997.
- [FSP99] Y Freund, RE Schapire, and Florham Park. Large margin classification using the perceptron algorithm. *Machine learning*, 296:277–296, 1999.
- [FSSSU06] Michael Fink, Shai Shalev-Shwartz, Yoram Singer, and Shimon Ullman. Online multiclass learning by interclass hypothesis sharing. In *Proceedings of the 23rd international conference on Machine learning*, pages 313–320. ACM, 2006.

- [FSST97] 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, September 1997.
- [FZL12] Yifan Fu, Xingquan Zhu, and Bin Li. A survey on instance selection for active learning. *Knowledge and Information Systems*, 35(2):249–283, June 2012.
- [GALH13] Quanquan Gu, Charu C. Aggarwal, Jialu Liu, and Jiawei Han. Selective sampling on graphs for classification. In *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013*, pages 131–139, 2013.
- [Gen02] Claudio Gentile. A new approximate maximal margin classification algorithm. *The Journal of Machine Learning Research*, 2:213–242, 2002.
- [GFT15] Robby Goetschalckx, Alan Fern, and Prasad Tadepalli. Multitask coactive learning. In Qiang Yang and Michael Wooldridge, editors, *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 3518–3524. AAAI Press, 2015.
- [GHP07] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007.
- [GR05] Amir Globerson and Sam T Roweis. Metric learning by collapsing classes. In *Advances in Neural Information Processing Systems 18 [Neural Information Processing Systems, NIPS 2005, December 5-8, 2005, Vancouver, British Columbia, Canada]*, pages 451–458, 2005.
- [GS08] Yuhong Guo and Dale Schuurmans. Discriminative batch mode active learning. In *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, pages 593–600, 2008.

- [GSS13] Alon Gonen, Sivan Sabato, and Shai Shalev-Shwartz. Efficient active learning of halfspaces: an aggressive approach. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, pages 480–488, 2013.
- [GWKP11] Ryan G Gomes, Peter Welinder, Andreas Krause, and Pietro Perona. Crowd-clustering. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain.*, pages 558–566, 2011.
- [GZFX11] Andrew B. Goldberg, Xiaojin Zhu, Alex Furger, and Jun-Ming Xu. OASIS: online active semi-supervised learning. 2011.
- [GZHD12] Quanquan Gu, Tong Zhang, Jiawei Han, and Chris H Ding. Selective labeling via error bound minimization. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.*, pages 323–331, 2012.
- [Han57] J. Hannan. Approximation to bayes risk in repeated plays. *Contributions to the Theory of Games*, 3:97–139, 1957.
- [Han07] Steve Hanneke. Teaching dimension and the complexity of active learning. In *Learning Theory*, pages 66–81. Springer, 2007.
- [Han12] Steve Hanneke. Activized learning : Transforming passive to active with improved label complexity *. *Journal of Machine Learning Research*, 13:1469–1587, 2012.
- [HFRS10] Robbie Haertel, Paul Felt, Eric Ringger, and Kevin Seppi. Parallel active learning: eliminating wait time with minimal staleness. In *Proceedings of the NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing*, pages 33–41. Association for Computational Linguistics, 2010.

- [HLLM06] Steven C.H. Hoi, Wei Liu, Michael R Lyu, and Wei-Ying Ma. Learning distance metrics with contextual constraints for image retrieval. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR2006)*, New York, US, 2006.
- [HLYG13] Gang Hua, Chengjiang Long, Ming Yang, and Yan Gao. Collaborative active learning of a kernel machine ensemble for recognition. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1209–1216. IEEE, 2013.
- [HWB98] M Herbster, MK Warmuth, and Applied Sciences Build. Tracking the best expert. *Machine Learning*, pages 1–29, 1998.
- [HWZ14] Steven C.H. Hoi, Jialei Wang, and Peilin Zhao. Libol: A library for online learning algorithms. *The Journal of Machine Learning Research*, 15(1):495–499, 2014.
- [HZHM15] Shuji Hao, Peilin Zhao, Steven C. H. Hoi, and Chunyan Miao. Learning relative similarity from data streams: Active online learning approaches. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015*, pages 1181–1190, 2015.
- [JKDG09] Prateek Jain, Brian Kulis, Inderjit S Dhillon, and Kristen Grauman. Online metric learning and fast similarity search. In *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada.*, pages 761–768, 2009.
- [JWZ09] Rong Jin, Shijun Wang, and Yang Zhou. Regularized distance metric learning: Theory and algorithm. In *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada.*, pages 862–870, 2009.

