ROBUST AND RELIABLE DECISION-MAKING SYSTEMS AND ALGORITHMS

BY

MEHMET ALI DONMEZ

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2017

Urbana, Illinois

Doctoral Committee:

Professor Andrew C. Singer, Chair
Professor Mark Hasegawa-Johnson
Associate Professor Maxim Raginsky
Assistant Professor Lav R. Varshney

ABSTRACT

We investigate robustness and reliability in decision-making systems and algorithms based on the tradeoff between cost and performance. We propose two abstract frameworks to investigate robustness and reliability concerns, which critically impact the design and analysis of systems and algorithms based on unreliable components.

We consider robustness in online systems and algorithms under the framework of online optimization subject to adversarial perturbations. The framework of online optimization models a rich class of problems from information theory, machine learning, game theory, optimization, and signal processing. This is a repeated game framework where, on each round, a player selects an action from a decision set using a randomized strategy, and then Nature reveals a loss function for this action, for which the player incurs a loss. Through a worst-case adversary framework to model the perturbations, we introduce a randomized algorithm that is provably robust even against such adversarial attacks. In particular, we show that this algorithm is Hannan-consistent with respect to a rich class of randomized strategies under mild regularity conditions.

We next focus on reliability of decision-making systems and algorithms based on the problem of fusing several unreliable computational units that perform the same task under cost and fidelity constraints. In particular, we model the relationship between the fidelity of the outcome and the cost of computing it as an additive perturbation. We analyze performance of repetition-based strategies that distribute cost across several unreliable units and fuse their outcomes. When the cost is a convex function of fidelity, the optimal repetition-based strategy in terms of minimizing total incurred cost while achieving a target mean-square error performance may fuse several computational units. For concave and linear costs, a single more reliable unit incurs lower cost compared to fusion of several lower cost and less reliable units while achieving the same mean-square error (MSE) performance. We show how our results give insight into problems from theoretical neuroscience, circuits, and crowdsourcing.

We finally study an application of a partial information extension of the cost-fidelity framework of this dissertation to a stochastic gradient descent problem, where the underlying cost-fidelity function is assumed to be unknown. We present a generic framework for trading off fidelity and cost in computing stochastic gradients when the costs of acquiring stochastic gradients of different quality are not known a priori. We consider a mini-batch oracle that distributes a limited query budget over a number of stochastic gradients and aggregates them to estimate the true gradient. Since the optimal mini-batch size depends on the unknown cost-fidelity function, we propose an algorithm, *EE-Grad*, that sequentially explores the performance of mini-batch oracles and exploits the accumulated knowledge to estimate the one achieving the best performance in terms of cost efficiency. We provide performance guarantees for EE-Grad with respect to the optimal mini-batch oracle, and illustrate these results in the case of strongly convex objectives.

To Ayca & Mert, for their love, support, and patience.

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor, Professor Andrew C. Singer, for his invaluable guidance throughout my doctoral studies. His insight, patience, and support have pushed me into exploring different ways of approaching any research problem, without which this thesis would be impossible.

I had the great pleasure of working closely with Professor Maxim Raginsky and Professor Lav R. Varshney, from whom I have received extensive guidance and help. I am also thankful to Professor Mark Hasegawa-Johnson for serving in my preliminary and final examination committees, and giving me valuable feedback.

I would also like to acknowledge the support and friendship I have received from my teammates: Noyan Sevuktekin, Gizem Tabak, Sijung Yang, Ryan Corey, and Jae Won Choi.

I cannot thank my mother Meral, my father Ahmet, and my sisters Fatma and Hatice enough for all the love, support, and encouragement they have given me throughout my entire life. Finally and most importantly, I am deeply thankful to my beautiful wife Ayca and my son Mert for being my source of motivation when things are both great and tough.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Consider consulting a medical expert in order to make a decision on a health concern, e.g., whether to accept a certain treatment or have some operation. Each medical expert has a certain level of competence associated with the *fidelity* of the advice he or she can provide, which in turn is highly correlated to the *cost* incurred to obtain the advice. Often, medical experts with higher competence may cost more than those with lower competence, where a patient seeking for advice may either spend his or her entire budget for a single and high competence medical expert, or distribute it across a number of less competent medical experts and make a final decision by aggregating their advice. Depending on the inherent decision-making problem and the nature of the cost-fidelity relationship, either approach may lead to a better result than the other in terms of the final decision quality. Similar arguments apply to the case where we consider getting investment advice subject to a limited budget. We might decide to either exhaust the entire budget on a single and expensive expert, or allocate it to several cheaper experts and fuse their outputs to make a decision. In both scenarios, the *optimal approach* in terms of final decision performance depends heavily on the underlying cost-fidelity function. Along these lines, we can also consider crowdsourcing, which assigns a single task to a number of cheaper but unreliable workers, instead of a single or smaller number of more expensive and reliable experts. In general, there is a tradeoff between cost (monetary payments, bonus) and fidelity (quality of work) in a wide range of crowdsourcing scenarios.

In this dissertation, we focus on decision-making systems and algorithms under uncertain environments from an abstract point of view. In particular, we focus on robustness and reliability, which are significant concerns for systems and algorithms built out of unreliable components in a wide range of applications including but not limited to machine learning and optimization, circuits and systems, neuroscience, crowdsourcing, communications, investment, and wireless sensor networks.

We propose and investigate two abstract frameworks to account for the robustness and reliability issues that critically impact the design and analysis of decision-making systems and algorithms subject to unreliable behavior, respectively. We first present the framework of *online optimization*, and show how it is connected to a large number of problems from different fields. We will use this framework to study robustness of online decision-making systems and algorithms under worst-case adversarial perturbations. We next propose a framework to study the fundamental cost-fidelity tradeoff inherent in decision-making engines subject to unreliable behavior. We demonstrate that this framework may have relevance in problems from a wide range of fields including circuits and systems, theoretical neuroscience, and crowdsourcing.

1.1 Online Optimization Setting

In the machine learning and optimization literature [1–8], *online optimization* has been introduced and used as an abstract framework that provides a unified approach to a number of problems including: prediction with expert advice and online classification/regression in online learning [9–13]; sequential investment and universal portfolios in mathematical finance [14–16]; universal prediction in information theory [17–19], and zero-sum repeated games in game theory [20].

To clarify the notion, we briefly describe an online optimization framework with a T-round repeated game, where on each round $t = 1, \ldots, T$, an *online player* chooses an action A_t from a set of feasible actions $\mathcal{A}$, which is treated as a comparison class. Then Nature selects a loss function $\ell_t(\cdot)$ from a class of loss functions $\mathcal{L}$, and the player suffers the loss $\ell_t(A_t)$. The goal of the online player is to minimize and control the *regret* it accumulates over T rounds with respect to the best action from the comparison class $\mathcal{A}$, which can be defined as

$$R_T \triangleq \sum_{t=1}^{T} \ell_t(A_t) - \inf_{U \in \mathcal{A}} \sum_{t=1}^{T} \ell_t(U).$$

Regret is a game-theoretic notion to assess the player's performance, which measures the difference between the cumulative performance of an online player and that of the best strategy from a class of competing strategies, the best of which can only be chosen in hindsight [13]. Under this framework, we investigate a natural question arising in different applications: Are online decision-making systems and algorithms robust against adversarial perturbations of external agents? Particularly in sequential systems, adversarial perturbations of the player's decisions can be catastrophic if not compensated properly since their effects will accumulate across iterations.

1.2 A Framework of Cost-Fidelity Tradeoff

In this framework, we consider the problem of fusing outcomes of several unreliable computational units subject to cost and fidelity constraints. We propose an analytical model for the output of an unreliable computational unit as an additive perturbation to its error-free result that captures the relationship between its fidelity and cost. In particular, suppose that a signal $\mathbf{X} \in \mathbb{R}^d$ is processed to compute some target function $f(\cdot)$ as

$$Y = f(\mathbf{X}),$$

where we model the output of any unreliable computational unit with fidelity $\theta > 0$ as

$$Z_\theta = Y + U_\theta,$$

where U_θ is a zero-mean perturbation with variance θ^{-1}. In our model, a cost $C(\theta)$ must be incurred to achieve the fidelity θ. Naturally, the cost is an increasing function of the fidelity. Note that by Chebyshev's inequality, the output Z_θ of the unreliable computational unit with fidelity $\theta > 0$ satisfies, for any $\varepsilon > 0$,

$$\Pr(|Z_\theta - Y| \geq \varepsilon) \leq \frac{1}{\varepsilon^2 \theta},$$

which suggests that the output of the unreliable unit converges to the error-free computation in probability as the fidelity increases at the expense of a larger cost. This introduces a *cost-fidelity tradeoff*, which will be extensively explored in this dissertation.

Under this framework, we study the performance of repetition-based strategies that distribute a limited cost budget across several unreliable computational units and fuse their outputs to form a final output. In many applications, the fusion operation also incurs some cost, which must also be taken into account in an effort to find the optimal approach in terms of cost-performance tradeoff.

One particular application of this framework is in modern signal processing systems based on unreliable circuit fabrics, such as nanoscale CMOS or spintronics, which exhibit a tradeoff between cost (such as area, complexity, power, or other resources) and performance (such as precision, accuracy, latency, or throughput). For instance, as CMOS technology scales beyond 10 nm, or the supply voltage scales below some threshold, the critical path lengths in a design may become too slow and their computation may not complete within a clock period, leading to static defects as well as dynamic operational non-determinism. This leads to artifacts such as process, voltage, and temperature variations, which results in unreliable behavior. Moreover, present implementations of spintronics, or electron spin-based electronics, exhibit unreliable behavior, where there is a tradeoff between reliability and energy consumption [21,22]. We emphasize that our framework of *cost-fidelity tradeoffs* also has connections to neuroscience, where typical central synapses are noisy devices, for instance, due to probabilistic transmitter release [23]. These unreliable synapses play essential roles in two principal tasks of the brain, namely, information storage and information processing.

1.3 Outline of the Dissertation

In Chapter 2, we consider robustness of online decision-making systems and algorithms under the framework of online optimization subject to adversarial perturbations. We investigate a repeated game framework where on each round, a player selects an action from a decision set using a randomized strategy, and then Nature reveals a loss function for this action, for which the player incurs a loss. The game then repeats for a total of T rounds, over which the player seeks to minimize the total incurred loss, or more specifically, the excess incurred loss with respect to a fixed comparison class.

The added challenge over traditional online optimization is that on certain rounds, which are unknown to the player, after the player selects an action, an adversarial agent perturbs this action arbitrarily. Through a worst-case adversary framework to model the perturbations, we introduce a randomized algorithm that is robust against such adversarial attacks. In particular, we show that this algorithm is Hannan-consistent with respect to a rich class of randomized strategies under mild conditions.

In Chapter 3, we turn our attention to the reliability issue, and study the problem of fusing several unreliable computational units that perform the same task under cost and fidelity constraints. Here we view any computational unit as a black box that produces results based on its unknown mechanism. More precisely, we propose an unreliable computational unit model, where instead of the error-free output, we observe a perturbed version while incurring an associated cost. We consider several cost models formalizing the relation between the fidelity of an unreliable computational unit and its cost. We analyze repetition-based strategies that distribute the cost across several unreliable units and fuse their outputs to make a final decision, and demonstrate limits of achievable performance within this framework. In particular, we demonstrate that a single and more reliable computational unit incurs less cost compared to a fusion of several less costly and less reliable computational units while achieving the same performance, under concave and linear costs. We also show that when the cost function is a convex function of fidelity, fusing several cheaper but less reliable computational units, instead of an expensive and reliable unit, may yield a better cost-performance tradeoff under certain conditions.

In Chapter 4, we consider an application of our cost-fidelity framework to a stochastic gradient descent problem, where the underlying cost-fidelity function is assumed to be unknown. In this case, the optimal repetition-based strategy is also unknown since it depends on the cost-fidelity function. In particular, we focus on an arbitrary unknown cost function satisfying some regularity conditions, and formulate an online learning problem, where we learn the optimal approach in terms of cost efficiency through sequential trials by using the paradigm of an *exploration-exploitation tradeoff*, which is heavily used the multi-armed bandit literature [24, 25]. More rigorously, we propose a novel algorithm that performs sequential trials over different repetition-based strategies, and prove that it performs almost as well as the optimal repetition-based approach in terms of cost-efficiency.

In Chapter 5, we present a number of extensions related to the problems investigated in this dissertation, and propose some new open problems. We conclude the dissertation with certain remarks in Chapter 6.

CHAPTER 2

ONLINE OPTIMIZATION UNDER ADVERSARIAL PERTURBATIONS

2.1 Introduction

In this chapter we consider the problem of *online optimization* [1,3,5] subject to adversarial perturbations. This online setting can be viewed as a repeated game between a decision maker (or player) and Nature. On each round, the player chooses a point from a decision set, possibly at random. Then, Nature reveals a loss function from a function class, and the player incurs a loss. Nature's actions are assumed to be adversarial, such that the revealed loss function sequence can even depend on the entire sequence of the player moves in a non-causal manner.

The online convex optimization framework was first introduced in [1], and has been extensively investigated in [2–8]. This framework provides a unified approach to many problems in online learning [10–13], mathematical finance [15, 16], and information theory [17–19]. In particular, recent survey papers by Hazan [26] and Shalev-Shwartz [27] show that this approach provides an abstraction for several problems including online classification and regression [9], online portfolio management [14–16], zero-sum repeated games [20], stochastic optimization [28], and online density estimation [29]. Moreover, the seminal book by Cesa-Bianchi and Lugosi [13] comprehensively studies the underlying connections between online learning, prediction, and repeated games. In particular, they demonstrate that results from these fields can be studied under the framework of prediction with expert advice. We emphasize that our repeated game framework fits into both frameworks, namely, *online convex optimization* and *prediction with expert advice*. Hence, in this sense, our results can be readily applied to problems from a number of different fields.

A natural question that arises in the frameworks of online optimization and prediction with expert advice is how various results regarding performance guarantees change if strategies are subject to adversarial actions of external agents. That is, are these strategies *robust* against adversarial environments? In this work, we introduce an extension of the online optimization problem where any online player's strategy is subject to perturbations. Here, as in the spirit of the repeated game we describe above, we study a generic model and produce results that hold in a worst-case setting, rather than assuming that such perturbations follow a stochastic model and designing players for *that* model.

We view perturbations in a player's strategy as acts of an *adversary*, who perturbs the player's strategy so as to prevent the player from achieving the goal, e.g., minimizing its cost function. We use the game-theoretic notion of *regret* to assess player's performance. Regret measures the difference between the cumulative performance of a player and that of the best strategy from a class of strategies, which can only be chosen in hindsight. In particular, since we investigate a randomized algorithm, we are interested in its expected regret. Also, the perturbation-generating mechanism of the adversary is completely unknown to the decision maker, we introduce a framework that models such perturbations from a worst-case perspective. More generally, we consider a worst-case oblivious adversary and a worst-case oblivious Nature, that is, their behavior are nonadaptive to the random decision of a player. Indeed, Cesa-Bianchi and Lugosi [13] establish that regret bounds that hold under *any* oblivious opponent hold also under an adaptive opponent who may adapt its actions based on the random decisions of a player. Hence, our results also hold against any strategies of a nonoblivious Nature and a nonoblivious adversary.

We note that this extended setting can also be seen as a repeated game, where the decision maker plays against two adversarial opponents, namely, Nature and the strategy-perturbing adversary. Evidently, performing well under this new framework is more challenging than performing under the standard setup, where the player is against only Nature. We emphasize that any perturbation in a player's strategy is especially harmful in online algorithms since uncompensated perturbations will accumulate across successive iterations, which can severely degrade the performance.

In this chapter, we propose a new randomized algorithm, which we call the *robust weighted average algorithm*. We note that there exists a deterministic version of this strategy that is computationally infeasible, which involves integrating over continuous decision sets in a high-dimensional Euclidean space [30]. We concentrate on how the worst-case expected regret of this algorithm depends on the number of rounds under our worst-case perturbation framework. Hence, this model allows us to measure how robust our strategies are against worst-case scenarios. Specifically, we prove a sub-linear worst-case expected regret bound that holds even under the worst-case adversarial perturbations and the worst-case action of Nature, when certain regularity conditions are satisfied.

2.1.1 Related Work

The work of Narayanan and Rakhlin [30] is related to ours as it investigates a random walk-based implementation of the randomized strategy that we study in this chapter. They consider a class of convex and bounded loss functions and a convex decision space, and later in [31], they extend their results to uniformly Lipschitz loss functions. However, both of their works [30, 31] focus on the problem of sampling from high-dimensional distributions and computational efficiency rather than robustness of the strategy to external agents. Our work extends and improves on [30] in the sense that we prove a worst-case regret bound under adversarial perturbations and show that this bound exactly matches the upper bound presented in [30], when there are no perturbations. Moreover, we prove that the algorithm analyzed in [30] performs poorly under our adversarial perturbations framework. We introduce a novel improved version of the algorithm that performs provably well even under worst-case scenarios.

In other related work, Weissmann [32] considers causal (sequential) filtering of a noisy sequence, where the underlying sequence is designed by a “well-informed antagonist” meaning that it may depend on past noise-free and noisy samples. He demonstrates that any deterministic filter is guaranteed to fail under some well-informed antagonist, and that there exists a randomized filter that can compete with any given finite class of filters, under every well-informed antagonist. Our work differs from his in several aspects. First, we consider a more general repeated game framework, where Nature can adversarially choose its actions based on observing the entire sequence of moves by a player in advance. Hence, the scheme of [32] can be framed as a special case of our framework. Second, in our framework, any player's strategy itself is also subject to actions of *another* adversary, who perturbs the decisions of the player in a certain worst-case manner.

More recently, Arora, et al. [33] and Cesa-Bianchi, et al. [34] consider adaptive and nonadaptive adversaries under a prediction with expert advice setting. They analyze strategies under different scenarios and specialize their results to the multi-armed bandit setting. Our results differ from theirs in the following sense. We emphasize adversarial *perturbations* of strategies, while the adversaries of this prior work are merely different versions of Nature in our setting. Hence, our extension to adversarial perturbations is novel.

2.1.2 Organization of the Chapter

The chapter is organized as follows. In Section 2.2, we present the online optimization framework. We provide the basic strategy of any randomized player and provide several performance-related quantities. We present the randomized *weighted average algorithm* and demonstrate its worst-case expected regret. In Section 2.3, we present our *worst-case adversarial perturbation framework*, where we specify a characterization of any adversary considered in this chapter. We next propose the randomized *robust weighted average algorithm* that combats adversarial perturbations by employing a local averaging scheme in Section 2.3.2. Section 2.4 provides the main results of this chapter, where we analyze the worst-case expected regret of the *robust weighted average algorithm*. We also provide some asymptotic results in Section 2.4.1, establishing Hannan consistency of this algorithm under mild regularity conditions. In Section 2.5, we present numerical experiments to illustrate our theoretical results. We conclude the chapter with certain remarks.

2.2 Problem Setup and Preliminaries

In this section we present our problem setup and some preliminary results. We first describe the online optimization setting as a repeated game between an online player and Nature. We next present a widely used player strategy, the randomized *weighted average* algorithm [30, 31]. Finally, we provide an upper bound on its worst-case expected regret under our optimization framework.

We first present the online optimization problem [1]. Let $\mathcal{X} \subset \mathbb{R}^m$ be a compact decision set with the diameter [1]

$$\operatorname{diam}(\mathcal{X}) = \sup_{x,y\in\mathcal{X}} \|x-y\| < \infty, \tag{2.1}$$

and let

$$\mathcal{L} = \{\ell : \mathcal{X} \to \mathbb{R}\}$$

be a class of uniformly Lipschitz loss functions, that is, for any $\ell \in \mathcal{L}$ we have

$$|\ell(x) - \ell(y)| \le C\|x-y\|,$$

for all $x, y \in \mathcal{X}$, where $C > 0$ is a constant. An online player produces a sequence of decisions $X^T = (X_1, \dots, X_T)$, where T is the time horizon, in a sequential manner as follows. On each round t, the player chooses a CDF W_t supported on $\mathcal{X}$ and produces its decision as

$$X_t \sim W_t.$$

Then, Nature reveals a loss function $\ell_t \in \mathcal{L}$ and the player incurs the loss $\ell_t(X_t)$. Here, we define the *strategy* of a player as a sequence of functions $S^T = (S_1, \dots, S_T)$, where

$$S_t : \mathcal{L}^{t-1} \to \mathcal{P}, \;\; S_t\big(\ell^{t-1}\big) = W_t, \;\; t = 1, \dots, T,$$

and $\mathcal{P}$ is the set of all probability distributions on $\mathcal{X}$. This generic online optimization setting is described in Algorithm 1.

We next describe the randomized weighted average (WA) algorithm [30], which is characterized by its distribution sequence

$$W^T = (W_1, \dots, W_T).$$

[1] $\|\cdot\|$ is the Euclidean norm.

Algorithm 1 Online Optimization

```
for $t = 1 : T$ do
    The player chooses a distribution $W_t$ on $\mathcal{X}$.
    The player generates $X_t \sim W_t$.
    Nature reveals a loss function $\ell_t \in \mathcal{L}$.
    The player incurs the loss $\ell_t(X_t)$.
end for
```

The algorithm picks its initial distribution W_1 such that

$$\operatorname{supp} W_1 = \mathcal{X}.$$

For each distribution W_t, we denote the corresponding density by w_t. Then, the decision of the player at round t is given by $X_{\mathrm{w},t} \sim W_t$. After initialization, the density w_t is determined in the following sequential manner:

$$w_{t+1}(x) = \frac{w_t(x) \exp(-\eta \ell_t(x))}{Z_t}, \quad x \in \mathcal{X}, \tag{2.2}$$

for all $t = 1, \ldots, T$, where $\eta > 0$ is the learning rate and

$$Z_t \triangleq \int_{\mathcal{X}} dW_t(u) \exp(-\eta \ell_t(u)) = \mathbb{E}[\exp(-\eta \ell_t(X_{\mathrm{w},t}))]$$

is the normalization term. After N rounds, for any $N \leq T$, the cumulative loss of the online player is defined as

$$L_N^{(\mathrm{o})}\left(X_{\mathrm{w}}^N; \ell^N\right) \triangleq \sum_{t=1}^{N} \ell_t(X_{\mathrm{w},t}).$$

Note that from (2.2), the density w_t can also be written as

$$w_t(x) = \frac{w_1(x) \exp\Big(-\eta L_{t-1}^{(\mathrm{o})}(x^{t-1}; \ell^{t-1})\Big)}{\int_{\mathcal{X}} dW_1(u) \exp\Big(-\eta L_{t-1}^{(\mathrm{o})}(u^{t-1}; \ell^{t-1})\Big)}, \quad x \in \mathcal{X},$$

where u^{t-1} is a constant sequence with the value u, for any $u \in \mathcal{X}$. Intuitively, the randomized WA algorithm chooses its distribution such that it puts *more* measure to the points in $\mathcal{X}$ that incurs *less* cumulative loss up to the round t by using a certain exponential mapping. We remark that this distribution is known as the *Boltzmann-Gibbs distribution* in statistical mechanics, where

Algorithm 2 Online Randomized WA Algorithm (η)

```
**Input:** A learning rate $\eta > 0$.
**Initialization:** Pick $W_1$ such that $\operatorname{supp} W_1 = \mathcal{X}$.
**for** $t = 1 : T$ **do**
    The player generates $X_{\mathrm{w},t} \sim W_t$ (with the density $w_t$).
    Nature reveals a loss function $\ell_t \in \mathcal{L}$.
    The player incurs the loss $\ell_t(X_{\mathrm{w},t})$ and updates $w_t$:
        $w_{t+1}(x) = \dfrac{w_t(x)\exp(-\eta\ell_t(x))}{\int_{\mathcal{X}} dW_t(u)\exp(-\eta\ell_t(u))}, \ \forall x \in \mathcal{X}.$
**end for**
```

it is used as a probability distribution of particles in a system over different states [35]. A description of this algorithm is given in Algorithm 2.

To introduce the *regret* [13], we first define a *comparison class* $\mathcal{C}$ as a set of probability distributions with the sample space $\mathcal{X}$. We characterize a "stationary P-strategy" as a strategy producing decisions U_t as i.i.d draws from a distribution $P \in \mathcal{C}$ on all rounds. That is,

$$U_t \overset{\text{i.i.d.}}{\sim} P$$

for all $t = 1, \dots, T$. We define the cumulative loss of a stationary P-strategy as

$$L_T^{(\mathrm{s})}\big(U^T; \ell^T, P\big) \triangleq \sum_{t=1}^{T} \ell_t(U_t).$$

Informally, the player's goal is to do almost as well as the best stationary randomized strategy in the comparison class $\mathcal{C}$ even if it could observe the entire loss function sequence $\ell_1, \dots, \ell_T$ ahead of time. Note that the best fixed randomized strategy can only be chosen in hindsight. Formally, given a sequence of loss functions ℓ^T, we define the *regret* [5, 13] with respect to a stationary P-strategy, $P \in \mathcal{C}$, as

$$R_T^{(\mathrm{o})}\big(X^T, U^T; \ell^T, P\big) \triangleq L_T^{(\mathrm{o})}\big(X^T; \ell^T\big) - L_T^{(\mathrm{s})}\big(U^T; \ell^T, P\big).$$

Since the decisions are randomized, we are particularly interested in the expected regret. We assume that Nature is *oblivious*, that is, the sequence

ℓ^T is chosen by Nature ahead of time without observing the random actions of the player. However, we will investigate the performance of the player under *any* loss function sequence so that our bound will hold even in the worst-case scenario. In particular, the player's goal is to guarantee that the worst-case *expected regret*

$$\mathbb{E}\Big[R_T^{(\mathrm{o})}\big(X^T;\mathcal{L},\mathcal{C}\big)\Big] \triangleq \sup_{\ell^T\in\mathcal{L}^T}\sup_{P\in\mathcal{C}}\mathbb{E}\Big[R_T^{(\mathrm{o})}\big(X^T,U^T;\ell^T,P\big)\Big]$$

is sublinear in T, where sublinearity is defined as follows.

Definition 2.2.1. *A function $f:\mathbb{Z}\to\mathbb{R}$ is sublinear in N if for any $c>0$, there exists N_0 such that $f(N)\leq cN$ for any $N\geq N_0$. See [36] for a thorough discussion.*

More generally, when the time horizon T is allowed to be unbounded, the player's goal is to achieve *Hannan consistency*, which is formally defined as follows.

Definition 2.2.2. *Any player strategy that satisfies*

$$\mathbb{E}\Big[R_T^{(\mathrm{o})}\big(X^T;\mathcal{L},\mathcal{C}\big)\Big] = o(T)$$

is said to be Hannan-consistent with respect to the comparison class $\mathcal{C}$; see Hannan's paper [37] and the book [13] for a detailed discussion.

We consider a particular comparison class of distributions on $\mathcal{X}$ to investigate the worst-case expected regret of the WA algorithm. We fix a parameter $r>0$, and let $\mathcal{P}(r)$ denote the set of all probability distributions P on $\mathcal{X}$, such that

$$D_{\mathrm{KL}}(P\|W_1)\leq r$$

(a "ball" of radius r around W_1 using the Kullback-Leibler divergence), that is,

$$\mathcal{P}(r)\triangleq\{P\in\mathcal{P}:D_{\mathrm{KL}}(P\|W_1)\leq r\}.$$

We first state and prove a lemma, which will be useful in the proof of Theorem 2.2.1.

