PREDICTION WITH EXPERT ADVICE: ON THE ROLE OF CONTEXTS, BANDIT FEEDBACK AND RISK-AWARENESS

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF

MASTER OF SCIENCE

IN

ELECTRICAL AND ELECTRONICAL ENGINEERING

By

Kubilay Ekşioğlu

December 2018

Prediction with Expert Advice: On the Role of Contexts, Bandit Feedback and Risk-Awareness
By Kubilay Ekşioğlu
December 2018

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Cem Tekin(Advisor)

Savaş Dayanık

Elif Vural

Approved for the Graduate School of Engineering and Science:

Ezhan Karaşan
Director of the Graduate School

To my father.

ABSTRACT

PREDICTION WITH EXPERT ADVICE: ON THE ROLE OF CONTEXTS, BANDIT FEEDBACK AND RISK-AWARENESS

Kubilay Ekşioğlu
M.S. in Electrical and Electronical Engineering
Advisor: Cem Tekin
December 2018

Along with the rapid growth in the size of data generated and collected over time, the need for developing online algorithms that can provide answers without any offline training has considerably increased. In this thesis, we consider the prediction with expert advice problem under the online learning framework. Specifically, we consider problems where experts have asymmetric information about the sample space. First, we propose an algorithm that selects a subset of the experts and makes predictions based on the advices of this subset. Then, we propose another algorithm that clusters samples in an online manner and makes predictions based on the history of observations and decisions within each cluster. Next, we consider the Safe Bandit, a variant of the Risk Aware Multi Armed Bandit, where the goal is to minimize the number of rounds in which a risky arm is chosen. Adopting mean-variance as the risk notion, we define an arm as risky if its mean-variance is higher than a given threshold. Using this, we define a new regret measure called Risk Violation Regret (RVR), which depends on the number of times risky arms are selected. Then, we propose a learning algorithm called Exploration and Exploitation with Risk Thresholds (EXERT), and prove that it achieves $O(1)$ RVR with high probability. Afterwards, we use EXERT in an expert selection problem, where each expert corresponds to a neural network with reject option. For this, we propose a method to train these neural networks and use them to evaluate the performance of EXERT in real-world datasets.

Keywords: Prediction with Expert Advice, Multi Armed Bandits, Online Learning, Neural Networks.

ÖZET

UZMAN ÖNERİLERİYLE TAHMİN: BAĞLAMLARIN, HAYDUT GERİBİLDİRİMİN VE RİSK FARKINDALIĞININ ROLÜ ÜZERİNE

Kubilay Ekşioğlu
Elektrik ve Elektronik Mühendisliği, Yüksek Lisans
Tez Danışmanı: Cem Tekin
Aralık 2018

Günlük olarak üretilen ve toplanan verinin büyüklüğü arttıkça, uzun bir eğitim süreci gerektirmeden çevrimiçi olarak çalışabilen tahmin algoritmalarına olan ihtiyaç da artmıştır. Bu tezde uzman önerisiyle tahmin problemi çevrimiçi öğrenme çerçevesinde ele alınmıştır. Daha detaylı olarak, Bölüm 3'te, bu sorun uzmanların örnek uzay hakkında asimetrik bilgiye sahip olduğu bir senaryoda ele alınmıştır. Bu senaryo için önce, uzmanların bir alt kümesini seçen ve bu alt kümenin önerilerine göre tahmin yapan bir algoritma önerilmiştir. Ardından, örnekleri çevrimiçi bir şekilde kümeleyen ve bu kümelerdeki seçimlerin ve gözlemlerin tarihçesini kullanarak tahmin yapmayı sağlayan bir algoritma önerilmiştir. Bölüm 4'te, riskli bir kolun seçilme sayısını en aza indirgemeyi amaçlayan, Riske Duyarlı Çok Kollu Haydut probleminin bir çeşidi olan Safe Bandit (Güvenli Haydut) problemi ele alınmıştır. Risk kavramı olarak ortalama-varyans seçilmiş ve ortalama-varyansı belirli bir eşikten yüksek olan kollar riskli olarak tanımlanmıştır. Riskli kolların toplam seçilme sayısı Risk İhlal Pişmanlığı (Risk Violation Regret, RVR) adında yeni bir pişmanlık kavramı olarak tanımlanmıştır. Ardından, Risk Eşikleriyle Keşif ve Faydalanma (Exploration and Exploitation with Risk Thresholds, EXERT) olarak adlandırılan bir öğrenme algoritması önerilmiş ve bu algoritmanın yüksek olasılıkla $O(1)$ RVR elde ettiği kanıtlanmıştır. EXERT algoritmasının performansı, tüm uzmanların reddetme opsiyonlu yapay sinir ağları olduğu bir uzman seçimi probleminde incelenmiş ve bu uzmanları eğitmek için bir yöntem önerilmiştir.

Anahtar sözcükler: Uzman Önerileriyle Tahmin, Çok Kollu Haydutlar, Çevrimiçi Öğrenme, Yapay Sinir Ağları.

Acknowledgement

First of all, I would like to thank my advisor Dr. Cem Tekin, for his guidance and support in supervision of this thesis. Without his patience, attention to detail and hard working attitude I would not be able to complete this work.

I would also like to thank my jury members Prof. Savaş Dayanık and Dr. Elif Vural for their time and valuable feedbacks. It was very kind of them to accept being in my jury despite of their busy schedule.

I am indebted to Zeynep Aygar for always being there for me (and especially for bringing "kelle-paça" soup when I fractured my wrist), to Eralp Turğay for eye-opening conversations, to Cem Bulucu for weird but always funny wordplays, to Safa Onur Şahin for enjoyable coffee breaks, to Ali Alp Akyol for his hospitality to a troubling housemate like me, to Anjum and Faiza Qureshi for their delicious food, to Ul Salman Hassan Dar for calming me in probably the most stressful evening in my education, to Ümitcan Şahin, Oytun Güneş, Alparslan Çelik, Andi Nika, Muhammad Nabi Yasinzai, Nima Akbarzadeh, Alireza Javanmardi, Barış Canatan, Muhammad Umar B. Niazi for friendly conversations, and to Ergün Hırlakoğlu for football nights.

Finally, I would like to thank my family, my mother Deniz Ekşioğlu and my sister Ezgi Ekşioğlu Arslan for encouraging me to pursue this degree and continuously supporting me along the way. I'm grateful for their unconditional love.

This work is partially supported by TÜBİTAK under the 2232 Scholarship Program (Project no: 116C043) and 2210-A Scholarship Program.

Contents

List of Figures

List of Tables

List of Publications

This thesis includes content from following publication:

- Kubilay Ekşioğlu, Muhammad Anjum Qureshi, and Cem Tekin. "Online classification with contextual exponential weights for disease diagnostics." Signal Processing and Communications Applications Conference (SIU), 2017 25th. IEEE, 2017.

Chapter 1

Introduction

Prediction is trying to guess the outcome of certain events. Forecasting what the price of a company's stock will be in a month, or whether it will rain tomorrow or not, can be provided as examples of prediction. In a nutshell, prediction is about making observations and deriving conclusions that were not available to us beforehand.

In classical *supervised learning*, the learner (also called the forecaster or the predictor) is given a *set* of observation and conclusion pairs, which it uses to construct a mapping from observations to outcomes. While supervised learning has a plethora of applications, this way of learning, in fact, contradicts with the natural way of learning. For instance, infants learn by observing their environment, taking actions according to their beliefs, and then, observing the feedback provided to them together with the changes in the environment (if there are any). The area of machine learning that models this process, which repeats over time, is called *reinforcement learning*. In this framework, a snapshot of the environment is called *state*, decision made by the learner is called *action*, and feedback provided to the learner is called *reward*. The state, action, reward, and next state quadruple forms the basis of all reinforcement learning problems, where the aim is to learn by interacting with the environment.

A very important area of reinforcement learning is *multi armed bandits* (MABs). MABs, also called bandits in the literature, are commonly used to model sequential decision making problems under uncertainty. In a MAB, every available action that learner can choose is modeled as a slot machine, with an underlying reward distribution that is unknown to the learner.[1] The name "one-armed bandit" originates from American slang, and is a synonym for "slot machine". In a MAB, the learner plays on a given set of slot machines sequentially over time, one at a time.

In a MAB, main objective of the learner is to maximize its total reward (with high probability or in expectation) over multiple rounds. Since the learner does not have the knowledge of the underlying reward distributions, it has to judiciously select arms based on the history of observations and feedbacks in order to maximize its long-term reward. In other words, it has to balance exploration (collecting information) and exploitation (utilizing information gained for reward maximization). Exploration can also be thought as a form of diversification, which prevents the learner getting trapped at pulling a suboptimal arm. On the other hand, exploitation can be thought as sticking with the best choice the learner has thus far, which enables the learner to accumulate high rewards. The trade-off between exploration and exploitation is the main concern of the MAB.

Since the maximum achievable cumulative reward depends on the arm reward distributions, in the MAB literature, it is customary to evaluate the performance of the learner with respect to a benchmark strategy. This strategy is usually taken to be the optimal "clairvoyant" strategy that always selects the arm with the highest expected reward. This relative performance measure is called the *regret*.

Recently, algorithms and models developed using the MAB formalism are used to solve a multitude of real-world problems. For instance, they are heavily used in content recommendation problems to maximize the user engagement [3], web

[1] This model is called the stochastic bandit [1]. There is also another model in which the rewards are generated by an adversary, which is called the adversarial (nonstochastic) bandit [2].

advertisement to maximize the click-through rate [4], expert crowd-sourcing [5] to optimally distribute tasks to different workers and clinical trials [6] to allocate patients to treatments. While some of these applications admit a global best arm, for others there may not be such an arm. Consider a scenario, where patients arrive to a clinic and the learner's task is to recommend a physician to each arriving patient with the goal of matching the patient with the best physician for that patient. In the context of online learning, these physicians can be modeled as functions that take user information as input and return a decision. Such entities are also called *experts*. Rather than recommending the best expert on average to every patient, recommending the best expert based on specific symptoms of each patient should provide a better solution. Such problems can be modeled by defining the patient as a *sample*, and the findings related to the patient as the *features* of this sample. In the literature, the features that are used to decide on which arm to pull (or expert to assign) are called the *context* of the sample and MAB models which take this information into account are called Contextual MABs. Contextual MABs (also called contextual bandits) are heavily used in recommender systems, for example, to recommend news articles [3], to select the format of the online advertisements [7], to choose the message to be published on online networks for maximizing the spread and influence [8].

For some other applications, instead of choosing a single expert at each round, the learner may choose multiple experts depending on its budget. Then, it needs to combine the predictions of these experts to reach a final conclusion. For this task, Littlestone and Warmuth [9] discuss the Weighted Majority Algorithm, which assigns weights to experts based on their performance history and then combines expert predictions using these weights. Instead of predictions, experts may also provide their *advices* reflecting the information they have for the current sample, generally as a probability distribution over possible labels. Regardless of whether experts return predictions or advices, the problems where the learner makes decisions sequentially using experts are called prediction with expert advice problems. In this thesis, we propose an algorithm that takes the contexts of the samples into account when choosing or combining the predictions or advices of the experts.

In the applications mentioned above, the main objective is to maximize the total reward collected by the learner. However, there exist many applications where the learner is interested in maximizing the expected reward as long as it can keep the variance low. The uncertainty about the reward of the chosen arm is called *risk* and models that take risk into account are called risk-aware models. In the same fashion, MAB models that choose arms using the risk information are called Risk-Aware MABs (RAMABs). While risk-neutral MABs maximize reward, RAMABs minimize a proxy to the risk model. Different risk models results in different RAMAB algorithms, some of which are discussed in Section 2.3. One of the most common risk notions is mean-variance, proposed by Markowitz [10], is widely used in finance, portfolio selection [11, 12], energy investment allocation [13], bankruptcy prevention [14], and many other fields.

In decision theory, another measure of risk prevention is to make sure that the chosen action has an expected reward higher than a predefined threshold, and this technique is called *satisficing* [15]. This approach can be thought as setting a target before the decision, and controlling whether the target is satisfied or not after the decision is made. Satisficing is especially useful if the resources are limited and a reasonable return over investments is expected by the stakeholders in a short amount of time. Reverdy et al. [16] study satisficing in the risk-neutral bandit setting by defining regret as the sum of expected suboptimality gaps of the chosen arms, where suboptimality gap of an arm is 0 if its mean reward is above the target and the distance between the threshold and mean reward of the arm otherwise. In this work, using a Bayesian framework, authors provide Upper Credible Limit (UCL) algorithms for Gaussian arm rewards. On the other hand, in this thesis we investigate satisficing on the mean-variance of the arm rewards, instead of mean. Furthermore, rather than using the distance of arm mean-variances to the threshold, we simply investigate the number of times this predefined threshold is violated.

1.1 Our Contributions

Contributions of this thesis are summarized as follows:

- In Online Contextual Expert Selection;
  - We consider a case of prediction with expert advice problem where experts have asymmetric information about the data samples.
  - We propose a variant of Weighted Average Forecaster algorithm for selecting multiple experts, called Selective Weighted Average Forecaster (Selective WAF).
  - We extend Selective WAF using contextual zooming [17] and propose Contextual Selective Weighted Average Forecaster (CS-WAF), an algorithm that adaptively create subsets of the context space and uses different weights for each set to make predictions.
  - We investigate performances of Selective WAF and CS-WAF algorithms numerically.
- In RAMAB;
  - We propose a new risk-aware multi-armed bandit problem, called the Safe Bandit.
  - We propose a new regret notion called Risk Violation Regret (RVR), which is the number of times risky arms are selected over all rounds where risky and risk-free arms are defined according to a risk threshold.
  - We propose Exploration and Exploitation using Risk Thresholds (EXERT), a new online learning algorithm that uses upper and lower confidence bounds to minimize RVR.
  - We show that the RVR of EXERT is $O(1)$ with high probability and $O(\log T)$ with expectation.
  - We investigate the performance of EXERT on both synthetic (variance minimization problem) and real-world (classifier selection with reject option problem) settings.

1.2 Organization of the Thesis

In Chapter 2, we provide a comprehensive literature review. In Chapter 3, we discuss the prediction with expert advice setting in detail. We also present two algorithms, Selective WAF, an expert selection algorithm, and CS-WAF, the adaptive contextual counterpart of Selective WAF. In Chapter 4, we introduce the Safe Bandit, a Risk Aware MAB problem where the objective is to minimize the total risk. In this chapter, we also describe an algorithm to solve the Safe Bandit, which is called EXERT. Finally, in Chapter 5, we present concluding remarks of the thesis and the ways to extend it in future.

Chapter 2

Literature Review

2.1 Stochastic (finite-armed) MAB

In the stochastic (finite-armed) MAB arm rewards are generated by an unknown stochastic process. In the very first work on the subject, Thompson [18] proposes a heuristic on how to make sequential decisions when there are two different actions (arms) with uncertain outcomes. This is followed by the works of Wald [19], Arrow et al. [20] and Robbins [21], which focus on the sequential experiment design problem, where the number of samples to be used in the experiment is not fixed beforehand, and at every round the learner has to decide whether to collect more samples or to finalize the experiment.

Lai [22] introduces a problem where the learner selects a single arm on every round and the arms' rewards are independent of other rounds and each other. This work shows that, under some conditions on the reward distributions, the minimum achievable regret is $O(\log T)$ with a constant dependent on the KL divergence between the optimum arm and suboptimal arms. Any policy that asymptotically achieves this regret bound is called an *asymptotically optimal* policy. This work also proposes policies for some specific exponential families of distributions, and shows that they are asymptotically optimal.

