7-2018

Online active learning with expert advice

Shuji HAO
Institute of High Performance of Computing

Peiying HU
Liaoning Normal University

Peilin ZHAO
South China University of Technology

Steven C. H. HOI
Singapore Management University, CHHOI@smu.edu.sg

Chunyan MIAO
Nanyang Technological University

DOI: https://doi.org/10.1145/3201604

Citation

HAO, Shuji; HU, Peiying; ZHAO, Peilin; HOI, Steven C. H.; and MIAO, Chunyan. Online active learning with expert advice. (2018). *ACM Transactions on Knowledge Discovery from Data*. 12, (5), 1-22. Research Collection School Of Information Systems.
Available at: https://ink.library.smu.edu.sg/sis_research/4185

Published in ACM Transactions on Knowledge Discovery from Data, Volume 12 Issue 5, July 2018, Article No. 58
https://doi.org/10.1145/3201604

Online Active Learning with Expert Advice

SHUJI HAO, Institute of High Performance of Computing
PEIYING HU, Liaoning Normal University
PEILIN ZHAO, South China University of Technology
STEVEN C. H. HOI, Singapore Management University
CHUNYAN MIAO, Nanyang Technological University

In literature, learning with expert advice methods usually assume that a learner always obtain the true label of every incoming training instance at the end of each trial. However, in many real-world applications, acquiring the true labels of all instances can be both costly and time consuming, especially for large-scale problems. For example, in the social media, data stream usually comes in a high speed and volume, and it is nearly impossible and highly costly to label all of the instances. In this article, we address this problem with active learning with expert advice, where the ground truth of an instance is disclosed only when it is requested by the proposed active query strategies. Our goal is to minimize the number of requests while training an online learning model without sacrificing the performance. To address this challenge, we propose a framework of active forecasters, which attempts to extend two fully supervised forecasters, Exponentially Weighted Average Forecaster and Greedy Forecaster, to tackle the task of online active learning (OAL) with expert advice. Specifically, we proposed two OAL with expert advice algorithms, named Active Exponentially Weighted Average Forecaster (AEWAF) and active greedy forecaster (AGF), by considering the difference of expert advices. To further improve the robustness of the proposed AEWAF and AGF algorithms in the noisy scenarios (where noisy experts exist), we also proposed two robust active learning with expert advice algorithms, named Robust Active Exponentially Weighted Average Forecaster and Robust Active Greedy Forecaster. We validate the efficacy of the proposed algorithms by an extensive set of experiments in both normal scenarios (where all of experts are comparably reliable) and noisy scenarios.

CCS Concepts: • **Theory of computation** → *Streaming models*;

Additional Key Words and Phrases: Online learning, active learning, expert advice, data streaming

ACM Reference format:
Shuji Hao, Peiying Hu, Peilin Zhao, Steven C. H. Hoi, and Chunyan Miao. 2018. Online Active Learning with Expert Advice. *ACM Trans. Knowl. Discov. Data.* 12, 5, Article 58 (June 2018), 22 pages.
https://doi.org/10.1145/3201604

Steven C. H. Hoi is supported by the National Research Foundation, Prime Minister's Office, Singapore under its International Research Centres in Singapore Funding Initiative.
Authors' addresses: S. Hao, Institute of High Performance Computing, 138632 Singapore; email: haosj@ihpc.a-star.edu.sg; P. Hu, School of Computer and Information Technology, Liaoning Normal University, Dalian 116021, China; email: pyhucs@gmail.com; P. Zhao (corresponding author), School of Software Engineering, South China University of Technology, Guangzhou 510630, China; email: peilinzhao@hotmail.com; S. C. H. Hoi, School of Information Systems, Singapore Management University, 188065 Singapore; email: chhoi@smu.edu.sg; C. Miao, School of Computer Engineering, Nanyang Technological University, 639798 Singapore; email: ascymiao@ntu.edu.sg.

1 INTRODUCTION

Learning with expert advice has been well studied for years in literature (Bousquet and Warmuth 2002; Cesa-Bianchi et al. 1997; Herbster and Warmuth 1998; Littlestone and Warmuth 1994). For a typical learning with expert advice task, it assumes that an online learner can access a pool of experts. At each trial, the pool of experts receive an incoming instance, and make their corresponding advices, based on which, the online learner makes the final predication. After that, the true label of the incoming instance will be disclosed by querying an external oracle (who is assumed to always give 100% correct label) at the end of the trial. The queried true label in turn can determine the losses suffered by both the online learner and the experts. The goal of this learning with expert advice framework is to enable the online learner make predictions as accurate as possible based on the advices. This framework was first introduced by Littlestone and Warmuth (1994), who proposed the well-known weighted majority voting algorithms. Over the past decades, the similar problem has been extensively explored by other studies in literature, including Cesa-Bianchi et al. (1997, 2006), Freund and Schapire (1997), Foster and Vohra (1993), Haussler et al. (1995), Vovk (1990), and so on.

The existing learning with expert advice methods usually assume that the true label of every incoming training instance will be *always* disclosed from an oracle. However, requesting the true label of an instance from the oracle is often expensive or time consuming in many real-world applications. For example, in disease diagnose task, it is usually more expensive to obtain advice from a senior doctor (oracle) than the advices from a pool of junior doctors. Unlike the conventional approaches, this article investigates a new framework of *active* learning with expert advice, in which the true label of an incoming instance may or may not be disclosed at each trial, depending on whether the active learner decides to query true label or not. The goal of active learning with expert advice is not only to train an accurate online learner, but also to minimize the number of queried instances. This problem is very challenging because we must design effective query strategies which cannot only minimize the number of queried instances but also do not sacrifice the performance of the online learner.

To overcome the challenges, we propose a framework of active forecasters by extending two regular forecasters, i.e., Exponentially Weighted Average Forecaster (EWAF) and Greedy Forecaster (GF), to tackle the task of active learning with expert advice. Specifically, we first proposed two active learning with expert advice algorithms, named Active Exponentially Weighted Average Forecaster (AEWAF) and Active Greedy Forecaster (AGF), by considering only the difference of expert advices. These two proposed algorithms perform well in the normal scenarios where all the experts are comparably reliable. However, the proposed query strategies would suffer when noisy experts exist, which is quite common in practical applications, such as in crowdsourcing, where there exist spammers who providing random annotations. To tackle this issue, we further proposed two robust active learning with expert advice algorithms, named Robust Active Exponentially Weighted Average Forecaster (RAEWAF) and Robust Active Greedy Forecaster (RAGF).

The rest of this article is organized as follows. Section 2 reviews the related works in literature. Section 3 introduces the problem setting of learning with expert advice and two well-known forecasters. Section 4 presents the active learning with expert advice framework and two active forecaster algorithms, AEWAF and AGF. Section 5 presents two robust online active learning (OAL) with expert advice algorithms, RAEWAF and RAGF. Section 6 shows our empirical results, and Section 7 concludes this work.

2 LITERATURE REVIEW

Our work is related to two groups of studies in machine-learning literature: online learning and active learning.

2.1 Online Learning

Online learning has been an active research topic in machine-learning community (Aleksandrov et al. 2015; Guo 2015; Hao et al. 2017a; Hayakawa et al. 2015; Hoi et al. 2018, 2014; Jahedpari 2015; Ruvolo and Eaton 2014; Veness et al. 2015; Zhang et al. 2018, 2016; Zhao et al. 2011b), in which a variety of online learning models has been proposed. Typically based on model updating strategy, the existing online learning algorithms can be categorized into two main groups: (i) first-order-based online learning, where only the first-order feature information is exploited, (ii) second-order-based online learning models, which not only maintains the first-order feature information, but also maintains the second-order information, such as the covariance matrix of the feature information.

In the first-order-based online learning algorithms, one of the most well-known algorithms is the Perceptron algorithm (Block 1962; Mohri and Rostamizadeh 2013), which updates the learner by adding the misclassified instance with a fixed weight to the current set of support vectors. Recently, several works also studied the first-order-based online learning algorithms by maximizing the margin value. One pioneer work is the Relaxed Online Maximum Margin Algorithm (Li and Long 2002), which repeatedly chooses the classifier which can correctly classify the existing training instances with a large margin. Another work is the Passive-Aggressive algorithms (PA) (Crammer et al. 2006), which updates the current model when the current instance is misclassified or its prediction value does not reach a predefined margin value. By examining the empirical performance of these first-order-based online learning algorithms, we can observe that the large margin algorithms can generally outperform the Perceptron algorithm. However, the performance of these large margin algorithms is still restricted as only the first-order information is adopted.

In recent years, researchers have been actively designing second-order-based online learning algorithms in order to overcome the limitation of first-order-based algorithms. Generally, the performance of second-order-based algorithms can be significantly improved by exploring the parameter confidence information (second-order information). One of the well-known second-order models is the second-order Perceptron algorithm (Cesa-Bianchi et al. 2005), which is usually viewed as a variant of the whitened Perceptron algorithm. The authors explore the online correlation matrices of the previously seen instances to achieve the whitened effect. Later, several large margin second-order online learning algorithms are also proposed, such as Confidence-Weighted (CW) learning (Dredze et al. 2008), which maintains a Gaussian distribution over the model parameters and uses the covariance of the parameters to guide the update of each parameter. Although CW is promising both in theory and empirical studies, it may overfit in some liner inseparable cases as the model assumes that the dataset is liner separable. To relax such strong assumption, researchers have proposed improved versions, such as the Adaptive Regularization Of Weights algorithm (AROW) (Crammer et al. 2013) and Soft CW algorithms (Wang et al. 2012) by employing an adaptive regularization for each training instance. In general, the second-order algorithms can consistently converge faster and perform better than the first-order-based algorithms.