Lemma 2.2.1. *Given any learning rate $\eta > 0$, the expected regret of the WA algorithm satisfies*[2]

$$\mathbb{E}\Big[R_T^{(\mathrm{o})}\big(X_{\mathrm{w}}^T;\ell^T,P\big)\Big] \leq \frac{1}{\eta}(D_{\mathrm{KL}}(P\|W_1) - D_{\mathrm{KL}}(P\|W_{T+1})) \\ + \frac{\eta T (C\,\mathrm{diam}(\mathcal{X}))^2}{8}.$$

Proof. To prove the desired result, we first write

$$\begin{aligned} \eta\,\mathbb{E}[\ell_t(X_t)] &= \eta \int_{\mathcal{X}} dW_t(x)\ell_t(x) \\ &= \int_{\mathcal{X}} dW_t(x)\ln(\exp(\eta\ell_t(x))) \\ &= \int_{\mathcal{X}} dW_t(x)\ln\!\left(\frac{w_t(x)}{w_{t+1}(x)}\right) - \ln(Z_t), \end{aligned} \tag{2.3}$$

which can be rewritten as

$$\begin{aligned} \int_{\mathcal{X}} dW_t(x)\ln\!\left(\frac{w_t(x)}{w_{t+1}(x)}\right) &= \eta\,\mathbb{E}[\ell_t(X_t)] + \ln(Z_t) \\ &= \ln\!\Big(\mathbb{E}\big[e^{-\eta(\ell_t(X_t)-\mathbb{E}[\ell_t(X_t)])}\big]\Big). \end{aligned}$$

Here, we note that by the Lipschitz continuity of the loss function $\ell_t(\cdot)$ and by the boundedness of the set $\mathcal{X}$, we have

$$\max_{x\in\mathcal{X}} \ell_t(x) - \min_{x\in\mathcal{X}} \ell_t(x) \leq C\,\mathrm{diam}(\mathcal{X}),$$

which implies that

$$\ell_t(X_t) - \mathbb{E}[\ell_t(X_t)]$$

is a zero-mean random variable supported on an interval of length at most $C\,\mathrm{diam}(\mathcal{X})$. Then, by Hoeffding's lemma [13], we obtain

$$\ln\!\Big(\mathbb{E}\big[e^{-\eta(\ell_t(X_t)-\mathbb{E}[\ell_t(X_t)])}\big]\Big) \leq \frac{(\eta C\,\mathrm{diam}(\mathcal{X}))^2}{8}.$$

When combined with (2.3), this result implies that

$$\mathbb{E}[\ell_t(X_t)] \leq -\frac{1}{\eta}\ln(Z_t) + \frac{\eta (C\,\mathrm{diam}(\mathcal{X}))^2}{8}. \tag{2.4}$$

[2] For any $P, Q \in \mathcal{P}$, $D_{\mathrm{KL}}(P\|Q)$ is the Kullback-Leibler divergence.

Following similar steps as in (2.3), we next write

$$\begin{aligned}\mathbb{E}[\ell_t(U_t)] &= \frac{1}{\eta}\int_{\mathcal{X}} dP(x)\ln\!\left(\frac{w_t(x)}{w_{t+1}(x)}\right) - \frac{\ln(Z_t)}{\eta}\\ &= -\frac{\ln(Z_t)}{\eta} - \frac{1}{\eta}\int_{\mathcal{X}} dP(x)\ln\!\left(\frac{dP(x)}{dW_t(x)}\right)\\ &+ \frac{1}{\eta}\int_{\mathcal{X}} dP(x)\ln\!\left(\frac{dP(x)}{dW_{t+1}(x)}\right)\\ &= -\frac{\ln(Z_t)}{\eta} - \frac{1}{\eta}(D_{\mathrm{KL}}(P\|W_t) - D_{\mathrm{KL}}(P\|W_{t+1})). \qquad (2.5)\end{aligned}$$

Hence, by summing (2.4) and (2.5) over $t = 1, \ldots, T$, we get

$$\begin{aligned}&\mathbb{E}\Big[R_T^{(\mathrm{o})}\big(X^T; \ell^T, P\big)\Big]\\ &\le \frac{1}{\eta}(D_{\mathrm{KL}}(P\|W_1) - D_{\mathrm{KL}}(P\|W_{T+1})) + \frac{T\eta(C\,\mathrm{diam}(\mathcal{X}))^2}{8}.\end{aligned}$$

□

In Theorem 2.2.1, we present a worst-case regret upper bound for the randomized WA algorithm and show that this algorithm is Hannan consistent with respect to the comparison class $\mathcal{P}(r)$, when the learning rate is chosen properly.

Theorem 2.2.1. *The worst-case expected regret of the WA algorithm satisfies*

$$\mathbb{E}\Big[R_T^{(\mathrm{o})}\big(X_{\mathrm{w}}^T; \mathcal{L}, \mathcal{P}(r)\big)\Big] \le \frac{r}{\eta} + \frac{T\eta(C\,\mathrm{diam}(\mathcal{X}))^2}{8},$$

for a learning rate $\eta > 0$ and $r > 0$. In particular, if the learning rate satisfies

$$\eta = O\Big(1/\sqrt{T}\Big),$$

then it follows that

$$\mathbb{E}\Big[R_T^{(\mathrm{o})}\big(X_{\mathrm{w}}^T; \mathcal{L}, \mathcal{P}(r)\big)\Big] = o(T).$$

Proof. We first note that by Lemma 2.2.1, for any distribution $P \in \mathcal{P}(r)$ and a loss function sequence $\ell^T \in \mathcal{L}^T$, the regret of the randomized WA algorithm satisfies

$$\mathbb{E}\Big[R_T^{(\mathrm{o})}\big(X^T;\ell^T,P\big)\Big] \leq \frac{1}{\eta}(D_{\mathrm{KL}}(P\|W_1) - D_{\mathrm{KL}}(P\|W_{T+1})) \\ + \frac{T\eta(C\,\mathrm{diam}(\mathcal{X}))^2}{8}. \tag{2.6}$$

By definition of $\mathcal{P}(r)$ and non-negativity of the KL divergence,

$$D(P\|W_1) - D(P\|W_{T+1}) \leq D(P\|W_1) \leq r,$$

for any $P \in \mathcal{P}(r)$. Therefore, we get

$$\mathbb{E}\Big[R_T^{(\mathrm{o})}\big(X^T;\mathcal{L},\mathcal{P}(r)\big)\Big] \leq \frac{r}{\eta} + \frac{T\eta(C\,\mathrm{diam}(\mathcal{X}))^2}{8}.$$

Moreover, if the learning rate η satisfies $\eta = O\Big(1/\sqrt{T}\Big)$, then it follows that

$$\frac{r}{\eta} + \frac{T\eta(C\,\mathrm{diam}(\mathcal{X}))^2}{8} = O\Big(\sqrt{T}\Big).$$

This yields the desired result

$$\mathbb{E}\Big[R_T^{(\mathrm{o})}\big(X^T;\mathcal{L},\mathcal{P}(r)\big)\Big] = o(T).$$

□

In this section, we provided the online regret minimization framework of this chapter, and presented the randomized WA algorithm. After demonstrating certain preliminary results, we showed that the worst-case expected regret of the randomized WA algorithm against all stationary P-strategies, $P \in \mathcal{P}(r)$, is sublinear in T, when the learning rate of the algorithm is chosen properly. In the next section, we will first introduce the worst-case perturbation framework. We will next show that the performance of the randomized WA algorithm can be arbitrarily poor in the presence of our adversarial perturbation model. We will propose a novel extension of this algorithm and demonstrate that it is robust to perturbations in its strategy under certain regularity conditions. That is, we will prove that the worst-case expected regret of the proposed algorithm is sublinear in T even under worst-case adversarial perturbations, when certain conditions are satisfied.

2.3 Online Optimization under Adversarial Perturbations

In this section, we first present our worst-case distribution perturbation framework to model the perturbations in a randomized player's strategy. We next introduce a new randomized decision strategy, the robust weighted average (WA) algorithm, which is robust to perturbations in a certain sense, as detailed in Section 2.4. This algorithm employs a local averaging technique to alleviate effects of perturbations on the player's regret, rather than explicitly trying to detect and fix the perturbations. In particular, we will show that this algorithm performs provably well even in the worst case.

2.3.1 Worst-Case Adversarial Perturbation Framework

Here, we propose a framework to model adversarial perturbations, where we view perturbations in the randomized player's strategy as actions of an adversary. We assume that the goal of the adversary is to maximize the expected regret. In this adversarial model, the goal is to better capture realistic adversarial environments and produce results that hold even under worst-case scenarios.

We first describe of our worst-case distribution perturbation model. An adversary $\mathcal{A}_k$ with k perturbations is characterized by the following two sequences:

- a distribution sequence

$$\Pi^k = (\Pi_1, \ldots, \Pi_k), \tag{2.7}$$

  defined over the set $\mathcal{X}$, i.e., $\Pi_j \in \mathcal{P}$ for $j = 1, \ldots, k$, with the corresponding densities $\pi^k = (\pi_1, \ldots, \pi_k)$,

- a sequence of time instants

$$\tau^k = (\tau_1, \ldots, \tau_k) \in \mathcal{Z}, \tag{2.8}$$

  where $\mathcal{Z} \subset \mathcal{N}^k$, $\mathcal{N} \triangleq \{1, \ldots, T\}$, is the set of monotonically increasing sequences (of length k) of the form

$$\mathcal{Z} = \big\{(t_1, \ldots, t_k) \in \mathcal{N}^k : \; t_j > t_{j-1} + 1, \; j = 2, \ldots, k\big\},$$

i.e., no elements of any sequence in $\mathcal{Z}$ are allowed to be consecutive.

We will denote any adversary as $\mathcal{A}_k\big(\Pi^k, \tau^k\big)$, which operates as follows. At each time instant τ_j, the adversary perturbs the player's distribution W_t to a new distribution Π_j (or, equivalently, w_t to π_j) for $j = 1, \ldots, k$. We observe that perturbing the distribution W_t is equivalent to reseting the algorithm to a new initial distribution. The adversary repeats the same process for all $j = 1, \ldots, k$. We call the resulting algorithm the randomized *Perturbed* WA (PWA) algorithm, and describe it in Algorithm 3.

Remark 2.3.1. *We make two observations regarding extreme cases. At one extreme, the number of perturbations is $k = 0$. In this case, the adversary does not perturb the algorithm, and the repeated game proceeds as usual. This yields the original randomized WA algorithm. At the other extreme, the number of perturbations is $k = T$. That is, the adversary perturbs the player's decisions on all rounds. It follows that the player's strategy has nothing to do with the final decisions, so the adversary may potentially disturb the player's entire strategy and maximize its expected regret. In this sense, our adversarial perturbations framework models a wide range of adversarial behavior.*

We next define some relevant performance measures and the *worst-case adversary*. First, we partition the time instants $\mathcal{N}$ into $k + 1$ disjoint sets as follows:

$$\mathcal{N}_j \triangleq \{\tau_j + 1, \ldots, \tau_{j+1}\}, \; j = 0, \ldots, k, \tag{2.9}$$

where we let $\tau_0 = 0$ and $\tau_{k+1} = T$. Hence, we have $\mathcal{N} = \cup_{j=0}^{k} \mathcal{N}_j$. Second, we define the total loss of the randomized PWA algorithm over the partition $\mathcal{N}_j$ as

$$\tilde{L}^{(j)}(X_{\mathrm{p}}; \ell) \triangleq \sum_{t=\tau_j+1}^{\tau_{j+1}} \ell_t(X_{\mathrm{p},t}),$$

where $X_{\mathrm{p},t}$ is the decision of the randomized PWA algorithm at time t. Then, the cumulative loss of the randomized PWA algorithm after T rounds can be

Algorithm 3 Online Randomized PWA Algorithm (η)

```
Input: A learning rate $\eta > 0$.
Initialization: Pick $W_1$ such that $\operatorname{supp} W_1 = \mathcal{X}$.
for $t = 1 : T$ do
    The player draws $X_{\mathrm{p},t} \sim W_t$ (with the density $w_t$).
    Nature reveals a loss function $\ell_t \in \mathcal{L}$.
    The player incurs the loss $\ell_t(X_{\mathrm{p},t})$ and updates $w_t$:
    $w_{t+1}(x) = \dfrac{w_t(x)\exp(-\eta\ell_t(x))}{\int_{\mathcal{X}} dW_t(u)\exp(-\eta\ell_t(u))}, \forall x \in \mathcal{X}.$
    if $t == \tau_j$ for some $j = 1, \ldots, k$ then
        The distribution is perturbed by an adversary:
        $W_{t+1} \leftarrow \Pi_j$,
        $w_{t+1} \leftarrow \pi_j$.
    end if
end for
```

defined as

$$L_T^{(\mathrm{o,p})}\big(X_{\mathrm{p}}^T; \ell^T, \tau^k, \Pi^k\big) \triangleq \sum_{j=0}^{k} \tilde{L}^{(j)}(X_{\mathrm{p}}; \ell).$$

Finally, using the definition of the worst-case expected regret, we can define the worst-case expected regret of the randomized PWA algorithm when subject to perturbations of an adversary as

$$\mathbb{E}\Big[R_T^{(\mathrm{o,p})}\big(X_{\mathrm{p}}^T; \mathcal{L}, \mathcal{P}(r), \tau^k, \Pi^k\big)\Big] \triangleq \sup_{\ell^T \in \mathcal{L}^T} \sup_{P \in \mathcal{P}(r)} \mathbb{E}\Big[R_T^{(\mathrm{o,p})}\big(X_{\mathrm{p}}^T; \ell^T, P, \tau^k, \Pi^k\big)\Big].$$

We define the *worst-case adversary* $\mathcal{A}_k^{\mathrm{w}} \triangleq \mathcal{A}_k\big(\Pi_{\mathrm{w}}^k, \tau_{\mathrm{w}}^k\big)$ as an adversary with:

- the distribution sequence Π_{w}^k, the *worst-case perturbation distributions*, that satisfies

  $$\Pi_{\mathrm{w}}^k = \arg\max_{\Pi^k \in \mathcal{P}^k} \mathbb{E}\Big[R_T^{(\mathrm{o,p})}\big(X_{\mathrm{p}}^T; \mathcal{L}, \mathcal{P}(r), \tau^k, \Pi^k\big)\Big], \qquad (2.10)$$

  given any sequence of time instants $\tau^k \in \mathcal{Z}$,

- the sequence of perturbation time instants τ_{w}^k, the *worst-case time in-*

stants, that satisfies

$$\begin{aligned}\tau^k_{\mathrm{w}} &= \arg\max_{\tau^k\in\mathcal{Z}} \max_{\Pi^k\in\mathcal{P}^k} \mathbb{E}\Big[R^{(\mathrm{o,p})}_T\big(X^T_{\mathrm{p}};\mathcal{L},\mathcal{P}(r),\tau^k,\Pi^k\big)\Big]\\ &= \arg\max_{\tau^k\in\mathcal{Z}} \mathbb{E}\Big[R^{(\mathrm{o,p})}_T\big(X^T_{\mathrm{p}};\mathcal{L},\mathcal{P}(r),\tau^k,\Pi^k_{\mathrm{w}}\big)\Big]. \qquad (2.11)\end{aligned}$$

Hence, the worst-case adversary's goal is to perturb the distribution of the algorithm such that its worst-case expected regret is maximized. More compactly, we denote the worst-case expected regret of the algorithm when subject to perturbations of the worst-case adversary $\mathcal{A}^{\mathrm{w}}_k$ as

$$\mathbb{E}\Big[R^{(\mathrm{o,p})}_T\big(X^T_{\mathrm{p}};\mathcal{L},\mathcal{P}(r),\mathcal{A}^{\mathrm{w}}_k\big)\Big] \equiv \mathbb{E}\Big[R^{(\mathrm{o,p})}_T\big(X^T_{\mathrm{p}};\mathcal{L},\mathcal{P}(r),\tau^k_{\mathrm{w}},\Pi^k_{\mathrm{w}}\big)\Big].$$

To illustrate the capabilities of adversaries of this framework, we consider a particular adversary $\mathcal{A}_k\big(\Pi^k_*,\tau^k\big)$ with the following distribution sequence:

$$\begin{aligned}\Pi^k_* &= (\Pi_{*,1},\ldots,\Pi_{*,k}),\\ \Pi_{*,j} &\triangleq \delta_{x^*_j},\ \forall j=1,\ldots,k,\end{aligned}$$

where x^*_j is given by

$$x^*_j = \arg\max_{u\in\mathcal{X}} \sum_{t=\tau_j+1}^{\tau_{j+1}} \ell_t(u),$$

and $\delta_{x^*_j}$ is the Dirac delta distribution[3] concentrated at x^*_j. Note that at the beginning of each time interval $\mathcal{N}_j$, the algorithm's distribution is perturbed to the distribution $\delta_{x^*_j}$ that puts all the measure on the single point x^*_j, for each $j=1,\ldots,k$. However, since the update rule (2.2) for the density w_t is multiplicative, the randomized PWA algorithm gets stuck at the distribution $\delta_{x^*_j}$ until the next perturbation time. It follows that $X_{\mathrm{p},t}\sim\delta_{x^*_j}$ for all $t\in\mathcal{N}_j$,

[3] For any point $x\in\mathcal{X}$, the Dirac delta distribution δ_x concentrated at x is defined as

$$\delta_x(A) = \begin{cases}1 & \text{if } x\in A\\ 0 & \text{if } x\notin A,\end{cases} \qquad (2.12)$$

for any Borel set $A\subseteq\mathcal{X}$.

i.e.,

$$X_{\mathrm{p},t} = x_j^*,\ \forall t = \tau_j + 1, \ldots, \tau_{j+1},\ \forall j = 1, \ldots, k,$$

with probability one. Hence, we have

$$\tilde{L}^{(j)}(X_{\mathrm{p}}; \ell) = \max_{u \in \mathcal{X}} \sum_{t=\tau_j+1}^{\tau_{j+1}} \ell_t(u),\ \forall j = 1, \ldots, k. \tag{2.13}$$

Then, the expected cumulative loss of the randomized PWA algorithm satisfies

$$\begin{aligned}\mathbb{E}\Big[L_T^{(\mathrm{o,p})}\big(X_{\mathrm{p}}^T; \ell^T, \Pi_*, \tau_k\big)\Big] &= \sum_{j=0}^{k} \mathbb{E}\Big[\tilde{L}^{(j)}(X_{\mathrm{p}}; \ell)\Big] \\ &= \sum_{j=1}^{k} \sum_{t=\tau_j+1}^{\tau_{j+1}} \ell_t\big(x_j^*\big),\end{aligned}$$

for any loss sequence $\ell^T \in \mathcal{L}^T$, when subject to perturbations of the adversary $\mathcal{A}_k\big(\Pi_*^k, \tau^k\big)$. Therefore, the worst-case expected regret of the algorithm satisfies

$$\begin{aligned}&\mathbb{E}\Big[R_T^{(\mathrm{o,p})}\big(X_{\mathrm{p}}^T; \mathcal{L}, \mathcal{P}(r), \Pi_*, \tau_k\big)\Big] \\ &\geq \sup_{\ell^T \in \mathcal{L}^T} \sup_{P \in \mathcal{P}(r)} \left\{ \sum_{j=1}^{k} \sum_{t=\tau_j+1}^{\tau_{j+1}} \ell_t\big(x_j^*\big) - \sum_{t=1}^{T} \ell_t(U_t) \right\}.\end{aligned}$$

We conclude that the worst-case expected regret performance of the randomized PWA algorithm can be arbitrarily poor in the presence of perturbations according to our model.

In this section, we first presented our worst-case distribution perturbation framework to model any adversary's actions from a worst-case perspective. We next showed a lower bound on how poor the performance of the randomized WA algorithm can be under this framework. In the next section, we will propose an algorithm we call the randomized robust WA algorithm. This algorithm is an improved version of the randomized WA algorithm so as to mitigate the effects of adversarial perturbations.

Remark 2.3.2. *One potential application of the worst-case adversarial per-*

turbation approach of this chapter is in the design of signal processing systems based on nanoscale beyond-CMOS circuit fabrics. As CMOS technology scales beyond 10 nm*, the operation of standard CMOS transistors begins to suffer from static defects as well as dynamic operational non-determinism [38]. Therefore, deeply scaled CMOS-based systems need to be capable of operating in the presence of both transient and fixed hardware errors [39]. Moreover, we emphasize that the computational errors caused by hardware defects may be catastrophic in online systems since the computation is performed recursively, so that the errors that are not corrected or compensated will propagate across successive iterations, leading to poor performance. This suggests that the adversarial perturbation model introduced in this chapter may be useful in modeling these computational errors, where the non-ideal computational fabric and the errors it causes can be perceived as an adversary and its actions, respectively. Hence, the designer can guarantee satisfactory performance even under the worst-case computational errors by utilizing the robust algorithm design approach proposed in this chapter.*

2.3.2 Randomized Robust Weighted Average Algorithm

In this section, we propose the randomized robust weighted average (RWA) algorithm, an extended version of the randomized WA algorithm to perform well under adversarial perturbations. To this end, this algorithm employs a local averaging scheme after it updates its distribution on each round to alleviate the effects of perturbations.

An explicit description of the randomized RWA algorithm is as follows. The algorithm maintains two different distributions:

- the intermediate distribution W_t (with density w_t)
- the actual distribution M_t (with density μ_t)

The algorithm chooses its initial intermediate distribution W_1 such that $\operatorname{supp} W_1 = \mathcal{X}$, and sets $\mathrm{M}_1 = W_1$. On each round t, the algorithm produces its decision as

$$X_{\mathrm{r},t} \sim \mathrm{M}_t. \tag{2.14}$$

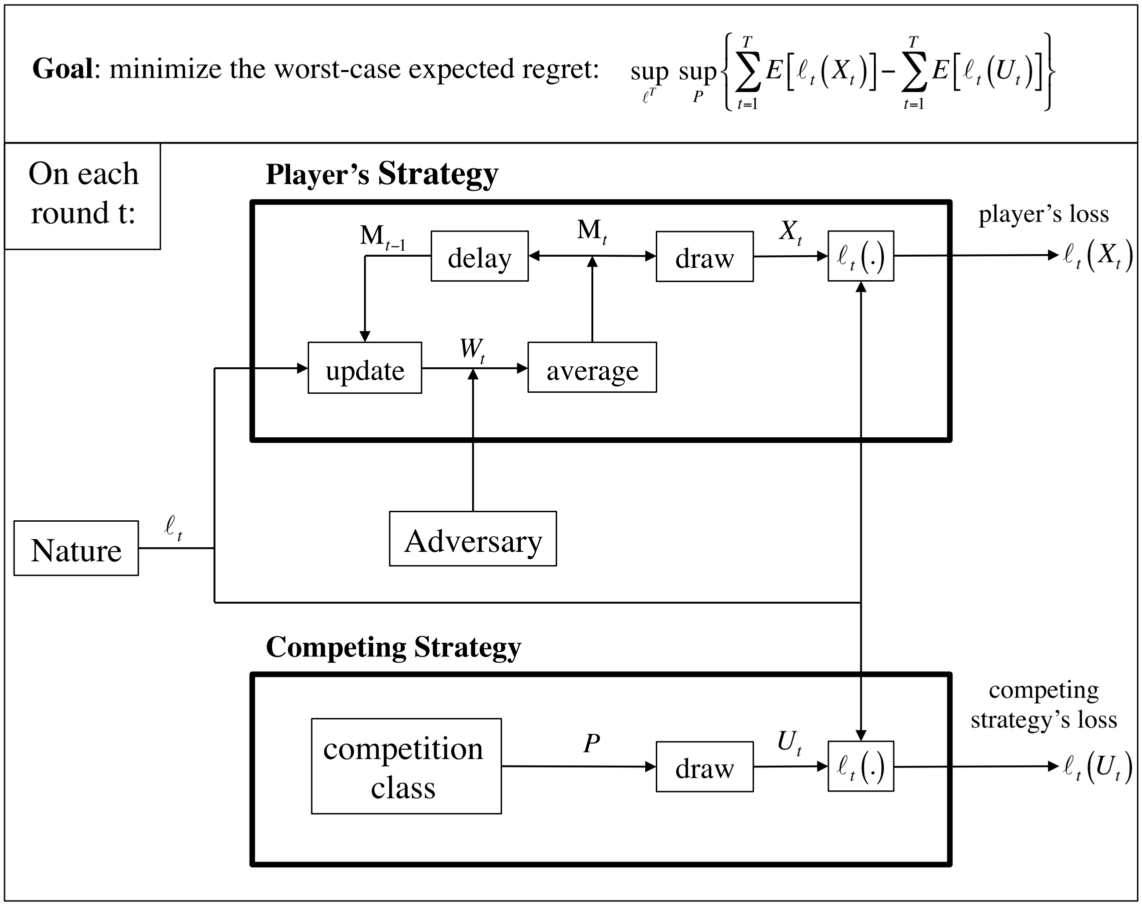


Figure 2.1: The randomized RWA algorithm subject to adversarial perturbations.

Then, Nature reveals its loss function $\ell_t(\cdot) \in \mathcal{L}$, and the player incurs the loss $\ell_t(X_{\mathrm{r},t})$. In response, the algorithm performs the update

$$w_{t+1}(x) = \frac{\mu_t(x)\exp(-\eta\ell_t(x))}{Z_t}, \tag{2.15}$$

for all $x \in \mathcal{X}$, where

$$Z_t \triangleq \int_{\mathcal{X}} d\mathrm{M}_t(u)\exp(-\eta\ell_t(u)) = \mathbb{E}[\exp(-\eta\ell_t(X_{\mathrm{r},t}))]$$

for all $t = 1, \ldots, T$. After this update, the intermediate distribution W_t is subject to perturbations of an adversary $\mathcal{A}_k\big(\Pi^k, \tau^k\big)$, as explained in Section 2.3.1. After this stage, the algorithm employs a local averaging scheme with time-varying averaging parameter $0 < \gamma_t < 1$. This scheme is assumed to be error-free. At each time t, this algorithm computes the weighted average of the two most recent values of the intermediate distribution W_t to evaluate the distribution M_t:

$$\mathrm{M}_{t+1} = \gamma_{t+1} W_{t+1} + (1 - \gamma_{t+1}) W_t. \tag{2.16}$$

We observe that (2.16) guarantees that $\mathrm{M}_{t+1} \in \mathcal{P}$ since $W_t, W_{t+1} \in \mathcal{P}$ and $\gamma_t \in (0,1)$ for all $t = 1, \ldots, T$. We present a block diagram description of this algorithm in Fig. 2.1, and a corresponding pseudocode in Algorithm 4.

We note that the averaging scheme in (2.16) incorporates the *new information*, i.e., W_{t+1}, gained after Nature reveals the loss function $\ell_t(\cdot)$ into the *history*, which is summarized in W_t, in order to protect the algorithm against perturbations. When the distribution W_{t+1} is perturbed, we observe that the distribution W_t is not perturbed, since the perturbation time instants are not allowed to be consecutive. Hence, the actual distribution M_{t+1} contains some information regarding the loss function sequence revealed in the previous rounds. When, on the other hand, that the distribution W_t is perturbed, W_{t+1} is not perturbed, so that M_{t+1} loses the past information while keeping the information gained on the round t, which is passed to the next rounds to improve performance.