A large set of learning algorithms developed for MABs use upper confidence bound (UCB) based arm selection strategies. The UCB of a probability distribution is a number that is larger than all samples that can be drawn from this distribution with a certain (generally high) probability. Conversely, a lower confidence bound (LCB) defines a number, which is smaller than any sample that can be drawn from this distribution with a certain probability. While preceding works mainly focused on upper confidence bound calculation based on the whole reward sequence of arms until current round, Agrawal [23] shows that for some parametric distribution families, UCB can be calculated simply based on the mean of the arm rewards and asymptotically optimal regret can still be achieved. While constant term of the regret found in this work is not optimal in general, the usage of sample mean based policies greatly reduces the number of operations needed on every round and decreases the computational complexity significantly. Auer et al. [1] discuss a UCB based index policy for all distributions with a bounded support, and prove order optimal regret bounds. Cappé et al. [24] generalize the policy in [1] and propose two algorithms, one for bounded and one for exponential family distributions, achieving the best known regret bounds for the UCB strategy.

2.2 Contextual MAB

In a Contextual MAB, the learner observes a d-dimensional context vector at the beginning of each round. It uses this context vector along with the reward history of the arms to select an arm in the current round. As expected, the main objective in this setting is to understand the relation between the context and the arm rewards. Woodroofe [25], Sarkar [26] and Wang et al. [27], study the two-armed bandit problem where the learner observes a context vector in each round before selecting an arm.

In general, sublinear regret learning is possible in a Contextual MAB only when further assumptions are imposed on the relation between the contexts and the rewards. There are three commonly used models. The first approach is to

assume that for a single action and any two contexts, mean reward distance is bounded by the distance in the context space times a constant. This assumption is called Lipschitz condition. Lu et al. [28] discuss a formal model under Lipschitz condition, and propose an optimal algorithm that learns context information by partitioning the context space into subspaces. Slivkins [4] studies another algorithm that jointly partitions the context and action spaces into non-uniform subspaces dynamically under Lipschitz condition.

In the second family of Contextual MAB problems, it is assumed that the reward of an arm is linearly dependent on the hidden parameters of the arm and context. Chu et al. [29] study a model under such assumption, while Li et al. [3] apply this idea to the content recommendation problem and discuss empirical results.

Another category of the contextual MAB is where the contexts and arm rewards are drawn from an unknown joint distribution. For this problem Langford and Zhang [30] propose the Epoch-Greedy algorithm, and Dudik et al. [31] introduce a more efficient algorithm with $O(\sqrt{T})$ regret. Agarwal et al. [32] improve the work in [31] with better constant terms and a simpler algorithm.

2.3 Risk Aware MAB

RAMABs are formed by introducing the risk notion to the stochastic MAB. While regret in the risk-neutral bandit setting is dependent on reward, regret in a RAMAB depends on the notion of risk. For instance, Audibert et al. [33] define the risk as the probability that the regret of the studied algorithm is much higher than its expected value, and introduce an algorithm performing trade-off between the expected reward and the risk.

One of the most widely studied models in the RAMABs is the mean-variance

model [10], which defines the risk of arm i as

$$\mathrm{mv}_i = \sigma_i^2 - \rho\mu_i$$

where $\rho \geq 0$ represents the risk trade-off factor, μ_i is the expectation and σ_i^2 is the variance of the reward distribution of arm i. Sani et al. [34] use this risk definition and introduce new a regret notion called mean-variance regret (MVR) as follows:

$$\mathrm{MVR}(t) = \hat{\mathrm{mv}}_L(t) - \hat{\mathrm{mv}}_{i^*_{\mathrm{mv}}}(t)$$

where $\hat{\mathrm{mv}}_L(t)$ denotes the empirical mean-variance of the learner by the beginning of round t and $i^*_{\mathrm{mv}} \in \arg\min_{i \in \Pi} \mathrm{mv}_i$ denotes the arm with the lowest mean-variance, which is called the *best arm in mean-variance*. In this work, MVRs of two different learning algorithms are analyzed. MV-LCB first calculates the mean-variance estimates of the arms, then on every round chooses the arm with lowest confidence bound. On the other hand, ExpExp, uses an explore-first strategy, plays every arm for a constant number of rounds and then commits to the arm with lowest estimated mean-variance at the end of exploration rounds. Vakili and Zhao [35] provide improved regret bounds for these two algorithms.

There exists many other definitions of risk in the literature. For instance, Maillard [36] uses cumulant generative function, a generalization of the mean-variance measure, as the risk notion. On the other hand, Galichet et al. [37] use conditional value at risk and propose an algorithm that selects the arm with maximal expected return given that its expected reward is in the target quantile. This algorithm takes a cautious approach and tends not to select the arms that are not well explored.

Risk-aversion and risk notions are also investigated in reinforcement learning problems. For example, Mannor and Tsitsiklis [38] and Moldovan and Abbeel [39, 40] deal with risk-aversion in the framework of Markov decision processes.

2.4 Adversarial Models and Prediction with Expert Advice

In a MAB problem it is generally assumed that the rewards of the arms are generated by well-behaved stochastic processes. Auer et al. [2] discuss a different scenario where the rewards are not drawn from such processes but instead generated by an adversary, akin to gambling in a rigged casino, and provide an algorithm called EXP3 that achieves $O(\sqrt{TM \log M})$ expected regret in a system with M arms. Auer et al. [2] also discuss adversarial bandit problem where on each round experts provide probability distributions over the available arms, and proposes EXP4 algorithm for this setting. EXP4 keeps a weight for each expert, calculates a probability distribution over available arms as a linear combination of these weights and expert advices, and then samples an arm from this probability distribution.

Cesa-Bianchi and Lugosi [41] study different versions of EXP4 algorithm where rewards of the unselected arms also become visible to the learner along with the reward of the selected arm. In online learning literature, the setting where rewards of all arms become visible to the learner at the end of the round is called *full-information feedback*, and the setting where the learner can only observe reward of the selected arm is called *bandit feedback*. Full-information feedback is applicable to the prediction with expert advice setting, where after the learner predicts a label for a sample, instead of the environment providing whether the prediction was correct or not, the correct label of the sample becomes visible to the learner.

In a prediction with expert advice system, it is also possible for the learner to use only some of the experts. Seldin et al. [42] discuss *prediction with limited advice* setting where on every round the learner is provided a budget, and according to this budget it has to select a subset of the arms. Kale [43] studies a similar problem where budget of the learner does not change between rounds.

Chapter 3

Online Contextual Expert Selection

3.1 Prediction with Expert Advice

Let $\mathcal{X}$ denote the set of samples, $\mathbf{x} \in \mathcal{X}$ denote a sample, $\mathcal{Y}$ denote the set of possible labels of the samples, and $y : \mathcal{X} \to \mathcal{Y}$ be the function mapping samples to their correct labels. In other terms, $y(\mathbf{x})$ denotes the true label for sample $\mathbf{x}$. In a classification problem $\mathcal{Y} = \{1, \ldots, J\}$, where J is the number of classes.[1]

In a prediction with expert advice system there are multiple experts (classifiers), and the learner may query one or more experts for their advices to classify a sample. Experts calculate the posterior probabilities over the possible labels. The vector of posterior probabilities of expert i for sample $\mathbf{x}$ is denoted by $\boldsymbol{p_i}(\mathbf{x}) = [p_{i,1}(\mathbf{x}), p_{i,2}(\mathbf{x}), \ldots, p_{i,J}(\mathbf{x})]^{\mathrm{T}}$ such that $\sum_{j=1}^{J} p_{i,j}(\mathbf{x}) = 1$. Here, $p_{i,j}(\mathbf{x})$ denotes the posterior probability that expert i assigns to label j for sample $\mathbf{x}$. Prediction of expert i, $f_i : \mathcal{X} \to \mathcal{Y}$ is a function mapping samples to the labels. In general, the prediction of expert i for $\mathbf{x}$ is given as $f_i(\mathbf{x}) = \arg\max_j p_{i,j}(\mathbf{x})$.

[1] In a regression problem both the label and the expert advices are real numbers. In this chapter, we focus on the classification problem.

3.2 Problem Description

The system consists of a set of M experts denoted by Π and a learner that operates sequentially over rounds indexed by $t \in \{1, 2, ...\}$. At the beginning of round t, the learner observes a sample $\mathbf{x}(t) \in \mathcal{X}$ and selects $\Pi(t) \subseteq \Pi$ such that $|\Pi(t)| = m$ and $1 \leq m \leq M$.

After experts are chosen, the learner gets advices of the selected experts, where advice of expert i is given as $\boldsymbol{p_i}(t) \triangleq \boldsymbol{p_i}(\mathbf{x}(t))$. Using the advices of the experts in $\Pi(t)$, the learner outputs a prediction $\hat{y}(t)$ for sample $\mathbf{x}(t)$, and afterwards observes the true label for sample $\mathbf{x}(t)$ denoted by $y(t) \triangleq y(\mathbf{x}(t))$, whose distribution given as $\mathrm{P}(\cdot \mid \mathbf{x}(t))$ is unknown. At the end of round t, the learner receives a loss that depends on the cost of incorrect classification. For this, let $C_{s,j}$ denote the cost of classifying a sample with label j as a sample with label s. Defining $f_i(t) \triangleq f_i(\mathbf{x}(t))$ as the prediction of expert i in round t, the loss of expert i in round t is given as $\ell_i(t) \triangleq C_{f_i(t),y(t)}$ and the loss of the learner in round t is given as $\ell(t) \triangleq C_{\hat{y}(t),y(t)}$. Since $y(t)$ is a random variable, $\ell(t)$ and $\ell_i(t)$ $\forall i \in \Pi$ are also random variables.

Losses can be translated into rewards by simply setting the reward of expert i as $r_i(t) = 1 - \ell_i(t)$ and setting the reward of the learner as $r(t) = 1 - \ell(t)$. Let T denote the time horizon, and $i^* = \arg\max_i \sum_{t=1}^{T} r_i(t)$ denote the best fixed expert over the time horizon. Performance of the learner is measured by the empirical regret, which is defined as:

$$R(T) = \sum_{t=1}^{T} (r_{i^*}(t) - r(t)). \tag{3.1}$$

The goal of the learner is to minimize its regret. In the following sections, we describe learning algorithms that will help the learner achieve its goal.

3.3 Algorithm

3.3.1 Exponentially Weighted Average Forecaster

Exponentially Weighted Average Forecaster (Exp. WAF) is a classical online learning algorithm proposed in [41]. It assigns weights to the experts by considering their past performances, and outputs prediction $\hat{y}(t)$ using the advices of all experts in Π. Let $w_i(t)$ denote the weight of the ith expert in the system in round t, then Exp. WAF calculates $p_{L,j}(t)$, the weighted posterior probability for label j in round t as:

$$p_{L,j}(t) = \sum_{i=1}^{M} w_i(t) p_{i,j}(t). \tag{3.2}$$

Then it predicts the true label as the label with the highest weighted posterior probability, i.e.,

$$\hat{y}(t) = \arg\max_{j} p_{L,j}(t). \tag{3.3}$$

Once the true label $y(t)$ is revealed, all experts receive a misclassification loss depending on how far their advices were from the correct label. For the regret bound provided in [41] to hold, the loss function needs to be convex in its first argument. In this work, we use a cost-aware and convex loss function. To define this loss, we first define the one-hot encoded version of the true label $y(t)$ as:

$$\boldsymbol{y}(t) = \left[\mathbb{1}(y(t) = 1), \mathbb{1}(y(t) = 2), \ldots, \mathbb{1}(y(t) = J)\right]^{\mathrm{T}}.$$

Then, the misclassification loss of expert i is defined as:

$$\tilde{\ell}_i(t) = \boldsymbol{p}_i(t)^{\mathrm{T}} \mathbf{C} \boldsymbol{y}(t) \tag{3.4}$$

where $\mathbf{C} = \begin{pmatrix} C_{1,1} & \ldots & C_{1,J} \\ \vdots & \ddots & \vdots \\ C_{J,1} & \ldots & C_{J,J} \end{pmatrix}$ is a $J \times J$ matrix holding the misclassification costs.[2] Correct predictions do not incur any loss, i.e., $C_{i,i} = 0$, $\forall i \leq J$.

[2]Note the difference between $\ell_i(t)$ and $\tilde{\ell}_i(t)$.

Cumulative misclassification loss of expert i at the end of round t is calculated as $L_i(t) = \sum_{n=1}^{t} \tilde{\ell}_i(n)$, and the weights of the experts are updated using the misclassification losses as follows:

$$w_i(t+1) = \frac{\exp(-\eta_t L_i(t))}{\sum_{k=1}^{M} \exp(-\eta_t L_k(t))} \tag{3.5}$$

where $\eta_t = \sqrt{\frac{8\ln(M)}{t}}$ is the learning rate. Moreover, the learner obtains the following reward:

$$r(t) = 1 - C_{\hat{y}(t),y(t)} \tag{3.6}$$

and the procedure described above repeats in every round.

3.3.2 Selective WAF

In this section we propose Selective WAF, an extension of Exp. WAF using $m \leq M$ expert advices in each round instead of the advices of all experts. This algorithm is proposed for scenarios in which the learner is limited to get advices from a subset of experts. When $m = M$, Selective WAF behaves exactly the same as Exp. WAF.

The operation of Selective WAF is similar to Exp. WAF once the experts are selected. Next, we describe how Selective WAF selects experts in round t. It divides the expert selection process in m slots in each round, where it sequentially selects one expert at a time in each slot. For convenience, slot s of round t is denoted by (t, s). Let $\Pi(t, s-1)$ be the set of experts selected before (t, s). At the beginning of round t, $\Pi(t, 0)$ is initialized as the empty set. In expert selection slot (t, s), Selective WAF randomly selects an expert from $\Pi - \Pi(t, s-1)$ according to following probability distribution:

$$\mathrm{P}(\pi(t,s) = i \mid \Pi(t,s-1)) = \begin{cases} 0, & \text{if } i \in \Pi(t,s-1) \\ w_i(t) / \sum_{k \in \Pi - \Pi(t,s-1)} w_k(t), & \text{otherwise} \end{cases}$$

where $\pi(t, s)$ denotes the expert selected at (t, s). Then, the selected expert is

Algorithm 1 Selective Weighted Average Forecaster

function SELECTIVEWAF($\mathcal{X}, \mathcal{Y}, \Pi, m, \mathbf{C}$)
  Init:
    for $i = 1, \ldots, M$ **do**
      $w_i(1) \leftarrow 1/M$
      $L_i(0) \leftarrow 0$
  for $t = 1, \ldots, T$ **do**
    Generate $\Pi(t)$ according to $\{w_1(t), \ldots, w_M(t)\}, m$
    for $j = 1, 2, \ldots, J$ **do**

$$p_{L,j}(t) \leftarrow \frac{\sum_{i \in \Pi(t)} w_i(t) p_{i,j}(t)}{\sum_{i \in \Pi(t)} w_i(t)}$$

    $\hat{y}(t) \leftarrow \arg\max_j p_{L,j}(t)$
    Obtain $r(t)$ according to (3.6)
    for $i = 1, \ldots, M$ **do**
      Calculate $q_i(t)$ according to (3.9)
      $L_i(t) \leftarrow L_i(t-1) + \mathbb{1}(i \in \Pi(t)) \frac{\boldsymbol{p}_i(t)^{\mathrm{T}} \mathbf{C} \boldsymbol{y}(t)}{q_i(t)}$
    $\eta_t \leftarrow \sqrt{\frac{8 \ln(M)}{t}}$
    for $i = 1, \ldots, M$ **do**

$$w_i(t+1) \leftarrow \frac{\exp(-\eta_t L_i(t))}{\sum_{k=1}^{M} \exp(-\eta_t L_k(t))}$$

added to the set of selected experts for next expert selection slot, i.e., $\Pi(t, s) = \Pi(t, s-1) \cup \{\pi(t, s)\}$. When m experts are selected this procedure terminates, and selected arms are fixed as $\Pi(t) = \Pi(t, m)$.

After expert selection is completed, Selective WAF computes the posterior distribution over the labels by taking a weighted combination of the advices of the selected experts as follows:

$$\begin{aligned} w_i(t,m) &\triangleq \frac{w_i(t)}{\sum_{k\in\Pi(t)} w_k(t)} \\ p_{L,j}(t) &= \sum_{i\in\Pi(t)} p_{i,j}(t) w_i(t,m). \end{aligned} \tag{3.7}$$

Once weighted posterior probabilities are calculated, $\hat{y}(t)$ is set according to (3.3), and then $r(t)$ is calculated according to (3.6).