Online learning with expert advice, as a special scheme in online learning area, typically study the problem where the online learner can make prediction based on the advices provided by several experts. This learning scheme was first introduced by Littlestone and Warmuth (1994), who proposed the well-known weighted majority voting algorithm. Following the similar idea, a serial of algorithms was proposed (Foster and Vohra 1993; Haussler et al. 1995; Vovk 1990), such as EWAF and GF (Cesa-Bianchi and Lugosi 2006) and so on. These algorithms are also extended to multi-task setting (Abernethy et al. 2007) and concept drifting scenario (Kolter and Maloof 2007). We recommend the reader to refer a comprehensive book written by Cesa-Bianchi and Lugosi (2006).

Most of these methods often assume a fully supervised learning settings, where the class label is always revealed to the learner at the end of each learning iteration, a scenario which is not always realistic for many real-world applications.

2.2 Active Learning

The goal of active learning is to train a well-performed predictive model by actively selecting a small subset of informative instances (whose labels are given). As active learning can largely reduce the labeling cost, it has been extensively studied in the batch-based learning scenarios (Hanneke 2014; Tong 2001). Existing active learning techniques could be generally grouped into four categories as follows: (1) uncertainty-based (Cesa-Bianchi et al. 2006; Hao et al. 2017b, 2015, 2016; Lu et al. 2016; Tong and Koller 2002) query strategies, and these strategies would query the instances on which the models have a lower certainty on their prediction. Uncertainty-based query strategies are the most widely used ones as they are simple and easy to implement; (2) query the instances on which the hypothesis space has the most disagreement degree on their predictions (Atlas et al. 1990; Freund and Mansour 1997; Hanneke 2016; Hanneke and Yang 2015); (3) label the instances which could minimize the expected error and variance on the pool of unlabeled instances (Guo and Schuurmans 2008); and (4) exploit the structure information (Sheng et al. 2008) among the instances. More about batch-based active learning studies can be found in the comprehensive survey (Hanneke 2012; Settles 2010).

Batch-based active learning algorithms are effective in reducing labeling cost in several applications, such as text classification, image recognitions, and abnormal detection. However, these algorithms typically require that all of the data should be collected first before the active learning process. This makes them infeasible in some real-world applications, such as in online social media platforms, where data usually comes in a sequential manner. To overcome this challenge, researchers have studied OAL (Cesa-Bianchi et al. 2006; Fujii and Kashima 2016; Lu et al. 2016; Zhao and Hoi 2013), also known as selective sampling, which aims to learn predictive models from a sequence of unlabeled data given limited label query budget. These OAL algorithms typically adopt first-order-based query strategies, such as margin-based query strategy. This will make the algorithms suffer from two major limitations. First, the performance (in terms of accuracy) of these algorithms is usually limited as most of them adopt first-order-based predictive models. Second, their active query strategy often strongly relies on the predictive model $\mathbf{w}_t$, which may not be precise in the early rounds of online learning.

Our work is closely related to the existing studies of OAL (Amin et al. 2015; Cesa-Bianchi et al. 2006; Lu et al. 2014; Zhao and Hoi 2013). However, the traditional OAL algorithms usually assume the input of online learner is the instance, and the designed query strategies are also focused on the prediction of the online learner. These characteristics make them hard to be applied to the learning with expert advice setting, where the input of online learner is advices from experts, and the instances are not accessible to the online learner. In this article, we are specifically interested in designing OAL with expert advice.

3 PROBLEM SETTING AND BACKGROUND

Specifically, we considered learning with expert advice for online classification tasks, which have been extensively studied in machine learning in the past few years (Crammer et al. 2006; Hoi et al. 2014; Rosenblatt 1958; Zhao et al. 2011a) for different problems, such as sentiment detection in social media (Li et al. 2010), cost-sensitive classification for malicious URL detection (Wang et al. 2014; Zhao and Hoi 2013), and online feature selection for high-dimensional tasks (Ruvolo and Eaton 2014; Wang et al. 2013). Learning with expert advice algorithm typically acts in a sequential manner. Consider a streaming of instances $\mathbf{x}_1, \ldots, \mathbf{x}_T \in \mathbb{R}^d$, an online learner termed as "forecaster" aims to predict the outcome (e.g., class label) of every incoming instance $\mathbf{x}_t$. The forecaster sequentially computes its predictions based on the advices from a set of N "experts." Specifically, at the tth round, after receiving an instance $\mathbf{x}_t$, the forecaster first accesses to the predictions

$f_i(\mathbf{x}_t) : \mathbb{R}^d \to [0,1] | i = 1, \ldots, N\}$ made by a set of experts, and then the forecaster makes its own prediction $p_t \in [0,1]$ based on these experts' predictions $f_i(\mathbf{x}_t)$. After p_t is computed, the true outcome $y_t \in \{0,1\}$ is disclosed by querying an oracle who is assumed to be capable to provide ground truth y_t.

Once the ground truth y_t is queried from the oracle, the performance of both the forecaster and the experts can be scored by some non-negative loss functions, e.g., the absolute loss

$$\ell(p_t, y_t) = |p_t - y_t| \tag{1}$$

for the forecaster and $\ell(f_i(\mathbf{x}_t), y_t) = |f_i(\mathbf{x}_t) - y_t|$ for each expert. The cumulative loss suffered by each expert and the forecaster, respectively, can be computed by summering the loss suffered on all historical instances, respectively, as follows:

$$L_{i,T} = \sum_{t=1}^{T} \ell(f_i(\mathbf{x}_t), y_t), \quad L_T = \sum_{t=1}^{T} \ell\left(p_t, y_t\right). \tag{2}$$

The goal of forecaster is to make prediction as good as the best expert in the pool, which is equivalent to minimizing the difference between the cumulative loss suffered by the forecaster and the one suffered by the best expert. Formally, we want to minimize the following term:

$$R_T = L_T - \min_{1 \le i \le N} L_{i,T},$$

where R_T is termed as "regret" which scores the loss difference between the forecaster and the best expert.

To solve the above task of learning with expert advice, a natural strategy for forecaster is weighted majority voting. Specifically, at time t, the forecaster makes its own prediction as follows:

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

where $w_{i,t-1}$ is the cumulative performance weight assigned to the ith expert at time $t-1$. The intuitive idea of learning the performance weights is to assign large weights for those experts with small $L_{i,t-1}$ and small weights for those with large $L_{i,t-1}$.

"EWAF" (Cesa-Bianchi and Lugosi 2006) is one of the forecasters which adopt this weighted majority voting strategy. In particular, by defining the cumulative performance weight as $w_{i,t-1} = \exp(\eta L_{i,t-1}) / \sum_{j=1}^{N} \exp(\eta L_{j,t-1})$, the EWAF forecaster makes the following prediction:

$$p_t = \frac{\sum_{i=1}^{N} \exp(-\eta L_{i,t-1}) f_i(\mathbf{x}_t)}{\sum_{i=1}^{N} \exp(-\eta L_{i,t-1})}, \tag{3}$$

where $\eta > 0$ is the learning rate and $L_{i,t-1}$ is the cumulative loss defined as in Equation (2).

In addition to the EWAF, another very effective forecaster, known as "GF" (Cesa-Bianchi and Lugosi 2006), makes the following prediction:

$$p_t = \pi_{[0,1]}\left(\frac{1}{2} + \frac{1}{2\eta} \ln \frac{\sum_{i=1}^{N} \exp(-\eta L_{i,t-1} - \eta \ell(f_i(\mathbf{x}_t), 1))}{\sum_{i=1}^{N} \exp(-\eta L_{i,t-1} - \eta \ell(f_i(\mathbf{x}_t), 0))}\right), \tag{4}$$

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

4 ACTIVE LEARNING WITH EXPERT ADVICE

Both the EWAF and GF forecasters are effective to make prediction as good as the best expert; however, to achieve the effectiveness, both of the forecasters requires to query the true labels of all instances in order to update the cumulative loss suffered by each expert. This will be costly and time consuming, especially for large-scale problems. In this section, we address this issue by proposing a novel framework of OAL with expert advice. Unlike the above regular learning with expert advice task where the true label of every incoming instance is *always* revealed to the forecaster, in an active learning with expert advice task, the true label of an incoming instance is *only* revealed whenever the learner has made a request from the oracle.

We introduce binary variables $z_t \in \{0, 1\}$ to indicate whether an active forecaster has decided to request the true label y_t of $\mathbf{x}_t$ or not. When $z_t = 1$, it means that the active forecaster decides to query the true label. We also denoted by $\widehat{L}_{i,T}$ the cumulative loss suffered by the ith expert on the queried instances with active forecaster, i.e., $\widehat{L}_{i,T} = \sum_{t=1}^{T} \ell(f_i(\mathbf{x}_t), y_t) \cdot z_t$.

Hence, the predication for $\mathbf{x}_t$ made 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 defined as follows based on different forecasters:

$$\bar{p}_t = \frac{\sum_{i=1}^{N} \exp(-\eta \widehat{L}_{i,t-1}) f_i(\mathbf{x}_t))}{\sum_{i=1}^{N} \exp(-\eta \widehat{L}_{i,t-1})} \text{ (EWAF)},$$

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

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

The key challenge for active forecaster is to decide when the forecaster should or should not make a request to acquire the true label. 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, our key motivation is to find some appropriate *confidence condition* such that it helps the active forecaster decide when we could skip the request of a true label whenever the *confidence condition* is satisfied. One intuitive idea is to seek the confidence condition by estimating the difference between p_t made by the fully supervised forecaster and $\widehat{p}_t$ made by the active forecaster. Intuitively, the smaller the difference, the more confident we have for the prediction made by the active forecaster. Before introducing our proposed confidence conditions, for convenience of presentation, we introduce

$$\widehat{H}_{i,T} = \sum_{t=1}^{\ell} (f_i(\mathbf{x}_t), y_t) \cdot (1 - z_t) \tag{5}$$

to denote the cumulative loss of ith expert suffered on the instances ($z_t = 0$) which are *not queried* to obtain ground truth. It is easy to see $L_{i,T} = \widehat{L}_{i,T} + \widehat{H}_{i,T}$.