We note that the choice of the averaging parameter γ_t is important for the performance of the randomized RWA algorithm. We let

$$\alpha^{T+1} = (\alpha_1, \ldots, \alpha_{T+1})$$

be a strictly decreasing sequence such that $\alpha_t \in (0,1)$ for all $t = 1, 2, \ldots, T+1$. At each time t, we define

$$\gamma_t = \frac{\alpha_{t+1}}{\alpha_t} \in (0,1),$$

where

$$0 < \Gamma_{\mathrm{l}} \leq \gamma_t \leq \Gamma_{\mathrm{u}} < 1.$$

Here the parameters Γ_{l} and Γ_{u} control the rate of decrease of the sequence α^{T+1}.

When subject to perturbations of the adversary $\mathcal{A}_k(\Pi_k, \tau_k)$, the cumulative expected loss of the randomized RWA algorithm after T rounds is defined as

$$\mathbb{E}\Big[L_T^{(\mathrm{o,p})}\big(X_\mathrm{r}^T;\ell^T,\Pi^k,\tau^k\big)\Big] \triangleq \sum_{t=1}^{T}\mathbb{E}[\ell_t(X_{\mathrm{r},t})].$$

Moreover, we define the worst-case expected regret of the randomized RWA algorithm as

$$\begin{aligned}&\mathbb{E}\Big[R_T^{(\mathrm{o,p})}\big(X_\mathrm{r}^T;\mathcal{L},\mathcal{P}(r),\Pi^k,\tau^k\big)\Big]\\ &\triangleq \sup_{\ell^T\in\mathcal{L}^T}\bigg\{\mathbb{E}\Big[L_T^{(\mathrm{o,p})}\big(X_\mathrm{r}^T;\ell^T,\Pi^k,\tau^k\big)\Big] - \inf_{P\in\mathcal{P}(r)}\mathbb{E}\Big[L_T^{(\mathrm{s})}\big(U^T;\ell^T,P\big)\Big]\bigg\}.\end{aligned}$$

For notational convenience, we denote the worst-case expected regret of the randomized RWA algorithm as

$$\mathbb{E}\Big[R_T^{(\mathrm{o,p})}\big(X_\mathrm{r}^T;\mathcal{L},\mathcal{P}(r),\mathcal{A}_k^\mathrm{w}\big)\Big] \equiv \mathbb{E}\Big[R_T^{(\mathrm{o,p})}\big(X_\mathrm{r}^T;\mathcal{L},\mathcal{P}(r),\Pi_\mathrm{w}^k,\tau_\mathrm{w}^k\big)\Big]$$

when subject to perturbations by the worst-case adversary $\mathcal{A}_k^\mathrm{w}$.

We note that, at each perturbation time instant τ_j, the intermediate distribution W_{t+1} (and the density w_t) is set to the distribution Π_j (and to the density π_j), for any $j = 1,\ldots,k$. We perceive this as the algorithm "losing" the information of the intermediate density w_{τ_j+1} for all $j = 1,\ldots,k$. We can express the "lost" density at each perturbation time $t = \tau_j$ as

$$f_j(x) \triangleq \frac{\mu_{\tau_j}(x)\exp\big(-\eta\ell_{\tau_j}(x)\big)}{Z_{\tau_j}},\ \forall x\in\mathcal{X}, \tag{2.17}$$

for each $j = 1,\ldots,k$.

In this section, we introduced the worst-case perturbation framework to model perturbations in the player's strategy as actions of an adversarial agent. We presented the randomized RWA algorithm subject to adversarial perturbations. This algorithm employs a local averaging scheme to mitigate adversarial effects of perturbations. We will next provide an upper bound on the worst-case cumulative expected regret of this algorithm under the worst-case scenario. In particular, we will show that the worst-case expected regret of the randomized RWA algorithm is sublinear in T even under the worst-case adversary, when some mild conditions are satisfied.

Algorithm 4 Online Randomized RWA Algorithm (η)

```
Input: A learning rate $\eta > 0$,
    A sequence $\gamma^T$ with $0 < \Gamma_\mathrm{l} \leq \gamma_t \leq \Gamma_\mathrm{u} < 1$, $\forall t$.
Initialization: Pick $W_1 = \mathrm{M}_1$ with $\operatorname{supp} W_1 = \mathcal{X}$.
for $t = 1 : T$ do
    The player draws $X_{\mathrm{r},t} \sim \mathrm{M}_t$, where $\mu_t$ is its density.
    Nature reveals a loss function $\ell_t \in \mathcal{L}$.
    The player incurs the loss $\ell_t(X_{\mathrm{r},t})$ and updates $w_t$:
    $w_{t+1}(x) = \dfrac{\mu_t(x)\exp(-\eta\ell_t(x))}{\int_{\mathcal{X}} d\mathrm{M}_t(u)\exp(-\eta\ell_t(u))}$, $\forall x \in \mathcal{X}$.
    if $t == \tau_j$ for some $j = 1, \dots, k$ then
        The distribution is perturbed by an adversary:
        $W_{t+1} \leftarrow \Pi_j$,
        $w_{t+1} \leftarrow \pi_j$.
    end if
    The algorithm computes the actual distribution:
    $\mathrm{M}_{t+1} = \gamma_{t+1} W_{t+1} + (1 - \gamma_{t+1}) W_t$.
end for
```

2.4 Worst-Case Expected Regret Analysis

In this section, we investigate the worst-case regret performance of the randomized RWA algorithm introduced in Section 2.3.2 in the presence of worst-case adversarial perturbations. We present an upper bound on the worst-case expected regret of the randomized RWA algorithm. We prove results that hold under *any* adversary of the form $\mathcal{A}_k\big(\Pi^k, \tau^k\big)$, so that they also hold under the worst-case adversary $\mathcal{A}_k^\mathrm{w}$. We first provide the following lemma:

Lemma 2.4.1. *Given any learning rate $\eta > 0$, the expected loss of the randomized RWA algorithm at any time t satisfies*

$$\mathbb{E}[\ell_t(X_{\mathrm{r},t})] \leq -\frac{1}{\eta}\ln(Z_t) + \frac{\eta(C\operatorname{diam}(\mathcal{X}))^2}{8}.$$

Proof. The proof is given in Appendix A.1. □

We next state and prove the main theorem of this section. This theorem provides an upper bound on the worst-case expected regret of the randomized RWA algorithm, when subject to the perturbations of the worst-case adversary. Later, we will use this theorem to prove that under certain conditions, the randomized RWA algorithm is *Hannan-consistent*.

Theorem 2.4.1. *Suppose that the randomized RWA algorithm is subject to the perturbations of the worst-case adversary* $\mathcal{A}_k\big(\Pi_{\mathrm{w}}^k, \tau_{\mathrm{w}}^k\big)$*, characterized by* (2.10) *and* (2.11)*. Then, for any learning rate* $\eta > 0$*, the worst-case expected regret of this algorithm satisfies*

$$\mathbb{E}\Big[R_T^{(\mathrm{o,p})}\big(X_{\mathrm{r}}^T; \mathcal{L}, \mathcal{P}(r), \mathcal{A}_k^{\mathrm{w}}\big)\Big] \leq \frac{r}{\eta} + \frac{T\eta(C\,\mathrm{diam}(\mathcal{X}))^2}{8} \\ + kF_\eta(\Gamma_{\mathrm{l}}, \Gamma_{\mathrm{u}}), \tag{2.18}$$

where

$$F_\eta(\Gamma_{\mathrm{l}}, \Gamma_{\mathrm{u}}) \triangleq C\,\mathrm{diam}(\mathcal{X}) + \frac{1}{\eta}\ln\bigg(\frac{1}{1-\Gamma_{\mathrm{u}}}\bigg) \\ + \frac{1}{\eta}\ln\bigg(\Gamma_{\mathrm{u}} + \exp(\eta C\,\mathrm{diam}(\mathcal{X}))\frac{1-\Gamma_{\mathrm{l}}}{\Gamma_{\mathrm{l}}}\bigg). \tag{2.19}$$

Remark 2.4.1. *We observe that the upper bound in* (2.18) *on the worst-case expected regret of the randomized RWA algorithm when subject to perturbations of the worst-case adversary is composed of two parts. The first part,*

$$\frac{r}{\eta} + \frac{T\eta(C\,\mathrm{diam}(\mathcal{X}))^2}{2},$$

is the upper bound in Theorem 2.2.1 on the worst-case expected regret of the randomized WA algorithm that is not subject to any perturbations. In this sense, the second part of the upper bound in (2.18)*,*

$$kF_\eta(\Gamma_{\mathrm{l}}, \Gamma_{\mathrm{u}}),$$

can be seen as an upper bound on the "extra" regret resulting from the perturbations by the worst-case adversary, which is an extension of the randomized WA algorithm where the only modification is the local averaging scheme in (2.16)*. Moreover, since* $kF_\eta(\Gamma_{\mathrm{l}}, \Gamma_{\mathrm{u}})$ *is a scaled version of* $F_\eta(\Gamma_{\mathrm{l}}, \Gamma_{\mathrm{u}})$*, scaled by the number of perturbations, we can perceive* $F_\eta(\Gamma_{\mathrm{l}}, \Gamma_{\mathrm{u}})$ *as an upper bound on the "cost" of any single perturbation to the algorithm in terms of the worst-case expected regret.*

Proof. Given a sequence of loss functions $\ell^T \in \mathcal{L}^T$ and a distribution $P \in$

$\mathcal{P}(r)$, let $U_t \overset{\text{i.i.d.}}{\sim} P$ for $t = 1, \dots, T$. To prove (2.18), we first derive an upper bound on $\mathbb{E}[\ell_t(X_{\mathrm{r},t})]$ for each t. We then sum these bounds to get an upper bound on the cumulative expected loss of the randomized RWA algorithm. Our analysis is based on $\mathbb{E}[\ell_t(X_{\mathrm{r},t})]$ for three different cases:

1. $t \neq \tau_j$ and $t \neq \tau_j + 1$ for any $j = 1, \dots, k$
2. $t = \tau_j$ for some $j = 1, \dots, k$
3. $t = \tau_j + 1$ for some $j = 1, \dots, k$

We first note that for each $t = 1, \dots, T$, we can write

$$\begin{aligned}
-\frac{1}{\eta}\ln(Z_t) &= -\frac{1}{\eta}\ln(Z_t) - \ell_t(U_t) + \ell_t(U_t) \qquad (2.20)\\
&= -\frac{1}{\eta}\ln(Z_t) + \frac{1}{\eta}\ln(\exp(-\eta\ell_t(U_t))) + \ell_t(U_t)\\
&= \frac{1}{\eta}\ln\left(\frac{\exp(-\eta\ell_t(U_t))}{Z_t}\right) + \ell_t(U_t)\\
&= \ell_t(U_t) + \frac{1}{\eta}\ln\left(\frac{\mu_t(U_t)\exp(-\eta\ell_t(U))}{\mu_t(U_t)Z_t}\right)\\
&= \ell_t(U_t) + \frac{1}{\eta}\ln\left(\frac{\mu_t(U_t)\exp(-\eta\ell_t(U))}{Z_t}\right) - \frac{1}{\eta}\ln(\mu_t(U_t)),
\end{aligned}$$

where in (2.20) we added and subtracted $\ell_t(U_t)$. Hence, by Lemma 2.4.1, we get

$$\begin{aligned}
\mathbb{E}[\ell_t(X_{\mathrm{r},t})] \le \ell_t(U_t) &+ \frac{1}{\eta}\ln\left(\frac{\mu_t(U_t)\exp(-\eta\ell_t(U_t))}{Z_t}\right)\\
&- \frac{1}{\eta}\ln(\mu_t(U_t)) + \frac{\eta(C\,\mathrm{diam}(\mathcal{X}))^2}{8}.
\end{aligned} \tag{2.21}$$

- Case 1: $t \neq \tau_j$ and $t \neq \tau_j + 1$ for any $j = 1, \dots, k$:
  In this case, we note that

  $$w_{t+1}(U_t) = \frac{\mu_t(U_t)\exp(-\eta\ell_t(U_t))}{Z_t},$$

  so that (2.21) is equivalent to

  $$\mathbb{E}[\ell_t(X_{\mathrm{r},t})] \le \ell_t(U_t) + \frac{1}{\eta}(\ln(w_{t+1}(U_t)) - \ln(\mu_t(U_t)))$$

$$+\frac{\eta(C\,\mathrm{diam}(\mathcal{X}))^2}{8}. \tag{2.22}$$

Due to the local averaging in (2.16), we have

$$\ln(\mu_t(U_t)) \geq \ln(\gamma_t) + \ln(w_t(U_t)), \tag{2.23}$$

since $1-\gamma_t \geq 0$ and $w_{t-1}(U_t) \geq 0$. Hence, by using (2.22) and (2.23), we can upper-bound $\mathbb{E}[\ell_t(X_{\mathrm{r},t})]$ as

$$\begin{aligned}\mathbb{E}[\ell_t(X_{\mathrm{r},t})] \leq \ell_t(U_t) &+ \frac{1}{\eta}(\ln(w_{t+1}(U_t)) - \ln(w_t(U_t))) \\ &+ \frac{1}{\eta}(\ln(\alpha_t) - \ln(\alpha_{t+1})) + \frac{\eta(C\,\mathrm{diam}(\mathcal{X}))^2}{8},\end{aligned}$$

where we used $\gamma_t = \alpha_{t+1}/\alpha_t$. We take expectations of both sides and get

$$\begin{aligned}&\mathbb{E}[\ell_t(X_{\mathrm{r},t})] \\ &\leq \mathbb{E}[\ell_t(U_t)] + \frac{1}{\eta}(\mathbb{E}[\ln(w_{t+1}(U_t))] - \mathbb{E}[\ln(w_t(U_t))]) \\ &+ \frac{1}{\eta}(\ln(\alpha_t) - \ln(\alpha_{t+1})) + \frac{\eta(C\,\mathrm{diam}(\mathcal{X}))^2}{8}.\end{aligned} \tag{2.24}$$

- Case 2: $t = \tau_j$ for some $j = 1, \ldots, k$:
  In this case, the density of the intermediate distribution, w_{t+1}, is perturbed to π_j. Therefore, we get

  $$\frac{\mu_t(U_t)\exp(-\eta\ell_t(U_t))}{Z_t} = f_j(U_t). \tag{2.25}$$

  As in the first case, we can write

  $$\ln(\mu_t(U_t)) \geq \ln(\gamma_t) + \ln(w_t(U_t)). \tag{2.26}$$

  Hence, by using (2.21), (2.25) and (2.26), the expected loss $\mathbb{E}[\ell_t(X_{\mathrm{r},t})]$ is upper bounded as

  $$\mathbb{E}[\ell_t(X_{\mathrm{r},t})] \leq \ell_t(U_t) + \frac{1}{\eta}(\ln(f_j(U_t)) - \ln(w_t(U_t)))$$

$$+\frac{1}{\eta}(\ln(\alpha_t)-\ln(\alpha_{t+1}))+\frac{\eta(C\operatorname{diam}(\mathcal{X}))^2}{8}.$$

By taking the expectation of both sides, we obtain

$$\begin{aligned}&\mathbb{E}[\ell_t(X_{\mathrm{r},t})]\\&\leq\mathbb{E}[\ell_t(U_t)]+\frac{1}{\eta}(\mathbb{E}[\ln(f_j(U_t))]-\mathbb{E}[\ln(w_t(U_t))])\\&+\frac{1}{\eta}(\ln(\alpha_t)-\ln(\alpha_{t+1}))+\frac{\eta(C\operatorname{diam}(\mathcal{X}))^2}{8}.\end{aligned}\tag{2.27}$$

- Case 3: $t=\tau_j+1$ for some $j=1,\ldots,k$:

  In this case, the density of the intermediate distribution, w_t, is perturbed to π_j. From the local averaging, we can write

  $$\mu_t(U_t)\geq(1-\gamma_t)w_{t-1}(U_t),$$

  since $\gamma_t\geq0$ and $\pi_j(U_t)\geq0$. This yields

  $$\ln(\mu_t(U_t))\geq\ln(1-\gamma_t)+\ln(w_{t-1}(U_t)).\tag{2.28}$$

  Therefore, by using (2.21) and (2.28), we get

  $$\begin{aligned}\mathbb{E}[\ell_t(X_{\mathrm{r},t})]\leq\ell_t(U_t)&+\frac{1}{\eta}(\ln(w_{t+1}(U_t))-\ln(w_{t-1}(U_t)))\\&+\frac{1}{\eta}\ln\left(\frac{1}{1-\gamma_t}\right)+\frac{\eta(C\operatorname{diam}(\mathcal{X}))^2}{8}.\end{aligned}$$

  We take expectation of both sides to get

  $$\begin{aligned}&\mathbb{E}[\ell_t(X_{\mathrm{r},t})]\\&\leq\mathbb{E}[\ell_t(U_t)]+\frac{1}{\eta}(\mathbb{E}[\ln(w_{t+1}(U_t))]-\mathbb{E}[\ln(w_{t-1}(U_t))])\\&+\frac{1}{\eta}\ln\left(\frac{1}{1-\gamma_t}\right)+\frac{\eta(C\operatorname{diam}(\mathcal{X}))^2}{8}.\end{aligned}\tag{2.29}$$

Hence, we have an upper bound on the expected loss of the randomized RWA algorithm for each time $t=1,\ldots,T$. We next sum these upper bounds over each set $\mathcal{N}_j$ to get upper bounds on $\mathbb{E}\left[\tilde{L}^{(j)}(X;\ell)\right]$ for each $j=0,\ldots,k$, which will be used to find a final upper bound on the cumulative expected loss. There are three cases depending on j.

1. Upper bound $\mathbb{E}\Big[\tilde{L}^{(0)}(X;\ell)\Big]$:
   From (2.24) and (2.27), we obtain

$$\begin{aligned}\mathbb{E}\Big[\tilde{L}^{(0)}(X;\ell)\Big] &= \sum_{t=1}^{\tau_1} \mathbb{E}[\ell_t(X_{\mathrm{r},t})] \\ &\le \sum_{t=1}^{\tau_1} \mathbb{E}[\ell_t(U_t)] + \frac{1}{\eta}(\mathbb{E}[\ln(f_1(U_{\tau_1}))] - \mathbb{E}[\ln(w_1(U_1))]) \\ &+ \frac{1}{\eta}\ln\bigg(\frac{\alpha_1}{\alpha_{\tau_1+1}}\bigg) + \tau_1 \frac{\eta(C\,\mathrm{diam}(\mathcal{X}))^2}{8}. \end{aligned} \tag{2.30}$$

2. Upper bound $\mathbb{E}\Big[\tilde{L}^{(j)}(X;\ell)\Big]$ for $j = 1,\ldots,k-1$:
   In this case, from (2.24), (2.27) and (2.29), we obtain

$$\begin{aligned}\mathbb{E}\Big[\tilde{L}^{(j)}(X;\ell)\Big] &= \sum_{t=\tau_j+1}^{\tau_{j+1}} \mathbb{E}[\ell_t(X_{\mathrm{r},t})] \\ &\le \sum_{t=\tau_j+1}^{\tau_{j+1}} \mathbb{E}[\ell_t(U_t)] + \frac{1}{\eta}\ln\bigg(\frac{1}{1-\gamma_{\tau_j+1}}\bigg) \\ &+ \frac{1}{\eta}\mathbb{E}\Bigg[\ln\Bigg(\frac{f_{j+1}\big(U_{\tau_{j+1}}\big)}{w_{\tau_j}\big(U_{\tau_j}\big)}\Bigg)\Bigg] + \frac{1}{\eta}\ln\bigg(\frac{\alpha_{\tau_j+2}}{\alpha_{\tau_{j+1}+1}}\bigg) \\ &+ (\tau_{j+1}-\tau_j)\frac{\eta(C\,\mathrm{diam}(\mathcal{X}))^2}{8}. \end{aligned} \tag{2.31}$$

3. Upper bound $\mathbb{E}\Big[\tilde{L}^{(k)}(X;\ell)\Big]$:
   From (2.24) and (2.29), we obtain

$$\begin{aligned}\mathbb{E}\Big[\tilde{L}^{(k)}(X;\ell)\Big] &= \sum_{t=\tau_k+1}^{T} \mathbb{E}[\ell_t(X_{\mathrm{r},t})] \\ &\le \sum_{t=\tau_k+1}^{T} \mathbb{E}[\ell_t(U_t)] + \frac{1}{\eta}\ln\bigg(\frac{1}{1-\gamma_{\tau_k+1}}\bigg) \\ &+ \frac{1}{\eta}\mathbb{E}\bigg[\ln\bigg(\frac{w_{T+1}(U_T)}{w_{\tau_k}(U_{\tau_k})}\bigg)\bigg] + \frac{1}{\eta}\ln\bigg(\frac{\alpha_{\tau_k+2}}{\alpha_{T+1}}\bigg) \\ &+ (T-\tau_k)\frac{\eta(C\,\mathrm{diam}(\mathcal{X}))^2}{8}. \end{aligned} \tag{2.32}$$

From (2.30), (2.31), and (2.32), an upper-bound on the expected cumulative loss can be obtained as

$$\mathbb{E}\Big[L_T^{(\mathrm{o,p})}\big(X^T;\ell^T,\Pi^k,\tau^k\big)\Big] \leq \sum_{t=1}^{T}\mathbb{E}[\ell_t(U_t)] + \frac{1}{\eta}\,\mathbb{E}\bigg[\ln\bigg(\frac{w_{T+1}(U_T)}{w_1(U_1)}\bigg)\bigg] + \frac{1}{\eta}\sum_{j=1}^{k}\mathbb{E}\Bigg[\ln\Bigg(\frac{f_j\big(U_{\tau_j}\big)}{w_{\tau_j}\big(U_{\tau_j}\big)}\Bigg)\Bigg] + \frac{1}{\eta}\sum_{j=1}^{k}\ln\bigg(\frac{1}{1-\gamma_{\tau_j+1}}\bigg) + \frac{T\eta(C\,\mathrm{diam}(\mathcal{X}))^2}{8}, \tag{2.33}$$

where in (2.33), we used that α_t is strictly decreasing. Moreover, since we have $\Gamma_\mathrm{l} \leq \gamma_t \leq \Gamma_\mathrm{u}$, we obtain

$$\frac{1}{\eta}\sum_{j=1}^{k}\ln\bigg(\frac{1}{1-\gamma_{\tau_j+1}}\bigg) \leq \frac{k}{\eta}\ln\bigg(\frac{1}{1-\Gamma_\mathrm{u}}\bigg). \tag{2.34}$$

From (2.33) and (2.34), we obtain

$$\mathbb{E}\Big[L_T^{(\mathrm{o,p})}\big(X_\mathrm{r}^T;\ell^T,\Pi^k,\tau^k\big)\Big] \leq \sum_{t=1}^{T}\mathbb{E}[\ell_t(U_t)] + \frac{1}{\eta}\,\mathbb{E}\bigg[\ln\bigg(\frac{w_{T+1}(U_T)}{w_1(U_1)}\bigg)\bigg] + \frac{1}{\eta}\sum_{j=1}^{k}\mathbb{E}\Bigg[\ln\Bigg(\frac{f_j\big(U_{\tau_j}\big)}{w_{\tau_j}\big(U_{\tau_j}\big)}\Bigg)\Bigg] + \frac{k}{\eta}\ln\bigg(\frac{1}{1-\Gamma_\mathrm{u}}\bigg) + \frac{T\eta(C\,\mathrm{diam}(\mathcal{X}))^2}{8}. \tag{2.35}$$

We next provide an upper bound on $\mathbb{E}\big[\ln\big(f_j\big(U_{\tau_j}\big)/w_{\tau_j}\big(U_{\tau_j}\big)\big)\big]$ for each $j = 1, \ldots, T$ as follows. We note that we can write

$$\mathbb{E}\Bigg[\ln\Bigg(\frac{f_j\big(U_{\tau_j}\big)}{w_{\tau_j}\big(U_{\tau_j}\big)}\Bigg)\Bigg] = \mathbb{E}\Bigg[\ln\Bigg(\frac{f_j\big(U_{\tau_j}\big)}{\mu_{\tau_j}\big(U_{\tau_j}\big)}\Bigg)\Bigg] + \mathbb{E}\Bigg[\ln\Bigg(\frac{\mu_{\tau_j}\big(U_{\tau_j}\big)}{w_{\tau_j}\big(U_{\tau_j}\big)}\Bigg)\Bigg]. \tag{2.36}$$

We will bound the term on the right-hand side of (2.36) separately. We first write from (2.17) that

$$\begin{aligned}\ln\Bigg(\frac{f_j\big(U_{\tau_j}\big)}{\mu_{\tau_j}\big(U_{\tau_j}\big)}\Bigg) &= -\eta\ell_{\tau_j}\big(U_{\tau_j}\big) - \ln\big(Z_{\tau_j}\big) \\ &\leq -\eta\ell_{\tau_j}\big(U_{\tau_j}\big) + \eta\min_{u\in\mathcal{X}}\ell_{\tau_j}(u)\end{aligned}$$

$$\leq \eta C \operatorname{diam}(\mathcal{X}), \tag{2.37}$$

where (2.37) follows from Lipschitz continuity of the loss function ℓ_{τ_j} and compactness of the set $\mathcal{X}$. Hence, by taking expectation of both sides, we obtain

$$\mathbb{E}\left[\ln\left(\frac{f_j(U_{\tau_j})}{\mu_{\tau_j}(U_{\tau_j})}\right)\right] \leq \eta C \operatorname{diam}(\mathcal{X}). \tag{2.38}$$

We note that from the local averaging, we have

$$\frac{\mu_{\tau_j}(U_{\tau_j})}{w_{\tau_j}(U_{\tau_j})} = \gamma_{\tau_j} + (1-\gamma_{\tau_j})\frac{w_{\tau_j-1}(U_{\tau_j})}{w_{\tau_j}(U_{\tau_j})}. \tag{2.39}$$

Here, we note that

$$\begin{aligned}\frac{w_{\tau_j-1}(U_{\tau_j})}{w_{\tau_j}(U_{\tau_j})} &= \frac{w_{\tau_j-1}(U_{\tau_j})Z_{\tau_j-1}}{\mu_{\tau_j-1}(U_{\tau_j})\exp(-\eta\ell_{\tau_j-1}(U_{\tau_j}))} \\ &\leq \exp\left(\eta\ell_{\tau_j-1}(U_{\tau_j}) - \eta\min_{u\in\mathcal{X}}\ell_{\tau_j-1}(u)\right)\frac{w_{\tau_j-1}(U_{\tau_j})}{\mu_{\tau_j-1}(U_{\tau_j})} \\ &\leq \exp(\eta C \operatorname{diam}(\mathcal{X}))\frac{1}{\gamma_{\tau_j-1}}, \end{aligned} \tag{2.40}$$

where in (2.40), we used

$$\frac{w_{\tau_j-1}(U_{\tau_j})}{\mu_{\tau_j-1}(U_{\tau_j})} \leq \frac{1}{\gamma_{\tau_j-1}},$$

which follows directly from (2.39). Hence, we can write

$$\mathbb{E}\left[\ln\left(\frac{\mu_{\tau_j}(U_{\tau_j})}{w_{\tau_j}(U_{\tau_j})}\right)\right] \leq \ln\left(\gamma_{\tau_j} + \exp(\eta C \operatorname{diam}(\mathcal{X}))\frac{(1-\gamma_{\tau_j})}{\gamma_{\tau_j-1}}\right). \tag{2.41}$$

Therefore, by combining (2.38) and (2.41), we obtain

$$\begin{aligned}\mathbb{E}\left[\ln\left(\frac{f_j(U_{\tau_j})}{w_{\tau_j}(U_{\tau_j})}\right)\right] &\leq \eta C \operatorname{diam}(\mathcal{X}) + \ln\left(\Gamma_{\mathrm{u}} + \exp(\eta C \operatorname{diam}(\mathcal{X}))\frac{1-\Gamma_{\mathrm{l}}}{\Gamma_{\mathrm{l}}}\right) \\ &\triangleq G_\eta(\Gamma_{\mathrm{l}}, \Gamma_{\mathrm{u}}), \end{aligned} \tag{2.42}$$

for each $j = 1, \dots, k$. Finally, by combining (2.35) and (2.42), we get the following upper bound:

$$\mathbb{E}\Big[L_T^{(\mathrm{o,p})}\big(X_\mathrm{r}^T;\ell^T,\Pi^k,\tau^k\big)\Big] \le \sum_{t=1}^{T}\mathbb{E}[\ell_t(U_t)] + \frac{1}{\eta}\,\mathbb{E}\bigg[\ln\bigg(\frac{w_{T+1}(U_T)}{w_1(U_1)}\bigg)\bigg] \\ + \frac{T\eta(C\,\mathrm{diam}(\mathcal{X}))^2}{8} + kF_\eta(\Gamma_\mathrm{l},\Gamma_\mathrm{u}), \qquad (2.43)$$

where $F_\eta(\Gamma_\mathrm{l},\Gamma_\mathrm{u})$ is given in (2.19). We observe that the second term in (2.43) can be written as

$$\begin{aligned}\frac{1}{\eta}\,\mathbb{E}\bigg[\ln\bigg(\frac{w_{T+1}(U_T)}{w_1(U_1)}\bigg)\bigg] &= \frac{1}{\eta}\bigg(\int_{\mathcal{X}} dP(u)\ln(w_{T+1}(u)) - \int_{\mathcal{X}} dP(u)\ln(w_1(u))\bigg)\\ &= \frac{1}{\eta}\int_{\mathcal{X}} dP(u)\ln\bigg(\frac{W_{T+1}(u)}{W_1(u)}\bigg)\\ &= \frac{1}{\eta}(D_{\mathrm{KL}}(P\|W_1) - D_{\mathrm{KL}}(P\|W_{T+1}))\\ &\le \frac{r}{\eta},\end{aligned}$$

since $P \in \mathcal{P}(r)$. Moreover, we note that (2.43) is true for all $\ell^T \in \mathcal{L}^T$, and for any adversary $\mathcal{A}_k\big(\Pi^k,\tau^k\big)$, it is also true for the worst-case adversary $\mathcal{A}_k^\mathrm{w}$, yielding

$$\mathbb{E}\Big[R_T^{(\mathrm{o,p})}\big(X_\mathrm{r}^T;\mathcal{L},\mathcal{P}(r),\mathcal{A}_k^\mathrm{w}\big)\Big] \le \frac{r}{\eta} + \frac{T\eta(C\,\mathrm{diam}(\mathcal{X}))^2}{8} \\ + kF_\eta(\Gamma_\mathrm{l},\Gamma_\mathrm{u}).$$

□

We proved an upper bound on the worst-case expected regret of the randomized RWA algorithm when subject to perturbations of an adversarial agent. We observed that this upper bound is intuitively related to the upper bound we presented in Theorem 2.2.1 on the worst-case expected regret of the randomized WA algorithm. This observation has two implications. First, when the adversary does not perturb the player's decisions, that is, when $k = 0$, then this result gives the same worst-case expected regret guarantee that we had for the randomized WA algorithm. Second, when the adversary *does* perturb the player's distribution, i.e., when $k > 0$, then we can introduce an intuitive notion of the "cost" of each single perturbation, and interpret the second part in the upper bound (2.18) as an upper bound on the total cost of k perturbations. In the next section, we will demonstrate some asymptotic results on the worst-case regret of the randomized RWA algorithm when T and k are allowed to be unbounded. Specifically, we will show that this algorithm is Hannan consistent under certain regularity conditions.