The expert selection rule of Selective WAF is random, therefore an expert is selected only in some rounds. Hence, we have to estimate the total loss experts would accumulate if they were selected in all rounds, by using their selection probability and the losses they received in the rounds where they were selected. Let $q_i(t)$ be the probability that expert i is selected by Selected WAF in round t, then the estimated loss of the expert i in round t is defined as:

$$\hat{\ell}_i(t) = \mathbb{1}(i \in \Pi(t)) \frac{\boldsymbol{p}_i(t)^{\mathrm{T}} \mathbf{C} \boldsymbol{y}(t)}{q_i(t)}. \tag{3.8}$$

To estimate this total loss, we need to calculate the $q_i(t)$. Let $\boldsymbol{e}$ be an ordered set of experts, $\boldsymbol{e}^{(j)}$ denotes the jth expert in $\boldsymbol{e}$. Let $\mathcal{E}$ be the set of all ordered sets that can be generated with M experts, and $\mathcal{E}_{i,s}$ be the set of all possible orderings such that any element of $\mathcal{E}_{i,s}$ contains s experts from Π and for any element in $\mathcal{E}_{i,s}$, sth expert is i, i.e. $\mathcal{E}_{i,s} = \{\boldsymbol{e} \in \mathcal{E} : \boldsymbol{e}^{(s)} = i, |\boldsymbol{e}| = s\}$. For example, if $\Pi = \{1,2,3\}$, then $\mathcal{E}_{1,3} = \{(3,2,1),(2,3,1)\}$.

Then $q_i(t)$ is calculated as follows:[3]

$$\begin{aligned} q_i(t) &\triangleq \mathrm{P}(i \in \Pi(t,m) \mid w_1(t), \ldots, w_M(t)) \\ &= \sum_{s=1}^{m} \mathrm{P}(\pi(t,s) = i \mid w_1(t), \ldots, w_M(t)) \\ &= \sum_{s=1}^{m} \sum_{\boldsymbol{e} \in \mathcal{E}_{i,s}} \mathrm{P}(\boldsymbol{e} \mid w_1(t), \ldots, w_M(t)) \\ &= \sum_{s=1}^{m} \sum_{\boldsymbol{e} \in \mathcal{E}_{i,s}} \prod_{k=0}^{s-1} \frac{w_{\boldsymbol{e}^{(s-k)}}(t)}{\sum_{j \in \Pi - \{\boldsymbol{e}^{(1)}, \ldots, \boldsymbol{e}^{(s-1-k)}\}} w_j(t)} \end{aligned} \tag{3.9}$$

Finally, the arm weights are updated according to (3.5), using $L_i(t) = \sum_{n=1}^{t} \hat{\ell}_i(n)$, and the algorithm proceeds to the next round. The pseudocode of Selective WAF is given in Algorithm 1.

3.3.3 Contextual Selective WAF

Contextual Selective WAF (CS-WAF) assumes that the experts have heterogeneous information and accuracy over the sample space. This may be the case when the experts are trained with different datasets and with different learning algorithms. For example, consider a sentiment analysis problem, where one classifier is trained with posts in social networks and the other classifier is trained with online reviews. In such case, understanding the context of text beforehand and weighting classifiers accordingly can greatly increase the accuracy of the system.

CS-WAF assumes the existence of a mapping $g : \mathcal{X} \to \mathcal{Z}$ that maps samples to contexts such that $\boldsymbol{z} = g(\mathbf{x})$ denotes the context of sample $\mathbf{x}$. It uses the *contextual zooming* approach proposed by Slivkins [17] to adaptively create subsets of the context space $\mathcal{Z}$.

CS-WAF works as follows: First, it creates a cover of $\mathcal{Z}$, where the cover consists of ball shaped sets. Each of these sets is called a *ball*. In each round, it

[3]It is assumed that $\{\boldsymbol{e}^{(1)}, \boldsymbol{e}^{(0)}\} = \emptyset$

selects a ball based on the sample received in current round and the history of the balls, where each ball holds its own weights and losses. CS-WAF combines advices according to weights in the selected ball, outputs a prediction, and updates weights along with the losses of the selected ball as described in Selective WAF. It observes the reward and updates its information about the selected ball. Finally, if the selected ball receives sufficient number of samples, CS-WAF creates a new ball shaped set, updates the cover for $\mathcal{Z}$ and proceeds to the next round. The pseudocode of CS-WAF is given in Algorithm 2. An example cover of a 2-dimensional context space is provided in Fig. 3.1.

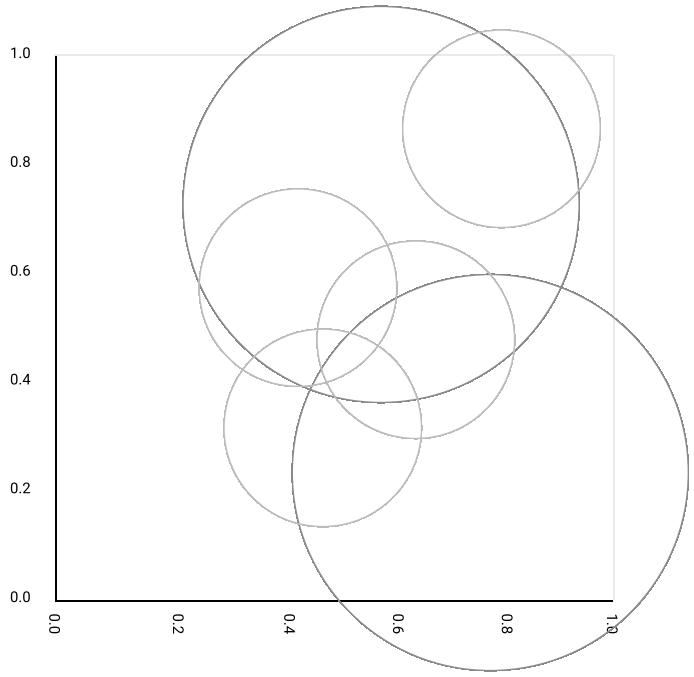

Figure 3.1: A Possible Cover of the 2-Dimensional Context Space[4]

Other than selecting and creating balls, the operation of CS-WAF is the same as Selective WAF. Hence, we will describe how CS-WAF selects the ball to be used, based on the received sample and how CS-WAF updates its cover. On round t, CS-WAF observes $\mathbf{x}(t)$ and calculates the context in round t as $\boldsymbol{z}(t) = g(\mathbf{x}(t))$.

Let $B \subseteq \mathcal{Z}$ be a ball shaped set in cover for $\mathcal{Z}$, and $\mathcal{B}(t)$ be the set of balls in

[4]The initial ball that covers the entire context space is not shown here.

the system at beginning of round t. Let o_B, rad_B denote the center and radius of the ball B respectively, $\mu_B(t)$ denote the mean of rewards received when ball B is selected until the beginning of round t, and $N_B(t)$ denote the number of times ball B is selected at the beginning of round t.

Once context $\boldsymbol{z}(t)$ arrives, CS-WAF calculates the set of balls that are *relevant* to $\boldsymbol{z}(t)$. A ball is defined as relevant to $\boldsymbol{z}(t)$ if it includes $\boldsymbol{z}(t)$ in its *domain*, where domain of ball B in round t, $dom_B(t)$ is defined as:

$$\begin{aligned} \mathcal{B}_B(t) &\triangleq \{A \in \mathcal{B}(t) : \mathrm{rad}_A < \mathrm{rad}_B\} \\ dom_B(t) &\triangleq B - \bigcup_{A \in \mathcal{B}_B(t)} A. \end{aligned} \tag{3.10}$$

Then, CS-WAF selects the ball with the highest UCB among the relevant balls. To calculate UCB, we first define a *confidence radius* of ball B in round t, $c_B(t)$ as:

$$c_B(t) \triangleq 4\sqrt{\frac{\log T}{1 + N_B(t)}}.$$

Then, UCB of ball B in round t, $U_B(t)$, is calculated as:

$$\begin{aligned} U_B^{pre}(t) &\triangleq \mu_B(t) + \mathrm{rad}_B + c_B(t) \\ U_B(t) &= \mathrm{rad}_B + min_{A \in \mathcal{B}(t)}(U_A^{pre}(t) + D_{\mathcal{Z}}(\mathrm{o}_B, \mathrm{o}_A)) \end{aligned} \tag{3.11}$$

where $D_{\mathcal{Z}} : (\mathcal{Z}, \mathcal{Z}) \to [0, 1]$ is a distance function in the context space. Finally, $B(t)$, the ball to be used by CS-WAF in round t, is selected according to following rule:

$$B(t) = \mathrm{argmax}_{B \in \mathcal{B}_{\mathrm{rel}}(t)} U_B(t) \tag{3.12}$$

where $\mathcal{B}_{\mathrm{rel}}(t) = \{B \in \mathcal{B}(t) : \boldsymbol{z}(t) \in dom_B(t)\}$ denotes the set of relevant balls in round t. This behavior can be defined as follows: If there is more than one ball that encapsulates $\boldsymbol{z}(t)$, then the ball with smallest radius is chosen. If the ball with the smallest radius is not unique, then the ball with highest UCB among the balls with the smallest radius is chosen.

Algorithm 2 Contextual Selective WAF (CS-WAF)

function InitBall($center, radius$)
  $\mathrm{o}_B \leftarrow center, \mathrm{rad}_B \leftarrow radius$
  $N_B(0) = \mu_B(1) = 0$
  for $i = 1, \ldots, M$ **do**
    $w_{B,i}(1) = 1/M$
    $L_{B,i}(0) = 0$
  return B
function ContextualSelectiveWAF($\mathcal{X}, \mathcal{Y}, \Pi, m, \mathbf{C}$)
  Init:
    $B \leftarrow$ InitBall($\boldsymbol{z}(1)$, 1)
    $\mathcal{B}(1) \leftarrow \{B\}$
  for t=1...T **do**
    Observe $\boldsymbol{z}(t)$
    $B(t) \leftarrow \arg\max_{B \in \mathcal{B}_{\mathrm{rel}}(t)} U_B(t)$
    Select $\Pi(t)$ according to $\{w_{B(t),1}(t), \ldots, w_{B(t),M}(t)\}$, m
    for $j = 1, 2, \ldots, J$ **do**

$$p_{L,j}(t) \leftarrow \frac{\sum_{i \in \Pi(t)} w_{B(t),i}(t) p_{i,j}(t)}{\sum_{i \in \Pi(t)} w_{B(t),i}(t)}$$

    $\hat{y}(t) \leftarrow \arg\max_j p_{L,j}(t)$
    Obtain $r(t)$, update $\mu_{B(t)}(t+1), N_{B(t)}(t+1)$ accordingly
    for $i = 1, \ldots, M$ **do**
      Calculate $q_i(t)$ according to (3.9)
      $L_{B(t),i}(t) \leftarrow L_{B(t),i}(t-1) + \mathbb{1}(i \in \Pi(t,m)) \frac{\boldsymbol{p}_i(t)^{\mathrm{T}} \mathbf{C} \boldsymbol{y}(t)}{q_i(t)}$
    $\eta_t \leftarrow \sqrt{\frac{8 \ln(M)}{N_{B(t)}(t)+1}}$
    for $i = 1, \ldots, M$ **do**

$$w_{B(t),i}(t+1) \leftarrow \frac{\exp(-\eta_t L_{B(t),i}(t))}{\sum_{l=1}^{M} \exp(-\eta_t L_{B(t),l}(t))}$$

    if $c_{B(t)}(t) \leq \mathrm{rad}_{B(t)}$ **then**
      $B \leftarrow$ InitBall($\boldsymbol{z}(t), \frac{\mathrm{rad}_{B(t)}}{2}$)
      $\mathcal{B}(t+1) \leftarrow \mathcal{B}(t) \cup \{B\}$

Fig. 3.2 shows the set $\mathcal{B}_{\text{rel}}(t)$ identified by CS-WAF in a round t for a 2-dimensional context space. Red dot marks $\boldsymbol{z}(t)$, the context of the sample at the current round. In this example, while 5 balls (2 large and 3 small) encapsulate the $\boldsymbol{z}(t)$, only the small ones are used as relevant balls.

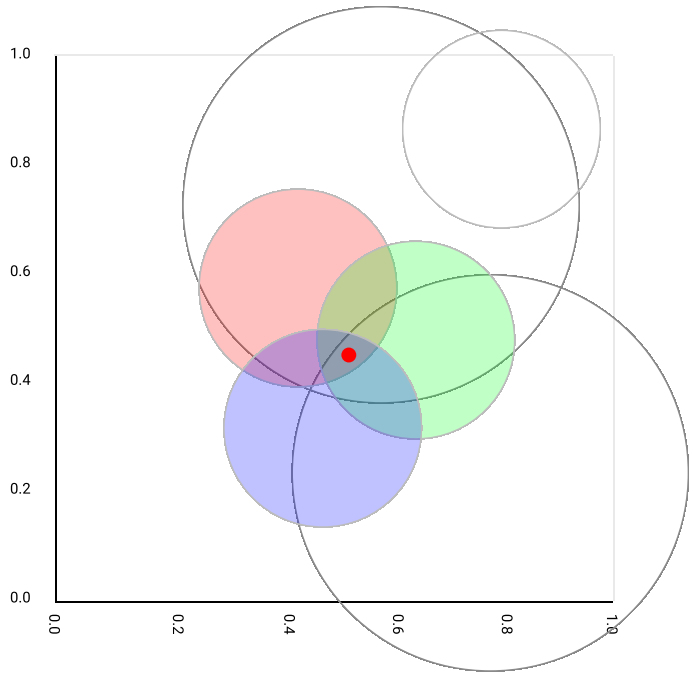

Figure 3.2: An Example of Relevant Balls in Contextual Zooming

Let $L_{B,i}(t)$ be the cumulative loss of the expert i in rounds where ball B is selected until the end of round t. Weight of expert i in ball B at the beginning of round t, $w_{B,i}(t)$, is calculated similar to (3.5) using $L_{B,i}(t)$. Then using these weights, experts are selected, their advices are combined, label is predicted and finally losses are updated in the same manner as Selective WAF. Once ground truth label is observed, $r(t)$ is obtained and $\mu_B(t+1)$ is updated. Weight updates of the selected experts are also made as described in Selective WAF.

Finally if the confidence radius of the $B(t)$ shrinks to a value smaller than $\text{rad}_{B(t)}$, a new ball created with the center $\boldsymbol{z}(t)$ and radius $\frac{\text{rad}_{B(t)}}{2}$. This final adjustment allows the algorithm to focus on contexts that arrive more frequently and to provide a more precise prediction for them.

3.4 Results

In this section, we illustrate the performance of the CS-WAF and Selective WAF algorithms over two different datasets, under various cost sensitivity settings.

3.4.1 Thyroid Disease

3.4.1.1 Dataset Description

In this section we compare Selective WAF and CS-WAF using the thyroid dataset from the UC Irvine data repository [44]. This dataset includes 3772 training and 3428 testing instances. Instances have 21 features, of which 15 are binary and 6 are real valued. Each instance belongs to one of the three classes: normal thyroid (healthy), hyper-functioning thyroid (unhealthy), and subnormal-functioning thyroid (unhealthy). 92.5% of the instances in the dataset belong to the healthy class.

3.4.1.2 Experiment Setup

Training set is first projected into 2-dimensional space using TSNE algorithm [45], which encodes samples such that two pairwise similar items in the original space are close to each other in the encoded space. To make sure that each expert in the system is informed about a different subset of the sample space, training data in 2-dimensional space is split into 10 different non-overlapping sets using K-Means algorithm [46].

Each of these sets is used to train a different classifier. In total, 10 experts are trained using Decision Tree algorithm [47], where Gini impurity is used as split criterion. No pre-pruning or post-pruning measures are used. In cases where a set did not include at least 1 sample from all three classes, a random sample from training data belonging to the missing class is added to samples in that set.