4.1 Active Exponentially Weighted Average Forecaster (AEWAF)

Based on fully supervised forecaster EWAF, we now present a *confidence condition* for active forecaster AEWAF in the following theorem, which guarantees a small difference between p_t and $\widehat{p}_t$.

THEOREM 1. *For a small constant $\delta > 0$, $\max_{1 \le i,j \le N} |f_i(\mathbf{x}_t) - f_j(\mathbf{x}_t)| \le \delta$ implies $|p_t - \widehat{p}_t| \le \delta$.*

Proof. For the AEWAF strategy, the distance between p_t and $\widehat{p}_t$ is computed as follows:

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

where

$$
\gamma_{i,j,t-1} = \exp(-\eta \widehat{L}_{i,t-1}) \exp(-\eta \widehat{H}_{i,t-1}) \exp(-\eta \widehat{L}_{j,t-1}).
$$

Thus, if $\max_{1 \le i,j \le N} |f_i(\mathbf{x}_t) - f_j(\mathbf{x}_t)| \le \delta$, it is easy to prove that $|p_t - \widehat{p}_t| \le \delta$. □

If the *confidence condition* is satisfied, it means that the active forecaster AEWAF can make a prediction $\widehat{p}_t$ which is very close to the prediction p_t made by the fully supervised forecaster EWAF, and we can safely skip to query the true label of this instance and continue to process the next instance. Otherwise, we should place a query to obtain the true label in order to update the active forecaster AEWAF.

4.2 Active Greedy Forecaster (AGF)

Based on the fully supervised forecaster GF, we now propose a *confidence condition* for active forecaster AGF in the theorem below, which also guarantees a small difference between $\widehat{p}_t$ and p_t.

Theorem 2. *For a small constant $\delta > 0$, $\max_{1 \le i \le N} |f_i(\mathbf{x}_t) - \bar{p}_t| \le \delta$ implies $|p_t - \widehat{p}_t| \le \delta$.*

Proof. We can bound p_t from the above as follows:

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

where

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

Since $\forall i \in [N]$

$$\begin{aligned}\frac{\alpha_{i,t}}{\beta_{i,t}} &= \frac{\sum_{j=1}^{N} \exp\left(-\eta\left(\widehat{L}_{j,t-1} + \ell\left(f_j\left(\mathbf{x}_t\right), 0\right)\right)\right)}{\sum_{j=1}^{N} \exp\left(-\eta\left(\widehat{L}_{j,t-1} + \ell\left(f_j\left(\mathbf{x}_t\right), 1\right)\right)\right)} \times \frac{\exp\left(-\eta\left(\widehat{L}_{i,t-1} + \ell\left(f_i\left(\mathbf{x}_t\right), 1\right)\right)\right)}{\exp\left(-\eta\left(\widehat{L}_{i,t-1} + \ell\left(f_i\left(\mathbf{x}_t\right), 0\right)\right)\right)}, \\ &= \frac{\exp\left(-\eta\left(\ell\left(f_i\left(\mathbf{x}_t\right), 1\right) - \ell\left(f_i\left(\mathbf{x}_t\right), 0\right)\right)\right)}{\exp\left(\eta\left(2\bar{p}_t - 1\right)\right)}, \\ &= \frac{\exp\left(\eta\left(2f_i\left(\mathbf{x}_t\right) - 1\right)\right)}{\exp\left(\eta\left(2\bar{p}_t - 1\right)\right)}, \\ &= \exp\left(2\eta\left(f_i(\mathbf{x}_t) - \bar{p}_t\right)\right) \le \exp\left(2\eta\delta\right),\end{aligned}$$

and $\ln x$ is an increasing function, we have

$$\ln \frac{\sum_{i=1}^{N} \alpha_{i,t} \exp\left(-\eta\widehat{H}_{i,t-1}\right)}{\sum_{j=1}^{N} \beta_{j,t-1} \exp\left(-\eta\widehat{H}_{j,t-1}\right)} \le 2\eta\delta,$$

and

$$p_t \le \widehat{p}_t + \delta. \tag{6}$$

Similar to the above analysis, we have p_t lower bounded as

$$p_t \ge \widehat{p}_t - \delta. \tag{7}$$

Combining Equations (6) and (7), it is easily to obtain $|p_t - \widehat{p}_t| \le \delta$. □

5 ROBUST ACTIVE LEARNING WITH EXPERT ADVICE

The proposed active learning 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, both Theorems 1 and 2, 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 bad or even completely noisy. To resolve this issue, we propose two robust active learning with expert advice algorithms by both considering the experts' advices on current instance and their cumulative performance.

Specifically, in the Theorem 1, the learner will place a query if the maximum difference of advices between any two experts is large enough, i.e., $\max_{1\le i,j\le N} |f_i(\mathbf{x}_t) - f_j(\mathbf{x}_t)| > \delta$, where $\delta > 0$ is a threshold for the query condition. If all the experts perform comparably well, this strategy is reasonable. It implies that two competing experts disagree very much on an instance which thus should be queried as this instance could be a difficult or informative sample to train the learner. However, if there are some noisy experts whose advices are very different from that of good experts, the strategy will result in many queries on non-informative instances. Similarly, Theorem 2 also suffers the same issue.

To tackle this issue, we propose strategies which attempt to weight the good experts' advices much higher than that of poor or noisy experts. More specifically, we propose to import the cumulative performance of each expert into the query strategy, so as to give good experts' advice higher weights than the poor or noisy ones.

5.1 Robust Active Exponentially Weighted Average Forecaster (RAEWAF)

We now provide a theorem as below which guarantees a small difference between $\widehat{p}_t$ and p_t, if the *confidence condition* is satisfied.

Theorem 3. *For a small constant $\delta > 0$, if the following condition is satisfied, then $|p_t - \widehat{p}_t| \le \delta$*

$$\left| \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} \gamma_{i,j,t} (f_i(\mathbf{x}_t) - f_j(\mathbf{x}_t))}{\sum_{i=1}^{N} \sum_{j=1}^{N} \gamma_{i,j,t}} \right| \le \delta, \tag{8}$$

where $\gamma_{i,j,t} = \exp(-\eta(\widehat{L}_{i,t-1} + \widehat{H}_{i,t-1} + \widehat{L}_{j,t-1}))$.

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

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

□

5.2 Robust Active Greedy Forecaster (RAGF)

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

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

$$\left| \frac{1}{2\eta} \ln \frac{\sum_{i=1}^{N} \mu_{i,t} \exp\left(2\eta(f_i(\mathbf{x}_t) - \bar{p}_t)\right)}{\sum_{i=1}^{N} \mu_{i,t}} \right| \le \delta, \tag{9}$$

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

Proof. We can bound p_t from above as follows:

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

ALGORITHM 1: A Framework of Online Active Learning with Expert Advice

Input: a pool of experts $f_i,\ i = 1, \ldots, N$.
Initialize tolerance threshold δ and $\widehat{L}_{i,t} = 0,\ i \in [N]$.
for $t = 1, \ldots, T$ **do**
  receive $\mathbf{x}_t$ and compute $f_i(\mathbf{x}_t)$, $i \in [N]$;
  compute $\bar{p}_t$ according to Equation (5) and set $\widehat{p}_t = \pi_{[0,1]}(\bar{p}_t)$;
  if the *confidence condition* defined in Theorem 1, 2, 3, or 4 is *satisfied* **then**
    skip the label request for instance $\mathbf{x}_t$
  else
    request true label y_t and update $\widehat{L}_{i,t} = \widehat{L}_{i,t-1} + \ell(f_i(\mathbf{x}_t), y_t),\ i \in [N]$;
  end if
end for

where $\beta_{i,t} = \frac{\exp(-\eta[\widehat{L}_{i,t-1}+\ell(f_i(\mathbf{x}_t),0)])}{\sum_{j=1}^{N}\exp(-\eta[\widehat{L}_{j,t-1}+\ell(f_j(\mathbf{x}_t),0)])}, \forall i \in [N]$ and $\alpha_{i,t} = \beta_{i,t} e^{2\eta(f_i(\mathbf{x}_t)-\bar{p}_t)}$.

Clearly, we can see, if

$$\left| \frac{1}{2\eta} \ln \frac{\sum_{i=1}^{N} \mu_{i,t} \exp(2\eta(f_i(\mathbf{x}_t) - \bar{p}_t))}{\sum_{i=1}^{N} \mu_{i,t}} \right| \leq \delta, \tag{10}$$

we have $|p_t - \widehat{p}_t| \leq \delta$. □

From Theorems 3 and 4, we can observe that, at the early stage, there is no much difference of $\gamma_{i,j,t}$ or $\mu_{i,t}$ among the experts, as $L_{i,t-1}$ is almost similar for each expert. 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 $\widehat{L}_{i,t-1}$ suffered by good experts would be much smaller than the one suffered by bad experts. So the values $\gamma_{i,j,t}$ or $\mu_{i,t}$ of good experts are much larger than those of bad experts. This characteristic makes the strategies pay more attention on 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, $\gamma_{i,j,t}$ and $\mu_{i,t}$ depends on unknown $\hat{H}_{i,t-1}$, which is defined in Equation (5). To solve this issue, we assume 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, we can verify that the cumulative loss of one expert on the unqueried examples is propositional to that on the queried examples. Formally, if there are m queried examples and n unqueried examples until time t, then we have $\widehat{H}_{i,t-1} = \frac{n}{m}\widehat{L}_{i,t-1}$.

Based on the above analysis of the confidence conditions, we can now present the general framework of active forecasters for OAL with expert advice, which is summarized in Algorithm 1.