2.4.1 Asymptotic Behavior of the Worst-Case Expected Regret

We present results on the asymptotic performance of the randomized RWA algorithm when subject to the perturbations of the worst-case adversary. In previous sections, we considered finite T and k. To study the asymptotic behavior of the worst-case regret of the randomized RWA algorithm, we allow T and k to be unbounded. We first prove the following result, which is a corollary to the Theorem 2.4.1.

Corollary 2.4.1. *Suppose that the randomized RWA algorithm is subject to the perturbations of the worst-case adversary $\mathcal{A}_k\big(\Pi_{\mathrm{w}}^k, \tau_{\mathrm{w}}^k\big)$ characterized by* (2.10) *and* (2.11)*. If the learning rate η is set to*

$$\eta_{\mathrm{o}} = \frac{\sqrt{2r}}{C\,\mathrm{diam}(\mathcal{X})}\frac{1}{\sqrt{T}},$$

then the worst-case expected regret of the randomized RWA algorithm satisfies

$$\begin{aligned}&\mathbb{E}\Big[R_T^{(\mathrm{o,p})}\big(X_{\mathrm{r}}^T; \mathcal{L}, \mathcal{P}(r), \mathcal{A}_k^{\mathrm{w}}\big)\Big]\\ &\leq \sqrt{T}(\mathrm{A}_1(r) + k\mathrm{A}_2(r, \Gamma_{\mathrm{l}}, \Gamma_{\mathrm{u}})) + 2kC\,\mathrm{diam}(\mathcal{X}),\end{aligned} \tag{2.44}$$

whenever $\Gamma_{\mathrm{l}} \leq 1 - \exp(-1)$, where

$$\mathrm{A}_1(r) \triangleq \sqrt{2r}C\,\mathrm{diam}(\mathcal{X}), \tag{2.45}$$

$$\mathrm{A}_2(r, \Gamma_{\mathrm{l}}, \Gamma_{\mathrm{u}}) \triangleq \sqrt{\frac{1}{2r}}C\,\mathrm{diam}(\mathcal{X})\ln\bigg(\frac{e(1-\Gamma_{\mathrm{l}})}{\Gamma_{\mathrm{l}}(1-\Gamma_{\mathrm{u}})}\bigg). \tag{2.46}$$

In particular, if

$$k = o\Big(\sqrt{T}\Big),$$

then it follows that

$$\mathbb{E}\Big[R_T^{(\mathrm{o,p})}\big(X_{\mathrm{r}}^T; \mathcal{L}, \mathcal{P}(r), \mathcal{A}_k^{\mathrm{w}}\big)\Big] = o(T).$$

Proof. The proof is presented in Appendix A.2. □

We note that in Corollary 2.4.1, the algorithm's learning rate must be a function of the time horizon T to achieve Hannan consistency. Here, we use a technique called the *doubling trick* to remove this dependence as follows [27]. We first divide time into periods

$$\mathcal{I}_n \triangleq \big[2^{n-1}, 2^n - 1\big]$$

of length 2^{n-1} for $n = 1, \ldots, N$, where

$$N \triangleq \lceil \log_2(T) \rceil.$$

Hence, we have

$$[1, T] \subseteq \cup_{n=1}^N \mathcal{I}_n.$$

Then, in each time period $\mathcal{I}_n$, the algorithm uses the learning rate

$$\eta_n = \frac{\sqrt{r}}{C \operatorname{diam}(\mathcal{X})} \frac{1}{\sqrt{2^{n-1}}}, \tag{2.47}$$

for $n = 1, \ldots, N$. We present a description of this algorithm in Algorithm 5. Corollary 2.4.2 proves that the worst-case expected regret of Algorithm 5 is sublinear in T when subject to the worst-case adversary's perturbations, under certain conditions.

Corollary 2.4.2. *Suppose that Algorithm 5 is subject to perturbations by the worst-case adversary $\mathcal{A}_k^{\mathrm{w}}$. Then, its worst-case expected regret satisfies*

$$\begin{aligned}&\mathbb{E}\Big[R_T^{(\mathrm{o,p})}\big(X_{\mathrm{r}}^T; \mathcal{L}, \mathcal{P}(r), \mathcal{A}_k^{\mathrm{w}}\big)\Big] \\ &\le \sqrt{T}(\beta \mathrm{A}_1(r) + k\mathrm{A}_2(r, \Gamma_{\mathrm{l}}, \Gamma_{\mathrm{u}})) + 2kC \operatorname{diam}(\mathcal{X}),\end{aligned}$$

where

$$\beta \triangleq \sqrt{2}/\Big(\sqrt{2} - 1\Big),$$

Algorithm 5 Online Randomized RWA Algorithm (η^N)

```
Input: A sequence of learning rates $\eta^N$, given in (2.47),
        A sequence $\gamma^T$ with $0 < \Gamma_{\mathrm{l}} \leq \gamma_t \leq \Gamma_{\mathrm{u}} < 1$, $\forall t$.
Initialization: Pick $W_1 = \mathrm{M}_1$ with $\operatorname{supp} W_1 = \mathcal{X}$, $n = 1$.
for $t = 1 : T$ do
    if $n == \log_2(t) - 1$ then
        Set $\eta = \eta_n$, and $n \leftarrow n + 1$.
        Reset $w_t(x) = \mu_t(x) = w_1(x)$, $\forall x \in \mathcal{X}$.
    end if
    The player draws $X_{\mathrm{r},t} \sim \mathrm{M}_t$ (with density $\mu_t$).
    Nature reveals a loss function $\ell_t \in \mathcal{L}$.
    The player incurs the loss $\ell_t(X_{\mathrm{r},t})$ and updates $w_t$:
    $w_{t+1}(x) = \dfrac{\mu_t(x)\exp(-\eta\ell_t(x))}{\displaystyle\int_{\mathcal{X}} d\mathrm{M}_t(u)\exp(-\eta\ell_t(u))}$, $\forall x \in \mathcal{X}$.
    if $t == \tau_j$ for some $j = 1, \ldots, k$ then
        The distribution is perturbed by an adversary:
        $W_{t+1} \leftarrow \Pi_j$, $(w_{t+1} \leftarrow \pi_j)$.
    end if
    The algorithm computes the actual distribution
    $\mathrm{M}_{t+1} = \gamma_{t+1},$
    $W_{t+1} + (1 - \gamma_{t+1})W_t$.
end for
```

A_1 *and* A_2 *are given in* (2.45) *and* (2.46), *respectively, whenever*

$$\Gamma_{\mathrm{l}} \leq 1 - \exp(-1),$$

and the learning rate sequence $\eta^N = (\eta_1, \ldots, \eta_N)$ *is given by*

$$\eta_n = \frac{\sqrt{r}}{C \operatorname{diam}(\mathcal{X})} \frac{1}{\sqrt{2^{n-1}}}, \; n = 1, \ldots, N.$$

Moreover, if

$$k = o\Big(\sqrt{T}\Big),$$

then it follows that

$$\mathbb{E}\Big[R_T^{(\mathrm{o},\mathrm{p})}\big(X_{\mathrm{r}}^T; \mathcal{L}, \mathcal{P}(r), \mathcal{A}_k^{\mathrm{w}}\big)\Big] = o(T). \tag{2.48}$$

Proof. To prove this result, we first upper-bound the worst-case expected regret of Algorithm 5 in each time period $\mathcal{I}_n$ for each $n = 1, \ldots, N$. We next combine these upper bounds to derive a final upper bound on the worst-case expected regret of the algorithm.

Suppose that the algorithm is subject to perturbations of an adversary $\mathcal{A}_k(\Pi_k, \tau_k)$ for some $k \in \mathbb{Z}$. Let k_n be the number of perturbation time instants in the time interval $\mathcal{I}_n$ for each $n = 1, \ldots, N$, that is,

$$\tau_k^{(n)} \triangleq \tau_k \cap \mathcal{I}_n,\ k_n \triangleq \left|\tau_k^{(n)}\right| \geq 0,\ n = 1, \ldots, N,$$

Hence, we can represent actions of the adversary $\mathcal{A}_k(\Pi_k, \tau_k)$ in each time interval $\mathcal{I}_n$ as $\mathcal{A}_{k_n}\left(\Pi_k^{(n)}, \tau_k^{(n)}\right)$ for each $n = 1, \ldots, N$.

Since the algorithm resets the intermediate density w_t to the initial density w_1 at the beginning of each time interval $\mathcal{I}_n$, the worst-case expected regret of this algorithm in $\mathcal{I}_n$ can be upper-bounded by the worst-case expected regret of the same algorithm that is run for a time horizon of 2^{n-1} using the learning rate η_n, where it is subject to k_n adversarial perturbations. We define, with mild abuse of notation, the worst-case expected regret of the algorithm in the time period $\mathcal{I}_n$ as

$$\begin{aligned}&\mathbb{E}\left[R_{\mathcal{I}_n}\left(X_{\mathrm{r}}; \mathcal{L}, \mathcal{P}(r), \tau_k^{(n)}, \Pi_k^{(n)}\right)\right]\\ &\triangleq \sup_{\ell^{2^{n-1}} \in \mathcal{L}^{2^{n-1}}} \left\{ \sum_{t=2^{n-1}}^{2^n-1} \mathbb{E}[\ell_t(X_{\mathrm{r},t})] - \inf_{P \in \mathcal{P}(r)} \sum_{t=2^{n-1}}^{2^n-1} \mathbb{E}[\ell_t(U_t)] \right\},\end{aligned} \tag{2.49}$$

for each $n = 1, \ldots, N$. Then, by Corollary 2.4.1, we upper-bound (2.49) as

$$\begin{aligned}&\mathbb{E}\left[R_{\mathcal{I}_n}\left(X_{\mathrm{r}}; \mathcal{L}, \mathcal{P}(r), \tau_k^{(n)}, \Pi_k^{(n)}\right)\right]\\ &\leq 2^{(n-1)/2}[\mathrm{A}_1(r) + k_n \mathrm{A}_2(r, \Gamma_{\mathrm{l}}, \Gamma_{\mathrm{u}})] + 2k_n C \operatorname{diam}(\mathcal{X}),\end{aligned}$$

for all $n = 1, \ldots, N$. Hence, we can upper-bound the worst-case expected regret of Algorithm 5 as

$$\begin{aligned}\mathbb{E}\left[R_T^{(\mathrm{o,p})}\left(X_{\mathrm{r}}^T; \mathcal{L}, \mathcal{P}(r), \tau_k, \Pi_k\right)\right] &\leq \sum_{n=1}^{N} \mathbb{E}\left[R_{\mathcal{I}_n}\left(X_{\mathrm{r}}; \mathcal{L}, \mathcal{P}(r), \tau_k^{(n)}, \Pi_k^{(n)}\right)\right]\\ &\leq \mathrm{A}_1(r) \sum_{n=0}^{N-1} 2^{n/2} + \mathrm{A}_2(r, \Gamma_{\mathrm{l}}, \Gamma_{\mathrm{u}}) \sum_{n=0}^{N-1} 2^{n/2} k_{n+1}\\ &\quad + 2kC \operatorname{diam}(\mathcal{X})\end{aligned}$$

$$
\begin{aligned}
&\leq \mathrm{A}_1(r)\frac{2^{N/2}-1}{\sqrt{2}-1} + k\mathrm{A}_2(r,\Gamma_{\mathrm{l}},\Gamma_{\mathrm{u}})2^{(N-1)/2} \\
&+ 2kC\,\mathrm{diam}(\mathcal{X}) \\
&\leq \sqrt{T}(\beta\mathrm{A}_1(r) + k\mathrm{A}_2(r,\Gamma_{\mathrm{l}},\Gamma_{\mathrm{u}})) + 2kC\,\mathrm{diam}(\mathcal{X})\,.
\end{aligned}
$$

We observe that this is true for all adversaries of the form $\mathcal{A}_k(\Pi_k,\tau_k)$. In particular, it is satisfied under the worst-case adversary, so that

$$\mathbb{E}\Big[R_T^{(\mathrm{o},\mathrm{p})}\big(X_{\mathrm{r}}^T;\mathcal{L},\mathcal{P}(r),\mathcal{A}_k^{\mathrm{w}}\big)\Big] \leq \sqrt{T}(\beta\mathrm{A}_1(r) + k\mathrm{A}_2(r,\Gamma_{\mathrm{l}},\Gamma_{\mathrm{u}})) + 2kC\,\mathrm{diam}(\mathcal{X})\,.$$

Moreover, when $k = o\Big(\sqrt{T}\Big)$, we get

$$\mathbb{E}\Big[R_T^{(\mathrm{o},\mathrm{p})}\big(X_{\mathrm{r}}^T;\mathcal{L},\mathcal{P}(r),\mathcal{A}_k^{\mathrm{w}}\big)\Big] = o(T).$$

This concludes the proof. □

Remark 2.4.2. *We observe from Corollary 2.4.2 that, when the time horizon T is unknown to the randomized RWA algorithm in advance, we can still guarantee a sublinear worst-case expected regret via the doubling trick. Moreover, this bound is of the same order as before, up to a constant factor.*

In this section, we investigated asymptotic performance of the randomized RWA algorithm when subject to perturbations of the worst-case adversary $\mathcal{A}_k^{\mathrm{w}}$. We showed that under certain conditions, the worst-case regret of this algorithm is sublinear in the time horizon T. We next proposed another version of this algorithm, where we removed the dependence of the algorithm to the time horizon T by using the so-called doubling trick. We demonstrated that this version of the randomized RWA algorithm enjoys an upper bound that is of the same order as before. In particular, we showed that it is also Hannan consistent under similar regularity conditions.

2.5 Experimental Results

In this section, we illustrate our theoretical results and performance of the proposed algorithms on synthetic data. For these experiments, we use the decision set $\mathcal{X} = [0,1]$, and Nature reveals affine loss functions, that is, for

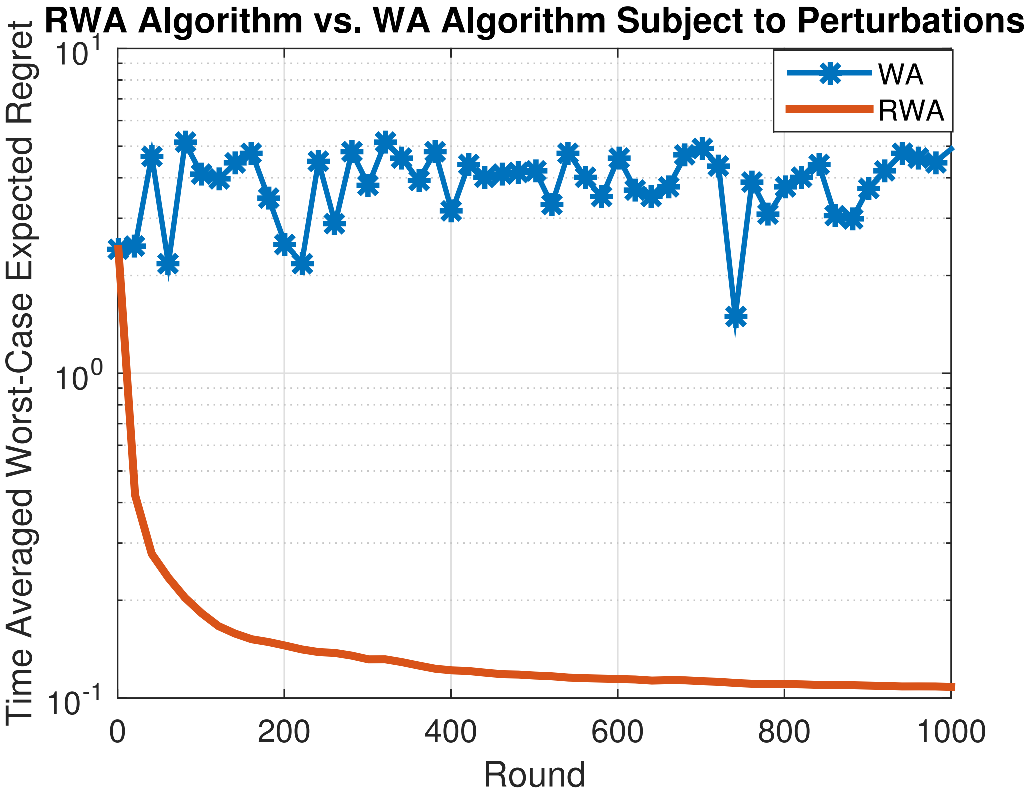


Figure 2.2: The randomized RWA algorithm (Algorithm 1) and the randomized WA algorithm when subject to adversarial perturbations with $k = \lfloor T^{1/4} \rfloor$.

each $x \in \mathcal{X}$ and $t = 1, \ldots, T$,

$$\ell_t(x) = \beta_t(x - \phi_t),$$

for some $\beta_t, \phi_t \in \mathbb{R}$.

We first present and compare the worst-case expected regret performance of the randomized WA algorithm and the randomized RWA algorithm, when both are subject to perturbations of the worst-case adversary with $k = \lfloor T^{1/4} \rfloor$. We use the *Independent Metropolis-Hastings algorithm* [40] to generate random decisions of the online player (using its distribution) in each case, where we run both algorithms for 10^3 rounds, and take averages over 10^4 realizations to experiment the expectations. In particular, we plot the time-averaged worst-case expected regret of both algorithms in Fig. 2.2. We observe that the performance of the WA algorithm is poor compared to that of the RWA algorithm. We next plot the worst-case expected regret curves of the RWA algorithm (Algorithms 1 and 2) when the time horizon T is known and unknown, respectively, under different regimes ($k = 0$, $\lfloor T^{1/3} \rfloor$, $\lfloor T^{1/5} \rfloor$) of adversarial perturbations in Fig. 2.3. We note that the randomized RWA algorithm performs satisfactorily in all cases, illustrating the sufficient condition for Hannan-consistency given in both Corollary 1 and Corollary 2, i.e., $k = o\left(\sqrt{T}\right)$. Hence, for these experiments, we observe a close agreement between our theoretical results and simulations.

2.6 Conclusion

We have introduced and investigated an adversarial worst-case perturbation framework for online optimization, where an online player's strategy is subject to perturbations by an adversary. We cast this problem as a new repeated game, where a randomized player is pitted against two opponents, namely, Nature and a strategy-perturbing adversary. We introduced a robust randomized algorithm and presented an upper bound on its worst-case expected regret under our worst-case model. In particular, we proved that this algorithm is Hannan consistent even under adversarial perturbations, when certain regularity conditions are satisfied. We presented some numerical experiments to illustrate our theoretical results.

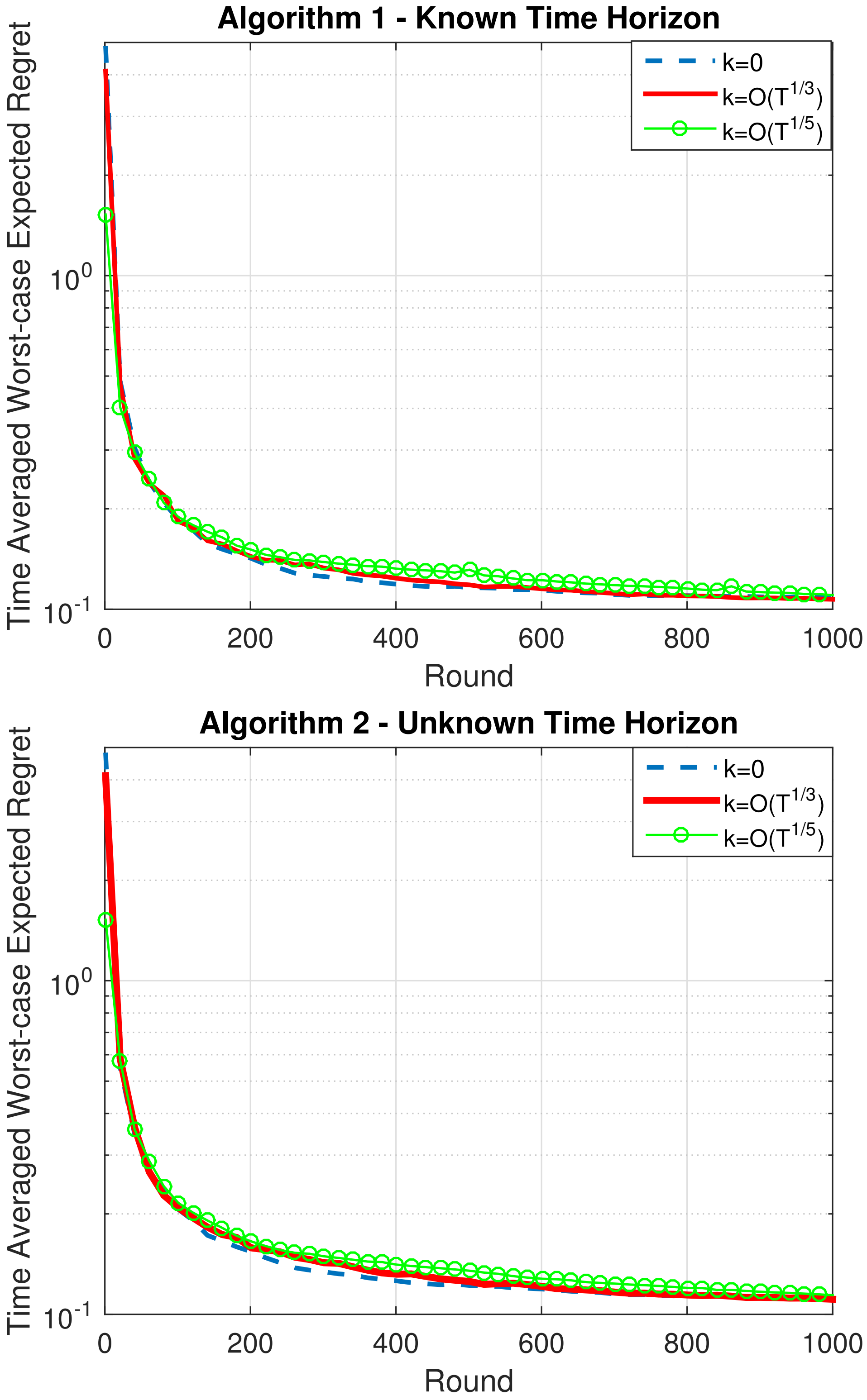


Figure 2.3: The worst-case expected regret performance of Algorithm 1 and Algorithm 2 under different regimes of adversarial perturbations.

CHAPTER 3

COST-PERFORMANCE TRADEOFFS IN FUSING UNRELIABLE COMPUTATIONAL UNITS

3.1 Introduction

We consider the problem of fusing outcomes of several unreliable computational units in order to form a reliable outcome from the individual contributions. In particular, we consider a case where each of the unreliable units performs the same computation. However, each of these units must operate under cost and fidelity constraints. We formalize the relationship between the fidelity of each unit and the cost associated with it, and explore this tradeoff in a number of practical problems. Consider, for instance, the capacity of an additive white Gaussian noise (AWGN) channel, which is a logarithmic function of the signal-to-noise (SNR) ratio. In this scenario, the capacity can be increased at the expense of requiring a higher SNR, which introduces a tradeoff between cost (SNR) and performance (rate). Note also that the Fisher information in estimation is often a linear function of SNR, leading to a different cost-performance tradeoff [41].