Principal components of the dataset are calculated using the training set, and the first 3 principal components of each sample are used as the context vector.

3.4.1.3 Results

Experiments are repeated with T values in $\{5000, 10000, 25000, 50000\}$. Before each experiment, T instances are sampled from the test set without replacement and provided to the learning algorithms. Cost matrix is selected as $\mathbf{C} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$. For each value of T, the experiment is repeated 24 times, and averages over these runs are reported.

Results of the experiments are given in Fig. 3.3. Solid lines correspond to the average rewards and the light shaded area correspond to the 95% confidence interval of the given metric.[5] When m and T are both small, average reward received by CS-WAF and Selective WAF are close to each other. For small values of m, the learner accumulates low initial reward in the beginning since it needs to learn to select the best experts from its history of observations and selections. As T increases, performance of Selective WAF converges to performance of the best expert. On the other hand, CS-WAF approximately learns the correct weights for each context, which results in substantial performance increase compared to Selective WAF. For large values of m, *e.g.*, $m = M$, CS-WAF performs better than Selective WAF for every T, since it requires smaller number of rounds to learn the near-optimum expert weights in each ball due to availability of the advices of all experts.

[5]Since rewards are real, normal distribution is assumed. 95% confidence interval is calculated as the interval covering the two standard deviation distance from the mean of the metric.

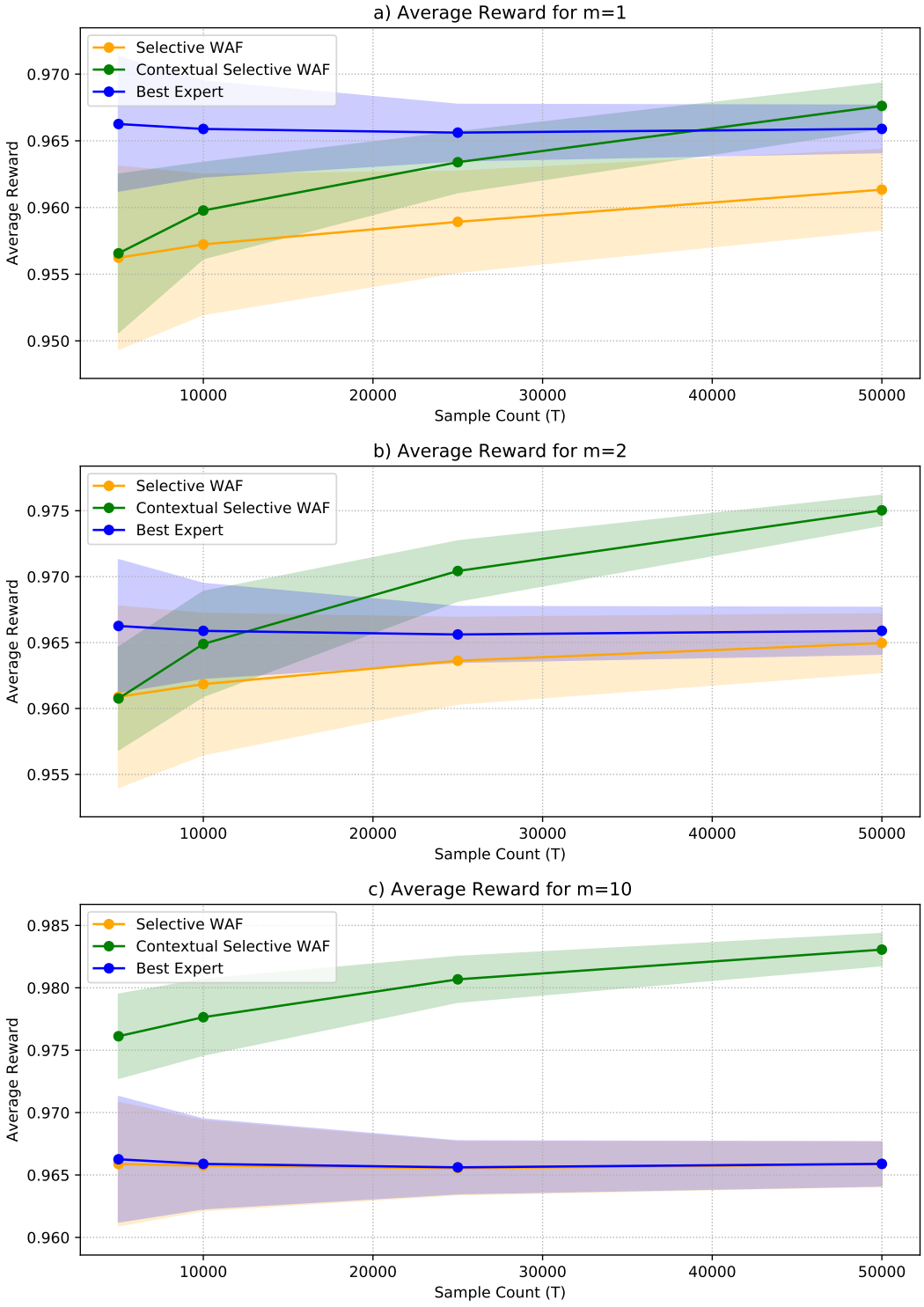

Figure 3.3: The average reward as a function of T for Thyroid Dataset

3.4.2 Mortality Detection in Intensive Care Unit (ICU)

3.4.2.1 Dataset Description

In this part, dataset obtained from PhysioNet/CinC Challenge 2012 [48] is used. This dataset is collected from 12000 ICU stays that lasted at least 48 hours. Each stay ends either with survival or in-hospital death (IHD). The aim is to predict patient mortality in ICU stays. Since final mortality labels are provided for only 4000 stays, other 8000 stays are omitted.

Due to the fact that records are collected from a real hospital, dataset includes invalid, missing, and duplicated measurements. In the pre-processing phase, non-invasive and invasive blood pressures (systolic, diastolic and mean-arterial pressure) are merged into a single measurement type. Measurements that are not within the valid maximum and minimum values provided in [49] are assumed to be invalid and they are removed. Variables are normalized and missing values are imputed as described in [50]. Finally mean, maximum and minimum values in first 24 and second 24 hours are calculated, and for each patient a 217×1 vector (3 descriptors, 6 values for 35 measurements and 4 categorical values) is generated. 6 patients whom did not have any measurements for a whole day are removed and final dataset with 3994 patient records is obtained. Random 1000 samples are selected as training set, while remaining 2994 samples are used as test set. 85% of the samples in the test set belong to the survival class.

Context vectors are calculated using an autoencoder neural network. Number of neurons in the hidden layers are 24, 6, 24, and 217, respectively. The autoencoder is trained with mean-squared error loss using preprocessed training data. For updates, mini-batch with a size of 32 samples is used and network is trained for 50 epochs. Once the training is completed, each sample in the test set is passed through the network and output of second layer, a 6×1 vector, is saved as the context of the related sample.

3.4.2.2 Results

Partition of the training set, expert training, and sampling from test set is made as described in 3.4.1. In this setup, the cost of classifying a IHD class as survival is set as 6 times (roughly the ratio of survival classes to IHD classes) of classifying survival as IHD, i.e. $\mathbf{C} = \left(\begin{smallmatrix} 0 & 1.714 \\ 0.285 & 0 \end{smallmatrix}\right)$. Similar to the 3.4.1 experiments are repeated for 24 runs and averages over these runs are reported.

Results of the experiments are provided in Fig. 3.4. Similar to the Fig. 3.3, solid lines correspond to the average rewards and the light shaded area correspond to the 95% confidence interval of the given metric. Similar to the results in Section 3.4.1, performance of both algorithms increase as T increases. For small values of T, average rewards collected by algorithms are very close to each other, while as T increases CS-WAF is able to collect more rewards by exploiting the contextual knowledge.

For $m = 1$, Selective WAF is unable to achieve the same average reward with the best expert, since half of the experts' suboptimality gap is very similar to the worst fixed expert over all rounds, but for $m = M$, it is able to achieve the performance of the best expert. As expected both algorithms perform better when the number of experts to be selected increases.

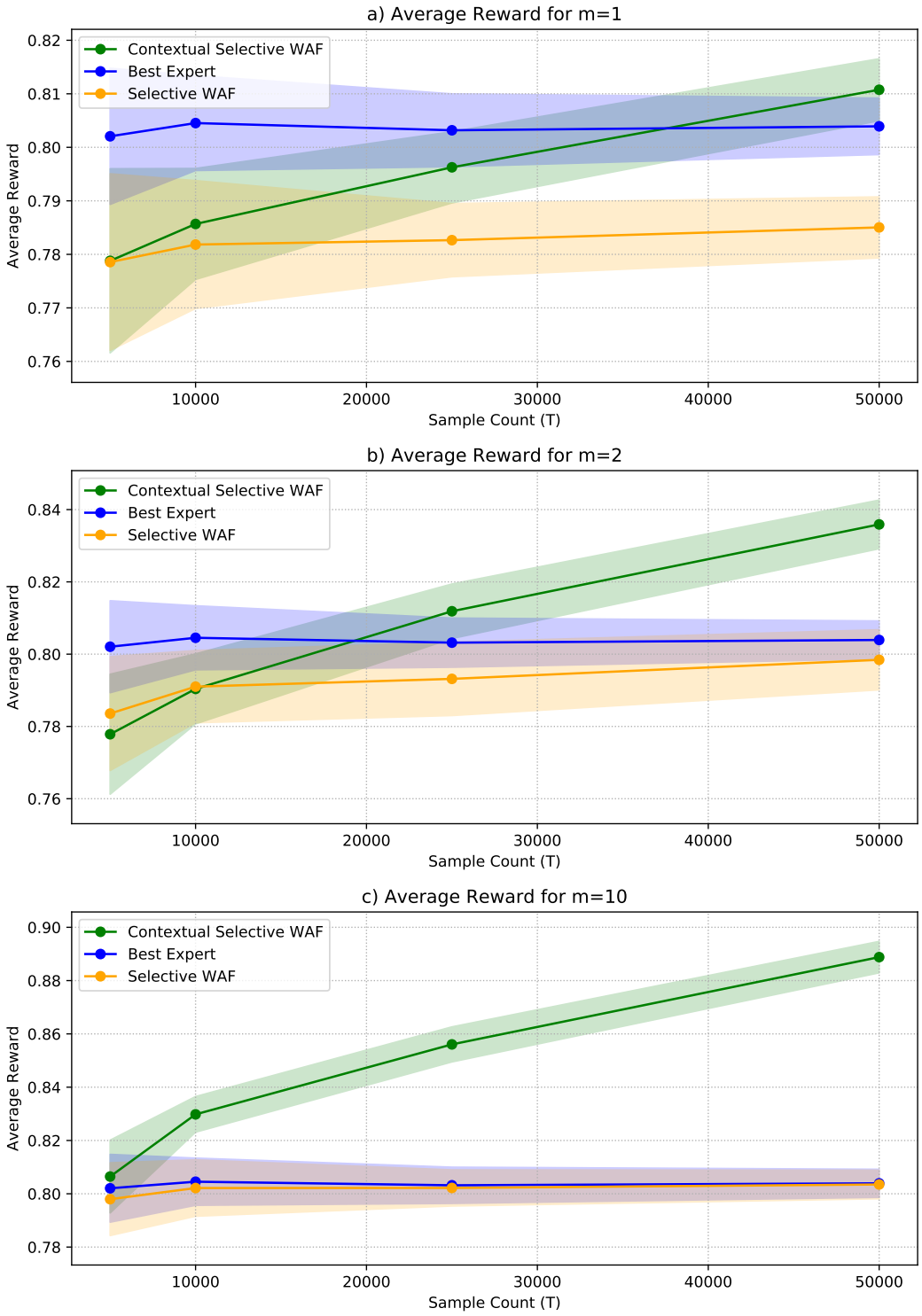

Figure 3.4: The average reward as a function of T for CinC Dataset

Chapter 4

The Safe Bandit

4.1 Problem Description

In the Safe Bandit, the system is comprised of two main elements, a finite set of M arms denoted by Π and a learner. The system operates in rounds, where in round $t \in \{1, 2, \ldots\}$ the learner selects an arm $\pi(t)$ from Π and observes a reward. The reward of arm i in round t is $r_i(t) = \mu_i + \mathcal{E}_i(t)$[1] where μ_i is the mean reward of arm i and $\mathcal{E}_i(t)$ denotes the zero mean random noise that comes from a fixed distribution with support in $[-1, +1]$.[2] The learner selects a single arm in every round, hence $r(t) = r_{\pi(t)}(t)$. The mean-variance of arm i is given as $\text{mv}_i = \sigma_i^2 - \rho\mu_i$, where $\rho \geq 0$ represents the risk trade-off factor, and σ_i^2 represents the variance of arm i.

In this setting, arms are clustered into two groups according to their mean-variance. Let κ denote the risk threshold, the set of arms with a mean-variance greater than κ are called *risky* arms, and the set of arms with a mean-variance lower than κ are called *risk-free* arms. The aim of the learner is to minimize the number of rounds where a risky arm is selected. The total loss of the learner due

[1] When it is clear to infer the referred round from the context, round index is dropped from the notation for all variables related to the current round.

[2] This result can be generalized to the case when the support of the noise is bounded.

to selecting risky arms is called *risk violation regret* (RVR) and is defined as:

$$\mathrm{RVR}_{\kappa,\rho}(T) \triangleq \sum_{t=1}^{T} \mathbb{1}(\mathrm{mv}_{\pi(t)} > \kappa) \tag{4.1}$$

where $\mathbb{1}(\cdot)$ is the indicator function.[3] The value of $\mathrm{RVR}_{\kappa,\rho}(T)$ depends on actual rewards of arms and choices of the learner, hence it is a random variable.

Different ρ values may result in different mean-variances based on ordering of the arms, and for a given ρ, changing κ changes the set of risky (and risk-free) arms. Thus, the sets of risky and risk-free arms depend both on ρ and κ, in addition to the actual mean and variances of the arms. In Section 4.2, an algorithm with an RVR depending on the size of the set of risky arms, and the distance between risky arms and the arm with minimum mean-variance is described.

4.2 A Learning Algorithm and its RVR

4.2.1 Algorithm

Exploration and Exploitation using Risk Thresholds (EXERT) forms empirical estimates of the mean-variances of the arms using empirical estimates of means and variances of the arms. $\hat{\mu}_i(t)$ denotes the empirical estimate of the mean of arm i at the beginning of round t and is defined as:

$$\hat{\mu}_i(t) = \frac{1}{N_i(t)} \sum_{n \in \mathcal{T}_i(t)} r(n)$$

where $N_i(t)$ denotes the number of times arm i is chosen until the beginning of round t and $\mathcal{T}_i(t)$ denotes the set of rounds in which arm i is chosen until the beginning of round t by the learner. Similarly, $\hat{\sigma}_i^2(t)$ denotes the empirical

[3]RVR can be written as a metric relative to the *best arm in mean-variance*, but since in our problem $\mathrm{mv}_{\pi^*} < \kappa$, that part is omitted.

estimate of the variance of arm i at the beginning of round t and is defined as:

$$\hat{\sigma}_i^2(t) = \frac{1}{N_i(t)} \sum_{n \in \mathcal{T}_i(t)} \left(r(n) - \hat{\mu}_i(t)\right)^2 .$$

Once $\hat{\mu}_i(t)$ and $\hat{\sigma}_i^2(t)$ are formed, the empirical mean-variance of arm i at the beginning of round t is calculated as:

$$\hat{\text{mv}}_i(t) = \hat{\sigma}_i^2(t) - \rho\hat{\mu}_i(t). \tag{4.2}$$

Rather than calculating confidence bounds on the mean reward as the risk-neutral algorithms like Lai [22] and Auer et al. [1] do, EXERT calculates the confidence bounds on the mean-variances of the arms. $U_i(t)$, upper confidence bound (UCB) of mean-variance of arm i in round t is calculated as:

$$U_i(t) = \hat{\text{mv}}_i(t) + (2 + \rho)c_i(t)$$

and $L_i(t)$ lower confidence bound (LCB) of arm i in round t is calculated as:

$$L_i(t) = \hat{\text{mv}}_i(t) - (1 + \rho)c_i(t)$$

where

$$c_i(t) = \sqrt{\frac{1 + N_i(t)}{N_i(t)^2} \left(1 + 2\log\left(\frac{2M(1 + N_i(t))^{1/2}}{\delta}\right)\right)} \tag{4.3}$$

and δ is the confidence parameter, controlling the probability of events where $\text{mv}_i < L_i(t)$ or $\text{mv}_i > U_i(t)$. Using UCBs of the arms, pessimistic estimate of the set of risk-free arms in round t is formed as follows:

$$\hat{\Pi}_{\text{rf}}(t) \triangleq \{i \in \Pi : U_i(t) \leq \kappa\}.$$

Once $\hat{\Pi}_{\text{rf}}(t)$ is calculated, $\pi(t)$ is selected according to the following rule: If $\hat{\Pi}_{\text{rf}}(t) \neq \emptyset$, then EXERT optimistically chooses the arm that is risk-free and has the lowest LCB, i.e., $\pi(t) \in \arg\min_{i \in \hat{\Pi}_{\text{rf}}(t)} L_i(t)$. However, if $\hat{\Pi}_{\text{rf}}(t) = \emptyset$, it is not possible to guarantee with a high probability that chosen arm will not violate

the risk threshold κ. Therefore, using the optimism under uncertainty approach, EXERT chooses the arm with lowest LCB in Π, i.e., $\pi(t) \in \arg\min_{i \in \Pi} L_i(t)$ where ties are broken randomly. Pseudocode of EXERT is given in Algorithm 3.