As shown in Algorithm 1, at each round, after receiving an input instance $\mathbf{x}_t$, we compute the prediction by each expert in the pool, i.e., $f_i(\mathbf{x}_t)$. Then, we examine if the *confidence condition* is satisfied. If so, we will skip the label request of $\mathbf{x}_t$; otherwise, the learner will request the true label of $\mathbf{x}_t$ from the oracle.

6 EXPERIMENTAL RESULTS

In this section, we evaluate the empirical performance of the proposed four active forecasters for OAL with expert advice tasks.

Table 1. Datasets Used in the Experiments

Dataset	# classes	# instances	# features	Sources
a8a	2	32,561	123	LIBSVM
Mushrooms	2	8,124	112	UCI
Spambase	2	4,601	57	UCI
svmguide1	2	7,089	4	LIBSVM
w8a	2	64,700	300	LIBSVM

6.1 Experts and Compared Algorithms

To construct experts for an online sequential prediction task, we choose to build the pool of experts by adopting five well-known online learning algorithms (Dredze et al. 2008; Orabona and Crammer 2010), which include the following (implemented as in Hoi et al. (2014)):

—PERCEPTRON: the classical Perceptron algorithm (Rosenblatt 1958).
—ROMMA: the Relaxed Online Maximum Margin Algorithm (Li and Long 1999).
—$\text{ALMA}_p(\alpha)$: the Approximate Maximal Margin Algorithm (Gentile 2001).
—PA: the PA online learning algorithm (Crammer et al. 2006).
—AROW: the Adaptive Regularization Of Weights algorithm (Crammer et al. 2009).

All the above 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 (Gentile 2001). The parameter C in PA is set to 5, and the parameter γ is set to 1 for AROW.

We compare the four proposed active forecasters (AEWAF and AGF, RAEWAF and RAGF) with the two regular forecasters (EWAF and GF) algorithm and their random variants as well, which are listed below:

—EWAF: the Exponentially Weighted Forecaster (Cesa-Bianchi and Lugosi 2006), which queries all of the instances.
—GF: the GF algorithm (Cesa-Bianchi and Lugosi 2006), 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 GF algorithm, a variant of GF, which will randomly select the instances to query according to a uniform distribution.
—**AEWAF:** the proposed Active Exponentially Weighted Forecaster algorithm shown in Theorem 1.
—**AGF:** the proposed AGF algorithm shown in Theorem 2.
—**RAEWAF:** the proposed RAEWAF shown in Theorem 3.
—**RAGF:** the proposed RAGF algorithm shown in Theorem 4.

6.2 Experimental Testbed and Setup

To evaluate the performance, we conduct experiments on a variety of benchmark datasets from web machine-learning repositories. Table 1 shows the details of datasets used in our experiments. All of them can be downloaded from LIBSVM website[1] and UCI machine learning repository.[2]

[1] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/.
[2] http://www.ics.uci.edu/~mlearn/MLRepository.html.

Table 2. Evaluation of Proposed Robust Active Learning Algorithms on the **Normal** Setting, Where All Experts are Comparably Reliable

Data	Algorithm	Regret	Query (%)	Time (s)	Algorithm	Regret	Query (%)	Time (s)
a8a	REWAF	1.303 ± 0.108	19.587 ± 0.396	0.160	RGF	1.275 ± 0.089	20.029 ± 0.208	0.240
	AEWAF	0.764 ± 0.059	20.210 ± 0.001	0.205	AGF	0.791 ± 0.062	20.370 ± 0.235	0.300
	RAEWAF	**0.638± 0.087**	19.667 ± 0.348	0.411	RAGF	**0.593±0.077**	19.930 ± 0.280	0.302
Mushrooms	REWAF	1.324 ± 0.042	19.269 ± 0.446	0.060	RGF	1.324 ± 0.042	19.269 ± 0.446	0.060
	AEWAF	0.512 ± 0.001	20.126 ± 0.001	0.087	AGF	0.512 ± 0.001	20.126 ± 0.001	0.087
	RAEWAF	**0.511 ± 0.001**	19.414 ± 0.358	0.172	RAGF	**0.511 ± 0.001**	19.414 ± 0.358	0.172
Spambase	REWAF	2.889 ± 0.188	19.689 ± 0.697	0.029	RGF	2.899 ± 0.207	19.993 ± 0.670	0.039
	AEWAF	0.844 ± 0.030	19.859 ± 0.001	0.029	AGF	0.922 ± 0.041	19.969 ± 0.116	0.038
	RAEWAF	**0.787 ± 0.010**	19.796 ± 0.1.2	0.061	RAGF	**0.785 ± 0.009**	19.955 ± 0.128	0.046
svmguide1	REWAF	2.922 ± 0.197	20.034 ± 0.005	0.061	RGF	2.940 ± 0.259	19.956 ± 0.004	0.131
	AEWAF	0.940 ± 0.045	20.046 ± 0.001	0.11	AGF	1.653 ± 0.156	20.331 ± 0.004	0.178
	RAEWAF	**0.615 ± 0.001**	20.070 ± 0.012	0.288	RAGF	**0.634 ± 0.001**	15.810 ± 0.008	0.199
w8a	REWAF	0.734 ± 0.016	19.933 ± 0.190	1.291	RGF	0.733 ± 0.021	20.141 ± 0.1.2	1.741
	AEWAF	0.241 ± 0.003	19.886 ± 0.001	1.362	AGF	0.321 ± 0.013	20.120 ± 0.068	1.888
	RAEWAF	**0.200 ± 0.001**	19.954 ± 0.202	1.159	RAGF	**0.200 ± 0.001**	20.156 ± 0.289	1.892

The bold face values indicate the best performance on each dataset.

To make fair comparisons, all the compared forecasters adopt the same setup. The learning rate η is set to $\sqrt{8 \ln N/T}$ (N is the number of the experts and T is the number of instances of each dataset), for all the datasets and forecasters. The querying ratio for obtaining true label instances made by the two random algorithms (REWAF and RGF) are set in the range of $[0, 1]$.

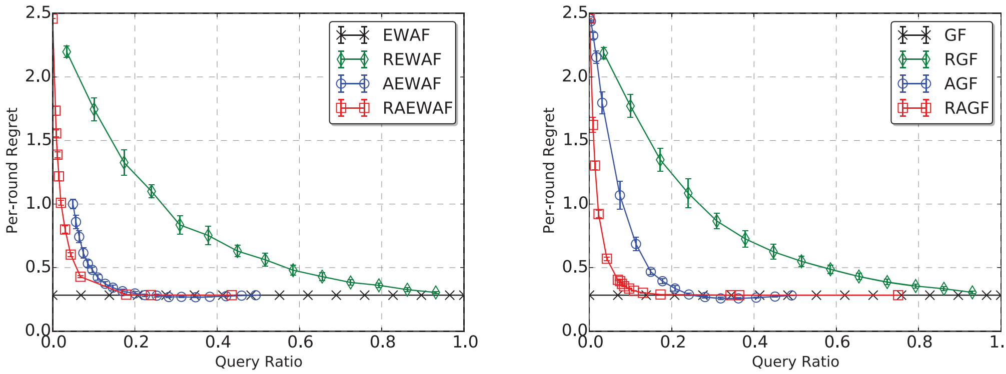
Each dataset is randomly divided into two subsets: a training set consisting of 20% of the entire data for training the experts algorithms; and a test set consisting of the remaining data for learning the forecasters. The five experts algorithms are applied on the training set to simulate the five reliable expert functions $\mathbf{u}_i \in \mathbb{R}^d,\ i \in [5]$, where d is the dimension of the instance. To satisfy the assumptions, we adopt $f_i(\mathbf{x}) = \pi_{[0,1]}(\mathbf{u}_i^\top \mathbf{x} + 0.5)$ as the expert functions. Then we test the forecasters on the test set. All the test experiments were conducted over 20 runs of different random permutations on the test dataset and reported by averaging these 20 runs. For performance metric, we evaluate the forecasters by measuring the *per-round regret, varied query ratio, varied number of noisy experts, and the running time cost* based on the absolute loss defined in Equation (1).

6.3 Evaluation on Normal Setting

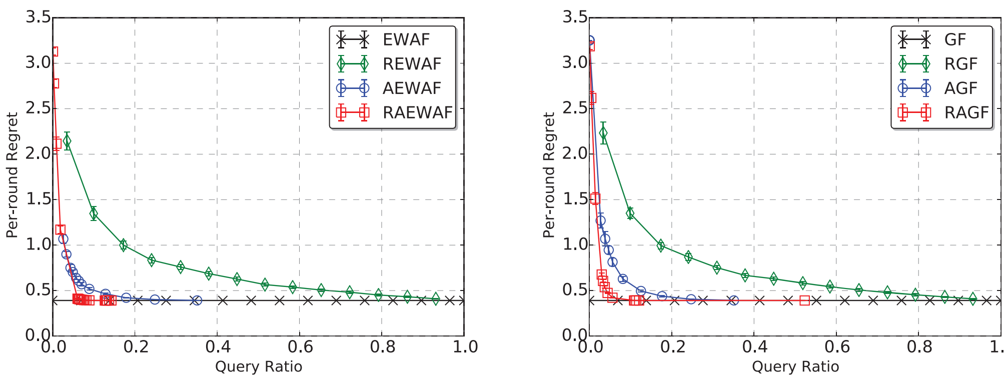
In this section, we evaluate the performance of the proposed active forecasters in the normal setting, where the experts pool are only consisted by the five reliable online learners. We first evaluate the performance with fixed query ratio, and then evaluate the forecasters with varied query ratio.

6.3.1 Fixed Query Ratio. Table 2 summarizes the average performance of the forecasters in the normal setting on all datasets, where the query ratios are roughly kept similar as it would be very difficulty to keep them exact same. From the experimental results, we can draw several observations as follows.