Building reliable systems out of unreliable components has attracted substantial interest in circuits and systems [42–44], information theory [45–47], and signal processing [48]. In [42], von Neumann investigated error in logic circuits from a statistical point of view and demonstrated that repeated computations followed by majority logic may yield reliable results even when the underlying components are unreliable. In [43], Tryon introduced a technique called quadded logic, which corrects errors by a redundant design of logic gates. Moreover, the authors of [45–47] investigated reliable computation by formulas in the presence of noise. More recently, the authors of [48] considered energy-reliability tradeoffs in computing linear transforms implemented on unreliable components.

Fusion of the outputs collected from several sensors has been considered in distributed detection, estimation, classification, and optimization in sensor networks [49–55]. Often, spatially distributed sensors locally perform a decision-making task and send their outputs, under bandwidth constraints, to a fusion center that forms a final decision. In most practical applications, these sensors are battery-powered devices with limited accuracy and computational capabilities, so their performance is critically affected by the resources allocated to them, introducing a cost-performance tradeoff. The authors of [54] studied tradeoffs between the number of sensors, resolution of the quantization at each sensor, and SNR. Similarly, [55] considered the tradeoff between reliability and efficiency in distributed source coding for field-gathering sensor networks. In general, the main goal is to make a reliable final decision in a cost-efficient manner based on these unreliable sensors subject to resource and reliability constraints.

A fundamental question that arises in fusing several unreliable computational units is how a limited budget should be allocated across several unreliable units, where adding a new unit incurs a baseline cost as well as an incremental cost, and also increases the cost of fusion. That is, what is an *optimal* approach for a given cost-performance tradeoff? Although existing work in fault-tolerant computing and sensor networks focuses on different pieces of this problem, a more general treatment that jointly considers cost and performance is necessary. This chapter is an attempt to combine insights from both fields into a unified framework that captures characteristics of a range of problems. In particular, we show how our framework and results are connected to problems from neuroscience, circuits, and crowdsourcing in Section 3.5.

In this chapter, we present an abstract framework to explore the fundamental tradeoff between cost and performance achievable through specific forms of redundancy. We model unreliability in any computational unit as an additive random perturbation, where the variance of the perturbation is inversely related to its fidelity. We cast the main task as one of inference of the error-free computation based on noisy computational outcomes. Each computational unit incurs a cost that is a function of fidelity and includes a baseline cost incurred to simply operate the unit.

We define a class of repetition-based strategies, where each strategy distributes the total cost across several unreliable computational units and fuses their outputs. We note that the fusion operation also incurs some cost, which is a function of the number of individual computational units to be fused. We measure the inference performance of each strategy in terms of MSE between its final output and the error-free computation.

We consider optimal repetition-based strategies under convex, linear, and concave cost functions rather than restricting to specific cost functions. For convex costs, there are two main cases. In the first case, we prove that using only a single and more reliable computational unit is more cost-efficient than the fusion of several lower cost but less reliable computational units. In the second case, however, we demonstrate that the optimal strategy uses several computational units instead of a single more reliable one. Intuitively, the convexity of the cost function disperses the cost across several less reliable computational units with smaller individual costs. For linear or concave costs, the optimal strategy is to use a single and more reliable computational unit.

3.1.1 Organization

This chapter is organized as follows. In Section 3.2, we describe the framework for unreliable computational units under cost and fidelity constraints. We model any unreliable computation in terms of the error-free computation and an additive random perturbation, where the fidelity is inversely related to the variance of the perturbation. Moreover, we describe the class of repetition-based strategies, and derive the optimal repetition-based strategies achieving the minimum MSE. In Section 3.4, we consider the cost-performance tradeoff of repetition-based strategies under classes of convex, linear, and concave cost functions, In particular, we characterize the optimal repetition-based strategy that incurs the smallest total cost while achieving a target MSE level under each class. Finally, we study application of our theoretical results into problems from neuroscience, circuits, and crowdsourcing in Section 3.5. We conclude with certain remarks and future research directions in Section 3.6.

3.2 Problem Description

We first introduce a model of an unreliable computational outcome as an additive perturbation to its error-free result. To provide a tradeoff between fidelity and cost, we assume the resource cost of the computational unit is inversely proportional to the variance of the additive perturbation. We next consider a class of repetition-based strategies that distribute cost across several parallel unreliable units and fuse their outcomes to produce a final estimate of the error-free computation.

Suppose a vector of input signals

$$\mathbf{X} = (X_1, \ldots, X_k)$$

is processed to yield the error-free computation,

$$Y = f(\mathbf{X}),$$

where $f(\cdot)$ is some arbitrary target function. Instead, we observe an unreliable computational outcome,

$$Z_\theta = Y + U_\theta,$$

where U_θ is a zero-mean perturbation with variance θ^{-1}. Here, θ is the fidelity of the unreliable computational outcome Z_θ. We assume that Y and U_θ are uncorrelated, that is,

$$\mathbb{E}[YU_\theta] = \mathbb{E}[Y]\,\mathbb{E}[U_\theta]$$

holds, whether or not Y is a random variable.

By Chebyshev's inequality, the unreliable outcome Z_θ with fidelity $\theta > 0$ satisfies, for any $\varepsilon > 0$,

$$\Pr(|Z_\theta - Y| \geq \varepsilon) \leq \frac{1}{\varepsilon^2 \theta}. \tag{3.1}$$

This implies the unreliable outcome Z_θ converges to the error-free computation in probability as the fidelity tends to infinity. However, as the fidelity parameter θ increases, the cost $C(\theta)$ incurred to guarantee that level of fidelity also increases, introducing a *cost-fidelity tradeoff*. Note that this holds both when X_i for $i = 1, \ldots, k$, or Y, are random as well as when they are purely deterministic.

In this model, we incur the cost $C(\theta)$ for the unreliable outcome Z_θ with fidelity $\theta > 0$, which we assume to be a strictly increasing function of θ. In particular, we assume

$$C(\theta) = c_{\min} + G(\theta),$$

where

$$c_{\min} \triangleq \inf_{\theta > 0} C(\theta) \geq 0$$

is the minimum (baseline) cost, and $G(\theta)$ is an increasing and twice differentiable incremental cost function with $G(0) = 0$. In the sequel, we focus on three classes of cost functions: convex, linear, and concave function of θ.

We define a class of repetition-based strategies that fuse the outputs of several computational units to estimate Y. For any positive integer N, a repetition-based strategy S_N, with weights

$$\mathbf{w} = (w_1, \ldots, w_N) \in \mathbb{R}^N$$

and fidelities

$$\boldsymbol{\theta} = (\theta_1, \ldots, \theta_N) \in (0, \infty)^N,$$

linearly combines the outcomes of N parallel unreliable units with fidelities $\boldsymbol{\theta}$ using the weights $\mathbf{w}$. That is, if we denote the outcome of a unit with fidelity θ_i and cost $C(\theta_i)$ as

$$Z_{\theta_i} = Y + U_{\theta_i},$$

for $i = 1, \ldots, N$, then the final output of this strategy S_N is

$$\hat{Y}_N(\mathbf{w}; \boldsymbol{\theta}) \triangleq \mathbf{w}^T \mathbf{Z}_{\boldsymbol{\theta}} = Y\big(\mathbf{w}^T \mathbf{1}\big) + \mathbf{w}^T \mathbf{U}_{\boldsymbol{\theta}}, \tag{3.2}$$

where $\mathbf{Z}_{\boldsymbol{\theta}} \triangleq (Z_{\theta_1}, \ldots, Z_{\theta_N})$, $\mathbf{U}_{\boldsymbol{\theta}} \triangleq (U_{\theta_1}, \ldots, U_{\theta_N})$, and $\mathbf{1} = (1, \ldots, 1) \in \mathbb{R}^N$ is a vector of ones. In particular, we assume that U_{θ_i}s are uncorrelated to each other.

The cost incurred by the strategy S_N with fidelities $\boldsymbol{\theta}$ is

$$\sum_{i=1}^{N} C(\theta_i) + D(N),$$

where $D(N)$ is the fusion cost, i.e., the cost of linear combination. We assume that the function

$$D : \mathbb{Z}_+ \to \mathbb{R}_+$$

is increasing, as fusing the outcomes of a larger number of computational units has higher cost than fewer. Note that the fusion cost is super-linear in N in that it requires at least $O(N)$ multiplications and additions. In particular, we assume that $D(N)$ is convex in N.

3.3 Performance Analysis

Here, we consider the MSE performance of each repetition-based strategy in estimating the error-free computation Y. For any positive integer N, the strategy S_N with a weight vector $\mathbf{w} \in \mathbb{R}^N$ and a fidelity vector $\boldsymbol{\theta} \in (0, \infty)^N$ achieves the MSE

$$\mathrm{MSE}(\mathbf{w}, \boldsymbol{\theta}) \triangleq \mathbb{E}\left[\left(\hat{Y}_N(\mathbf{w}; \boldsymbol{\theta}) - Y\right)^2\right]. \tag{3.3}$$

In particular, we derive the minimum MSE (MMSE) achievable by this strategy S_N while producing an unbiased output:

$$\mathrm{MSE_o}(\boldsymbol{\theta}) \triangleq \min_{\mathbf{w}^T \mathbf{1} = 1} \mathrm{MSE}(\mathbf{w}, \boldsymbol{\theta}),$$

where $\mathbf{w}_\mathrm{o}$ is the corresponding minimizer.

Lemma 3.3.1. *Suppose that for any positive integer N, the strategy S_N fuses the outcomes of N parallel computational units with fidelities $\boldsymbol{\theta} \in (0, \infty)^N$. Then the MMSE achievable by this strategy S_N while producing an unbiased estimate of Y, and the corresponding weights are*

$$\mathrm{MSE_o}(\boldsymbol{\theta}) = \frac{1}{\boldsymbol{\theta}^T \mathbf{1}}, \quad \mathbf{w}_\mathrm{o} = \frac{\boldsymbol{\theta}}{\boldsymbol{\theta}^T \mathbf{1}}, \tag{3.4}$$

respectively.

Proof. The MSE of the strategy S_N with a given $\boldsymbol{\theta} \in (0,\infty)^N$ is

$$\mathrm{MSE}(\mathbf{w},\boldsymbol{\theta}) = \mathbb{E}\Big[\big(Y\big(\mathbf{w}^T\mathbf{1}-1\big)+\mathbf{w}^T\mathbf{U}_{\boldsymbol{\theta}}\big)^2\Big],$$

where (3.2) is substituted in (3.3). Since Y and $\mathbf{U}_{\boldsymbol{\theta}}$ are uncorrelated:

$$\mathrm{MSE}(\mathbf{w},\boldsymbol{\theta}) = \mathbb{E}\big[Y^2\big]\big(\mathbf{w}^T\mathbf{1}-1\big)^2+\mathbf{w}^T\Sigma_{\mathbf{U}_{\boldsymbol{\theta}}}\mathbf{w}, \tag{3.5}$$

where $\Sigma_{\mathbf{U}_{\boldsymbol{\theta}}}$ is the covariance matrix of the perturbation vector $\mathbf{U}_{\boldsymbol{\theta}}$. If we impose the condition that $\mathbf{w}^T\mathbf{1}=1$ in (3.5), then

$$\mathrm{MSE}(\mathbf{w},\boldsymbol{\theta}) = \mathbf{w}^T\Sigma_{\mathbf{U}_{\boldsymbol{\theta}}}\mathbf{w}.$$

To minimize this over weights that satisfy $\mathbf{w}^T\mathbf{1}=1$, we first form the Lagrangian

$$J(\mathbf{w},\lambda) = \frac{1}{2}\mathbf{w}^T\Sigma_{\mathbf{U}_{\boldsymbol{\theta}}}\mathbf{w}+\lambda\big(1-\mathbf{w}^T\mathbf{1}\big),$$

and then compute the gradient with respect to $\mathbf{w}$ to get

$$\Sigma_{\mathbf{U}_{\boldsymbol{\theta}}}\mathbf{w}-\lambda\mathbf{1}=0,$$

which is satisfied if and only if

$$\mathbf{w}=\lambda\Sigma_{\mathbf{U}_{\boldsymbol{\theta}}}^{-1}\mathbf{1}.$$

With $\mathbf{w}^T\mathbf{1}=1$, we obtain

$$\lambda=\frac{1}{\mathbf{1}^T\Sigma_{\mathbf{U}_{\boldsymbol{\theta}}}^{-1}\mathbf{1}},$$

which yields the optimal weights

$$\mathbf{w}_{\mathrm{o}}=\frac{1}{\mathbf{1}^T\Sigma_{\mathbf{U}_{\boldsymbol{\theta}}}^{-1}\mathbf{1}}\Sigma_{\mathbf{U}_{\boldsymbol{\theta}}}^{-1}\mathbf{1}.$$

When we substitute this result in $\mathrm{MSE}(\mathbf{w},\boldsymbol{\theta})$, we achieve

$$\mathrm{MSE}_{\mathrm{o}}(\boldsymbol{\theta})=\mathbf{w}_{\mathrm{o}}^T\Sigma_{\mathbf{U}_{\boldsymbol{\theta}}}\mathbf{w}_{\mathrm{o}}$$

$$= \frac{1}{\mathbf{1}^T \Sigma_{\mathbf{U}_{\boldsymbol{\theta}}}^{-1} \mathbf{1}}.$$

We finally note that

$$\Sigma_{\mathbf{U}_{\boldsymbol{\theta}}} = \operatorname{diag}\bigl(\theta_1^{-1}, \ldots, \theta_N^{-1}\bigr),$$

which leads to the desired result. □

Thus, Lemma 3.3.1 provides the strategy S_N achieving the MMSE for a given fidelity vector $\boldsymbol{\theta} \in (0, \infty)^N$. For any positive integer N, whenever we refer to the strategy S_N, we use the optimal weights given in (3.4), so that its output is

$$\hat{Y}_N(\mathbf{w}_{\mathrm{o}}; \boldsymbol{\theta}) = \mathbf{w}_{\mathrm{o}}^T \mathbf{Z}_{\boldsymbol{\theta}} = \frac{\boldsymbol{\theta}^T \mathbf{Z}_{\boldsymbol{\theta}}}{\boldsymbol{\theta}^T \mathbf{1}}.$$

We next study a particular scenario, where U_θ is sub-Gaussian.

3.3.1 Sub-Gaussian Perturbations

Here, we consider a case where the perturbation U_θ is sub-Gaussian with parameter θ^{-1}, which implies [56]

$$\mathbb{E}\bigl[e^{\lambda U_\theta}\bigr] \leq \exp\biggl(\frac{\lambda^2}{2\theta}\biggr), \quad \forall \lambda \in \mathbb{R}, \tag{3.6}$$

or equivalently, the probability of absolute deviation of Z_θ from Y satisfies, for any $\varepsilon > 0$,

$$\Pr(|Z_\theta - Y| \geq \varepsilon) \leq 2 \exp\bigl(-\varepsilon^2 \theta / 2\bigr). \tag{3.7}$$

The tail bound in (3.7) decreases faster (with increasing θ) than the bound in (3.1). Sub-Gaussian distributions can be used to model a wide range of stochastic phenomena including Gaussian and uniform distributions, or distributions with finite or bounded support. Note that a weighted sum of finitely many sub-Gaussian random variables is also sub-Gaussian [56]. By applying this result to the output of a strategy S_N with $\mathbf{w} \in \mathbb{R}^N$ and

$\boldsymbol{\theta} \in (0,\infty)^N$, we get, for any $\varepsilon > 0$,

$$\Pr\left(\left|\hat{Y}_N(\mathbf{w};\boldsymbol{\theta}) - Y\right| \geq \varepsilon\right) \leq 2\exp\left(-\frac{\varepsilon^2}{\sum_{i=1}^N w_i^2/\theta_i}\right).$$

The weights minimizing the upper bound under $\mathbf{w}^T\mathbf{1} = 1$, and the resulting bound are known to be

$$\mathbf{w}_\mathrm{o} = \frac{\boldsymbol{\theta}}{\boldsymbol{\theta}^T\mathbf{1}},$$

and

$$\Pr\left(\left|\hat{Y}_N(\mathbf{w}_\mathrm{o};\boldsymbol{\theta}) - Y\right| \geq \varepsilon\right) \leq 2\exp\left(-\varepsilon^2\boldsymbol{\theta}^T\mathbf{1}/2\right),$$

for any $\varepsilon > 0$, respectively.

We emphasize that, in this case, even though the performance is measured in terms of probability of absolute deviation from the error-free computation, the optimal weights are exactly the same as the ones minimizing the MSE. Hence, same results apply to both cases when comparing the cost-performance tradeoff of the repetition-based strategies.

In this section, we analyzed the MSE performance of repetition-based strategies. More precisely, for any positive integer N and a fidelity vector $\boldsymbol{\theta} \in (0,\infty)^N$, we derived the optimal weights for the strategy S_N in terms of minimizing the MSE. Based on these results, we next investigate the cost-performance tradeoff for a wide variety of repetition-based strategies.

3.4 Cost-Performance Tradeoff

We investigate the performance of repetition-based strategies under convex, linear, and concave cost functions in terms of the tradeoff between the total incurred cost and the final MSE performance in estimating the error-free computation.

We first analyze the case where the cost $C(\theta)$ is a convex function of the fidelity θ. We characterize the optimal strategy, based on the desired MSE performance as well as the baseline and fusion cost functions. In particular, we show that the optimal cost-performance tradeoff may be achieved by some strategy S_N with $N > 1$ under certain conditions.

We next consider the case where the cost $C(\theta)$ is a linear function of the fidelity parameter θ, and show that strategy S_1 is optimal among repetition-based strategies. We finally study the concave cost scenario, and demonstrate results similar to the linear cost function case.

To compare cost-performance tradeoffs of repetition-based strategies, we constrain each strategy to guarantee the same MSE performance. More precisely, given some $\tau > 0$, we assume that the strategy S_N with $\boldsymbol{\theta} \in (0, \infty)^N$ satisfies

$$\tau = \mathrm{MSE}_{\mathrm{o}}(\boldsymbol{\theta}) = \frac{1}{\boldsymbol{\theta}^T \mathbf{1}},$$

or equivalently, $\tau^{-1} = \boldsymbol{\theta}^T \mathbf{1}$, for any positive integer N. We also define the total cost incurred by strategy S_N, which achieves $\mathrm{MSE}_{\mathrm{o}}(\boldsymbol{\theta}) = \tau$, as

$$\mathrm{Cost}_\tau(N) \triangleq \sum_{i=1}^{N} C(\theta_i) + D(N).$$

3.4.1 Convex Cost Functions

We study the cost-performance tradeoff for the class of repetition-based strategies under a convex cost function. This case turns out to correspond to a *law of diminishing returns* between cost and fidelity, which may drive the dispersion of cost across several less reliable computational units with smaller individual costs. We show that there are two main cases, where, in the first case, some strategy S_N with $N > 1$ may incur the minimum total cost achievable by the repetition-based strategies while achieving the same MSE, whereas in the second case, the strategy S_1 is optimal in terms of cost-performance tradeoff, i.e., no repetition or fusion is required.

Consider a uniform fidelity distribution across several unreliable computational outcomes, given by

$$\theta_i \triangleq \frac{1}{\tau N}, \quad i = 1, \ldots, N, \tag{3.8}$$

which implies that the constraint $\mathrm{MSE}_{\mathrm{o}}(\boldsymbol{\theta}) = \tau$ is satisfied. In fact, the following lemma shows that the optimal fidelity distribution satisfying the MSE constraint in terms of minimizing the total cost is in fact uniform.

Lemma 3.4.1. *For any $\tau > 0$, the uniform fidelity distribution given by* (3.8) *is the unique solution to the optimization problem:*

$$\min_{\boldsymbol{\theta} \in \mathbb{R}_+^N} \sum_{i=1}^{N} C(\theta_i)$$

subject to $\boldsymbol{\theta}^T \mathbf{1} = \tau^{-1}$ when the cost function $C(\theta)$ is convex.

Proof. The proof is given in Appendix B.1. □

Hence, we only consider the case where the strategy S_N, for each positive integer N, uses the fidelities in (3.8). The total cost incurred by this strategy S_N is

$$\begin{aligned} \mathrm{Cost}_\tau(N) &= \sum_{i=1}^{N} C\left(\frac{1}{\tau N}\right) + D(N) \\ &= NG\left(\frac{1}{\tau N}\right) + Nc_{\min} + D(N). \end{aligned} \tag{3.9}$$

To investigate the behavior of the total cost, we define its continuous relaxation as

$$\begin{aligned} \mathrm{Cost}_\tau &: [1, \infty) \to (0, \infty) \\ \mathrm{Cost}_\tau(a) &\triangleq aG\left(\frac{1}{\tau a}\right) + ac_{\min} + D(a), \end{aligned}$$

where $D(a)$ is a twice differentiable continuous relaxation of the fusion cost function $D(N)$. We first demonstrate that $\mathrm{Cost}_\tau(a)$ is a convex function in a.

Lemma 3.4.2. *The total cost function* $\mathrm{Cost}_\tau(a)$ *is convex in a.*

Proof. The proof is provided in Appendix B.2. □

Convexity of $\mathrm{Cost}_\tau(a)$ implies that it has a unique minimizer on any given compact subset of its domain $[1, \infty)$. In particular, note that

$$\mathrm{Cost}_\tau(1) = G\big(\tau^{-1}\big) + c_{\min},$$

and

$$\mathrm{Cost}_\tau(a) \to \infty$$

as $a \to \infty$. Therefore, the total cost function $\mathrm{Cost}_\tau(a)$ has a unique and finite minimizer $a_\mathrm{o}(\tau) \in [1, \infty)$. Also, there exists a corresponding unique optimal repetition-based strategy, which we denote as the strategy $S_{N_\mathrm{o}(\tau)}$ where

$$N_\mathrm{o}(\tau) = \underset{N \in \{\lfloor a_\mathrm{o}(\tau) \rfloor, \lceil a_\mathrm{o}(\tau) \rceil\}}{\arg\min} \ \mathrm{Cost}_\tau(N) \tag{3.10}$$

is a finite positive integer (a function of τ), that minimizes the total incurred cost while achieving the desired MSE of $\tau > 0$.

We next characterize conditions under which the optimal repetition-based strategy either uses a single but more reliable computational unit, that is, $N_\mathrm{o}(\tau) = 1$, or distributes the cost across several unreliable computational units and fuses their outcomes, that is, $N_\mathrm{o}(\tau) > 1$. In the latter case, we implicitly derive the optimal strategy as a function of the desired MSE level τ, the baseline cost $c_{\min}$, and the fusion cost function $D(\cdot)$. The next theorem characterizes these cases in terms of the first derivative of the fusion cost and the baseline cost.

Theorem 3.4.1. *For any given $\tau > 0$, the minimizer of $\mathrm{Cost}_\tau(a)$ satisfies $a_\mathrm{o}(\tau) > 1$ if and only if*

$$c_{\min} + D'(1) < V(\tau)$$

where

$$V(\tau) \triangleq \tau^{-1} G'\big(\tau^{-1}\big) - G\big(\tau^{-1}\big). \tag{3.11}$$

Proof. We define

$$\kappa_\tau(a) \triangleq \partial \mathrm{Cost}_\tau(a) / \partial a,$$

and observe that from Lemma 3.4.2, $\kappa_\tau(a)$ is nondecreasing and continuous in a since $\mathrm{Cost}_\tau(a)$ is a twice differentiable and convex function of a. Hence, whenever

$$\kappa_\tau(1) \geq 0,$$

we have $\kappa_\tau(a) \geq 0$ for any $a > 1$. It implies that $\mathrm{Cost}_\tau(a)$ is a nondecreasing function of a on $[1, \infty)$, and minimized at $a_\mathrm{o}(\tau) = 1$. When

$$\kappa_\tau(1) < 0,$$

$\mathrm{Cost}_\tau(a)$ is minimized at some finite $a_\mathrm{o}(\tau) > 1$, since

$$\mathrm{Cost}_\tau(a) \to \infty$$

as $a \to \infty$. The proof follows by noting that

$$\kappa_\tau(1) = G\big(\tau^{-1}\big) - \tau^{-1}G'\big(\tau^{-1}\big) + c_{\min} + D'(a) < 0$$

if and only if

$$c_{\min} + D'(1) < V(\tau),$$

where $V(\tau)$ is defined in (3.11). □

Based on these results, we can characterize the optimal repetition-based strategy. If

$$c_{\min} + D'(1) \geq V(\tau),$$

then

$$N_\mathrm{o}(\tau) = 1$$

since $a_\mathrm{o}(\tau) = 1$. Otherwise, we get $a_\mathrm{o}(\tau) > 1$, which is in this case implicitly given by

$$\begin{aligned}\frac{\partial \mathrm{Cost}_\tau(a)}{\partial a}\Big|_{a=a_\mathrm{o}(\tau)} &= G\bigg(\frac{1}{\tau a_\mathrm{o}(\tau)}\bigg) - \frac{1}{\tau a_\mathrm{o}(\tau)}G'\bigg(\frac{1}{\tau a_\mathrm{o}(\tau)}\bigg) \\ &\quad + c_{\min} + D'(a_\mathrm{o}(\tau)) \\ &= 0. \end{aligned} \tag{3.12}$$

If $1 < a_\mathrm{o}(\tau) < 2$, then we may get

$$N_\mathrm{o}(\tau) = 1 \text{ or } N_\mathrm{o}(\tau) = 2,$$

based on (3.10). When $a_\mathrm{o}(\tau) \geq 2$, we get

$$N_\mathrm{o}(\tau) > 1.$$

We finally consider the optimal repetition-based strategy as the target MSE τ changes. In the following lemma, we investigate the function $V(\tau)$ defined in (3.11) as τ changes.

Lemma 3.4.3. *$V(\tau)$ is non-negative and nonincreasing on $(0,\infty)$, and in particular, we have $\lim_{\tau\to\infty} V(\tau)=0$, and*

$$L \triangleq \lim_{\tau\to 0} V(\tau) > 0, \tag{3.13}$$

if $V(\tau)$ is bounded as $\tau \to 0$, or else, the limit does not exist.

Proof. We first observe that from (3.11)

$$\begin{aligned} V'(\tau) &= -\frac{1}{\tau^2}G'\bigl(\tau^{-1}\bigr) - \frac{1}{\tau^3}G''\bigl(\tau^{-1}\bigr) + \frac{1}{\tau^2}G'\bigl(\tau^{-1}\bigr) \\ &= -\frac{1}{\tau^3}G''\bigl(\tau^{-1}\bigr) \le 0, \end{aligned}$$

for any $\tau > 0$, as $G(\cdot)$ is convex and twice differentiable. Thus, the function $V(\tau)$ is decreasing on $(0,\infty)$. We next note that

$$\begin{aligned} \lim_{\tau\to\infty} V(\tau) &= \lim_{\tau\to\infty}\bigl(\tau^{-1}G'\bigl(\tau^{-1}\bigr) - G\bigl(\tau^{-1}\bigr)\bigr) \\ &= \lim_{\tau\to\infty} \tau^{-1}G'\bigl(\tau^{-1}\bigr) - G(0) = 0, \end{aligned}$$

since $G(0)=0$ and $G'(0)$ is finite. Therefore, $V(\tau)$ is non-negative on $(0,\infty)$. This implies that the function $V(\tau)$ either converges to a finite limit (if and only if $V(\tau)$ is bounded on $(0,\infty)$), or is unbounded as $\tau \to 0$. □

It may appear that from (3.10) and (3.12), as the target MSE τ decreases, the optimal repetition-based strategy may need to fuse more units, i.e., $N_{\mathrm{o}}(\tau)$ may increase. More rigorously, we next characterize the behavior of the minimizer $a_{\mathrm{o}}(\tau)$ of the total cost $\mathrm{Cost}_\tau(a)$ as the target MSE τ changes.