- [KHH12] Ece Kamar, Severin Hacker, and Eric Horvitz. Combining human and machine intelligence in large-scale crowdsourcing. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 467–474. International Foundation for Autonomous Agents and Multiagent Systems, 2012.
- [KJ00] Ralf Klinkenberg and Thorsten Joachims. Detecting concept drift with support vector machines. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29 - July 2, 2000*, pages 487–494, 2000.
- [Kli04] Ralf Klinkenberg. Learning drifting concepts: Example selection vs. example weighting. *Intelligent Data Analysis*, 8(3):281–300, 2004.
- [KOS11a] David R Karger, Sewoong Oh, and Devavrat Shah. Budget-optimal crowdsourcing using low-rank matrix approximations. In *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on*, pages 284–291. IEEE, 2011.
- [KOS11b] David R Karger, Sewoong Oh, and Devavrat Shah. Iterative learning for reliable crowdsourcing systems. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain.*, pages 1953–1961, 2011.
- [KOS14] David R Karger, Sewoong Oh, and Devavrat Shah. Budget-optimal task allocation for reliable crowdsourcing systems. *Operations Research*, 62(1):1–24, 2014.
- [KTK12] Hiroshi Kajino, Yuta Tsuboi, and Hisashi Kashima. A convex formulation for learning from crowds. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, AAAI’12*, pages 73–79. AAAI Press, 2012.
- [KW97] Jyrki Kivinen and Manfred K Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1–63, 1997.

- [KZB⁺14] Georg Krempf, Indre Zliobaite, Dariusz Brzeziński, Eyke Hüllermeier, Mark Last, Vincent Lemaire, Tino Noack, Ammar Shaker, Sonja Sievi, Myra Spiliopoulou, et al. Open challenges for data stream mining research. *ACM SIGKDD Explorations Newsletter*, 16(1):1–10, 2014.
- [LC94] David D. Lewis and Jason Catlett. Heterogenous uncertainty sampling for supervised learning. In *Machine Learning, Proceedings of the Eleventh International Conference, Rutgers University, New Brunswick, NJ, USA, July 10-13, 1994*, pages 148–156, 1994.
- [Lew91] David D. Lewis. Learning in intelligent information retrieval. In *Proceedings of the Eighth International Workshop (ML91), Northwestern University, Evanston, Illinois, USA*, pages 235–239, 1991.
- [LHCJ10] Guangxia Li, Steven C.H. Hoi, Kuiyu Chang, and Ramesh Jain. Microblogging sentiment detection by collaborative online learning. *2010 IEEE International Conference on Data Mining*, pages 893–898, December 2010.
- [LIS13] Qiang Liu, Alexander T. Ihler, and Mark Steyvers. Scoring workers in crowdsourcing: How many control questions are enough? In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 1914–1922. 2013.
- [Lit88] Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2(4):285–318, April 1988.
- [LL02] Y Li and PM Philip M Long. The relaxed online maximum margin algorithm. *Machine Learning*, pages 361–387, 2002.
- [LM13] Joel Lehman and Risto Miikkulainen. Leveraging human computation markets for interactive evolution. In *International Conference on Machine Learning 2013 Workshop: Machine Learning Meets Crowdsourcing*, 2013.

- [LW89] Nick Littlestone and MK M.K. Warmuth. The weighted majority algorithm. In *30th Annual Symposium on Foundations of Computer Science*, pages 256–261. IEEE, 1989.
- [LYZ13] Hongwei Li, Bin Yu, and Dengyong Zhou. Error rate analysis of labeling by crowdsourcing. In *ICML Workshop: Machine Learning Meets Crowdsourcing. Atalanta, Georgia, USA*, 2013.
- [LZH16] Jing Lu, Peilin Zhao, and Steven C. H. Hoi. Online passive-aggressive active learning. *Machine Learning*, 103(2):141–183, 2016.
- [MMK00] Ion Muslea, Steven Minton, and Craig A. Knoblock. Selective sampling with redundant views. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence, July 30 - August 3, 2000, Austin, Texas, USA.*, pages 621–626, 2000.
- [MSSV09] Justin Ma, Lawrence K Saul, Stefan Savage, and Geoffrey M Voelker. Identifying suspicious urls: an application of large-scale online learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 681–688. ACM, 2009.
- [NWHD13] Feiping Nie, Hua Wang, Heng Huang, and Chris H. Q. Ding. Early active learning via robust representation and structured sparsity. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, pages 1572–1578, 2013.
- [OC10] Francesco Orabona and Koby Crammer. New adaptive algorithms for on-line classification. In *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada.*, pages 1840–1848, 2010.
- [OCB11] Francesco Orabona and Nicolo Cesa-Bianchi. Better algorithms for selective sampling. In *Proceedings of the 28th international conference on Machine learning (ICML-11)*, pages 433–440, 2011.