First, compared to the regular forecasters EWAF and GF, the proposed active forecasters (AEWAF, AGF, RAEWAF, and RAGF) could achieve comparable performance by querying a small fraction (less than 20%) of true labels. This validates the effectiveness of the proposed *confidence conditions* (Theorems 1–4) in saving labeling cost while maintaining comparable performance. This also proves that the proposed *confidence conditions* are capable to query informative instances by considering the difference of advices provided by reliable experts.



a8a



mushrooms

Fig. 1. Evaluation with varied querying ratio in **normal** setting, where all experts are comparably reliable.

Second, compared with the random forecasters REWAF and RGF, the proposed active forecasters AEWAF and AGF can achieve statistically lower per-round regret rates, respectively. This validates the effectiveness of the proposed query strategies by considering the difference of reliable experts' advices. By further incorporating the expert cumulative performance information, the proposed robust active forecasters RAEWAF and RAGF can achieve the lowest regret and variance. This verify the importance of considering the experts' cumulative performances.

Finally, our proposed forecasters cost more time when comparing the regular algorithms (EWAF and GF) and the random algorithms (REWAF and RGF) due to the computation of *confidence conditions*. Compared to AEWAF and AGF, the robust active forecasters RAEWAF and RAGF cost more time due to computation of ($\gamma_{i,j,t}$ and $\mu_{i,t}$) in Theorems 3 and 4. However, the extra time cost could be ignored considering the efficiency of the online learning scheme.

6.3.2 Varied Query Ratio. In this section, we investigate the performance of proposed active forecasters with varied query ratio in the normal scenario, where the trained experts generally share competing performance. Figure 1 presents the per-round regret w.r.t. varied query ratios on *a8a* and *mushrooms* datasets. Figure 4 shows the results on the other datasets with same experiment setup. For each dataset, left figure shows the results with algorithms based on EWAF forecaster and right figure shows the results of algorithms based on GF forecaster. From the figures, we can made several observations as follows.

Table 3. Evaluation of the Proposed Algorithms *RAEWAF* and *RAGF* on the **Noise-Added** Setting, Where Query Ratio is Fixed Around 20%

Data	Algorithm	Regret	Query(%)	Time (s)	Algorithm	Regret	Query(%)	Time (s)
a8a	REWAF	1.851 ± 0.140	19.493 ± 0.341	0.166	RGF	1.818 ± 0.098	19.786 ± 0.299	0.254
	AEWAF	2.228 ± 0.196	19.968 ± 0.247	0.219	AGF	11.043 ± 0.225	19.955 ± 0.526	0.324
	RAEWAF	**0.537 ± 0.107**	19.626 ± 0.389	0.656	RAGF	**0.508 ± 0.082**	19.768 ± 0.285	0.333
Mushrooms	REWAF	2.240 ± 0.096	19.652 ± 0.536	0.065	RGF	2.317 ± 0.088	19.161 ± 0.523	0.100
	AEWAF	1.863 ± 0.145	20.156 ± 0.408	0.090	AGF	15.116 ± 4.330	22.132 ± 7.874	0.126
	RAEWAF	**0.636 ± 0.017**	19.650 ± 0.452	0.268	RAGF	**0.633 ± 0.016**	19.343 ± 0.465	0.129
Spambase	REWAF	2.639 ± 0.235	19.997 ± 0.619	0.030	RGF	2.650 ± 0.212	20.367 ± 0.679	0.040
	AEWAF	4.436 ± 0.315	19.999 ± 0.113	0.034	AGF	10.906 ± 0.824	20.274 ± 1.586	0.044
	RAEWAF	**1.060 ± 0.087**	20.005 ± 0.131	0.097	RAGF	**1.041 ± 0.098**	20.276 ± 1.556	0.052
svmguide1	REWAF	2.084 ± 0.113	20.282 ± 0.006	0.065	RGF	2.150 ± 0.084	19.926 ± 0.004	0.135
	AEWAF	**1.600 ± 0.099**	20.271 ± 0.004	0.114	AGF	4.146 ± 0.507	25.843 ± 0.020	0.184
	RAEWAF	1.621 ± 0.132	15.794 ± 0.017	0.469	RAGF	**1.528 ± 0.106**	17.779 ± 0.015	0.22
w8a	REWAF	0.804 ± 0.031	19.743 ± 0.158	0.973	RGF	0.813 ± 0.029	19.744 ± 0.111	1.281
	AEWAF	0.691 ± 0.034	20.439 ± 0.162	1.142	AGF	13.600 ± 3.066	20.968 ± 5.185	1.534
	RAEWAF	**0.214 ± 0.005**	19.781 ± 0.198	2.670	RAGF	**0.213 ± 0.006**	19.741 ± 0.194	1.557

The bold face values indicate the best performance on each dataset.

First, similar observations could be made as in the Table 2. With similar query ratio (x-axis), the proposed active forecasters could achieve the best performance (y-axis) on most of the datasets. The proposed robust active forecasters, RAEWAF and RAGF, can consistently outperform the random forecasters REWAF and RGF, respectively. This again validates the effectiveness of the proposed *confidence condition* in identifying informative instances.

Second, by increasing the query ratio, the per-round regret consistently decreases, and this indicates that more training instances in general will improve the performance of the forecasters. Furthermore, the regrets of the proposed active forecasters decrease with an exponential speed when query ratio increases. This further confirms the effectiveness of the proposed active forecasters. More importantly, the regrets of proposed robust active forecasters RAEWAF and RAGF decrease faster than AEWAF and AGF, respectively. With less than 20% query ratio, RAEWAF and RAGF could achieve the lowest regret achieved by the regular forecasters EWAF and GF with 100% query ratio.

In sum, in the normal setting, all of the proposed active forecasters could achieve the best performance compared to the random forecasters REWAF and RGF. With less than 20% of query ratio, they also can achieve comparable performance with EWAF and GF forecasters which query all of the instances. All these observations validate that the proposed forecasters are effective in saving labeling cost while maintaining comparable performance. In the next section, we will further evaluate the proposed active forecasters in noisy scenarios.

6.4 Evaluation on Noisy Setting

In this subsection, we investigate the performance of the proposed forecasters in noisy scenarios, where two extra random noisy experts are added to the pool of experts (there are five reliable experts and two noisy experts).

6.4.1 Fixed Query Ratio. We first evaluate all the forecasters when the query ratio is fixed. Table 3 shows the results of all forecasters in terms of query ratio, per-round regret, and time cost. Based on the results, we can make several observations.

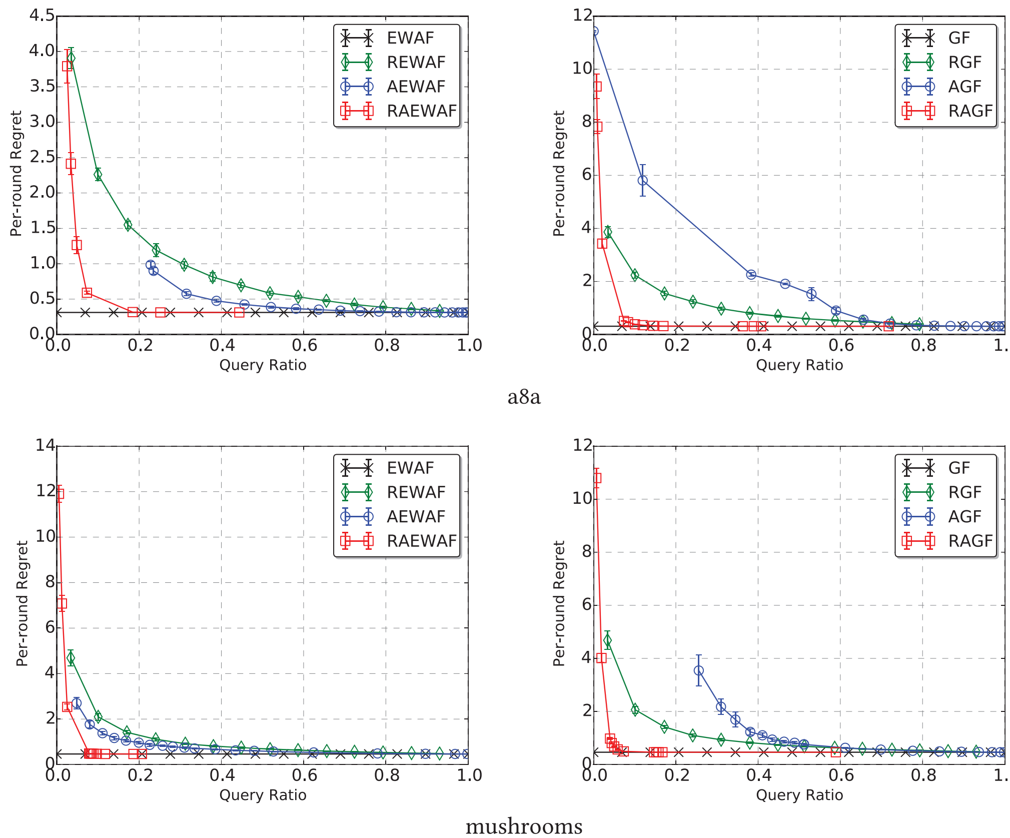
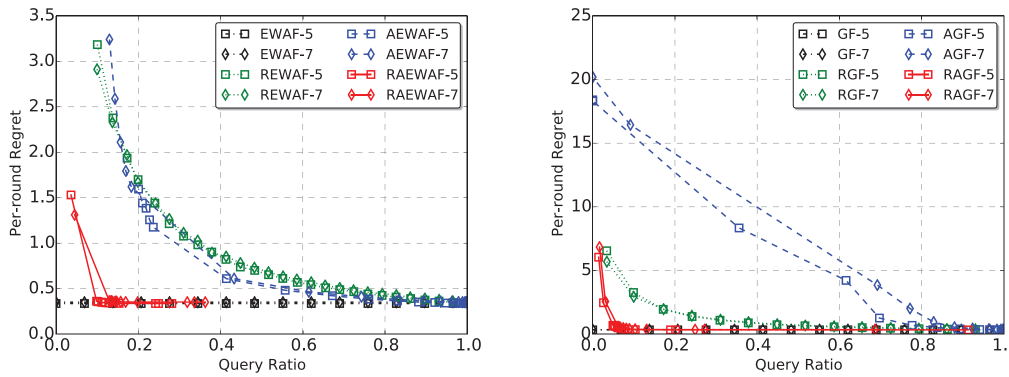


Fig. 2. Evaluation with varied querying ratio in **noisy** setting, where two extral noisy experts are added.