Theorem 3.4.2. *If the limit in (3.13) exists, and*

$$L \le c_{\min} + D'(1),$$

then $a_{\mathrm{o}}(\tau)=1$ for all $\tau>0$. If, on the other hand, the limit does not exist, or it exists and

$$L > c_{\min} + D'(1),$$

we define

$$T \triangleq \inf V^{-1}(c_{\min} + D'(1)) > 0,$$

where $V^{-1}(x)$ is the inverse image of a point x under the function V for any $x > 0$. Then we obtain $a_{\mathrm{o}}(\tau) = 1$ whenever $\tau \geq T$, and $a_{\mathrm{o}}(\tau) > 1$ whenever $0 < \tau < T$.

Proof. Suppose the limit in (3.13) exists, and $L \leq c_{\min} + D'(1)$. Then

$$V(\tau) \leq c_{\min} + D'(1),$$

and $a_{\mathrm{o}}(\tau) = 1$, for all $\tau > 0$.

Suppose next that the limit in (3.13) either does not exist, or it exists and $L > c_{\min} + D'(1)$. Since $V(\tau)$ is a monotone function,

$$V^{-1}(c_{\min} + D'(1))$$

is either a singleton or an interval. Then for any $\tau \geq T$, we have

$$V(\tau) \leq c_{\min} + D'(1),$$

which implies $a_{\mathrm{o}}(\tau) = 1$, and when $0 < \tau < T$, we have

$$c_{\min} + D'(1) < V(\tau),$$

which implies $a_{\mathrm{o}}(\tau) > 1$. □

In this section, we investigated the cost-performance tradeoff for repetition-based strategies under convex cost functions. In particular, we characterized the optimal repetition-based strategy in terms of the baseline cost, the behaviors of the incremental and fusion cost functions with different parameters, for different values of the target MSE level τ. We next study the cost-performance tradeoff under linear cost functions.

3.4.2 Linear Cost Functions

We consider the optimal repetition-based strategy in terms of cost-efficiency when the underlying cost function is linear, where we can express it as

$$C(\theta) = c_{\min} + \alpha\theta, \quad \theta > 0,$$

where $\alpha > 0$ is an application-dependent constant. This case corresponds to a *law of proportional returns.* We show that the strategy S_1 is the optimal repetition-based strategy for any target MSE $\tau > 0$. There is no gain in repetition-based approaches in terms of cost-efficiency for linear cost functions.

Theorem 3.4.3. *When the cost function $C(\theta)$ is linear, that is, $C(\theta) = c_{\min} + \alpha\theta$ for some $\alpha > 0$, then the optimal repetition-based strategy in terms of minimizing the incurred cost while achieving the same MSE is the strategy S_1.*

Proof. Let $\tau > 0$ be given. The total cost of the strategy S_N, for any positive integer N, is given by

$$\begin{aligned}\mathrm{Cost}_\tau(N) &= Nc_{\min} + \alpha \sum_{i=1}^{N} \theta_i + D(N),\\ &= Nc_{\min} + \alpha\tau^{-1} + D(N)\\ &> c_{\min} + \alpha\tau^{-1} = \mathrm{Cost}_\tau(1).\end{aligned}$$

This implies the cost incurred by the strategy S_1 is smaller than that of the strategy S_N for any $N > 1$ and $\tau > 0$. □

For proportional costs a single more reliable unit is always more cost-efficient than a fusion of several less reliable units in the sense that it incurs a smaller cost while achieving the same MSE.

3.4.3 Concave Cost Functions

We consider the cost-performance tradeoff of each strategy in the class of strategies when the cost function is concave. This case corresponds to a *law of increasing returns*, as opposed to a law of diminishing returns. That is, the incremental cost for performance decreases, making single, high-cost, high-performance elements more attractive. Before proving the main theorem of this section, we present a lemma that proves that the concave incremental cost function is sub-additive.

Lemma 3.4.4. *If a function f with the domain $[0, \infty)$ is concave, and $f(0) \geq 0$, then it is sub-additive, i.e., for any $x, y \geq 0$,*

$$f(x) + f(y) \geq f(x + y).$$

Proof. We provide the proof in Appendix B.3. □

The next theorem characterizes the optimal repetition-based strategy in terms of minimizing the total incurred cost while achieving the same MSE performance for a given $\tau > 0$.

Theorem 3.4.4. *When the cost function $C(\theta)$ is concave, and each repetition-based strategy achieves the same MSE level $\tau > 0$, then the strategy S_1 is always the optimal strategy in terms of incurring the smallest cost for any $\tau > 0$.*

Proof. Let $\tau > 0$ be given. Then, for any positive integer N, the total cost incurred by the strategy S_N is given by

$$\begin{aligned}\mathrm{Cost}_\tau(N) &= \sum_{i=1}^{N} C(\theta_i) + D(N) \\ &= \sum_{i=1}^{N} G(\theta_i) + N c_{\min} + D(N).\end{aligned}$$

We note that by Lemma 3.4.4, the incremental cost function is sub-additive, since it is concave and $G(0) \geq 0$, implying that

$$\sum_{i=1}^{N} G(\theta_i) \geq G\left(\sum_{i=1}^{N} \theta_i\right) = G\big(\tau^{-1}\big). \tag{3.14}$$

Note that the cost incurred by the strategy S_1 is given by

$$\mathrm{Cost}_\tau(1) = G\big(\tau^{-1}\big) + c_{\min},$$

implying $\mathrm{Cost}_\tau(N) > \mathrm{Cost}_\tau(1)$ for any $N > 1$. Hence, the strategy S_1 is the optimal strategy for any desired MSE. □

Strategy S_1, which is formed by exhausting all available budget for a single computational unit, is more cost-efficient as compared to any strategy

S_N with $N > 1$, which allocates available cost across several less reliable computational units.

In this section, we considered the cost-performance tradeoff of repetition-based strategies under convex, linear, and concave cost function classes. We showed that under convex cost functions the optimal cost-performance tradeoff may be achieved either by the strategy S_1 or by some strategy S_N with $N > 1$ under certain conditions. For linear and concave costs, optimality is always achieved by strategy S_1 for any target MSE performance. In the next section, we consider applications of our results into a number of contexts.

3.5 Applications

Here, we show how our cost-fidelity formulation and theoretical results are connected to problems from different fields.

3.5.1 Neuroscience

We review a particular application of our framework in a theoretical neuroscience context. We focus on two principal tasks of the brain where synapses play essential roles, namely, information storage and information processing. Typical central synapses exhibit noisy behavior due, for instance, to probabilistic transmitter release. The firing of the presynaptic neuron is inherently stochastic and occasionally fails to evoke an excitatory postsynaptic potential (EPSP). In this sense, we can cast each noisy synapse as an unreliable computational unit, contributing to the overall neural computation carried out by its efferent neuron. We focus on two distinct cost-fidelity formulations, where we show that experimental results [23, 57] agree with our theoretical predictions. We note that recall corresponds to a form of “in-memory computing” whereas processing corresponds to a form of “in-sensor computing”.

In-Memory Computing:
Revisiting [23], we first consider an information-theoretic framework to study the information storage capacity of synapses under resource constraints, where memory is seen as a communication channel subject to several sources of noise. Each synapse has a certain SNR, where increasing the SNR in-

creases the information storage capacity in a logarithmic fashion. However, this increase comes at a cost, namely, the synaptic volume. Hence, from an information storage perspective, we cast capacity as the fidelity of a noisy synapse and the volume as the cost. If we denote the information storage capacity of a synapse and its average volume by C_{I} and V, respectively, then taking Shannon's AWGN channel capacity formula [58] for concreteness:

$$C_{\mathrm{I}} = \frac{1}{2}\ln\left(1 + \frac{V}{V_N}\right),$$

where V_N is the volume of a synapse with a unit SNR. This relationship assumes the power law

$$\frac{V}{V_N} = \left(\frac{A}{A_N}\right)^2,$$

which is supported by experimental measurements [23], where A is the mean EPSP amplitude and A_N is the noise amplitude. We rewrite the volume as a function of capacity as

$$V = V_N\left(e^{2C_{\mathrm{I}}} - 1\right),$$

and observe that this is an exponential cost function, a particular example of convex costs. For exponential costs, fusion of several less reliable computational units may lead to better cost-efficiency than a single more reliable computational unit. Therefore, our cost-fidelity framework applied to information recall under resource constraints recovers the principle that several small and noisy synapses should be present in brain regions performing storage and recall, rather than large and isolated synapses [23, 59].

Moreover, [60–64] show that the noisiness of the synapses leads to efficient information transmission. That is, transmitting the same information over several less reliable but metabolically cheaper synapses requires less energy, as compared to the case where the information is transmitted over a single, more reliable but metabolically more expensive synapse. The idea that noise can facilitate information transmission is also present in neuronal networks. In particular, the authors in [65] show that a neuron is a noise-limited device of restricted bandwidth, and an energy-efficient nervous system will split the information and transmit it over a large number of relatively noisy neurons of lower information capacity.

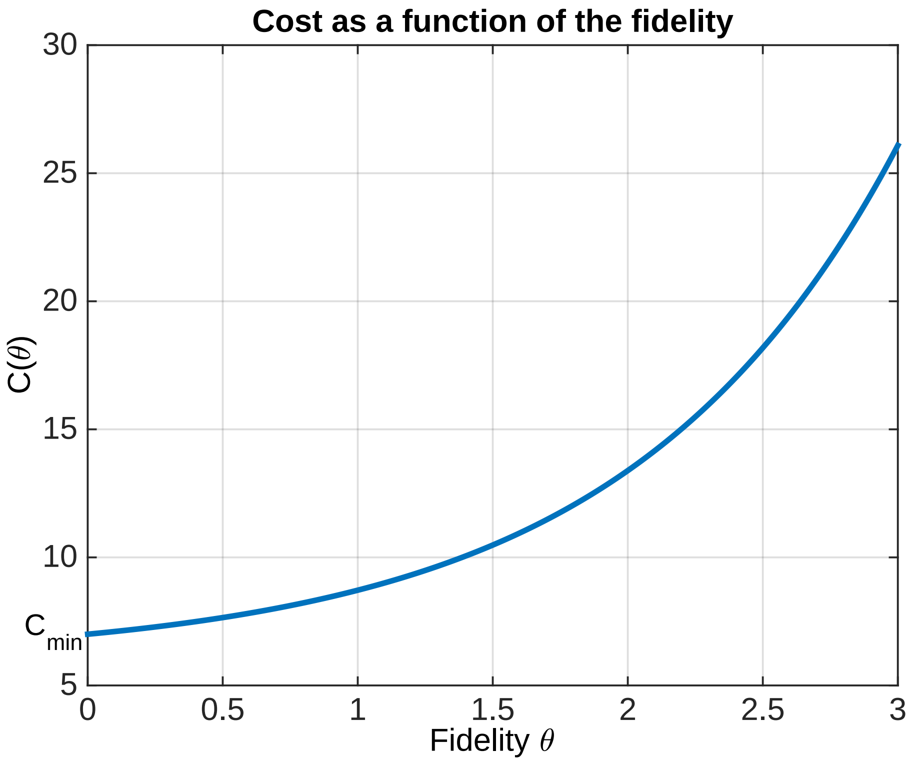


Figure 3.1: Exponential cost function (3.15).

In-Sensor Computing:
We next consider an information processing perspective, and view the SNR of a synapse itself as its fidelity and the synaptic volume as the cost. We adopt a data-driven approach using two different data sets. This joining is necessary since joint electrophysiology and imaging experiments are technically difficult, where electrophysiology experiments to measure voltages require live tissue while electron micrograph imaging experiments to measure volumes require fixing and slicing the tissue [57].

The first data set [23] includes EPSP measurements across 637 distinct synapses over 43 trials for each synapse. Based on these measurements, we generate an empirical distribution of the mean EPSP measurements of a synapse. The second data set [57] includes volume measurements across 357 synapses, which is used to compute a distribution of a synapse volume.[1]

We first generate $T = 500$ random variables $\{Y_t\}_{t=1}^{T}$ from the calculated volume distribution. We next generate T random variables from the calculated mean EPSP distribution, and sort them assuming a monotonic rela-

[1] We thank Dmitri B. Chklovskii for providing data from [57].

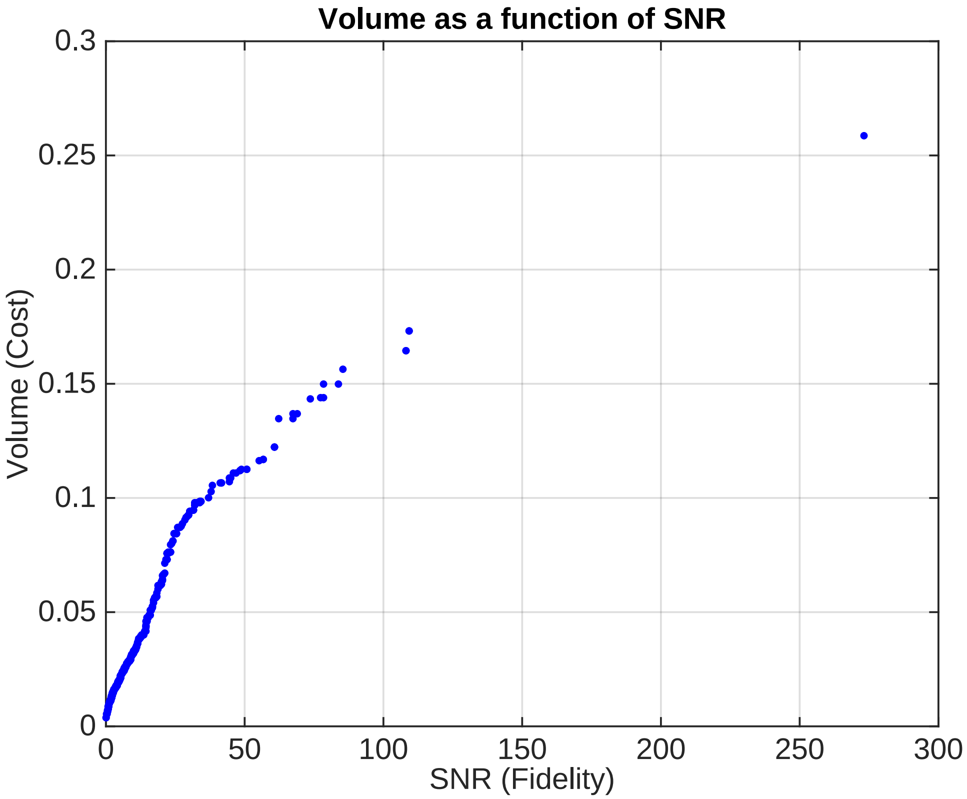


Figure 3.2: A data-driven cost (volume in μm^3) versus fidelity (SNR) function.

tionship between the mean EPSP and the volume of synapses [23]. From the sorted mean EPSP amplitudes, we compute the corresponding SNRs $\{X_t\}_{t=1}^{T}$. We plot the resulting pairs $\{(X_t, Y_t)\}_{t=1}^{T}$ in Fig. 3.2. This plot indicates that the cost function is approximately concave as a function of SNR. More rigorously, we assess convexity using a nonparametric hypothesis test based on a simplex statistic, a descriptive measure of curvature described in [66]. When applied to this data, the test yields a p-value of 3.25×10^{-4}, which can be interpreted as a strong evidence in favor of the hypothesis that the cost (volume) is a concave function of the SNR (fidelity). This suggests that the brain may achieve cost-efficiency by using a single large and reliable synapse, instead of several smaller and less reliable synapses, from an information processing perspective.

To compare this prediction with experimental findings, we focus on a particular synapse called the *calyx of Held*, the largest synapse in the mammalian auditory central nervous system that connects principal neurons within the

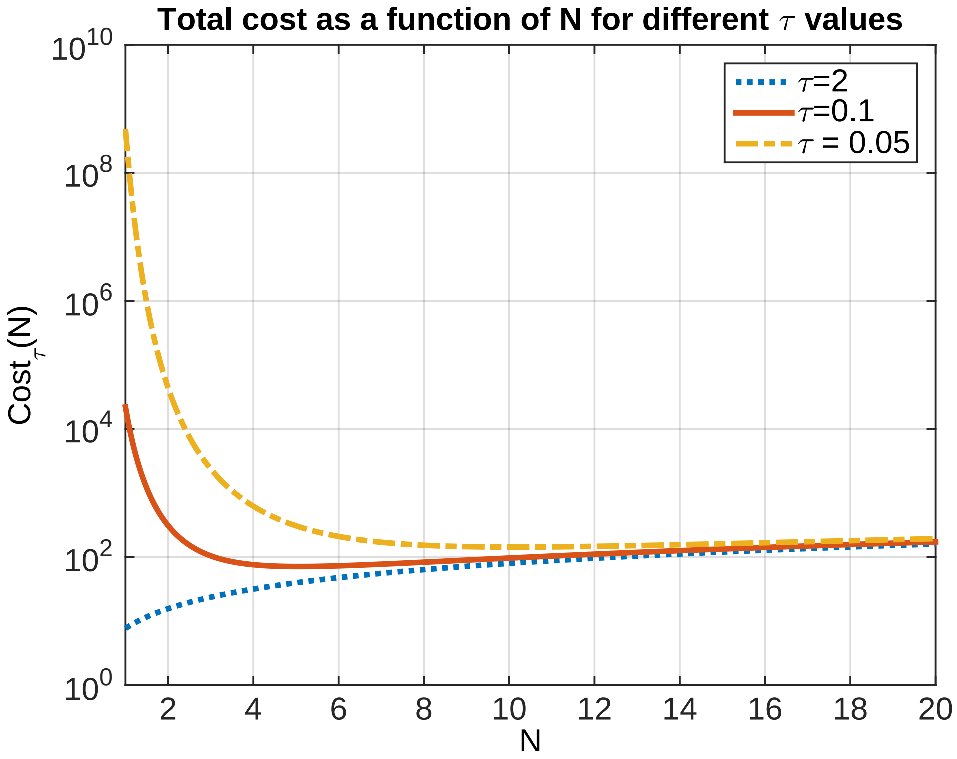


Figure 3.3: Total cost function (3.16).

auditory system [67–69]. The calyx of Held plays a crucial role in certain information processing tasks of the brain. For instance, the principal cells connected by the calyx of Held enable interaural level detection, a vital role in high-frequency sound localization [70, 71]. The signals derived from the calyx of Held generate large excitatory postsynaptic currents with a short synaptic delay, where the transmission speed and fidelity of the calyx is very reliable in mature animals [72].

Hence, the calyx of Held may be regarded as a very reliable but costly synapse, as compared to the ones performing information storage tasks, which are noisier and less costly in terms of brain resources. We observe that these experimental findings agree with our prediction that the cost-efficiency results from employing a single reliable and costly synapse (calyx of Held), outperform several less reliable and metabolically cheaper synapses, under a concave cost function.

3.5.2 Circuits

Next, let us consider signal processing systems implemented on unreliable circuit fabrics. As CMOS technology scales beyond 10 nm, the operation of CMOS devices begins to suffer from static defects as well as dynamic operational non-determinism [38,39,73]. Moreover, spintronics, which use electron spin for computing, exhibit an unreliable behavior, where there is a tradeoff between reliability and energy consumption [21, 22]. That is, probability of failure is smaller when more energy is used. Hence, deeply scaled CMOS and spintronics based systems must operate in the presence computational errors.

In [42], von Neumann studied noise in circuits and showed that even when circuit components are unreliable, reliable computations can be performed by using repetition-based schemes. Repeated computations followed by a majority vote have also been used extensively in error-tolerant circuit design [74,75]. Also, Hadjicostis [76] investigated redundancy-based approaches to build fault-tolerant dynamical systems out of cheap but unreliable components.

Moreover, a statistical error compensation technique called Algorithmic Noise Tolerance (ANT) has been studied in [77, 78], and compensates for errors in computation in a statistical manner by fusing outcomes of several unreliable computational branches that operate at different points along energy-reliability tradeoffs. The ANT framework can also be cast as a CEO problem in multiterminal source coding [79].

Stochastic behavior in circuit fabrics may also arise when computation is embedded into low-sensing, analog parts of a system such as either memory, which leads to in-memory computing [80], or sensing, which leads to in-sensor computing [81], to achieve cost-efficiency [82]. Note that in-memory computing and in-sensor computing may lead to fundamentally different cost-performance tradeoffs. In particular, we demonstrate that the difference between in-memory computing and in-sensor computing may be modeled through our framework by using different cost-fidelity function classes.

Example Case:

Here, we present an application of the results of this section to spintronics. In particular, an exponential cost has been shown to approximately model the functional dependence between energy and reliability for a typical spin device [22]. Consider the exponential cost

$$C(\theta) = c_{\min} + \alpha\big(e^{\beta\theta} - 1\big), \quad \theta > 0, \tag{3.15}$$

for some $\alpha, \beta > 0$. We illustrate this cost function in Fig. 3.1. Moreover, for illustration purposes, we assume that the fusion cost function is

$$D(N) = \gamma(N - 1),$$

for $N \geq 1$ and $\gamma > 0$. Then the total cost function is given by

$$\mathrm{Cost}_\tau(N) = \alpha N\Big(e^{\frac{\beta}{\tau N}} - 1\Big) + N(c_{\min} + \gamma) - \gamma, \tag{3.16}$$

for any positive integer N. In Fig. 3.3, we plot this total cost function with parameters

$$\alpha = 1,\ \beta = 1,\ \gamma = 1,\ c_{\min} = 7$$

for different values of the target MSE $\tau > 0$. We observe that Fig. 3.3 illustrates how $N_{\mathrm{o}}(\tau)$ increases as τ decreases, as discussed in this section. In particular, we note that $N_{\mathrm{o}}(\tau) = 1, 6, 13$ for $\tau = 2, 0.1, 0.05$, respectively.

Finally, the total cost function (3.16) yields

$$V(\tau) = \alpha \exp\big(\beta\tau^{-1}\big)\big(\beta\tau^{-1} - 1\big) + \alpha, \tag{3.17}$$

implying $V(\tau) \to \infty$ as $\tau \to 0$. Hence there exists a threshold

$$T = V^{-1}(c_{\min} + \gamma) > 0$$

such that $a_{\mathrm{o}}(\tau) = 1$ when $\tau \geq T$, and $a_{\mathrm{o}}(\tau) > 1$ when $\tau < T$. These cases are illustrated in Fig. 3.4 for $c_{\min} = 7, \gamma = 1$.

3.5.3 Crowdsourcing

Crowdsourcing assigns a task to a large number of less expensive but unreliable workers, instead of a small number of more expensive and reliable experts. Monetary payment to incentivize workers has been shown to affect the quality and the quantity of work in such scenarios [83]. Recently, motivated

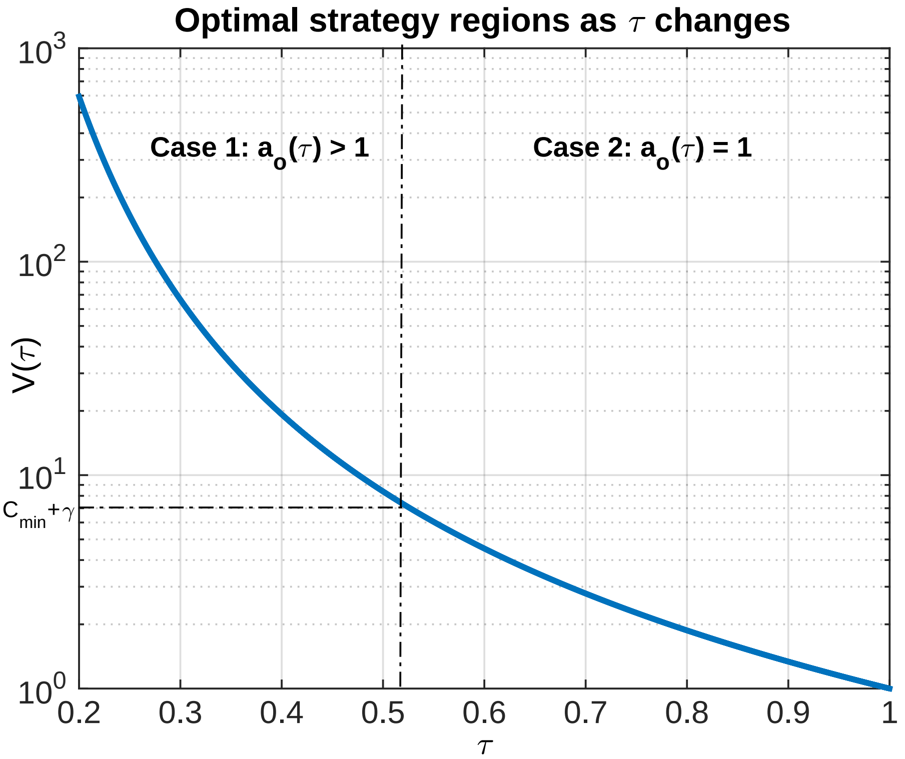


Figure 3.4: The function $V(\tau)$ (3.17) to illustrate the optimal strategy regions.

by reliability issues of crowdsourced workers and limited budgets, several researchers have pursued the limits of achievable performance from estimation-theoretic [83], information-theoretic [84], optimization [85, 86], and empirical [87] perspectives.

The authors of [87] studied the relation between monetary incentives and work quality in a knowledge task. More precisely, they performed an experiment on 451 unique workers on Amazon Mechanical Turk, and investigated the effect of bonus payments on the work quality in the task of proofreading an article. They measured the quality by the number of typographical errors found in a given article. In this scenario, each worker is paid a base salary (minimum cost), and an additional bonus (incremental cost), which is shown to yield an improvement in the work quality. In this sense, the bonus payment, i.e., the incremental cost, can be viewed as a function of the number of errors found. In particular, experiments in [87] showed that increasing the bonus payment has diminishing returns in terms of the work quality. That is, the incremental cost is a convex function of the work quality.

More recently, Lahouti and Hassibi [84] considered the crowdsourcing problem as a human-based computation problem where the main task is inference. They formulated an information-theoretic framework, where unreliable workers are modeled as parallel noisy communication channels. They represented the queries of the workers and the final inference using a joint source channel encoding/decoding scheme. Similarly, Khetan and Oh [86] studied the tradeoff between budget and accuracy in crowdsourcing scenarios under the generalized Dawid-Skene model, where they introduced an adaptive scheme to allocate a budget across unreliable workers.

We observe that there is a tradeoff between cost (monetary payments, bonus) and fidelity (quality of work) in a wide range of crowdsourcing scenarios. In particular, assigning a task to several workers, distributing the limited budget among them, and fusing their unreliable outputs have been problems of interest in the crowdsourcing literature. In this sense, our cost-fidelity formulation and repetition-based approaches may have relevance in crowdsourcing problems.

3.6 Conclusion and Future Directions

We considered fusing outcomes of several unreliable computational units that perform the same task. We modeled unreliability in a computational outcome using an additive perturbation, where the fidelity is inversely related to the variance of the perturbation. We investigated cost-performance tradeoffs achievable through repetition-based approaches. Here, each computational unit incurs a baseline cost as well as an incremental cost, which is a function of its fidelity.