- [Ols09] Fredrik Olsson. A literature survey of active machine learning in the context of natural language processing. 2009.
- [RHC99] Yong Rui, Thomas S Huang, and Shih-Fu Chang. Image retrieval: Current techniques, promising directions, and open issues. *Journal of visual communication and image representation*, 10(1):39–62, 1999.
- [RM01] Nicholas Roy and Andrew McCallum. Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001)*, Williams College, Williamstown, MA, USA, June 28 - July 1, 2001, pages 441–448, 2001.
- [RU08] Tom Richardson and Ruediger Urbanke. *Modern coding theory*. Cambridge University Press, 2008.
- [Set10a] Burr Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114, June 2010.
- [Set10b] Burr Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52:55–66, 2010.
- [SHL14] Doyen Sahoo, Steven CH Hoi, and Bin Li. Online multiple kernel regression. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 293–302. ACM, 2014.
- [SHV⁺10] Shiliang Sun, David R. Hardoon, Marley Vellasco, Marcilio C.P. Souto, and André C.P.L.F. de Carvalho. Active learning with extremely sparse labeled examples. *Neurocomputing*, 73(16-18):2980–2988, October 2010.
- [SHX15] Mrinmaya Sachan, Eduard H. Hovy, and Eric P. Xing. An active learning approach to coreference resolution. In Qiang Yang and Michael Wooldridge, editors, *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 1312–1318. AAAI Press, 2015.

- [SJ03] Matthew Schultz and Thorsten Joachims. Learning a distance metric from relative comparisons. In *Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS 2003, December 8-13, 2003, Vancouver and Whistler, British Columbia, Canada]*, pages 41–48, 2003.
- [SJHL06] Luo Si, Rong Jin, Steven C.H. Hoi, and Michael R Lyu. Collaborative image retrieval via regularized metric learning. *ACM Multimedia Systems Journal (MMSJ)*, 12(1):34–44, 2006.
- [SPI08] 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*, pages 614–622. ACM, 2008.
- [SS07] Shai Shalev-Shwartz and Yoram Singer. A unified algorithmic approach for efficient online label ranking. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics, AISTATS 2007, San Juan, Puerto Rico, March 21-24, 2007*, pages 452–459, 2007.
- [SS11] Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends? in Machine Learning*, 4(2):107–194, 2011.
- [SSS05] Shai Shalev-Shwartz and Yoram Singer. A new perspective on an old perceptron algorithm. In *Learning Theory*, pages 264–278. Springer, 2005.
- [SSS07] Shai Shalev-Shwartz and Yoram Singer. *Online learning: Theory, algorithms, and applications*. PhD thesis, Hebrew University, 2007.
- [SSSN04] Shai Shalev-Shwartz, Yoram Singer, and Andrew Y Ng. Online and batch learning of pseudo-metrics. In *Proceedings of the twenty-first international conference on Machine learning*, page 94. ACM, 2004.
- [SYTC11] Noam Slonim, Elad Yom-Tov, and Koby Crammer. Active online classification via information maximization. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 1498, 2011.

- [TCS13] Zaid J. Towfic, Jianshu Chen, and Ali H. Sayed. On distributed online classification in the midst of concept drifts. *Neurocomputing*, 112:138–152, July 2013.
- [TK02] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research*, pages 45–66, 2002.
- [TTHR⁺14] Long Tran-Thanh, Trung Dong Huynh, Avi Rosenfeld, Sarvapali D Ramchurn, and Nicholas R Jennings. Budgetfix: budget limited crowdsourcing for interdependent task allocation with quality guarantees. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 477–484. International Foundation for Autonomous Agents and Multiagent Systems, 2014.
- [VGK⁺14] Matteo Venzani, John Guiver, Gabriella Kazai, Pushmeet Kohli, and Milad Shokouhi. Community-based bayesian aggregation models for crowdsourcing. In *Proceedings of the 23rd international conference on World wide web*, pages 155–164. International World Wide Web Conferences Steering Committee, 2014.
- [Vov95] Vladimir G Vovk. A game of prediction with expert advice. In *Proceedings of the eighth annual conference on Computational learning theory*, pages 51–60. ACM, 1995.
- [WBBP10] Peter Welinder, Steve Branson, Serge J. Belongie, and Pietro Perona. The multidimensional wisdom of crowds. In *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada.*, pages 2424–2432, 2010.
- [WBS05] Kilian Q. Weinberger, John Blitzer, and Lawrence K. Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in Neural Information Processing Systems 18 [Neural Information Processing*