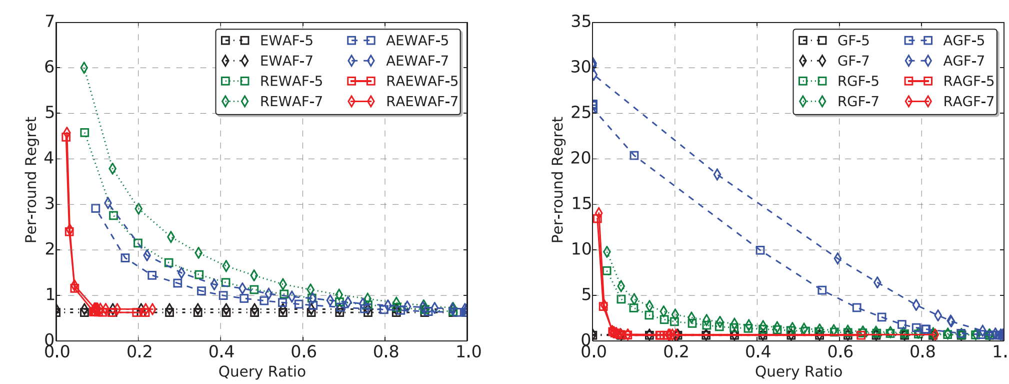
First, the proposed robust active forecasters RAEWAF and RAGF can still achieve the best performance even there is noisy experts in the pool. Both RAEWAF and RAGF could statistically achieve the smallest regrets compared to their random variants (REWAF and RGF) and the other active forecasters (AEWAF and AGF). What is more, with less than 20% query ratio, RAEWAF and RAGF could achieve similar performance as the fully supervised forecasters EWAF and GF, respectively. These observations first confirm the importance of considering the cumulative performance of each expert, and also validate the effectiveness of the proposed robust active forecasters in reducing label cost while achieving comparable performance.

Second, the proposed algorithm AEWAF can still outperform its random variant REWAF, but its performance has been largely reduced when compared to its performance in the normal scenario shown in Figure 1. This observation would be more clear in the following sections. More importantly, the proposed algorithm AGF even performs worse than its random variant RGF. These observations confirm our analysis that the *confidence conditions* considering experts advices alone would be easily misled by noisy experts.

6.4.2 Varied Query Ratio. Figure 2 shows the per-round regret w.r.t. varied query ratio on two datasets in this noisy-experts added scenario. Figure 5 shows the results on the other datasets with same setting.



a8a



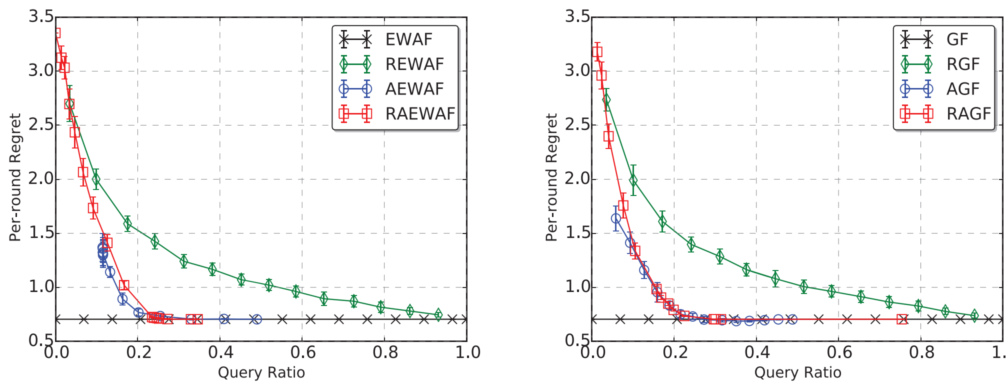
mushrooms

Fig. 3. Evaluation with varied querying ratio in **noisy** setting, where {5, 7} extral noisy experts are added.

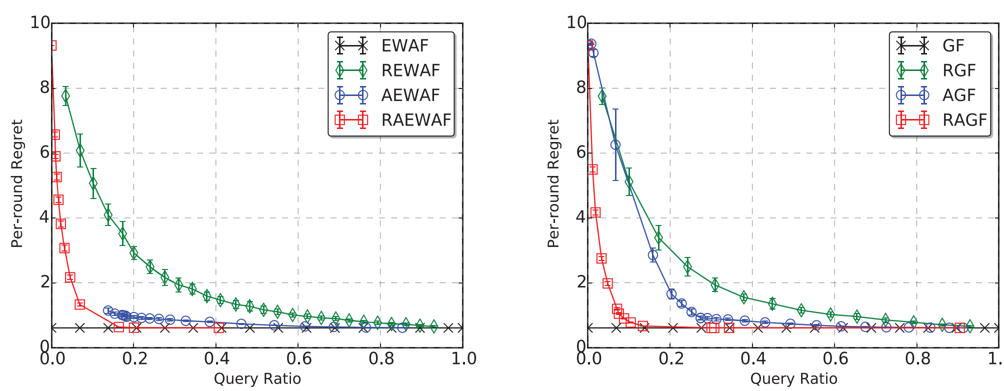
The first observation we can make is that the proposed two robust active forecasters RAEWAF and RAGF can consistently outperform the first two proposed active forecasters (AEWAF and AGF) and their random algorithms (REWAF and RGF), respectively. This observation first confirms that it is necessary to consider both the difference of advices and reliability of each expert, also suggests that the robust forecasters 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 the less number of queried instances to obtain the smaller regret.

The second observation we can make is that the performances of both AEWAF and AGF forecasters are largely reduced compared to the ones in normal setting. The random forecasters REWAF and RGF could outperform them on nearly all of the datasets, respectively. Especially for the AGF, the aggressive characteristic makes it perform worse than RGF. These observations may indicate that the active forecasters which only consider the difference of advice may consider the advice from noisy experts more than the one from reliable experts, which are also consistent with our analysis in Section 3 and confirm the importance of considering the experts cumulative performance.

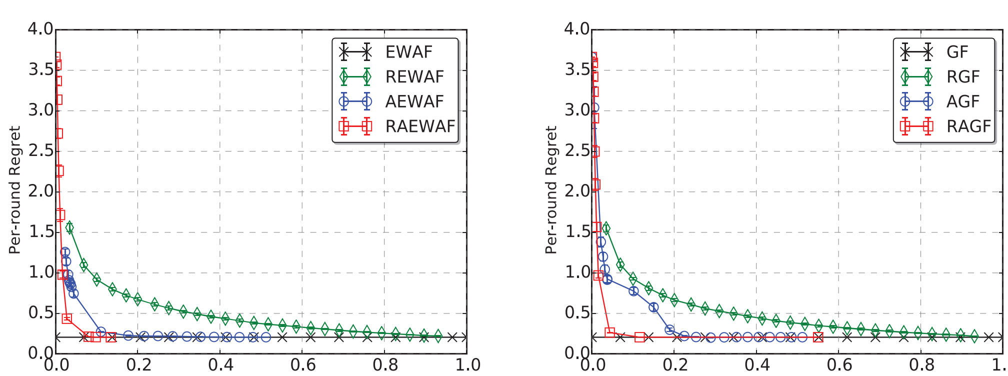
6.4.3 Evaluation on Varied Number of Noisy Experts. In this section, we evaluate the algorithms in different noisy scenarios. Figure 3 shows the performance of proposed forecasters on two datasets, and Figure 6 shows the results on the other datasets. For the legend in the figures,



spambase



svmguide1



w8a

Fig. 4. Evaluation with varied querying ratio in **normal** setting, where all experts are comparably reliable.