Algorithm 3 EXERT

Input: ρ, δ, κ
Initialize: Select all arms once, and observe their rewards to initialize $\hat{\text{mv}}_\pi$, set $t = M + 1$, and $N_i = 1 \; \forall i \in \Pi$
while $t > M$ **do**
  for $i \in \Pi$ **do**
    $L_i \leftarrow \hat{\text{mv}}_i - (1 + \rho)c_i$
    $U_i \leftarrow \hat{\text{mv}}_i + (2 + \rho)c_i$
  $\hat{\Pi}_{\text{rf}} \leftarrow \{i \in \Pi : U_i \leq \kappa\}$
  if $\hat{\Pi}_{\text{rf}} \neq \emptyset$ **then**
    $\pi \leftarrow \arg\min_{i \in \hat{\Pi}_{\text{rf}}} L_i$
  else
    $\pi \leftarrow \arg\min_{i \in \Pi} L_i$
  Receive reward r
  $N_\pi \leftarrow N_\pi + 1$
  Update $\hat{\text{mv}}_\pi$ using (4.2) and c_π using (4.3)
  $t \leftarrow t + 1$

4.2.2 Analysis

In this section, RVR of EXERT is analyzed and it is shown to be independent of T. First of all, following lemma provides a high probability tail bound on the mean-variances of the arms using empirical mean-variances.

Lemma 1. *With probability at least* $1 - \delta$*, we have*

$$\hat{mv}_i(t) - (1 + \rho)c_i(t) \leq mv_i \leq \hat{mv}_i(t) + (2 + \rho)c_i(t) \; \forall i \in \Pi, \forall t > M.$$

Proof. Initially, two lemmas that will be used in the proof are presented.

Lemma 1.1. *With probability at least* $1 - \delta/2$*, we have*

$$|\hat{\mu}_i(t) - \mu_i| \leq c_i(t) \; \forall i \in \Pi \; \textit{and} \; \forall t > M$$

Proof. Given that $\mathcal{E}_i(t)$ is 1-sub-Gaussian, this lemma directly follows from Lemma 6 in [51]. □

Let

$$\tilde{\sigma}_i^2(t) = \frac{1}{N_i(t)} \sum_{n \in \mathcal{T}_i(t)} (r(n) - \mu_i)^2$$

The next lemma gives a tail bound on $\tilde{\sigma}_i^2(t)$ for all arms.

Lemma 1.2. *With probability at least* $1 - \delta/2$*, we have*

$$|\tilde{\sigma}_i^2(t) - \sigma_i^2| \leq c_i(t) \ \forall i \in \Pi \ \textit{and} \ \forall t > M$$

Proof. First we write $r(n) - \mu_i$ in terms of $\mathcal{E}_i(n)$:

$$\tilde{\sigma}_i^2(t) = \frac{1}{N_i(t)} \sum_{n \in \mathcal{T}_i(t)} (\mathcal{E}_i(n))^2 .$$

Since $\mathcal{E}_i \in [-1, 1]$, we have $\mathcal{E}_i^2 \in [0, 1]$, and $\mathcal{E}_i^2 - \sigma_i^2 \in [-\sigma_i^2, 1 - \sigma_i^2] \subseteq [-1, 1]$. Since $\mathrm{E}[\mathcal{E}_i] = \sigma_i^2$, we know that $\mathrm{E}[\mathcal{E}_i^2 - \sigma_i^2] = 0$, which implies that $\mathcal{E}_i^2 - \sigma_i^2$ is 1-sub-Gaussian. The rest of the proof is similar to the proof of Lemma 1.1. □

We have

$$\begin{aligned}
\hat{\sigma}_i^2(t) &= \frac{1}{N_i(t)} \sum_{n \in \mathcal{T}_i(t)} (r(n) - \hat{\mu}_i(t))^2 \\
&= \frac{1}{N_i(t)} \sum_{n \in \mathcal{T}_i(t)} (r(n) - \mu_i + \mu_i - \hat{\mu}_i(t))^2 \\
&= \frac{1}{N_i(t)} \sum_{n \in \mathcal{T}_i(t)} \left[(r(n) - \mu_i)^2 + (\hat{\mu}_i(t) - \mu_i)^2 \right] - \frac{2}{N_i(t)} \sum_{n \in \mathcal{T}_i(t)} (r(n) - \mu_i)(\hat{\mu}_i(t) - \mu_i) \\
&= \tilde{\sigma}_i^2(t) + (\hat{\mu}_i(t) - \mu_i)^2 - \frac{2}{N_i(t)} \sum_{n \in \mathcal{T}_i(t)} (r(n)\hat{\mu}_i(t) - r(n)\mu_i - \mu_i\hat{\mu}_i(t) + \mu_i^2) \\
&= \tilde{\sigma}_i^2(t) + (\hat{\mu}_i(t) - \mu_i)^2 - 2(\hat{\mu}_i(t)^2 - 2\hat{\mu}_i(t)\mu_i + \mu_i^2) \\
&= \tilde{\sigma}_i^2(t) - (\hat{\mu}_i(t) - \mu_i)^2.
\end{aligned}$$

Using the equality above, we obtain

$$\begin{aligned}
\hat{\text{mv}}_i(t) - \text{mv}_i &= \hat{\sigma}_i^2(t) - \rho\hat{\mu}_i(t) - (\sigma_i^2 - \rho\mu_i) \\
&= \tilde{\sigma}_i^2(t) - (\hat{\mu}_i(t) - \mu_i)^2 - \rho\hat{\mu}_i(t) - \sigma_i^2 + \rho\mu_i \\
&= \tilde{\sigma}_i^2(t) - (\hat{\mu}_i(t) - \mu_i)^2 - \sigma_i^2 - \rho(\hat{\mu}_i(t) - \mu_i) \\
&= (\tilde{\sigma}_i^2(t) - \sigma_i^2) - (\hat{\mu}_i(t) - \mu_i)^2 - \rho(\hat{\mu}_i(t) - \mu_i).
\end{aligned}$$

Lemma 1.1 shows that

$$\text{P}(|\hat{\mu}_i(t) - \mu_i| \geq c_i(t)) \leq \frac{\delta}{2} \quad \forall i \in \Pi \text{ and } \forall t > M,$$

and Lemma 1.2 shows that

$$\text{P}(|\tilde{\sigma}_i^2(t) - \sigma_i^2| \geq c_i(t)) \leq \frac{\delta}{2} \quad \forall i \in \Pi \text{ and } \forall t > M.$$

Combining these two events, we have

$$\text{P}(|\hat{\mu}_i(t) - \mu_i| \geq c_i(t) \cup |\tilde{\sigma}_i^2(t) - \sigma_i^2| \geq c_i(t)) \leq \delta \quad \forall i \in \Pi \text{ and } \forall t > M,$$

and we have

$$\text{P}(|\hat{\mu}_i(t) - \mu_i| \leq c_i(t) \cap |\tilde{\sigma}_i^2(t) - \sigma_i^2| \leq c_i(t)) \geq 1 - \delta \quad \forall i \in \Pi \text{ and } \forall t > M.$$

Next, we bound mv_i under the event where both $|\hat{\mu}_i(t) - \mu_i| \leq c_i(t)$ and $|\tilde{\sigma}_i^2(t) - \sigma_i^2| \leq c_i(t)$ hold. First, the lower bound for mv_i is obtained as follows:

$$\begin{aligned}
\hat{\text{mv}}_i(t) - \text{mv}_i &= (\tilde{\sigma}_i^2(t) - \sigma_i^2) - (\hat{\mu}_i(t) - \mu_i)^2 - \rho(\hat{\mu}_i(t) - \mu_i) \\
&\leq (\tilde{\sigma}_i^2(t) - \sigma_i^2) - \rho(\hat{\mu}_i(t) - \mu_i) \\
&\leq |\tilde{\sigma}_i^2(t) - \sigma_i^2| + \rho|\hat{\mu}_i(t) - \mu_i| \\
&\leq (1 + \rho)c_i(t) \Rightarrow \\
\text{mv}_i &\geq \hat{\text{mv}}_i(t) - (1 + \rho)c_i(t).
\end{aligned}$$

We know that $\hat{\mu}_i(t) - \mu_i \subseteq [-1, 1]$, which implies that $(\hat{\mu}_i(t) - \mu_i)^2 \leq |\hat{\mu}_i(t) - \mu_i|$.

Using this information, we can obtain the upper bound for mv_i as follows:

$$\begin{aligned}
\mathrm{mv}_i - \hat{\mathrm{mv}}_i(t) &= -(\tilde{\sigma}_i^2(t) - \sigma_i^2) + (\hat{\mu}_i(t) - \mu_i)^2 + \rho(\hat{\mu}_i(t) - \mu_i) \\
&\leq |\tilde{\sigma}_i^2(t) - \sigma_i^2| + (\hat{\mu}_i(t) - \mu_i)^2 + \rho|\hat{\mu}_i(t) - \mu_i| \\
&\leq |\tilde{\sigma}_i^2(t) - \sigma_i^2| + |\hat{\mu}_i(t) - \mu_i| + \rho|\hat{\mu}_i(t) - \mu_i| \\
&\leq |\tilde{\sigma}_i^2(t) - \sigma_i^2| + (1+\rho)|\hat{\mu}_i(t) - \mu_i| \\
&\leq (2+\rho)c_i(t) \Rightarrow \\
\mathrm{mv}_i &\leq \hat{\mathrm{mv}}_i(t) + (2+\rho)c_i(t).
\end{aligned}$$

which completes the proof. □

It should be noted that when the statement in Lemma 1 holds, we have $L_i(t) \leq \mathrm{mv}_i \leq U_i(t)\ \forall i \in \Pi, \forall t > M$. The following theorem bounds the RVR of EXERT.

Theorem 1. *For a given risk trade-off factor $\rho \geq 0$ and risk threshold κ assume that there exists an arm whose mean-variance is at most κ (otherwise the RVR is linear). When EXERT is run with $0 < \delta < 1$, with probability at least $1 - \delta$ its RVR is bounded by*

$$RVR_{\kappa,\rho}(T) \leq 5|\Pi_r| + \sum_{i \in \Pi_r} \frac{(3+2\rho)^2 4}{\Delta_i^2} \left[\log\left(\frac{2M(3+2\rho)e^{1/2}}{\Delta_i \delta}\right)\right]$$

where $\Pi_r \coloneqq \{i \in \Pi : mv_i > \kappa\}$ denotes the set of risky arms, $\Delta_i \coloneqq mv_i - mv_{i^*_{mv}}$ is the suboptimality gap of arm i, and $i^*_{mv} \in \arg\min_{i \in \Pi} mv_i$.

Proof. Since theorem states that provided RVR bound holds with a $1 - \delta$ probability, it is possible to ignore the event in which the confidence interval in Lemma 1 does not hold and instead consider the event in which it holds. Let $\mathcal{T}_{\mathrm{r}}$ denote the set of rounds where $\hat{\Pi}_{\mathrm{rf}}(t) = \emptyset$ for $t > M$. Similarly let $\mathcal{T}_{\mathrm{rf}}$ denote the set of rounds where $\hat{\Pi}_{\mathrm{rf}}(t) \neq \emptyset$ for $t > M$. We have

$$\pi(t) \in \hat{\Pi}_{\mathrm{rf}}(t) \quad \forall t \in \mathcal{T}_{\mathrm{rf}},$$

which implies:

$$\mathrm{mv}_{\pi(t)} \leq \kappa \quad \forall t \in \mathcal{T}_{\mathrm{rf}}.$$

Thus, RVR of EXERT can be written as follows:

$$\begin{aligned}
\mathrm{RVR}_{\kappa,\rho}(T) &= \sum_{t=1}^{M} \mathbb{1}(\mathrm{mv}_{\pi(t)} > \kappa) + \sum_{t\in\mathcal{T}_{\mathrm{r}}} \mathbb{1}(\mathrm{mv}_{\pi(t)} > \kappa) + \sum_{t\in\mathcal{T}_{\mathrm{rf}}} \mathbb{1}(\mathrm{mv}_{\pi(t)} > \kappa) \\
&= |\Pi_{\mathrm{r}}| + \sum_{t\in\mathcal{T}_{\mathrm{r}}} \mathbb{1}(\mathrm{mv}_{\pi(t)} > \kappa) \\
&= |\Pi_{\mathrm{r}}| + \sum_{t\in\mathcal{T}_{\mathrm{r}}} \sum_{i\in\Pi_{\mathrm{r}}} \mathbb{1}(\pi(t) = i). \qquad (4.4)
\end{aligned}$$

If arm $i \in \Pi_{\mathrm{r}}$ is selected in round $t \in \mathcal{T}_{\mathrm{r}}$, selected arm must have the lowest LCB, which implies:

$$\begin{aligned}
\hat{\mathrm{mv}}_i(t) - (1+\rho)c_i(t) &\leq \hat{\mathrm{mv}}_{i^*_{\mathrm{mv}}}(t) - (1+\rho)c_{i^*_{\mathrm{mv}}}(t) \\
\hat{\mathrm{mv}}_i(t) - (1+\rho)c_i(t) &\leq \mathrm{mv}_{i^*_{\mathrm{mv}}}.
\end{aligned}$$

Using this and $\mathrm{mv}_i - (2+\rho)c_i(t) \leq \hat{\mathrm{mv}}_i(t)$ together, we obtain

$$\begin{aligned}
&\mathrm{mv}_i - (3+2\rho)c_i(t) \leq \mathrm{mv}_{i^*_{\mathrm{mv}}} \Rightarrow \\
&c_i(t) \geq \frac{\Delta_i}{3+2\rho}. \qquad (4.5)
\end{aligned}$$

Replacing $c_i(t)$ with the value in (4.5) we get:

$$\begin{aligned}
(3+2\rho)\sqrt{\frac{1+N_i(t)}{N_i(t)^2}\left[1+2\log(\frac{2M(1+N_i(t))^{1/2}}{\delta})\right]} &\geq \Delta_i \\
(3+2\rho)^2\frac{1+N_i(t)}{N_i(t)^2}\left[1+2\log(\frac{2M(1+N_i(t))^{1/2}}{\delta})\right] &\geq \Delta_i^2 \\
(3+2\rho)^2\frac{1+N_i(t)}{N_i(t)^2-1}\left[1+2\log(\frac{2M(1+N_i(t))^{1/2}}{\delta})\right] &> \Delta_i^2 \\
\frac{(3+2\rho)^2}{\Delta_i^2}\left[1+2\log(\frac{2M(1+N_i(t))^{1/2}}{\delta})\right] &> N_i(t)-1.
\end{aligned}$$