We defined a class of repetition-based strategies, where any repetition-based strategy distributes the cost across several unreliable computational units and fuses their outcomes to produce a final output, where it incurs cost to perform the fusion operation. We considered the MSE of each strategy in estimating the error-free computation. In particular, we defined the optimal repetition-based strategy as the one incurring the smallest cost while achieving the desired MSE performance.

When the cost is a convex function of fidelity, the optimal repetition-based strategy may distribute cost across several less reliable computational units instead of using a single more reliable unit under certain conditions. For the classes of concave and linear cost functions we preserved that the optimal strategy uses only a single and relatively reliable computational unit, instead of a fusion of several less costly but less reliable units.

We assumed that outcomes produced by different computational units are uncorrelated. This framework can be extended to a correlated outcome setting, where a model that captures both the cost-fidelity tradeoff and the correlation between computational units may be employed. When studying the fundamental tradeoff between cost and performance, we assumed that the fusion operation is error-free. We can extend this to the case where the fusion operation also produces noisy results under cost and fidelity constraints by considering the tradeoff in allocating a budget to the fusion operation as well. Moreover, we focused on a particular fusion operation, i.e., linear combination, which is common in certain applications. More generally, we can consider nonlinear fusion rules to compute the final estimate of the error-free computation. For instance, midrange [83] and median-of-means [88] estimators have been considered as alternatives to linear estimators under different scenarios to improve performance. Extension to this framework would be of interest for different network topologies, as opposed to the centralized fusion setting of this chapter, as in [89].

CHAPTER 4

EE-GRAD: EXPLORATION AND EXPLOITATION FOR COST-EFFICIENT MINI-BATCH SGD

4.1 Introduction

Stochastic gradient methods are widely used to solve large-scale optimization problems in machine learning. Given a differentiable objective function

$$F : \mathbb{R}^d \to \mathbb{R}$$

with a gradient ∇F, a stochastic gradient descent (SGD) algorithm chooses an initial iterate $\mathbf{w}_1 \in \mathbb{R}^d$, and, on each iteration $k = 1, \ldots, K$, it uses a noisy gradient $\mathbf{G}(\mathbf{w}_k)$ instead of $\nabla F(\mathbf{w}_k)$ to set the next iterate as

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \eta_k \mathbf{G}(\mathbf{w}_k),$$

where $\eta_k > 0$ is a step size. The overall performance of stochastic gradient methods is controlled by the noise in $\mathbf{G}(\mathbf{w}_k)$ with respect to $\nabla F(\mathbf{w}_k)$ [90]. Often, noisy gradients with large variances lead to slower convergence and degraded performance [91].

Mini-batch stochastic gradient methods, as well as their distributed or parallelized variants, have been proposed to tackle some of these issues [92, 93]. Recently, federated learning [94] has been proposed as a decentralized optimization framework, where SGD runs on a large dataset distributed across a number of devices performing local model updates and sending them to a centralized server that aggregates them, under privacy and communication constraints. In typical resource- and budget-constrained applications, as the mini-batch size increases, the cost available to be allocated to each single stochastic gradient in the mini-batch decreases, so that its quality degrades, i.e., its noise variance increases. A common approach is to focus on the tradeoff between the rate of convergence and the computational complexity of stochastic gradient methods, where the dependence of the noise variance on the cost allocated to stochastic gradients is often omitted.

In this chapter, we propose an alternative framework and consider the tradeoff between *fidelity* and *cost* of computing a stochastic gradient. In particular, we model a noisy gradient as an unbiased estimate of the true gradient, where the noise variance depends on the incurred cost, and this dependence is formalized through a *cost-fidelity* function. We focus on mini-batch oracles, where each mini-batch oracle distributes a limited budget across a mini-batch of stochastic gradients and aggregates them to form a final gradient estimate. We assume that the aggregation operation also incurs a cost from a fixed budget, as does each of the noisy gradients in the mini-batch. The optimal mini-batch size in minimizing the noise variance depends on the underlying cost-fidelity function.

We focus on determining the optimal mini-batch oracle in terms of the cost-fidelity tradeoff when the cost-fidelity function is unknown. In particular, we propose and analyze EE-Grad: an algorithm that, on each iteration, performs sequential trials over different mini-batch oracles to *explore* the performance of each mini-batch oracle with high precision and *exploit* the current knowledge to focus on the one that seems to provide the best performance, i.e, the smallest noise variance. We demonstrate that the proposed algorithm performs almost as well as the optimal mini-batch oracle on each iteration in expectation. We apply this result to the case of strongly convex objectives, and prove performance guarantees in terms of the rate of convergence.

4.1.1 Organization

This chapter is organized as follows. In Section 4.2, we propose a model for stochastic gradients in terms of the true gradient and the noise variance. In particular, we formalize the dependence of the cost incurred to compute a stochastic gradient and its fidelity. We next describe mini-batch stochastic gradient oracles subject to budget constraints. In Section 4.3, we propose an algorithm that, on each iteration of the SGD, aggregates stochastic gradients computed over sequential trials, where at each trial, based on an estimate of the optimal mini-batch size, allocates the per-round budget to query the corresponding mini-batch oracle. We provide performance guarantees for the proposed algorithm in Section 4.4, where we prove an upper bound on its noise variance, and compare it to the noise variance achieved by the optimal mini-batch oracle. We next apply these results to the case of strongly convex objective functions with Lipschitz continuous gradients in Section 4.5. In Section 4.6, we finally provide a numerical example to illustrate our theoretical results. We conclude the chapter with certain remarks in Section 4.7.

4.2 Cost-Fidelity Tradeoff and Mini-Batch Stochastic Gradient Oracles

Suppose that, on each iteration, a stochastic gradient $\mathbf{g}(\mathbf{w}, \theta)$ and the gradient $\nabla F(\mathbf{w})$ are related as

$$\mathbf{g}(\mathbf{w}, \theta) = \nabla F(\mathbf{w}) + \mathbf{U}(\mathbf{w}, \theta), \tag{4.1}$$

where $\mathbf{U}(\mathbf{w}, \theta)$ is a zero-mean perturbation with a positive definite and diagonal covariance matrix $\theta^{-1}\mathbf{M}(\mathbf{w})$ for $\theta > 0$. That is,

$$\begin{aligned} \mathbb{E}_{\mathbf{w}}[\mathbf{U}(\mathbf{w}, \theta)] &= 0, \\ \mathbb{E}_{\mathbf{w}}\Big[\mathbf{U}(\mathbf{w}, \theta)\mathbf{U}(\mathbf{w}, \theta)^T\Big] &= \theta^{-1}\mathbf{M}(\mathbf{w}), \end{aligned}$$

where $\mathbb{E}_{\mathbf{w}}[\cdot]$ is the conditional expectation given $\mathbf{w}$. Here, θ is the *fidelity* of the stochastic gradient $\mathbf{g}(\mathbf{w}, \theta)$. We assume that ith element of $\mathbf{U}(\mathbf{w}, \theta)$ is sub-Gaussian with the parameter $\theta^{-1}\mathbf{M}(\mathbf{w})_{i,i}$, i.e.,

$$\mathbb{E}_{\mathbf{w}}\left[e^{\lambda \mathbf{U}(\mathbf{w}, \theta)_i}\right] \leq e^{\lambda^2 \mathbf{M}(\mathbf{w})_{i,i}/2\theta}, \quad \forall \lambda \in \mathbb{R}, \tag{4.2}$$

for $i \in [d]$.[1] A mini-batch stochastic gradient is computed by averaging n independent noisy gradients

$$\mathbf{g}_i(\mathbf{w}, \theta) = \nabla F(\mathbf{w}) + \mathbf{U}_i(\mathbf{w}, \theta),$$

[1] For any positive integer N, $[N] \triangleq \{1, \ldots, N\}$.

$i \in [n]$, each with fidelity θ:

$$\mathbf{G}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{g}_i(\mathbf{w}, \theta), \tag{4.3}$$

which has the covariance matrix $\mathbf{M}(\mathbf{w})/n\theta$, and satisfies

$$\mathbb{E}_{\mathbf{w}}\left[\|\nabla F(\mathbf{w}) - \mathbf{G}(\mathbf{w})\|_2^2\right] = \frac{S(\mathbf{w})}{n\theta},$$

where

$$S(\mathbf{w}) = \mathrm{Tr}(\mathbf{M}(\mathbf{w}))$$

is the trace of the covariance matrix.

A stochastic gradient $\mathbf{g}(\mathbf{w}, \theta)$ with fidelity $\theta > 0$ incurs a cost $C(\theta)$, which is a strictly increasing function of θ with

$$\lim_{\theta \to 0} C(\theta) = c_{\min} \geq 0.$$

We assume that the cost function $C(\theta)$ is unknown. There is also an aggregation cost $D(n)$ to perform the averaging operation, where $D(n)$ is increasing with $D(1) = 0$. Hence, given a budget $B > 0$, the maximum feasible mini-batch size is

$$N = \max\{n \in \mathbb{Z}_+ \mid B > nc_{\min} + D(n)\}.$$

Here, we define, for each $n \in [N]$, a mini-batch oracle $\mathrm{MBO}(n, B, \mathbf{w})$ that computes a mini-batch stochastic gradient $\mathbf{G}(\mathbf{w}, n)$ as in (4.3) using the fidelity

$$\theta_n \triangleq C^{-1}\left(\frac{B - D(n)}{n}\right).$$

That is, each individual stochastic gradient in the mini-batch is allocated $(B - D(n))/n$ in cost. Therefore, the covariance matrix of $\mathbf{G}(\mathbf{w}, n)$ is $\sigma_n^2 \mathbf{M}(\mathbf{w})$, where

$$\sigma_n^2 \triangleq \frac{1}{n\theta_n}$$

is unknown, since the cost function $C(\theta)$ is assumed unknown. Note that, given $\nabla F(\mathbf{w})$, the concentration of $\mathbf{G}(\mathbf{w}, n)$ around $\nabla F(\mathbf{w})$ is completely governed by σ_n^2 for each $n \in [N]$. The optimal mini-batch size in terms of the cost-fidelity tradeoff is given by

$$n_* \triangleq \arg\min_{n=1,\dots,N} \sigma_n^2,$$

and

$$\sigma_*^2 \triangleq \sigma_{n_*}^2.$$

In particular, we define the suboptimality gap of each mini-batch oracle $\mathrm{MBO}(n, B, \mathbf{w})$

$$\Delta_n \triangleq \sigma_n^2 - \sigma_*^2 \geq 0.$$

Since the cost function is unknown, the optimal mini-batch size n_* and σ_*^2, and hence the optimal mini-batch oracle

$$\mathrm{MBO}(n_*, B, \mathbf{w}),$$

are unknown. In the next section, we propose an algorithm that attempts to *learn* the optimal mini-batch oracle over sequential trials in the sense that its noise variance is almost as small as the optimal mini-batch oracle on each iteration.

4.3 The EE-Grad Algorithm

In this section, we present EE-Grad: an algorithm that, on each iteration of the SGD, aggregates stochastic gradients computed over sequential trials, where at each trial it estimates the optimal mini-batch size and uses the available per-round budget to query the corresponding mini-batch oracle. EE-Grad constructs a high confidence bound on the variance estimate of each mini-batch oracle by exploiting the sub-Gaussian assumption on the noisy gradients. We demonstrate that, in expectation, the algorithm performs almost as well as the optimal mini-batch oracle at each iteration.

On each SGD iteration, EE-Grad performs the following T-round procedure. On round $t = 1, \dots, T$, it picks a mini-batch size $n_t \in [N]$ based on a strategy introduced later in this section, and uses the per-round budget B

to query the mini-batch oracle $\mathrm{MBO}(n_t, B, \mathbf{w})$. The oracle returns

$$\mathbf{G}_t(\mathbf{w}) = \mathbf{G}_t(\mathbf{w}, n_t),$$

an unbiased estimate of $\nabla F(\mathbf{w})$, with covariance matrix $\sigma_{n_t}^2 \mathbf{M}(\mathbf{w})$. After T rounds, the algorithm outputs the stochastic gradient

$$\mathbf{G}(\mathbf{w}) = \frac{1}{T} \sum_{t=1}^{T} \mathbf{G}_t(\mathbf{w}).$$

We denote the number of rounds the algorithm picks $\mathrm{MBO}(n, B, \mathbf{w})$ up to round t as $\gamma_t(n)$, index its outputs as

$$\mathbf{G}_1(\mathbf{w}, n), \ldots, \mathbf{G}_{\gamma_t(n)}(\mathbf{w}, n),$$

and write its sample mean and sample covariance matrix as

$$\begin{aligned} \mathbf{m}_t(n) &= \frac{1}{\gamma_t(n)} \sum_{i=1}^{\gamma_t(n)} \mathbf{G}_i(\mathbf{w}, n), \\ \mathbf{Cov}_t(n) &= \frac{1}{\gamma_t(n) - 1} \sum_{i=1}^{\gamma_t(n)} (\mathbf{G}_i(\mathbf{w}, n) - \mathbf{m}_t(n))(\mathbf{G}_i(\mathbf{w}, n) - \mathbf{m}_t(n))^T, \end{aligned}$$

respectively, for $n \in [N]$. The algorithm computes the trace of the sample covariance matrix, denoted by

$$\begin{aligned} V_t(n) &= \mathrm{Tr}(\mathbf{Cov}_t(n)) \\ &= \frac{1}{\gamma_t(n) - 1} \sum_{i=1}^{\gamma_t(n)} (\mathbf{G}_i(\mathbf{w}, n) - \mathbf{m}_t(n))^T (\mathbf{G}_i(\mathbf{w}, n) - \mathbf{m}_t(n)) \end{aligned}$$

for each $n \in [N]$. Note that

$$\mathbb{E}_{\mathbf{w}}[V_t(n)] = \sigma_n^2 S(\mathbf{w}),$$

which implies that for each $\mathrm{MBO}(n, B, \mathbf{w})$, the trace of its sample covariance matrix is an unbiased estimate of $\sigma_n^2 S(\mathbf{w})$.

We emphasize that this framework is similar to the stochastic multi-armed bandit setup that involves an exploration/exploitation tradeoff when picking different arms over sequential trials [25]. In particular, algorithms that exploit the available knowledge on the current best arm and explore the other arms to estimate the actual best arm with higher precision have been shown to yield satisfactory performance [24,25]. We adopt a similar approach here, and propose an algorithm that simultaneously performs exploration and exploitation. More precisely, EE-Grad is initialized by picking each mini-batch oracle exactly twice, so that $\gamma_t(n) = 2$ for each $n \in [N]$ at trial $t = 2N$, and then picks the mini-batch oracle at trial $t = 2N + 1, \dots, T$ according to

$$n_t \in \arg\min_{n=1,\dots,N} \left[V_t(n) - f\left(\frac{\alpha \ln(t)}{\gamma_t(n) - 1} \right) \right], \tag{4.4}$$

for some $\alpha > 2$, where

$$f(x) \triangleq \beta P \sqrt{\frac{xd}{c}} \max\left(1, \sqrt{\frac{x}{cd}}\right), \tag{4.5}$$

and $c > 0$ is a universal constant that comes from the use of Hanson-Wright inequality, as detailed in the proof of Theorem 4.4.1, and we assume that β and P are known constants such that

$$\sigma_n^2 \leq \beta$$

for each $n \in [N]$, and

$$S(\mathbf{w}) \leq P.$$

This algorithm constructs an upper confidence bound (UCB) on the trace of the sample covariance matrix of each mini-batch oracle, and picks the one with the best estimate. The overall scheme, presented as Algorithm 6, will be analyzed using techniques similar to those used in UCB strategies [24,95,96], as explained in the proof of Theorem 4.4.1.

Algorithm 6 EE-Grad

```
Input: Number of mini-batch oracles $N > 1$, number of sequential trials $T > 1$, per-round budget $B$.
Initialization:
for $t = 1 : 2N$ do
    Set $n = \lceil t/2 \rceil$, and use the mini-batch size $n_t = n$.
    Distribute the budget $B$ to MBO$(n_t, B, \mathbf{w})$, which reveals $\mathbf{G}_t(\mathbf{w}, n)$, and set
        $\mathbf{G}_t(\mathbf{w}) = \mathbf{G}_t(\mathbf{w}, n).$
end for
Main Loop:
for $t = 2N + 1 : T$ do
    Compute $V_t(n)$ for each $n \in [N]$, and pick a mini-batch size $n_t$ based on (4.4).
    Distribute the budget $B$ to MBO$(n_t, B, \mathbf{w})$, which reveals $\mathbf{G}_t(\mathbf{w}, n)$, and set
        $\mathbf{G}_t(\mathbf{w}) = \mathbf{G}_t(\mathbf{w}, n).$
end for
Compute the final gradient estimate as
    $\mathbf{G}(\mathbf{w}) = \frac{1}{T} \sum_{t=1}^{T} \mathbf{G}_t(\mathbf{w}).$
```

4.4 EE-Grad Performance Guarantees

In this section, we investigate the performance of EE-Grad. In particular, we prove an upper bound on its noise variance, and compare it to the noise variance achieved by the optimal mini-batch oracle.

Theorem 4.4.1. *On each iteration, the stochastic gradient computed by EE-Grad satisfies*

$$\mathbb{E}_{\mathbf{w}}\left[\|\mathbf{G}(\mathbf{w}) - \nabla F(\mathbf{w})\|_2^2\right] \leq Z_T(\mathbf{w}) S(\mathbf{w}),$$

where

$$Z_T(\mathbf{w}) = \frac{\sigma_*^2}{T} + \left(\frac{\ln T}{T^2}\right)\mathrm{C}_1(\mathbf{w}) + \left(\frac{1}{T^2}\right)\mathrm{C}_2,$$

and

$$\mathrm{C}_1(\mathbf{w}) \triangleq \sum_{n:\Delta_n>0} \frac{\alpha\Delta_n}{\phi(\Delta_n S(\mathbf{w})/2)},$$
$$\mathrm{C}_2 \triangleq \left(\sum_{n=1}^{N} \Delta_n\right)\frac{2(\alpha-1)}{\alpha-2},$$
$$\phi(\varepsilon) \triangleq \frac{c\varepsilon}{\beta P}\min\left(1, \frac{\varepsilon/d}{\beta P}\right).$$

Also, the stochastic gradient $\mathbf{G}^*(\mathbf{w})$ *computed by the optimal mini-batch oracle satisfies*

$$\mathbb{E}_{\mathbf{w}}\left[\|\mathbf{G}^*(\mathbf{w}) - \nabla F(\mathbf{w})\|_2^2\right] = \frac{\sigma_*^2}{T}S(\mathbf{w}).$$

Proof. We prove this theorem in several steps. We first analyze the difference between the noise variance of the stochastic gradient generated by EE-Grad and that of the optimal mini-batch oracle. We next show that this quantity is related to the pseudo-regret term that appears in stochastic multi-armed bandit problems, where UCB-type strategies are used to achieve upper bounds on the pseudo-regret by leveraging concentration inequalities. We present a similar formulation to analyze the behavior of the proposed algorithm with respect to the optimal mini-batch oracle. To prove the upper bound, we first demonstrate that the trace of the sample covariance matrix for each mini-batch oracle, which is used to pick an oracle on each trial in (4.4), can be written as a quadratic form of independent sub-Gaussian random variables. We combine this observation with the Hanson-Wright inequality [97] to prove a high probability tail bound on the estimate of the optimal mini-batch size. This result also is the derivation of the rule in (4.4). Based on these results, we prove a pseudo-regret bound and connect this bound to the noise variance achieved by EE-Grad.

Note that, on each iteration, the stochastic gradient of the optimal mini-batch oracle after T rounds is

$$\mathbf{G}^*(\mathbf{w}) \triangleq \frac{1}{T}\sum_{t=1}^{T} \mathbf{G}_t(\mathbf{w}, n_*),$$

where

$$\mathbf{G}_1(\mathbf{w}, n_*), \ldots, \mathbf{G}_T(\mathbf{w}, n_*)$$

are independent. We observe that

$$\begin{aligned}
&\mathbb{E}_{\mathbf{w}}\big[\|\mathbf{G}(\mathbf{w}) - \nabla F(\mathbf{w})\|_2^2\big] - \mathbb{E}_{\mathbf{w}}\big[\|\mathbf{G}^*(\mathbf{w}) - \nabla F(\mathbf{w})\|_2^2\big] \\
&= \frac{1}{T^2}\left(\sum_{t=1}^{T} \mathbb{E}_{\mathbf{w}}\big[\|\mathbf{G}_t(\mathbf{w}) - \nabla F(\mathbf{w})\|_2^2\big] - \sum_{t=1}^{T} \mathbb{E}_{\mathbf{w}}\big[\|\mathbf{G}_t(\mathbf{w}, n_*) - \nabla F(\mathbf{w})\|_2^2\big]\right) \\
&= \frac{1}{T^2}\left(\sum_{t=1}^{T} \mathbb{E}_{\mathbf{w}}\big[\|\mathbf{G}_t(\mathbf{w}) - \nabla F(\mathbf{w})\|_2^2\big] - T\sigma_*^2 S(\mathbf{w})\right), \qquad (4.6)
\end{aligned}$$

where in (4.6) we used

$$\mathbb{E}_{\mathbf{w}}\big[\|\mathbf{G}_t(\mathbf{w}, n_*) - \nabla F(\mathbf{w})\|_2^2\big] = \sigma_*^2 S(\mathbf{w})$$

for each $t \in [T]$. We next observe that

$$\begin{aligned}
\mathbb{E}_{\mathbf{w}}\big[\|\mathbf{G}_t(\mathbf{w}) - \nabla F(\mathbf{w})\|_2^2\big] &= \mathbb{E}_{\mathbf{w}}\big[\mathbb{E}_{\mathbf{w}}\big[\|\mathbf{G}_t(\mathbf{w}) - \nabla F(\mathbf{w})\|_2^2 \mid n_t\big]\big] \\
&= \mathbb{E}_{\mathbf{w}}\big[\sigma_{n_t}^2\big] S(\mathbf{w}), \qquad (4.7)
\end{aligned}$$

where in (4.7) the expectation is with respect to the randomness in n_t. In particular, we can write

$$\mathbb{E}_{\mathbf{w}}\big[\sigma_{n_t}^2\big] = \sum_{n=1}^{N} \sigma_n^2 \Pr(n_t = n) \qquad (4.8)$$

for each $t \in [T]$. If we substitute (4.8) into (4.7) and use the result in (4.6), then we obtain

$$\begin{aligned}
&\mathbb{E}_{\mathbf{w}}\big[\|\mathbf{G}(\mathbf{w}) - \nabla F(\mathbf{w})\|_2^2\big] - \mathbb{E}_{\mathbf{w}}\big[\|\mathbf{G}^*(\mathbf{w}) - \nabla F(\mathbf{w})\|_2^2\big] \\
&= \frac{1}{T^2}\left(\sum_{n=1}^{N} \sigma_n^2 \sum_{t=1}^{T} \Pr(n_t = n) - T\sigma_*^2\right) S(\mathbf{w}),
\end{aligned}$$

$$= \frac{1}{T^2} \left(\sum_{n=1}^{N} \sigma_n^2 \sum_{t=1}^{T} \mathbb{E}_{\mathbf{w}}[\mathbb{1}\{n_t = n\}] - T\sigma_*^2 \right) S(\mathbf{w}) \tag{4.9}$$

$$= \frac{1}{T^2} \left(\sum_{n=1}^{N} \sigma_n^2 \, \mathbb{E}_{\mathbf{w}}[\gamma_T(n)] - \sigma_*^2 \sum_{n=1}^{N} \mathbb{E}_{\mathbf{w}}[\gamma_T(n)] \right) S(\mathbf{w}) \tag{4.10}$$

$$= \frac{1}{T^2} \, \mathbb{E}_{\mathbf{w}} \left[\sum_{n=1}^{N} \Delta_n \gamma_T(n) \right] S(\mathbf{w}), \tag{4.11}$$

where in (4.9) we used

$$\Pr(n_t = n) = \mathbb{E}_{\mathbf{w}}[\mathbb{1}\{n_t = n\}],$$

in (4.10) we used

$$\gamma_T(n) = \sum_{t=1}^{T} \mathbb{1}\{n_t = n\}$$

and

$$\sum_{n=1}^{N} \gamma_T(n) = T,$$

and in (4.11) we used

$$\Delta_n = \sigma_n^2 - \sigma_*^2.$$

We note that the term

$$\mathbb{E}_{\mathbf{w}} \left[\sum_{n=1}^{N} \Delta_n \gamma_T(n) \right] S(\mathbf{w})$$

is similar to the pseudo-regret term that appears in stochastic multi-armed bandit problems, where there are N arms with unknown reward distributions [25]. We derive the strategy in (4.4) based on similar arguments, where we leverage a novel application of the Hanson-Wright inequality to the trace of the sample covariance matrix of each mini-batch oracle to prove concentration inequalities.

To prove an upper bound on (4.11), we first show in Lemma C.1.1 that $V_t(n)$ can be written as a quadratic form of sub-Gaussian random variables as

$$V_t(n) = \mathbf{s}_{t,n}^T \mathbf{A}_{t,n} \mathbf{s}_{t,n}, \;\; n \in [N],$$

where

$$\mathbf{s}_{t,n} \triangleq \left(\mathbf{G}_1(\mathbf{w}, n)^T, \ldots, \mathbf{G}_{\gamma_t(n)}(\mathbf{w}, n)^T\right)^T,$$

and

$$\mathbf{A}_{t,n} = \frac{1}{\gamma_t(n) - 1}\left(\mathbf{I} - \frac{1}{\gamma_t(n)}\mathbf{E}\right),$$

$\mathbf{I} \in \mathbb{R}^{d\gamma_t(n) \times d\gamma_t(n)}$ is an identity matrix, and

$$\mathbf{E} \in \mathbb{R}^{d\gamma_t(n) \times d\gamma_t(n)}$$

is a block matrix with $d \times d$ identity blocks. We next apply the Hanson-Wright inequality [97,98] to $V_t(n)$ for each $n \in [N]$ to obtain high confidence bounds. This inequality provides a tail probability bound for an arbitrary quadratic function of independent sub-Gaussian random variables. We present this inequality in the appendix for completeness. Moreover, Lemma C.3.1 shows that the tail probability of the trace of the sample covariance matrix of each mini-batch oracle satisfies, for any $\varepsilon > 0$,

$$\Pr\left(V_t(n) - \sigma_n^2 S(\mathbf{w}) > \varepsilon\right) \leq \exp(-(\gamma_t(n) - 1)\phi(\varepsilon)), \tag{4.12}$$

where

$$\phi(\varepsilon) \triangleq \frac{c\varepsilon}{\beta P} \min\left(1, \frac{\varepsilon/d}{\beta P}\right),$$

for each $n \in [N]$. We observe that $\phi = f^{-1}$, where f is defined in (4.5).