- Systems, NIPS 2005, December 5-8, 2005, Vancouver, British Columbia, Canada*], pages 1473–1480, 2005.
- [WHZL13] Jialei Wang, Steven CH Hoi, Peilin Zhao, and Zhi-Yong Liu. Online multi-task collaborative filtering for on-the-fly recommender systems. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 237–244. ACM, 2013.
- [WIP12] De Wang, Danesh Irani, and Calton Pu. Evolutionary study of web spam: Webb spam corpus 2011 versus webb spam corpus 2006. In *CollaborateCom*, pages 40–49, 2012.
- [WJH⁺09] Lei Wu, Rong Jin, Steven C. H. Hoi, Jianke Zhu, and Nenghai Yu. Learning bregman distance functions and its application for semi-supervised clustering. In *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada.*, pages 2089–2097, 2009.
- [WLX13] Wei Wu, Hang Li, and Jun Xu. Learning query and document similarities from click-through bipartite graph with metadata. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 687–696. ACM, 2013.
- [WP10] Peter Welinder and Pietro Perona. Online crowdsourcing: rating annotators and obtaining cost-effective labels. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 25–32. IEEE, 2010.
- [WRW⁺09] Jacob Whitehill, Paul Ruvolo, Tingfan Wu, Jacob Bergsma, and Javier R. Movellan. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada.*, pages 2035–2043, 2009.

- [WZH12] Jialei Wang, Peilin Zhao, and Steven C.H. Hoi. Exact soft confidence-weighted learning. In John Langford and Joelle Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, ICML '12, pages 121–128, New York, NY, USA, July 2012. Omnipress.
- [WZH14] Jialei Wang, Peilin Zhao, and Steven CH Hoi. Cost-sensitive online classification. *Knowledge and Data Engineering, IEEE Transactions on*, 26(10):2425–2438, 2014.
- [XYT⁺03] Zhao Xu, Kai Yu, Volker Tresp, Xiaowei Xu, and Jizhi Wang. Representative sampling for text classification using support vector machines. In *Advances in Information Retrieval, 25th European Conference on IR Research, ECIR 2003, Pisa, Italy, April 14-16, 2003, Proceedings*, pages 393–407, 2003.
- [YFRD11] 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 (ICML-11)*, pages 1161–1168, 2011.
- [YJ06] Liu Yang and Rong Jin. Distance metric learning: A comprehensive survey. *Michigan State University*, 2, 2006.
- [YSL⁺15] Rui Yan, Yiping Song, Cheng-Te Li, Ming Zhang, and Xiaohua Hu. Opportunities or risks to reduce labor in crowdsourcing translation? characterizing cost versus quality via a pagerank-hits hybrid model. In Qiang Yang and Michael Wooldridge, editors, *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 1025–1032. AAAI Press, 2015.
- [ZCZJ14] Yuchen Zhang, Xi Chen, Dengyong Zhou, and Michael I Jordan. Spectral methods meet em: A provably optimal algorithm for crowdsourcing. *arXiv preprint arXiv:1406.3824*, 2014.
- [ZH13] Peilin Zhao and Steven C. H. Hoi. Cost-sensitive online active learning with application to malicious URL detection. In *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013*, pages 919–927, 2013.

BIBLIOGRAPHY

- [ZHJ11] Peilin Zhao, Steven C.H. Hoi, and Rong Jin. Double updating online learning. *The Journal of Machine Learning Research*, 12:1587–1615, 2011.
- [ZHZ13] Peilin Zhao, Steven CH Hoi, and Jinfeng Zhuang. Active learning with expert advice. In *Uncertainty in Artificial Intelligence*, page 704. Citeseer, 2013.
- [ZJYH11] Peilin Zhao, Rong Jin, Tianbao Yang, and Steven C.H. Hoi. Online auc maximization. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 233–240, 2011.