To ease calculation, let $1 + N_i(t) = Y$. Then,

$$\begin{aligned}
&\frac{(3+2\rho)^2}{\Delta_i^2}\left[1 + 2\log(\frac{2MY^{1/2}}{\delta})\right] > Y - 2 \\
&\left[1 + 2\log(\frac{2MY^{1/2}}{\delta})\right] > \frac{\Delta_i^2}{(3+2\rho)^2}Y - 2\frac{\Delta_i^2}{(3+2\rho)^2} \\
&\log Y + 2\log\frac{2Me^{1/2}}{\delta} > \frac{\Delta_i^2}{(3+2\rho)^2}Y - 2\frac{\Delta_i^2}{(3+2\rho)^2} \\
&\log Y > \frac{\Delta_i^2}{(3+2\rho)^2}Y - 2\frac{\Delta_i^2}{(3+2\rho)^2} - 2\log\frac{2Me^{1/2}}{\delta}.
\end{aligned}$$

Proposition 4 in [52], requires $aY + b > \log Y$ for any $Y > \frac{2}{a}(\log(\frac{1}{a}) - b)$. Let $a = \frac{\Delta_i^2}{(3+2\rho)^2}$ and $b = -(2\frac{\Delta_i^2}{(3+2\rho)^2} + 2\log\frac{2Me^{1/2}}{\delta})$, then following must be true:

$$\begin{aligned}
Y &\leq \frac{2}{\frac{\Delta_i^2}{(3+2\rho)^2}}\left[\log(\frac{1}{\frac{\Delta_i^2}{(3+2\rho)^2}}) + 2\frac{\Delta_i^2}{(3+2\rho)^2} + 2\log\frac{2Me^{1/2}}{\delta}\right] \\
&\leq \frac{2}{\frac{\Delta_i^2}{(3+2\rho)^2}}\left[2\log(\frac{(3+2\rho)}{\Delta_i}) + 2\log\frac{2Me^{1/2}}{\delta}\right] + 4 \\
&\leq \frac{(3+2\rho)^2 2}{\Delta_i^2}\left[2\log(\frac{2M(3+2\rho)e^{1/2}}{\Delta_i\delta})\right] + 4 \\
&\leq \frac{(3+2\rho)^2 4}{\Delta_i^2}\left[\log(\frac{2M(3+2\rho)e^{1/2}}{\Delta_i\delta})\right] + 4
\end{aligned}$$

Plugging back $Y = 1 + N_i(t)$, a bound for $N_i(t)$ where $i \in \Pi_{\mathrm{r}}$ is obtained:

$$N_i(t) \leq \frac{(3+2\rho)^2 4}{\Delta_i^2}\left[\log(\frac{2M(3+2\rho)e^{1/2}}{\Delta_i\delta})\right] + 3. \tag{4.6}$$

Thus, if $i \in \Pi_{\mathrm{r}}$ is selected in round $t \in \mathcal{T}_{\mathrm{r}}$, then (4.6) must hold. Since $N_i(t)$ is incremented by 1 after each round in which arm i is selected, we have

$$\sum_{t\in\mathcal{T}_{\mathrm{r}}} \mathbb{1}(\pi(t) = i) \leq \frac{(3+2\rho)^2 4}{\Delta_i^2}\left[\log(\frac{2M(3+2\rho)e^{1/2}}{\Delta_i\delta})\right] + 4 \quad \forall i \in \Pi_{\mathrm{r}}.$$

Finally, we use the above inequality in (4.4) to obtain the RVR bound:

$$\begin{aligned}
\mathrm{RVR}_{\kappa,\rho}(T) &= |\Pi_{\mathrm{r}}| + \sum_{t\in\mathcal{T}_{\mathrm{r}}}\sum_{i\in\Pi_{\mathrm{r}}} \mathbb{1}(\pi(t)=i) \\
&= |\Pi_{\mathrm{r}}| + \sum_{i\in\Pi_{\mathrm{r}}}\sum_{t\in\mathcal{T}_{\mathrm{r}}} \mathbb{1}(\pi(t)=i) \\
&\leq |\Pi_{\mathrm{r}}| + \sum_{i\in\Pi_{\mathrm{r}}} \frac{(3+2\rho)^2 4}{\Delta_i^2}\left[\log(\frac{2M(3+2\rho)e^{1/2}}{\Delta_i\delta})\right] + 4 \\
&\leq 5|\Pi_{\mathrm{r}}| + \sum_{i\in\Pi_{\mathrm{r}}} \frac{(3+2\rho)^2 4}{\Delta_i^2}\left[\log(\frac{2M(3+2\rho)e^{1/2}}{\Delta_i\delta})\right].
\end{aligned}$$

□

Theorem 1 proves that the number of rounds where EXERT selects a risky arm is bounded independently of T, i.e., $O(1)$, with probability at least $1-\delta$. We can also obtain a $O(\log T)$ bound on the expected RVR of EXERT by setting $\delta = 1/T$. For $\rho = 0$ the problem becomes an online variance minimization problem, and as $\rho \to \infty$ it becomes a reward maximization problem since the contribution of variance on the mean-variance goes to 0. Increasing κ generally results a decrease in $|\Pi_{\mathrm{r}}|$, which causes EXERT to act less risk-aware. Finally, it should be noted that, $\Delta_i \geq \kappa - \mathrm{mv}_{i^*_{\mathrm{mv}}} \quad \forall i \in \Pi_{\mathrm{r}}$ and a suboptimal arm $i \in \Pi_{\mathrm{r}}$ is pulled at most $\tilde{O}(\Delta_i^{-2})$ times. Following corollary shows that RVR of EXERT can be bounded using the distance of between κ and $\mathrm{mv}_{i^*_{\mathrm{mv}}}$, instead of a function of suboptimality gaps of risky arms.

Corollary 1. *When $mv_{i^*_{mv}} < \kappa$, with probability at least $1-\delta$ the RVR of EXERT is bounded by*

$$RVR_{\kappa,\rho}(T) \leq 5|\Pi_r| + \sum_{i\in\Pi_r} \frac{(3+2\rho)^2 4}{(\kappa - mv_{i^*_{mv}})^2}\left[\log(\frac{2M(3+2\rho)e^{1/2}}{(\kappa - mv_{i^*_{mv}})\delta})\right]$$

Proof. For any $i \in \Pi_{\mathrm{r}}$ we know that $\Delta_i \geq \kappa - \mathrm{mv}_{i^*_{\mathrm{mv}}}$. Which implies for a risky arm i, $\Delta_i^{-2} \leq (\kappa - \mathrm{mv}_{i^*_{\mathrm{mv}}})^{-2}$ and $\Delta_i^{-1} \leq (\kappa - \mathrm{mv}_{i^*_{\mathrm{mv}}})^{-1}$. Starting the RVR bound in Theorem 1 and replacing the suboptimality gap terms with the distance of optimal arm to the κ, we reach the inequality. □

Δ_i is measured with respect to the best arm and it is not affected by κ, hence changing κ will not affect the RVR upper bound as long as the number of risk-free arms does not change. But a change in κ may change the actual RVR acquired by the EXERT since it may take longer or shorter for EXERT to find an arm that is risk-free with high probability. κ-sensitivity of the EXERT is investigated in Section 4.3.3.5.

While bound on the RVR depends on unknown parameters of the Safe Bandit similar to the regret bounds derived in the classical MAB [1], in practice the learner can estimate $\text{RVR}_{\kappa,\rho}(T)$ by $\sum_{n=1}^{T} \mathbb{1}(\hat{\text{mv}}_{\pi(n)}(T) > \kappa)$, where $\hat{\text{mv}}_i(T)$ is the empirical mean-variance of arm i defined in (4.2). This follows from Lemma 1 in [35], which shows that the empirical mean-variance converges to the true mean-variance exponentially fast for a class of random variables that includes random variables with bounded support.

Using $\hat{\Pi}_{\text{rf}}(t)$ decreases the number of explorations significantly, especially in cases where M is large. In Section 4.3 it is shown that EXERT achieves a competitive MVR performance, albeit it is not primarily designed for this metric. It is possible to explain such phenomenon as follows: Since EXERT chooses one arm every round, $\hat{\Pi}_{\text{rf}}(t)$ generally either is empty or contains a single arm.[4] When $|\hat{\Pi}_{\text{rf}}(t)| = 1$, EXERT only chooses this arm, and this approach is similar to how ExpExp behaves during its exploitation phase. When $|\hat{\Pi}_{\text{rf}}(t)| = 0$, either in initial rounds or when an arm leaves $\hat{\Pi}_{\text{rf}}(t)$, EXERT uses optimism under uncertainty principle and explores arms based on their LCBs, similar to the MV-LCB. In the event that $|\hat{\Pi}_{\text{rf}}(t)| > 1$, EXERT chooses the arm potentially with lowest mean-variance in estimated risk-free arm set. These varying behaviors in three different settings results in low MVR.

[4] In some cases $\hat{\Pi}_{\text{rf}}(t)$ can contain more than one arm due to the initialization phase of EXERT. This event also can occur, if EXERT is initiated with prior knowledge on the mean-variances of the arms.

4.3 Illustrative Results

In this section, we evaluate the performance of EXERT and compare it with other RAMAB algorithms (ExpExp and MV-LCB [34]) that use mean-variance as the risk notion and a risk-neutral MAB algorithm (UCB1 [1]) in both synthetic and real world datasets. In the first setting, we examine a variance minimization problem on a synthetic dataset. Then, we consider the expert selection problem on a real-world breast cancer dataset, where experts are classifiers with reject option. How to generate such classifiers using neural networks is described in Section 4.3.2.

4.3.1 Variance Minimization

In this experiment the number of arms is set to 100, and reward distributions of the arms are set as Gaussian.[5] For an arm $i \in \{1, \ldots, 100\}$, μ_i is sampled from $\mathcal{N}(\mathrm{E}[\mu_i], 0.1)$ and σ_i^2 is sampled from $\mathcal{N}(\mathrm{E}[\sigma_i^2], 0.1)$, where $\mathrm{E}[\mu_i]$ and $\mathrm{E}[\sigma_i^2]$ are given in Table 4.3.1.

Sani et al. [34] proves that $1/T^2$ is the optimum confidence term for the MV-LCB algorithm. With this confidence term, MVR bound of MV-LCB holds with at least $1 - 600/T$ probability, hence we set $\delta = 600/T$ for EXERT. With the aim of examining the variance minimization case, we set the $\rho = 0$, and $\kappa = 0.1$ which corresponds to a setting where roughly 10% of the arms are risk-free. This experiment is repeated for 50 times for T values in $\{1000, 2500, 5000, 10000, 25000, 50000, 100000\}$. Results are averaged for every T over 50 runs.

In Fig. 4.1, RVR and MVR of the algorithms as a function of T is provided. UCB1, since it is not risk-aware, has higher RVR and MVR than other algorithms. Note that it accumulates a linear RVR, because of risk-neutrality of the algorithm.

[5] Although it is assumed that the rewards are bounded, similar to [34], in numerical results Gaussian distribution is used.

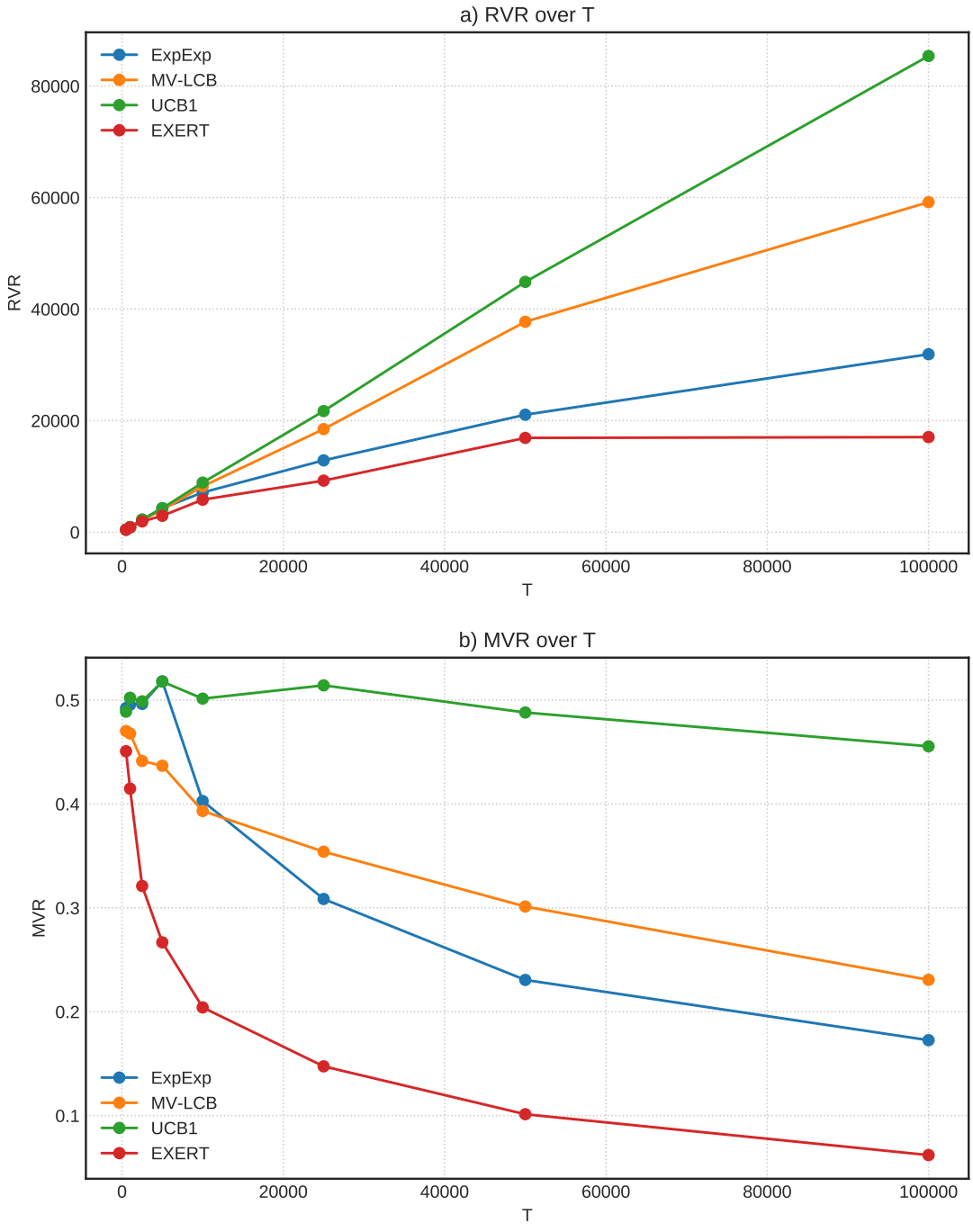

Figure 4.1: The RVR, number of risky arm selection and MVR as a function of T for the variance minimization problem for $\kappa = 0.1$.

Table 4.1: Expected Means and Variances of the Arms for the Variance Minimization problem

	$\mathrm{E}[\mu_i]$	$\mathrm{E}[\sigma_i^2]$
$i \in [1-10]$	0.5	0.05
$i \in [11-20]$	0.5	0.15
$i \in [21-30]$	0.5	0.25
$i \in [31-40]$	0.5	0.35
$i \in [41-50]$	0.5	0.45
$i \in [51-60]$	0.5	0.55
$i \in [61-70]$	0.5	0.65
$i \in [71-80]$	0.5	0.75
$i \in [81-90]$	0.5	0.85
$i \in [91-100]$	0.5	0.95

In terms of both MVR, ExpExp performs worse than MV-LCB for small values of T, and better for large values of T. Coherent with the experiments provided in [34], MVR of both ExpExp and MV-LCB decrease as T increase.