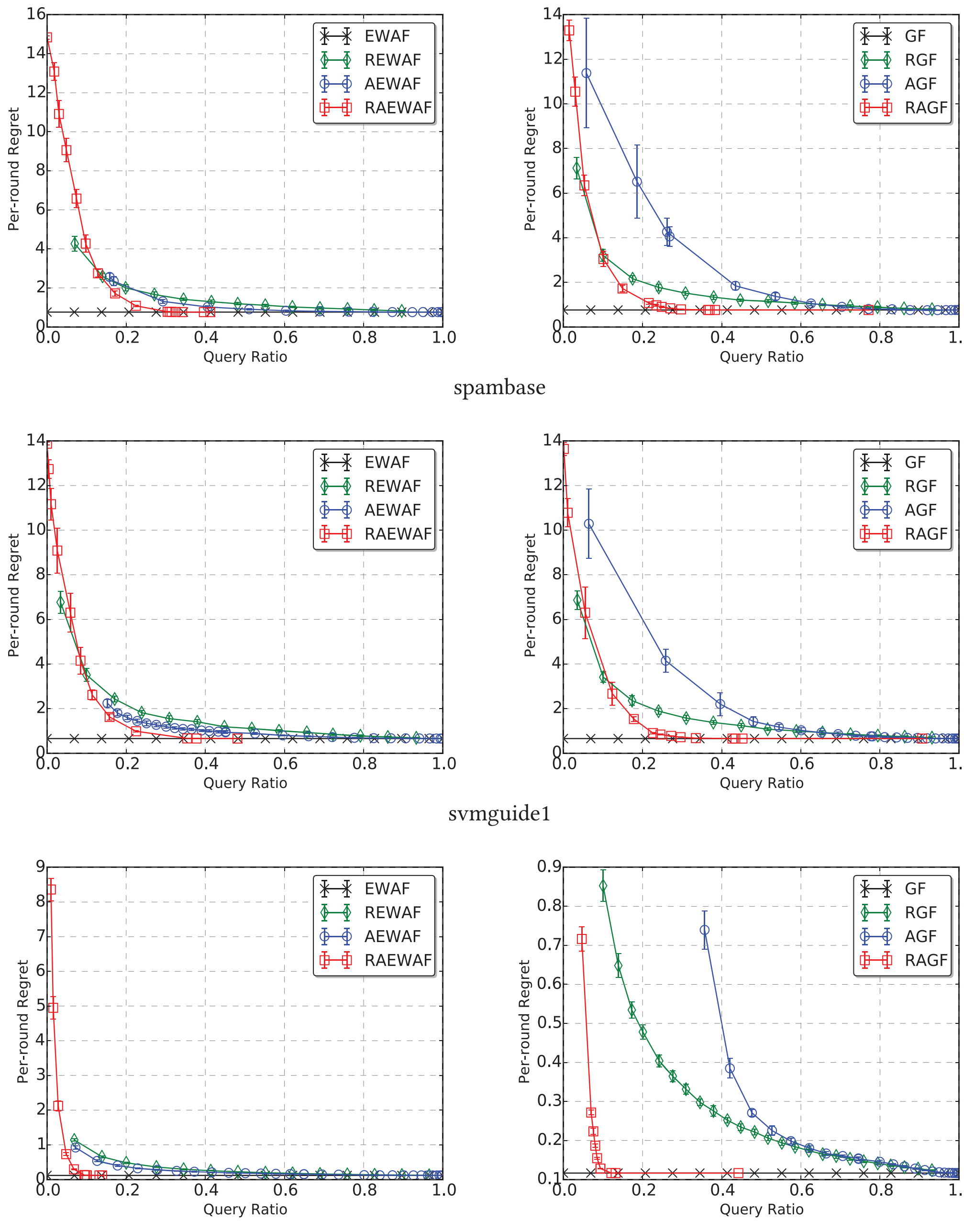
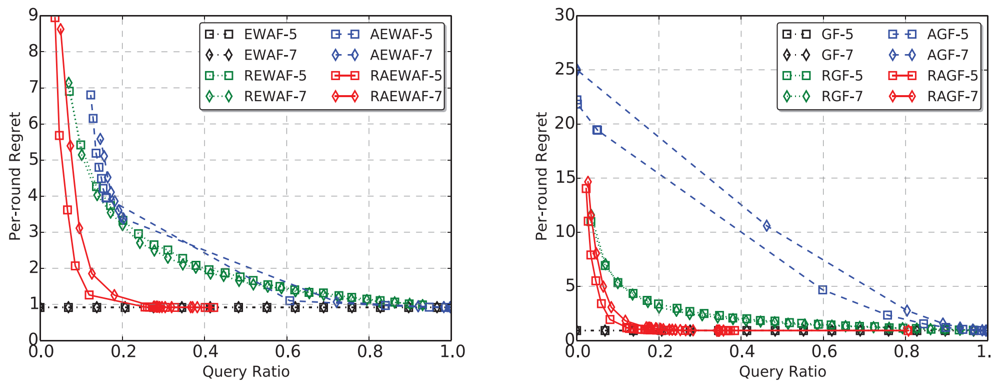
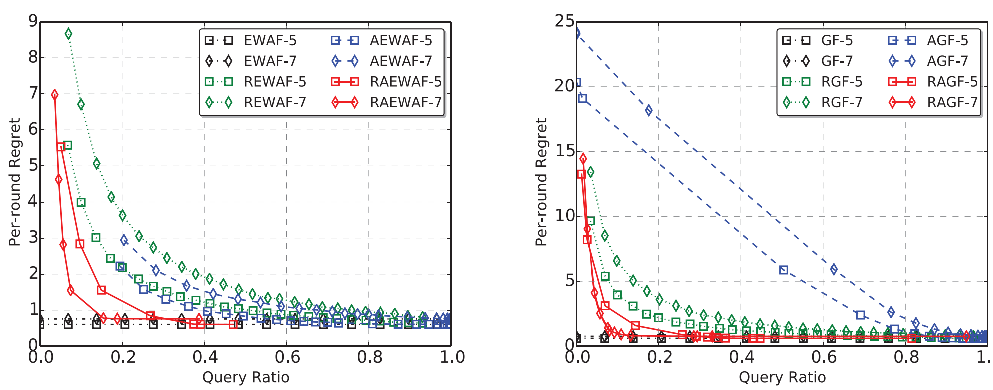


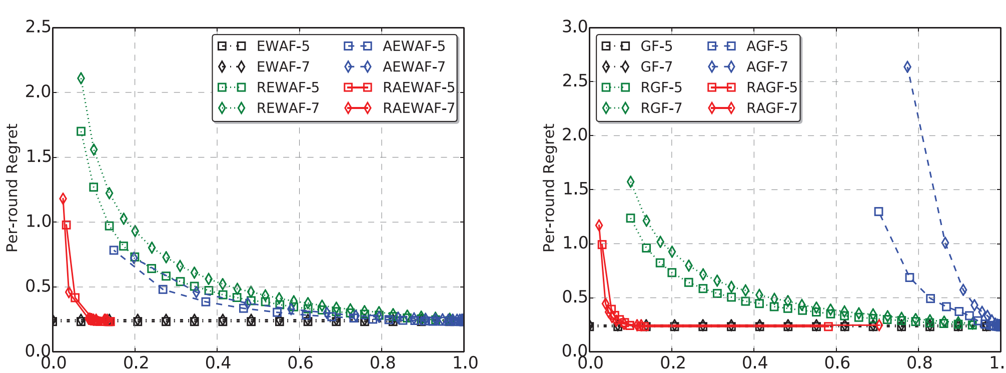
Fig. 5. Evaluation with varied querying ratio in **noisy** setting, where two extral noisy experts are added.



spambase



svmguide1



w8a

Fig. 6. Evaluation with varied querying ratio in **noisy** setting, where $\{5, 7\}$ extra noisy experts are added.

the digital number (5,7) corresponds to different number of noisy experts added. As there are totally five reliable experts, these numbers of noisy experts could simulate all the noisy scenarios. For example, setting with $\{5, 7\}$ noisy experts added into the pool can simulate the scenarios where number of noisy experts are equal, larger than the number of reliable experts, respectively. From the figures and previous experiments, we can make several observations. First, as the number of noisy experts increases, it becomes challenging to learn the tasks. On all datasets, the per-round regret of all forecasters increases when the number of noisy experts is increasing. The convergence speed of random forecasters (REWAF and RGF) and active forecasters AEWAF and AGF decreases as the number of noisy experts increases. Even for the regular forecasters EWAF and GF which query 100% of true labels, their per-round regret also increases when number of noisy experts increases. These observations indicate that adding more noisy experts generally increases the difficulty of learning a good forecaster.

Second, consistent with previous observations, the proposed robust active forecasters RAEWAF and RAGF can robustly achieve the best performance on all noisy scenarios. On all of the datasets, we can observe that RAEWAF and RAGF could converge fastest with similar query ratio no matter how many noisy experts are added, when comparing the random forecasters and active forecasters AEWAF and AGF. On the contrary, the convergence speed of AEWAF and AGF decreases fast, and they require more queried instances to converge to the lowest regret. Even worse, AGF performs worse than the random forecaster RGF on all dataset. One possible reason is that the AGF forecaster usually queries the true label on which noisy experts disagree with the forecaster's predication $\bar{p}_t$.

In sum, these observations on noisy setting confirm our argument in Section 5. When there are no noisy experts, the proposed active forecasters AEWAF and AGF, which only consider the difference of advices but do not care about who give the advices, can be used to decide when to query the true label. However, when there are noisy experts in the pool, it would be better to also consider the reliability of each expert whose advice is considered in the *confidence conditions*.

7 CONCLUSION

This article addressed a new problem of active learning with expert advice for online sequential prediction tasks. We proposed a new framework of OAL with expert advice, specifically, we first proposed two active learning approaches by considering expert advice, and then proposed two robust active forecasters by incorporating the cumulative performance of expert. We have conducted an extensive set of experiments both in normal and noisy setting to evaluate the efficacy of the proposed algorithms. Promising empirical results validate the effectiveness of our technique.

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.

REFERENCES

Jacob D. Abernethy, Peter L. Bartlett, and Alexander Rakhlin. 2007. Multitask learning with expert advice. In *Proceedings of the 20th Annual Conference on Learning Theory (COLT'07).* San Diego, CA. 484–498.

Martin Aleksandrov, Haris Aziz, Serge Gaspers, and Toby Walsh. 2015. Online fair division: Analysing a food bank problem. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI'15).* AAAI Press, Buenos Aires, Argentina, 2540–2546.

Kareem Amin, Satyen Kale, Gerald Tesauro, and Deepak S. Turaga. 2015. Budgeted prediction with expert advice. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence.* Austin, Texas. 2490–2496.

Les E. Atlas, David A. Cohn, and Richard E. Ladner. 1990. Training connectionist networks with queries and selective sampling. In *Advances in Neural Information Processing Systems.* 566–573.

H. D. Block. 1962. The Perceptron: A model for brain functioning. I. *Reviews of Modern Physics* 34, 1 (1962), 123–135.

Olivier Bousquet and Manfred K. Warmuth. 2002. Tracking a small set of experts by mixing past posteriors. *Journal of Machine Learning Research* 3 (2002), 363–396.

Nicolò Cesa-Bianchi, Alex Conconi, and Claudio Gentile. 2005. A second-order Perceptron algorithm. *SIAM Journal on Computing* 34, 3 (Jan. 2005), 640–668.