In this experiment EXERT manages to obtain the best RVR and MVR between algorithms, but the latter is not guaranteed and is a problem specific phenomena. While the RVR performance of ExpExp is close to the EXERT for small values of T, ExpExp has the highest standard deviation between the algorithms because of its explore-then-exploit strategy.

In Fig. 4.3.1, the change in RVR as a function of the risk threshold κ is presented, for $T = 10000$ and $\rho = 0$. In this experiment, since the other algorithms do not use any risk threshold, κ for EXERT is fixed to 0.1. $\mathrm{RVR}_{\kappa,\rho}(T)$ is reported for all κ values such that $\kappa \in \{\mathrm{mv}_i \ \forall i \in \Pi\}$. Note that, If $\kappa < \mathrm{mv}_{i^*_{\mathrm{mv}}}$, all algorithms would achieve a linear RVR and if $\kappa > \max_{i \in \Pi} \mathrm{mv}_i$ all algorithms would achieve zero RVR.

In this experiment, EXERT achieves the lowest RVR for all values of $\kappa > 0.03$. This is due to the fact that, κ input of EXERT is fixed to 0.1 and EXERT assumes that any arm with a mean-variance lower than 0.1 is risk-free and can be selected

without any problems. It should be noted that, as κ increase, the number of risky arms decrease, which results in lower RVR for all algorithms.

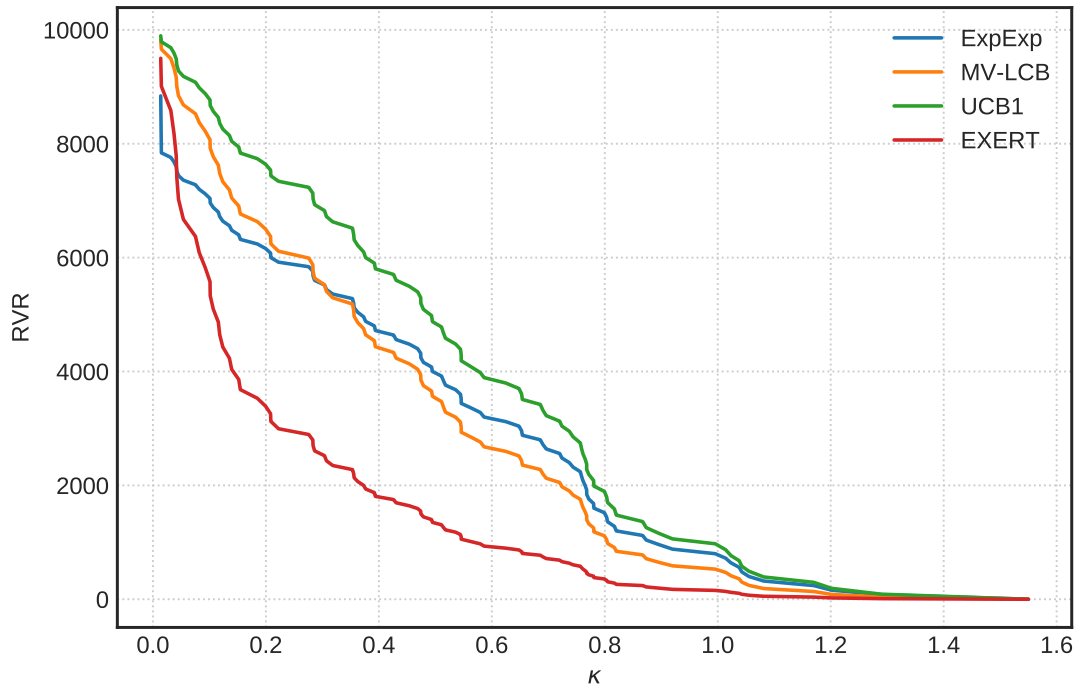

Figure 4.2: RVR at $T = 10000$ for different κ values for the variance minimization problem.

4.3.2 Expert Selection for Classification With Reject Option

4.3.2.1 Problem Definition

Consider an expert selection for binary classification problem, where on each round the learner selects an expert. Reward of the learner depends on the classification accuracy of the selected expert: it is 1 if expert selects the correct class and 0 otherwise. In such case, r_i, the reward distribution of arm i, can be written

as:

$$P(r_i = r) = \begin{cases} s_i & \text{if } r = 1 \\ 1 - s_i & \text{if } r = 0 \end{cases},$$

where $1 \geq s_i \geq 0.5$ is the probability that expert i will classify a given sample correctly. Then, mv_i can be calculated as:

$$\begin{aligned} \text{mv}_i &= s_i(1 - s_i) - \rho s_i \\ &= s_i - s_i^2 - \rho s_i \\ &= s_i(1 - s_i - \rho) \end{aligned}$$

Since $\frac{\partial \text{mv}_i}{\partial s_i}$ and $\frac{\partial^2 \text{mv}_i}{\partial s_i^2}$ is negative for all $s_i > 0.5$, increasing s_i would decrease mv_i. Hence in such scenario, the arm with highest correct classification probability is also the arm with lowest mean variance.

Consider another setting where each expert is a classifier with reject option, meaning that if classifiers do not have enough confidence in their predictions, they can reject to classify given sample. This scenario arises in cases where the cost of wrong prediction is high (false negative in cancer diagnostic) compared to the cost of rejection (asking for another expert opinion or collecting further test). Such problems can be modeled using MABs where every arm is an expert which classifies current sample only when selected by the learner. The reward of the learner is 1 if classification is correct, -1 if classification is wrong, and 0 if the selected classifier rejects to classify given sample. Similar to the case without reject option, r_i, reward distribution of arm i, can be written as:

$$P(r_i = r) = \begin{cases} s_i'(1 - q_i) & \text{if } r = 1 \\ q_i & \text{if } r = 0 \\ (1 - s_i')(1 - q_i) & \text{if } r = -1 \end{cases},$$

where q_i is the probability that expert i rejects a sample and s_i' denotes the probability that expert i will classify current sample correctly, given that classifier

did not reject the sample. We know that:

$$\begin{aligned}\mathrm{E}[r_i] &= (1-q_i)s'_i + q_i 0 + -1(1-q_i)(1-s'_i) \\ &= (1-q_i)(2s'_i - 1)\end{aligned}$$

and

$$\begin{aligned}\mathrm{E}[r_i^2] &= (1-q_i)s'_i + q_i 0 + 1(1-q_i)(1-s'_i) \\ &= (1-q_i).\end{aligned}$$

Then mean-variance of arm i, mv_i can be written as:

$$\begin{aligned}\mathrm{mv}_i &= \mathrm{Var}[r_i] - \rho\mathrm{E}[r_i] \\ &= (\mathrm{E}[r_i^2] - \mathrm{E}[r_i]^2) - \rho\mathrm{E}[r_i] \\ &= (1-q_i) - (1-q_i)^2(2s'_i-1)^2 - \rho(1-q_i)(2s'_i-1) \\ &= (1-q_i)\left[1-(1-q_i)(2s'_i-1)^2\right] - \rho(1-q_i)(2s'_i-1) \\ &= (1-q_i)\left[1-(1-q_i)(2s'_i-1)^2 + (1-2s'_i)\rho\right].\end{aligned}$$

Consider following setup, where $\rho = 0.5$, $s'_1 = 0.5$, $q_1 = 0.6$, $s'_2 = 0.6$, $q_2 = 0.5$. Then, $\mu_1 = 0$, $\mu_2 = 0.4$, $\mathrm{mv}_1 = 0.4$ and $\mathrm{mv}_2 = 0.44$, which shows that under an expert selection problem, where experts can reject to classify, the expert with highest mean does not always correspond to the expert with lowest mean-variance.

4.3.3 Training The Experts

In this experiment, each expert is a classifier with reject option. The set of available responses of classifiers is defined as $\mathcal{Y}_R = \mathcal{Y} \cup \{R\}$, where R denotes the decision where classifier rejects given sample. Consequently, classifiers are functions mapping samples to the labels or rejection, i.e., $f_i : \mathcal{X} \to \mathcal{Y}_R$. Let τ denote the rejection threshold, and X denote the random sample from the set of samples, Chow [53] shows that a classifier $f : \mathcal{X} \to \mathcal{Y}_R$ achieves the optimum

reject-error trade-off when

$$f(\mathbf{x}) = \begin{cases} \arg\max_j \mathrm{P}(j \mid \mathbf{x}) & \text{if } \max_j \mathrm{P}(j \mid \mathbf{x}) > \tau \\ R & \text{else} \end{cases} \tag{4.7}$$

where $\mathrm{P}(j \mid \mathbf{x}) \triangleq \mathrm{P}(y(\mathbf{x}) = j \mid X = \mathbf{x})$. In such model, classifier labels a given sample only if posterior probability of a label is greater than a rejection threshold. This principle is used to define neural network based classifiers. Using posterior probabilities, Eq. 4.7 can be rewritten as follows:

$$f_i(\mathbf{x}) = \begin{cases} \arg\max_j p_{i,j}(\mathbf{x}) & \text{if } \max_j p_{i,j}(\mathbf{x}) > \tau \\ R & \text{else} \end{cases}. \tag{4.8}$$

Let f_i be a neural network, then the reject notion can be integrated into the loss function used for training this neural network as:

$$\ell_i(\boldsymbol{p_i}(\mathbf{x}), y(\mathbf{x})) = \begin{cases} \ell'(\boldsymbol{p_i}(\mathbf{x}), y(\mathbf{x})) & \text{if } \max_j p_{i,j}(\mathbf{x}) > \tau \\ c_i & \text{else} \end{cases} \tag{4.9}$$

where c_i is the cost of rejection for expert i and $\ell'(\boldsymbol{p_i}(\mathbf{x}), y(\mathbf{x}))$ is the loss due to the differences between $\boldsymbol{p_i}(\mathbf{x})$ and $\boldsymbol{y}(\mathbf{x})$. Generally when training neural networks for classification task, this loss is defined as:

$$\ell'(\boldsymbol{p_i}(\mathbf{x}), y(\mathbf{x})) = -\sum_{j=1}^{J} \mathbb{1}(y(\mathbf{x}) = j) \log p_{i,j}(\mathbf{x}) \tag{4.10}$$

and is also called a log loss. In a real-world problem, while it is easy to infer c_i from the domain experts or the model it is not easy to infer τ, hence a model that can learn τ given c_i is needed. Finally, loss function of the classifier can be written as a smooth version of Eq. 4.9:

$$\begin{aligned} \mathrm{sgm}_i &\triangleq \mathrm{sigmoid}(\max_{j \in \{1,\ldots,J\}} p_{i,j}(\mathbf{x}) - \tau) \\ \ell_i(\boldsymbol{p_i}(\mathbf{x}), y(\mathbf{x})) &= \mathrm{sgm}_i \ell'(\boldsymbol{p_i}(\mathbf{x}), y(\mathbf{x})) + (1 - \mathrm{sgm}_i) c_i. \end{aligned} \tag{4.11}$$

In following sections the usage and behavior of such networks for different c_i values is presented.

4.3.3.1 Dataset Description

In this experiment Wisconsin Breast Cancer (Diagnostic) Dataset [54] is used. In this dataset, there exist 569 patients, with their relative identifiers, 30 real valued features and breast cancer diagnostics. Dataset is split into training and testing parts which include 250 and 319 samples respectively. Real valued features are scaled between [0,1] to decrease standard deviation and increase the learning speed. Each feature is scaled independently such that the highest value of a single feature becomes 1 and the lowest value becomes 0. Then, a random noise independently sampled from $\mathcal{N}(0, 0.1)$ is added to these scaled features to increase the complexity of the problem.

4.3.3.2 Training of the Experts

11 neural networks are trained in this experiment using the loss function defined in (4.11). Each of the networks are trained with 200 samples, randomly drawn from the training set, and for 200 epochs. No early stopping criterion is used. Each network received a different rejection cost as input, and as a result each network achieved different classification accuracies and reject rates. Selected rejection costs with corresponding rejection rates, classification accuracies, and learned τ values are provided in Table 4.2.

4.3.3.3 Simulation Set

50000 samples are selected with replacement from the test set. These values are scaled by using the minimum and maximum values for each feature in the training set. Similar to the creation of training dataset provided to networks, independent noise sampled from $\mathcal{N}(0, 0.1)$ is added to each feature in every sample, and the

Table 4.2: For Each Expert i, τ_i Value, Accuracy and Rejection Rate on the Simulation Set Given Rejection Cost c_i

Reject Cost c_i	Accuracy	Reject Rate	τ_i
0.205	0.947	0.305	0.977
0.210	0.941	0.221	0.945
0.215	0.929	0.178	0.913
0.220	0.922	0.144	0.880
0.225	0.920	0.125	0.847
0.230	0.913	0.109	0.814
0.235	0.908	0.091	0.779
0.240	0.905	0.077	0.745
0.245	0.903	0.069	0.710
0.250	0.899	0.056	0.675
0.255	0.892	0.042	0.64

resulting dataset is saved as the simulation set.

4.3.3.4 Results

Following the same approach in Section 4.3.1, the confidence term for MV-LCB is selected as $1/50000^2$ and δ, confidence term for EXERT, is set to $\frac{6M}{T} = 0.00132$, to make sure that regret bounds of both algorithms hold with the same probability. Once the learning algorithm selects an expert, based on the response of the selected expert it receives a reward. Reward is 0 if selected expert rejects to classify the given sample, 1 if classification is correct and -1 if the classification is wrong.

Empirical means of the experts over the 50000 samples in the simulation set is between [0.62-0.75] and empirical variances of the experts over the simulation set are found to be between [0.31-0.39]. Mean-variances to be used in MVR and RVR are calculated using the empirical mean-variances over the simulation set.

For all ρ values in $\{0, 0.1, \ldots, 1\}$ experiments are repeated for 50 times and averages over these 50 experiments are reported. Before each experiment, order of

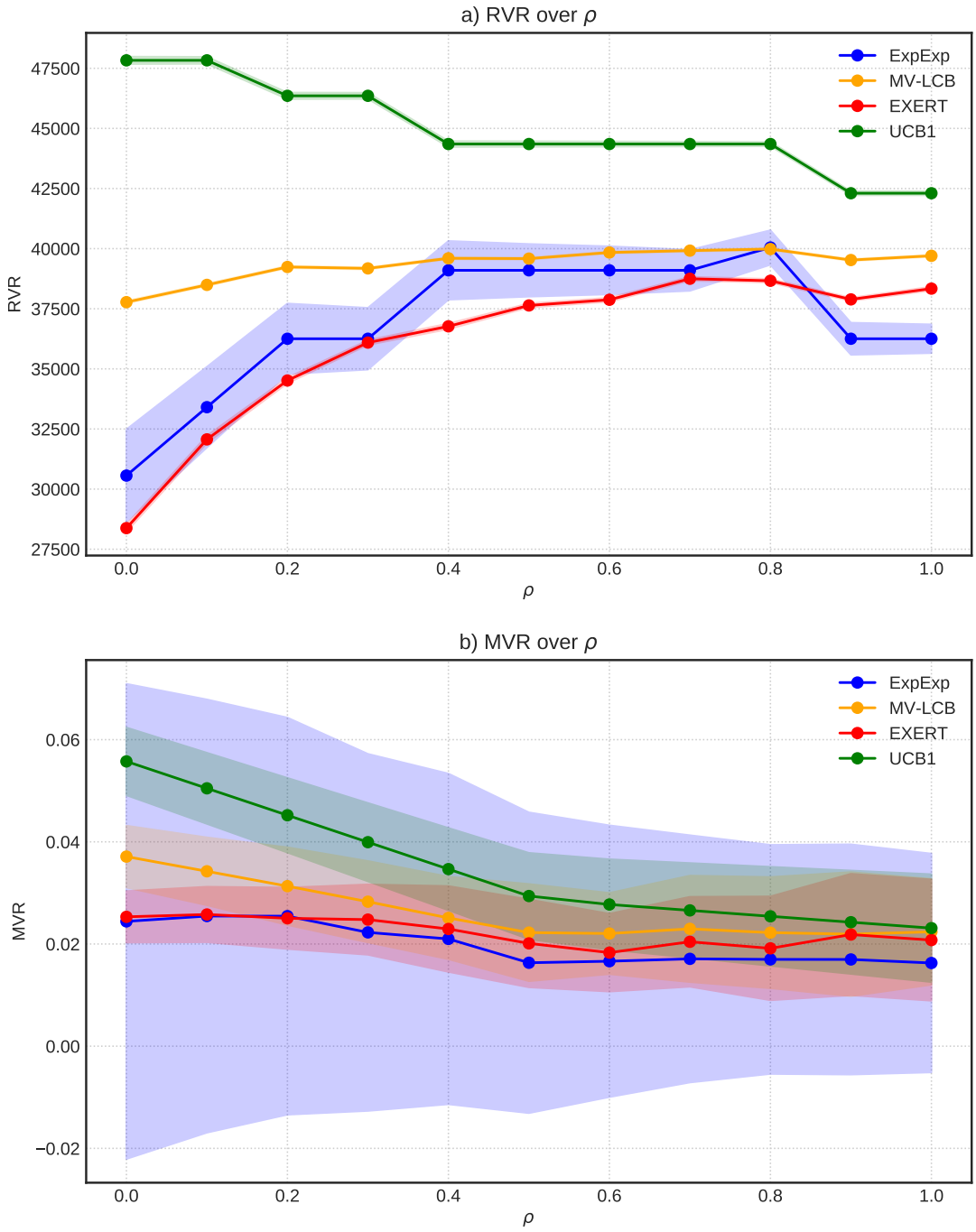

Figure 4.3: The RVR and MVR as a function of ρ.

the samples in the simulation set is randomly shuffled. For every ρ, κ is selected such that the empirical mean-variances of exactly two experts are below κ.