Nicolò Cesa-Bianchi, Yoav Freund, David Haussler, David P. Helmbold, Robert E. Schapire, and Manfred K. Warmuth. 1997. How to use expert advice. *Journal of the Association for Computing Machinery* 44, 3 (May 1997), 427–485.

Nicolò Cesa-Bianchi, Claudio Gentile, and Luca Zaniboni. 2006. Worst-case analysis of selective sampling for linear classification. *The Journal of Machine Learning Research* 7 (Dec. 2006), 1205–1230.

N. Cesa-Bianchi and G. Lugosi. 2006. *Prediction, Learning, and Games*. Cambridge University Press.

K. Crammer, O. Dekel, Joseph Keshet, and Shai Shalev-shwartz. 2006. Online Passive-Aggressive algorithms. *The Journal of Machine Learning* 7 (2006), 551–585.

Koby Crammer, Alex Kulesza, and Mark Dredze. 2009. Adaptive regularization of weight vectors. In *Proceedings of the 22nd International Conference on Neural Information Processing Systems*. 414–422.

Koby Crammer, Alex Kulesza, and Mark Dredze. 2013. Adaptive regularization of weight vectors. *Machine Learning* 91, 2 (March 2013), 155–187.

Mark Dredze, Koby Crammer, and Fernando Pereira. 2008. Confidence-weighted linear classification. In *Proceedings of the 25th International Conference on Machine Learning*. ACM, 264–271.

Dean P. Foster and Rakesh V. Vohra. 1993. A randomization rule for selecting forecasts. *Operations Research* 41, 4 (1993), 704–709.

Yoav Freund and Yishay Mansour. 1997. Learning under persistent drift. In *Proceedings of the European Conference on Computational Learning Theory*. 109–118.

Yoav Freund and Robert E. Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences* 55, 1 (1997), 119–139.

Kaito Fujii and Hisashi Kashima. 2016. Budgeted stream-based active learning via adaptive submodular maximization. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*. D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Eds.). 514–522.

Claudio Gentile. 2001. A new approximate maximal margin classification algorithm. *Journal of Machine Learning Research* 2 (2001), 213–242.

Xiaojie Guo. 2015. Online robust low rank matrix recovery. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI'15)*. AAAI Press, Buenos Aires, Argentina, 3540–3546.

Yuhong Guo and Dale Schuurmans. 2008. Discriminative batch mode active learning. In *Proceedings of the International Conference on Neural Information Processing Systems*. 593–600.

Steve Hanneke. 2012. Activized learning: Transforming passive to active with improved label complexity ∗. *Journal of Machine Learning Research* 13 (2012), 1469–1587.

Steve Hanneke. 2014. Theory of disagreement-based active learning. *Foundations and Trends® in Machine Learning* 7, 2–3 (2014), 131–309.

Steve Hanneke. 2016. The optimal sample complexity of PAC learning. *Journal of Machine Learning Research* 17, 38 (2016), 1–15.

Steve Hanneke and Liu Yang. 2015. Minimax analysis of active learning. *Journal of Machine Learning Research* 16 (2015), 3487–3602.

Shuji Hao, Jing Lu, Peilin Zhao, Chi Zhang, Steven C. H. Hoi, and Chunyan Miao. 2017a. Second-order online active learning and its applications. *IEEE Transactions on Knowledge and Data Engineering* 30 (2017), 1338–1351.

Shuji Hao, Peilin Zhao, Steven C. H. Hoi, and Chunyan Miao. 2015. Learning relative similarity from data streams: Active online learning approaches. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM'15)*. ACM, 1181–1190.

Shuji Hao, Peilin Zhao, Yong Liu, Steven C. H. Hoi, and Chunyan Miao. 2017b. Online multi-task relative similarity learning. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17)*.

Shuji Hao, Peilin Zhao, Jing Lu, Steven C. H. Hoi, Chunyan Miao, and Chi Zhang. 2016. SOAL: Second-order online active learning. In *Proceedings of the 16th International Conference on Data Mining (ICDM'16)*. IEEE, 931–936.

David Haussler, Jyrki Kivinen, and Manfred K. Warmuth. 1995. Tight worst-case loss bounds for predicting with expert advice. In *Proceedings of the European Conference on Computational Learning Theory*. Springer, 69–83.

Keiichiro Hayakawa, Enrico H. Gerding, Sebastian Stein, and Takahiro Shiga. 2015. Online mechanisms for charging electric vehicles in settings with varying marginal electricity costs. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI'15)*. AAAI Press, Buenos Aires, Argentina, 2610–2616.

Mark Herbster and Manfred K. Warmuth. 1998. Tracking the best expert. *Machine Learning* 32, 2 (1998), 151–178.

Steven C. H. Hoi, Doyen Sahoo, Jing Lu, and Peilin Zhao. 2018. Online learning: A comprehensive survey. arXiv:1802.02871.

Steven C. H. Hoi, Jialei Wang, and Peilin Zhao. 2014. LIBOL: A library for online learning algorithms. *Journal of Machine Learning Research* 15, 1 (2014), 495–499.
Fatemeh Jahedpari. 2015. Artificial prediction markets for online prediction. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI'15)*. AAAI Press, Buenos Aires, Argentina, 4371–4372.
J. Zico Kolter and Marcus A. Maloof. 2007. Dynamic weighted majority: An ensemble method for drifting concepts. *The Journal of Machine Learning Research* 8 (2007), 2755–2790.
Guangxia Li, Steven C. H. Hoi, Kuiyu Chang, and Ramesh Jain. 2010. Micro-blogging sentiment detection by collaborative online learning. In *Proceedings of the 10th International Conference on Data Mining (ICDM'10)*. IEEE, 893–898.
Yi Li and Philip M. Long. 1999. The relaxed online maximum margin algorithm. In *Proceedings of the Neural Information Processing Systems (NIPS'99)*. 498–504.
Yi Li and Philip M. Long. 2002. The relaxed online maximum margin algorithm. *Machine Learning* 46, 1–3 (2002), 361–387.
Nick Littlestone and Manfred K. Warmuth. 1994. The weighted majority algorithm. *Information and Computation* 108, 2 (1994), 212–261.
Jing Lu, Zhao Peilin, and C. H. Steven Hoi. 2014. Online Passive-Aggressive active learning and its applications. In *Proceedings of the Asian Conference on Machine Learning Research (ACML'14)*.
Jing Lu, Zhao Peilin, and C. H. Steven Hoi. 2016. Online Passive-Aggressive active learning and its applications. 103, 2 (2016), 141–183.
Mehryar Mohri and Afshin Rostamizadeh. 2013. Perceptron mistake bounds. arxiv:1305.0208v2.
Francesco Orabona and Koby Crammer. 2010. New adaptive algorithms for online classification. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems*. 1840–1848.
F Rosenblatt. 1958. The Perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review* 65, 6 (Nov. 1958), 386–408.
Paul Ruvolo and Eric Eaton. 2014. Online multi-task learning via sparse dictionary optimization. In *Proceedings of the 28th Conference on Artificial Intelligence*. AAAI Press, Québec City, Québec, Canada, 2062–2068.
Burr Settles. 2010. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 6, 1 (Jun. 2010), 1–114.
Victor S. Sheng, Foster Provost, and Panagiotis G. Ipeirotis. 2008. Get another label? Improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 614–622.
S. Tong. 2001. *Active Learning: Theory and Applications*. Ph.D. Dissertation. Stanford University.
Simon Tong and Daphne Koller. 2002. Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research* 2, 3/1 (2002), 45–66.
Joel Veness, Marcus Hutter, Laurent Orseau, and Marc G. Bellemare. 2015. Online learning of k-CNF Boolean functions. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI'15)*, AAAI Press, Buenos Aires, Argentina, 3865–3873.
Volodimir G. Vovk. 1990. Aggregating strategies. In *Proceedings of the 3rd Annual Workshop on Computational Learning Theory (COLT'90)*. 371–386.
Jialei Wang, Steven CH Hoi, Peilin Zhao, and Zhi-Yong Liu. 2013. Online multi-task collaborative filtering for on-the-fly recommender systems. In *Proceedings of the 7th Conference on Recommender Systems*. ACM, 237–244.
Jialei Wang, Peilin Zhao, and Steven C. Hoi. 2012. Exact soft confidence-weighted learning. In *Proceedings of the 29th International Conference on Machine Learning (ICML'12)*. J. Langford and J. Pineau (Eds.), ACM, New York, NY, 121–128.
Jialei Wang, Peilin Zhao, and Steven CH Hoi. 2014. Cost-sensitive online classification. *IEEE Transactions on Knowledge and Data Engineering* 26, 10 (2014), 2425–2438.
Chi Zhang, Peilin Zhao, Shuji Hao, and Steven Hoi. 2018. Distributed multi-task classification: A decentralized online learning approach. *Machine Learning* 107, 4 (2018), 727–747.
Chi Zhang, Peilin Zhao, Shuji Hao, Yeng Chai Soh, and Bu Sung Lee. 2016. ROM: A robust online multi-task learning approach. In *Proceedings of the16th International Conference on Data Mining (ICDM'16)*. IEEE, 1341–1346.
Peilin Zhao and Steven C. H. Hoi. 2013. Cost-sensitive online active learning with application to malicious URL detection. In *Proceedings of the 19th SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'13)*. ACM, New York, NY, 919. DOI: http://dx.doi.org/10.1145/2487575.2487647
Peilin Zhao, Steven C. H. Hoi, and Rong Jin. 2011a. Double updating online learning. *Journal of Machine Learning Research* 12 (2011), 1587–1615.
Peilin Zhao, Rong Jin, Tianbao Yang, and Steven C. Hoi. 2011b. Online AUC maximization. In *Proceedings of the 28th International Conference on Machine Learning (ICML'11)*. 233–240.

Received February 2017; revised March 2018; accepted March 2018