In Fig. 4.3 performances of 4 algorithms (UCB1, MV-LCB, ExpExp and EXERT) over the simulation set are provided. Solid lines correspond to the averages of the metric and the light shaded area correspond to the 95% confidence interval of the given metric.[6]

UCB1 incurs very high RVR and MVR, especially for small values of ρ, due to its risk-neutral nature. For larger values of ρ, MVR and RVR of UCB1 decrease and the performance of the risk-neutral algorithm gets closer to the risk-aware counterparts. It can be inferred that, benefit of using a risk-aware algorithm instead of a risk-neutral alternative diminishes as the ρ increase.

EXERT outperforms MV-LCB in terms of RVR for all values of ρ and outperforms ExpExp in terms of RVR for all values of $\rho \leq 0.8$. While minimizing MVR is not the main objective of EXERT, it performs very well in terms of this metric, too. It outperforms MV-LCB in terms of MVR for all values of ρ, which has been designed to optimize MVR, and it performs on-par in terms of MVR with ExpExp, which is another algorithm designed to optimize MVR. It should be noted that, EXERT may accumulate some MVR worse than MV-LCB during the rounds where it selects risk-free arms, especially if the κ is large, compared to the mean-variance of the best arm.

4.3.3.5 κ-Sensitivity of EXERT

$\text{RVR}_{\kappa,\rho}(T)$ and arms selected by EXERT (and their selection order) depends on the choice of κ. To investigate the effect of κ over EXERT, 10 different κ values between [0.17-0.25] are selected such that for every selected κ the number of risky arms were different. [0.17-0.25] interval corresponds to the interval of empirical mean-variances of the arms for $\rho = 0.2$. For every experiment, selected

[6]Since the metric value is real, normal distribution is assumed. 95% confidence interval is calculated as the interval covering the two standard deviation distance from the mean of the metric.

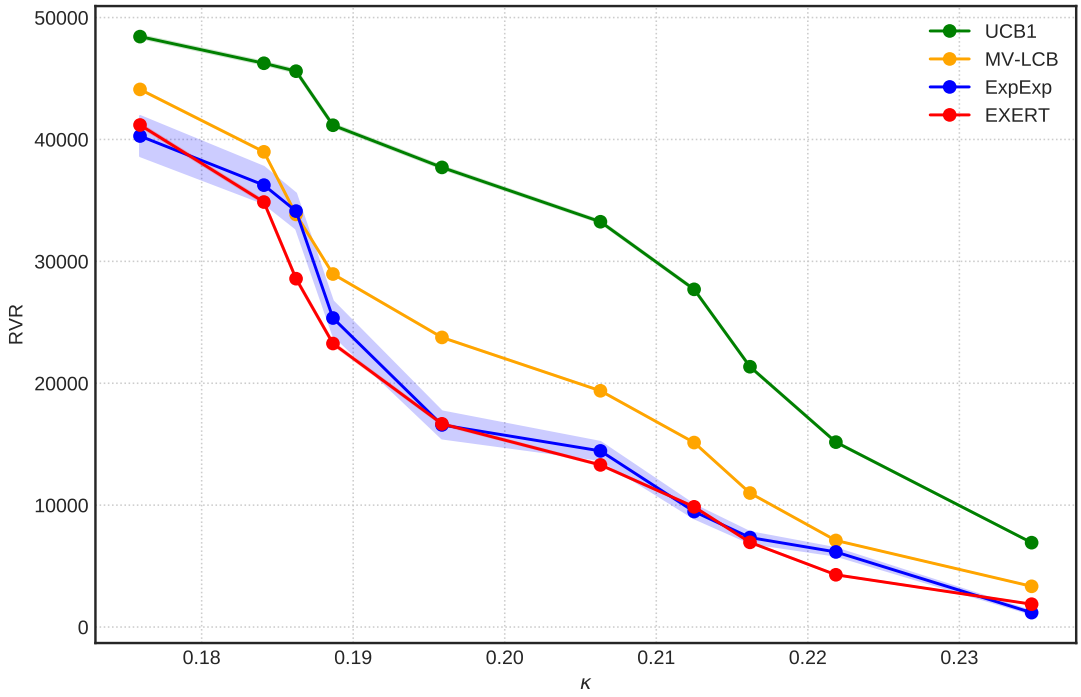

Figure 4.4: The RVR as a function of κ for $\rho = 0.2$.

κ is provided as an input to the EXERT, while other algorithms do not use κ. Results of this experiment is provided in Fig. 4.4, where solid lines correspond to the average RVR over 50 rounds, and light shaded area covers the 95% confidence interval of the metric.

In this experiment, the lowest RVR for the majority of all considered κ values is achieved by EXERT. UCB1 performs worst and MV-LCB performs second worst for all κ values. While the average performance of ExpExp is very close to the EXERT, the high variance in the RVR makes it a worse alternative, considering these algorithms principally designed to perform under risky scenarios.

Chapter 5

Conclusion and Future Work

In this thesis, two different problems in the online learning domain have been considered. In the first problem, we have examined prediction with expert advice setting where experts are trained with different parts of the sample space. We have proposed Selective WAF, a variant of the WAF algorithm that can select a subset of the experts instead of using all of them, and CS-WAF, an altered version of this selective algorithm using contextual zooming to handle the cases where experts are not uniformly knowledgeable about the sample space. In the experiments section it has been shown that CS-WAF performs better than the non-contextual version. We have also showed that when these algorithms use all experts, they perform significantly better than all the experts in the system. We are considering to extend this study by using a combinatorial bandit approach to exploit the unknown relationship between the experts. In the current work, we assume that advices of the experts are independent of each other, while in real life it is possible that such a relationship exists.

In the second problem, we have considered a special case of the Risk-Aware MAB, called the Safe Bandit. We have proposed a new model that separates the set of available arms into risky and risk-free clusters based on a threshold. We have defined a new regret measure (RVR), which is equal to the number of times the learner chooses a risky arm, and have proposed a new algorithm,

called EXERT, which minimizes the RVR by estimating the set of risky arms using upper and lower confidence bounds on the mean-variance of the arms. We have proven that RVR of EXERT is $O(1)$ with a high probability, and expected RVR of EXERT is $O(\log T)$. We have shown how neural networks can be used to train classifiers with reject option and used these neural networks to show that RAMAB algorithms can be used in an expert selection problem. Results of our experiments imply that EXERT can achieve good performance in terms of both mean-variance regret and RVR.

Our work on the Safe Bandit can be extended by assuming an unbounded noise distribution on the arm rewards. In such a case, estimation error on variances of the arms would not be sub-Gaussian, which would require a different theoretical analysis. It should also be noted that EXERT expects two different parameters κ and ρ. Due to the fact that detection of κ requires domain knowledge, a mathematical formulation that can approximate a good κ after some exploration over the rewards of the arms would greatly enhance the usage of EXERT in real world situations.

Bibliography

[1] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multi-armed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002.

[2] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32 (1):48–77, 2002.

[3] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670. ACM, 2010.

[4] A. Slivkins. Dynamic ad allocation: Bandits with budgets. *arXiv preprint arXiv:1306.0155*, 2013.

[5] L. Tran-Thanh, S. Stein, A. Rogers, and N. R. Jennings. Efficient crowd-sourcing of unknown experts using bounded multi-armed bandits. *Artificial Intelligence*, 214:89–111, 2014.

[6] Sofía S Villar, Jack Bowden, and James Wason. Multi-armed bandit models for the optimal design of clinical trials: benefits and challenges. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 30(2): 199, 2015.

[7] Liang Tang, Romer Rosales, Ajit Singh, and Deepak Agarwal. Automatic ad format selection via contextual bandits. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 1587–1594. ACM, 2013.

[8] Ricardo Lage, Ludovic Denoyer, Patrick Gallinari, and Peter Dolog. Choosing which message to publish on social networks: A contextual bandit approach. In *Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on*, pages 620–627. IEEE, 2013.

[9] Nick Littlestone and Manfred K Warmuth. The weighted majority algorithm. *Information and computation*, 108(2):212–261, 1994.

[10] H. Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952.

[11] X. Y. Zhou and D. Li. Continuous-time mean-variance portfolio selection: A stochastic LQ framework. *Applied Mathematics & Optimization*, 42(1): 19–33, 2000.

[12] D. Li and W. L. Ng. Optimal dynamic portfolio selection: Multiperiod mean-variance formulation. *Mathematical Finance*, 10(3):387–406, 2000.

[13] F. A. Roques, D. M. Newbery, and W. J. Nuttall. Fuel mix diversification incentives in liberalized electricity markets: A mean–variance portfolio theory approach. *Energy Economics*, 30(4):1831–1849, 2008.

[14] S. S. Zhu, D. Li, and S. Y. Wang. Risk control over bankruptcy in dynamic portfolio selection: A generalized mean-variance formulation. *IEEE Transactions on Automatic Control*, 49(3):447–457, 2004.

[15] Herbert A Simon. Rational choice and the structure of the environment. *Psychological review*, 63(2):129, 1956.

[16] Paul Reverdy, Vaibhav Srivastava, and Naomi Ehrich Leonard. Satisficing in multi-armed bandit problems. *IEEE Transactions on Automatic Control*, 62(8):3788–3803, 2017.

[17] Aleksandrs Slivkins. Contextual bandits with similarity information. *Journal of Machine Learning Research*, 15(1):2533–2568, 2014.

[18] William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4): 285–294, 1933.

[19] Abraham Wald. Sequential tests of statistical hypotheses. *The annals of mathematical statistics*, 16(2):117–186, 1945.

[20] Kenneth J Arrow, David Blackwell, and Meyer A Girshick. Bayes and minimax solutions of sequential decision problems. *Econometrica, Journal of the Econometric Society*, pages 213–244, 1949.

[21] Herbert Robbins. Some aspects of the sequential design of experiments. In *Herbert Robbins Selected Papers*, pages 169–177. Springer, 1985.

[22] H. Lai, T. L.and Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.

[23] Rajeev Agrawal. Sample mean based index policies by o (log n) regret for the multi-armed bandit problem. *Advances in Applied Probability*, 27(4): 1054–1078, 1995.

[24] Olivier Cappé, Aurélien Garivier, Odalric-Ambrym Maillard, Rémi Munos, Gilles Stoltz, et al. Kullback–leibler upper confidence bounds for optimal sequential allocation. *The Annals of Statistics*, 41(3):1516–1541, 2013.

[25] Michael Woodroofe. A one-armed bandit problem with a concomitant variable. *Journal of the American Statistical Association*, 74(368):799–806, 1979.

[26] Jyotirmoy Sarkar. One-armed bandit problems with covariates. *The Annals of Statistics*, pages 1978–2002, 1991.

[27] Chih-Chun Wang, Sanjeev R Kulkarni, and H Vincent Poor. Bandit problems with side observations. *IEEE Transactions on Automatic Control*, 50 (3):338–355, 2005.

[28] Tyler Lu, Dávid Pál, and Martin Pál. Contextual multi-armed bandits. In *Proceedings of the Thirteenth international conference on Artificial Intelligence and Statistics*, pages 485–492, 2010.

[29] Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 208–214, 2011.

[30] John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in neural information processing systems*, pages 817–824, 2008.

[31] Miroslav Dudik, Daniel Hsu, Satyen Kale, Nikos Karampatziakis, John Langford, Lev Reyzin, and Tong Zhang. Efficient optimal learning for contextual bandits. *arXiv preprint arXiv:1106.2369*, 2011.

[32] Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. In *International Conference on Machine Learning*, pages 1638–1646, 2014.

[33] J. Y. Audibert, R. Munos, and C. Szepesvári. Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19):1876–1902, 2009.

[34] A. Sani, A. Lazaric, and R. Munos. Risk-aversion in multi-armed bandits. In *Proc. Advances in Neural Information Processing Systems*, pages 3275–3283, 2012.

[35] S. Vakili and Q. Zhao. Risk-averse multi-armed bandit problems under mean-variance measure. *IEEE Journal of Selected Topics in Signal Processing*, 10 (6):1093–1111, 2016.

[36] O. A. Maillard. Robust risk-averse stochastic multi-armed bandits. In *Proc. International Conference on Algorithmic Learning Theory*, pages 218–233. Springer, 2013.

[37] N. Galichet, M. Sebag, and O. Teytaud. Exploration vs exploitation vs safety: Risk-aware multi-armed bandits. In *Proc. Asian Conference on Machine Learning*, pages 245–260, 2013.

[38] S. Mannor and J. N. Tsitsiklis. Mean-variance optimization in Markov decision processes. In *Proc. 28th International Conference on Machine Learning*, pages 177–184, 2011.

[39] T. M. Moldovan and P. Abbeel. Risk aversion in Markov decision processes via near optimal Chernoff bounds. In *Proc. Advances in Neural Information Processing Systems*, pages 3131–3139, 2012.

[40] T. M. Moldovan and P. Abbeel. Safe exploration in Markov decision processes. In *Proc. 29th International Conference on Machine Learning*, pages 1451–1458, 2012.

[41] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

[42] Yevgeny Seldin, Peter L Bartlett, Koby Crammer, and Yasin Abbasi-Yadkori. Prediction with limited advice and multiarmed bandits with paid observations. In *ICML*, pages 280–287, 2014.

[43] Satyen Kale. Multiarmed bandits with limited expert advice. In *Conference on Learning Theory*, pages 107–122, 2014.

[44] Dua Dheeru and Efi Karra Taniskidou. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

[45] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[46] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[47] Leo Breiman, Jerome Friedman, RA Olshen, and Charles J Stone. Classification and regression trees. 1984.

[48] Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. Physiobank, physiotoolkit, and physionet. *Circulation*, 101(23):e215–e220, 2000.

[49] Caleb Hug. *Detecting hazardous intensive care patient episodes using real-time mortality models*. PhD thesis, 2009.

[50] Luca Citi and Riccardo Barbieri. Physionet 2012 challenge: Predicting mortality of icu patients using a cascaded svm-glm paradigm. In *Computing in Cardiology (CinC), 2012*, pages 257–260. IEEE, 2012.

[51] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320, 2011.

[52] A. Antos, V. Grover, and C. Szepesvári. Active learning in heteroscedastic noise. *Theoretical Computer Science*, 411(29-30):2712–2728, 2010.

[53] C. Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16(1):41–46, 1970.

[54] M. Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.