Note that (4.12) is equivalent to stating that, for any $\delta \in (0, 1)$,

$$V_t(n) - f\left(\frac{1}{\gamma_t(n) - 1} \ln\left(\frac{1}{\delta}\right)\right) \leq \sigma_n^2 S(\mathbf{w}) \tag{4.13}$$

with probability at least $1 - \delta$. Using this result, we propose the UCB-type strategy in (4.4) to pick the mini-batch oracle on round t. In particular, we show in Lemma C.4.1 that, for any $\alpha > 2$, we have

$$\mathbb{E}_{\mathbf{w}}\left[\sum_{n=1}^{N} \Delta_n \gamma_T(n)\right] S(\mathbf{w}) \leq (\mathrm{C}_1(\mathbf{w}) \ln(T) + \mathrm{C}_2) S(\mathbf{w}), \tag{4.14}$$

where

$$\mathrm{C}_1(\mathbf{w}) \triangleq \sum_{n:\Delta_n>0} \frac{\alpha\Delta_n}{\phi(\Delta_n S(\mathbf{w})/2)},$$
$$\mathrm{C}_2 \triangleq \left(\sum_{n=1}^{N} \Delta_n\right)\frac{2(\alpha-1)}{\alpha-2}.$$

Finally, if we use (4.14) in (4.11), then we obtain

$$\begin{aligned}&\mathbb{E}_{\mathbf{w}}\left[\|\mathbf{G}(\mathbf{w}) - \nabla F(\mathbf{w})\|_2^2\right] - \mathbb{E}_{\mathbf{w}}\left[\|\mathbf{G}^*(\mathbf{w}) - \nabla F(\mathbf{w})\|_2^2\right]\\ &\leq \frac{1}{T^2}(\mathrm{C}_1(\mathbf{w})\ln(T) + \mathrm{C}_2)S(\mathbf{w}),\end{aligned} \tag{4.15}$$

where substituting

$$\mathbb{E}_{\mathbf{w}}\left[\|\mathbf{G}^*(\mathbf{w}) - \nabla F(\mathbf{w})\|_2^2\right] = \frac{\sigma_*^2 S(\mathbf{w})}{T}$$

in (4.15) yields the desired result. □

4.5 SGD Performance under Strongly Convex Objectives

In this section, we investigate the performance of EE-Grad with strongly convex objective functions with Lipschitz continuous gradients. That is, we assume that the gradient ∇F is Lipschitz continuous with Lipschitz constant $L > 0$, i.e.,

$$\|\nabla F(\mathbf{w}) - \nabla F(\overline{\mathbf{w}})\|_2 \leq L\|\mathbf{w} - \overline{\mathbf{w}}\|_2, \quad \forall \mathbf{w}, \overline{\mathbf{w}} \in \mathbb{R}^d,$$

and there exists $m > 0$ such that

$$F(\overline{\mathbf{w}}) \geq F(\mathbf{w}) + \nabla F(\mathbf{w})^T(\overline{\mathbf{w}} - \mathbf{w}) + \frac{1}{2}m\|\overline{\mathbf{w}} - \mathbf{w}\|_2^2, \quad \forall \overline{\mathbf{w}}, \mathbf{w} \in \mathbb{R}^d.$$

Let

$$\mathbf{w}_* = \arg\min_{\mathbf{w}\in\mathbb{R}^d} F(\mathbf{w})$$

be the global minimizer. We first describe the *optimal* mini-batch SGD algorithm that uses the optimal mini-batch oracle on each iteration. We next compare its performance to EE-Grad in terms of the rate of convergence to the global solution $\mathbf{w}_*$. Note that the cost function $C(\theta)$, and hence the optimal mini-batch size, is allowed to vary across iterations of the SGD algorithm. We use the subscript k, which denotes the SGD iteration, for the quantities introduced in Section 4.2 and Section 4.3 to emphasize the iteration dependence whenever necessary.

On each iteration $k = 1, \dots, K$, the optimal mini-batch SGD algorithm that knows the optimal mini-batch oracle

$$\mathrm{MBO}(n_{*,k}, B, \mathbf{w}_k^{\mathrm{o}})$$

distributes the per-round budget B_k to it producing

$$\mathbf{G}_t^{\mathrm{o}}(\mathbf{w}_k^{\mathrm{o}}) = \mathbf{G}_t^{\mathrm{o}}(\mathbf{w}_k^{\mathrm{o}}, n_*)$$

on each trial $t = 1, \dots, T$. After T trials, it computes its final stochastic gradient as

$$\mathbf{G}^{\mathrm{o}}(\mathbf{w}_k^{\mathrm{o}}) = \frac{1}{T}\sum_{t=1}^{T} \mathbf{G}_t^{\mathrm{o}}(\mathbf{w}_k^{\mathrm{o}}),$$

and sets the next iterate as

$$\mathbf{w}_{k+1}^{\mathrm{o}} = \mathbf{w}_k^{\mathrm{o}} - \eta \mathbf{G}^{\mathrm{o}}(\mathbf{w}_k^{\mathrm{o}}).$$

We observe that $\mathbf{w}_k$ and $\mathbf{w}_k^{\mathrm{o}}$ may be different over iterations, so the true gradients $\nabla F(\mathbf{w}_k)$ and $\nabla F(\mathbf{w}_k^{\mathrm{o}})$ also may differ. Also, note that $\mathbf{G}^{\mathrm{o}}(\mathbf{w}_k^{\mathrm{o}})$ satisfies

$$\mathbb{E}_{\mathbf{w}}\big[\|\mathbf{G}^{\mathrm{o}}(\mathbf{w}_k^{\mathrm{o}}) - \nabla F(\mathbf{w}_k^{\mathrm{o}})\|_2^2\big] = \frac{\sigma_*^2 S(\mathbf{w}_k^{\mathrm{o}})}{T},$$

where $S(\mathbf{w}_k^{\mathrm{o}}) \triangleq \mathrm{Tr}(\mathbf{M}(\mathbf{w}_k^{\mathrm{o}}))$ for each $k \in [K]$. In this section, we focus on the case where

$$\mathbf{M}(\mathbf{w}) \triangleq \mathrm{diag}\big(\nabla F(\mathbf{w})_1^2, \dots, \nabla F(\mathbf{w})_d^2\big)$$

for any $\mathbf{w} \in \mathbb{R}^d$, which implies that

$$S(\mathbf{w}) = \|\nabla F(\mathbf{w})\|_2^2.$$

We define the expected gaps of EE-Grad and of the optimal mini-batch SGD algorithm with respect to the global minimizer $\mathbf{w}_*$ on each iteration k as

$$\begin{aligned} J_{k,\eta} &\triangleq \mathbb{E}[F(\mathbf{w}_k)] - F(\mathbf{w}_*), \\ J^{\mathrm{o}}_{k,\eta} &\triangleq \mathbb{E}[F(\mathbf{w}^{\mathrm{o}}_k)] - F(\mathbf{w}_*), \end{aligned} \tag{4.16}$$

respectively. The next theorem shows how these expected gaps evolve over iterations.

Theorem 4.5.1. *Suppose that the step size η_k is sufficiently small so that it satisfies*

$$0 < \eta_k < \frac{2}{L(1 + Z_T(\mathbf{w}_k))}. \tag{4.17}$$

Then, on each iteration k, the expected gap of the optimal mini-batch SGD algorithm satisfies

$$J^{\mathrm{o}}_{k+1,\eta} \le \tau^{\mathrm{o}}_k(\eta_k) J^{\mathrm{o}}_{k,\eta},$$

where

$$0 < \tau^{\mathrm{o}}_k(\eta_k) \triangleq mL\eta_k^2\big(1 + \sigma^2_{*,k}/T\big) - 2m\eta_k + 1 < 1.$$

Moreover, the expected gap of the EE-Grad algorithm on iteration k satisfies

$$J_{k+1,\eta} \le \tau_k(\eta_k) J_{k,\eta},$$

where

$$0 < \tau_k(\eta_k) \triangleq \tau^{\mathrm{o}}_k(\eta_k) + mL\eta_k^2 O_{T,k} < 1,$$

and

$$O_{T,k} \triangleq Z_T(\mathbf{w}_k) - \frac{\sigma^2_{*,k}}{T} = \frac{\mathrm{C}_{1,k}(\mathbf{w}) \ln T}{T^2} + \frac{\mathrm{C}_{2,k}}{T^2} > 0,$$

where $O_{T,k} \to 0$ *as* $T \to \infty$.

Proof. First note that since ∇F is Lipschitz continuous with Lipschitz constant $L > 0$, it satisfies [90]

$$F(\mathbf{w}) \leq F(\overline{\mathbf{w}}) + \nabla F(\overline{\mathbf{w}})^T(\mathbf{w} - \overline{\mathbf{w}}) + \frac{1}{2}L\|\mathbf{w} - \overline{\mathbf{w}}\|_2^2, \quad \forall \mathbf{w}, \overline{\mathbf{w}} \in \mathbb{R}^d,$$

which implies that on each iteration k, we have

$$F(\mathbf{w}_{k+1}) - F(\mathbf{w}_k) \leq -\eta_k \nabla F(\mathbf{w}_k)^T \mathbf{G}(\mathbf{w}_k) + \frac{1}{2}L\eta_k^2\|\mathbf{G}(\mathbf{w}_k)\|_2^2. \tag{4.18}$$

By taking conditional expectations of both sides and rearranging the terms, we obtain

$$\mathbb{E}_k[F(\mathbf{w}_{k+1})] - F(\mathbf{w}_k) \leq -\eta_k S(\mathbf{w}_k)\bigg(1 - \frac{1}{2}\eta_k L(1 + Z_T(\mathbf{w}_k))\bigg). \tag{4.19}$$

Performing the same steps on the optimal mini-batch SGD algorithm yields

$$\mathbb{E}_k\big[F\big(\mathbf{w}_{k+1}^{\mathrm{o}}\big)\big] - F(\mathbf{w}_k^{\mathrm{o}}) \leq -\eta_k S(\mathbf{w}_k^{\mathrm{o}})\bigg(1 - \frac{1}{2}\eta_k L\big(1 + \sigma_{*,k}^2/T\big)\bigg). \tag{4.20}$$

Since F is assumed to be m-strongly convex, the optimality gap for any $\mathbf{w} \in \mathbb{R}^d$ satisfies [90]

$$F(\mathbf{w}) - F(\mathbf{w}_*) \leq \frac{1}{2m}\|\nabla F(\mathbf{w})\|_2^2. \tag{4.21}$$

The assumption in (4.17) guarantees that

$$1 - \eta_k L(1 + Z_T(\mathbf{w}_k))/2 > 0.$$

Thus, using (4.21) in (4.19), subtracting $F(\mathbf{w}_*)$ on both sides, and rearranging terms give

$$\begin{aligned}\mathbb{E}_k[F(\mathbf{w}_{k+1})] - F(\mathbf{w}_*) &\leq F(\mathbf{w}_k) - F(\mathbf{w}_*) - \eta_k S(\mathbf{w}_k)\bigg(1 - \frac{1}{2}\eta_k L(1 + Z_T(\mathbf{w}_k))\bigg) \\ &\leq F(\mathbf{w}_k) - F(\mathbf{w}_*) \\ &\quad - 2m\eta_k(F(\mathbf{w}_k) - F(\mathbf{w}_*))\bigg(1 - \frac{1}{2}\eta_k L(1 + Z_T(\mathbf{w}_k))\bigg) \\ &= \big(mL\eta_k^2(1 + Z_T(\mathbf{w}_k)) - 2m\eta + 1\big)(F(\mathbf{w}_k) - F(\mathbf{w}_*)),\end{aligned}$$

$$= \tau_k(\eta_k)(F(\mathbf{w}_k) - F(\mathbf{w}_*)).$$

Here if we take expectations of both sides and note the definition in (4.16), then we obtain

$$J_{k+1,\eta} \leq \tau_k(\eta_k) J_{k,\eta}.$$

Similar steps for the optimal mini-batch SGD algorithm imply

$$J^{\mathrm{o}}_{k+1,\eta} \leq \tau^{\mathrm{o}}_k(\eta_k) J^{\mathrm{o}}_{k,\eta},$$

where $\tau_k(\eta_k) = \tau^{\mathrm{o}}_k(\eta_k) + mL\eta_k^2 O_{T,k}$, so that

$$\tau_k(\eta_k) - \tau^{\mathrm{o}}_k(\eta_k) \to 0$$

as $T \to \infty$. □

Here, we note that $\tau_k(\eta_k)$ is a quadratic function of η_k, minimized at

$$\eta_k = \frac{1}{1 + Z_T(\mathbf{w}_k)},$$

and $\tau_k(\eta_k) < 1$ for all η_k satisfying (4.17). Similarly, $\tau^{\mathrm{o}}_k(\eta_k)$ is a quadratic function of η_k, minimized at

$$\eta_k = \frac{1}{1 + \frac{\sigma^2_{*,k}}{T}},$$

and $\tau^{\mathrm{o}}_k(\eta_k) < 1$ for all η_k satisfying (4.17). Also, we observe that

$$\tau_k(\eta_k) = \tau^{\mathrm{o}}_k(\eta_k) + mL\eta_k^2 O_{T,k} > \tau^{\mathrm{o}}_k(\eta_k)$$

for all $\eta_k > 0$, i.e., $\tau_k(\eta_k)$ is uniformly larger than $\tau^{\mathrm{o}}_k(\eta_k)$, which implies that the optimal mini-batch SGD algorithm enjoys faster convergence rate than the proposed algorithm. However, the gap between them is proportional to $O_{T,k}$ for any given step size $\eta_k > 0$, which is the gap between EE-Grad and the optimal mini-batch SGD algorithm, as shown in Theorem 4.4.1. Finally, we note that this gap diminishes as the number of trials T increases, at the expense of larger total incurred cost. In the next section, we illustrate our theoretical results with numerical examples.

4.6 Numerical Results

In this section, we present a numerical example based on synthetic data to illustrate our main results. We consider the $d = 2$ dimensional case, where the per-iteration budget is $B = 1$, and the objective function and its gradient are

$$F(\mathbf{w}) = \frac{\mathbf{w}^T\mathbf{w}}{2}$$

and $\nabla F(\mathbf{w}) = \mathbf{w}$, respectively, where $F(\mathbf{w}_*) = 0$ with $\mathbf{w}_* = (0,0)^T$.

We assume that

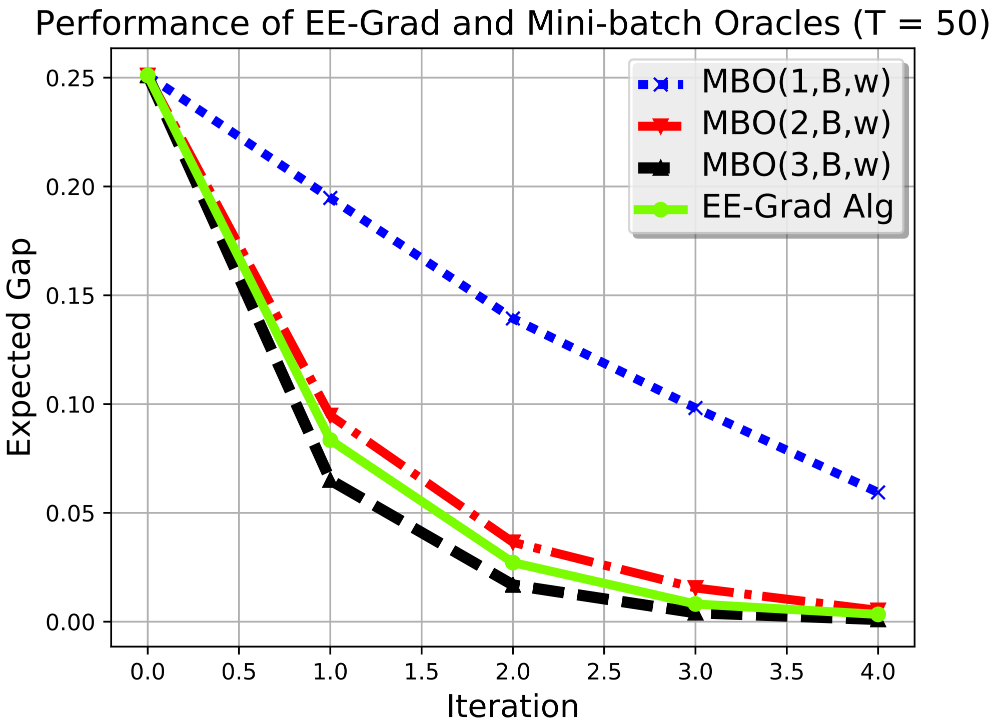
$$\mathbf{M}(\mathbf{w}) = \mathrm{diag}\big(w_1^2, w_2^2\big),$$

and each stochastic gradient $\mathbf{g}(\mathbf{w}, \theta)$ with fidelity $\theta > 0$ has uncorrelated Gaussian components with the parameters w_1^2/θ and w_2^2/θ, respectively. We next assume that the unknown parameters of the mini-batch oracles are given by $\sigma_1^2 = 50, \sigma_2^2 = 26, \sigma_3^2 = 16.7$, and run the EE-Grad algorithm and the mini-batch oracles with a randomly generated initial iterate for $T = 50$ trials and $K = 5$ iterations by using the constant step size $\eta = 0.85$, where we obtain expected results over 2000 independent realizations. We plot the resulting expected gaps achieved by EE-Grad and the mini-batch oracles in Fig. 4.1a. We repeat the same procedure for $T = 200$ and $T = 3000$ and plot the results in Fig. 4.2a and Fig. 4.2b, where we note that σ_i^2 are scaled accordingly, so that the results over different Ts are comparable.

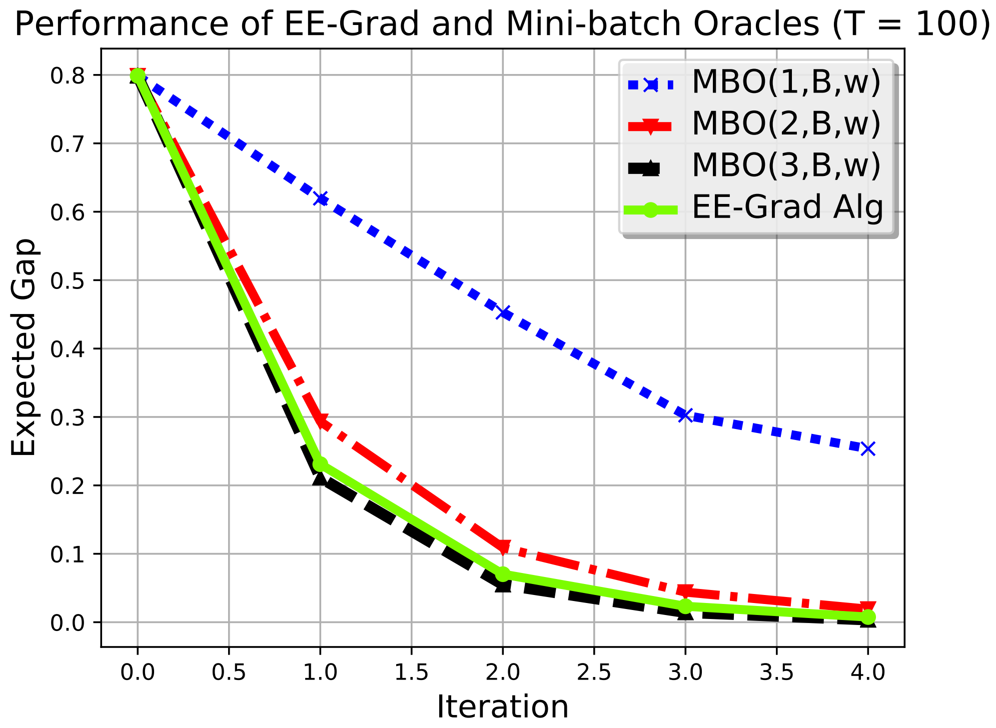
We observe that for this numerical example, the expected gap achieved by the EE-Grad algorithm is close to that of the optimal mini-batch oracle, where the performance difference between them shrinks with increasing T at the expense of increased total cost, as we proved in Theorem 4.5.1.

4.7 Discussion

We presented a new framework to analyze the tradeoff between *fidelity* and *cost* of computing a stochastic gradient, where we modeled a noisy gradient as an unbiased estimate of the true gradient such that the noise variance depends on the cost incurred to compute it. We investigated mini-batch oracles that distribute a limited budget to a mini-batch of stochastic gradients and averages them to estimate the true gradient, where the averaging opera-

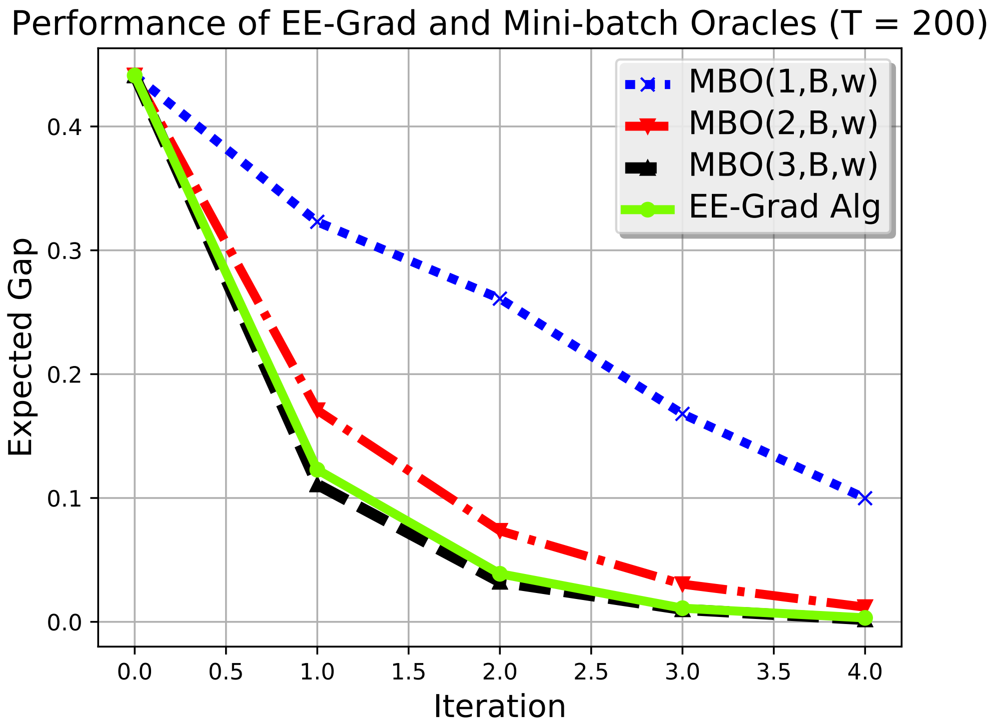


(a) $T = 50$.

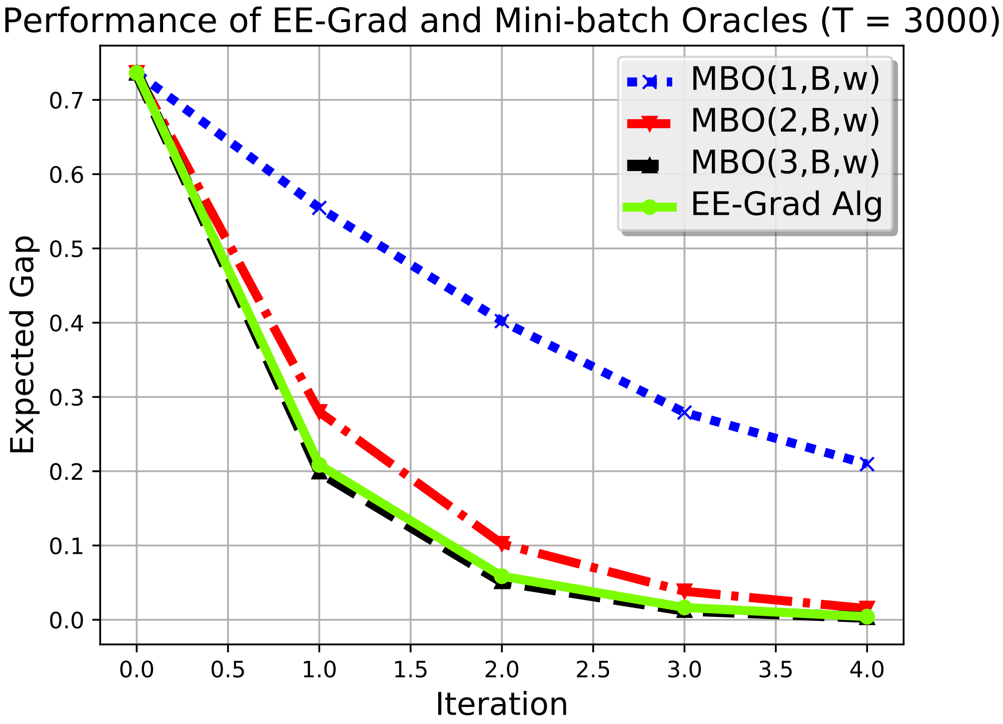


(b) $T = 100$.

Figure 4.1: Expected gaps achieved by the EE-Grad algorithm and the mini-batch oracles for different values of $T = 50, 100$ over $K = 5$ iterations.



(a) $T = 200$.



(b) $T = 3000$.

Figure 4.2: Expected gaps achieved by the EE-Grad algorithm and the mini-batch oracles for different values of $T = 200, 3000$ over $K = 5$ iterations.

tion is also assumed to be costly (i.e., aggregation cost). In this framework, the optimal mini-batch size in minimizing the noise variance depends on the underlying cost-fidelity function, which is assumed to be unknown.

We proposed the EE-Grad algorithm that performs sequential trials over different mini-batch oracles to explore the performance of each mini-batch oracle with high precision and exploit the current knowledge to allocate the budget to the one that seems to provide the best performance. We demonstrated that the proposed algorithm performs almost as well as the optimal mini-batch oracle on each iteration in expectation. We next applied this result to the strongly convex objectives with Lipschitz continuous gradients, and provided a performance guarantee on the rate of convergence with respect to the optimal mini-batch oracle. We finally illustrated our theoretical results through numerical experiments on synthetic data.

CHAPTER 5

EXTENSIONS AND OPEN PROBLEMS

In this chapter we first present some extensions to the problems considered in the earlier chapters of this dissertation. We next propose new problems that may be explored in future work.

In Chapter 3, we investigated cost-performance tradeoffs in fusing unreliable computational units subject to cost and fidelity constraints, where we assume that outputs produced by different computational units are uncorrelated. However, this assumption may not hold in certain applications such as crowdsourcing [99]. This setting can be extended to a more general case where outputs produced by different unreliable computational units are allowed to be correlated, where the correlation structure may depend on the underlying application.

We next note that when studying the fundamental tradeoff between cost and performance, we assumed that the fusion operation is error-free. This assumption can be relaxed to a case where the fusion operation may also produce noisy results under cost and fidelity constraints. In this case, the fusion operation may also be subject to a cost-fidelity tradeoff.

Moreover, we focused on a particular fusion operation, i.e., linear combination, which is common in certain applications. More generally, we can consider nonlinear fusion rules to compute the final estimate of the error-free computation. For instance, midrange [83] and median-of-means [88] estimators have been considered as alternatives to linear estimators under different scenarios to improve performance. Extension of the centralized fusion setting of Chapter 3 and Chapter 4 to decentralized settings under different network topologies, as in [89], is another potential extension.

We next present a new problem formulation where the error-free computation and outputs of the unreliable computational units are binary-valued, and where it is later extended to M-ary alphabet case.

5.1 A Cost-Fidelity Framework under a Binary Alphabet

Suppose that an error-free computation is given by

$$Y = f(\mathbf{X}) \in \{0, 1\},$$

where $\mathbf{X} \in \mathbb{R}^d$ is an input signal and $f(\cdot)$ is a desired function. An unreliable computational unit outputs Z_c incurring a cost $c \geq c_{\min} \geq 0$, where $c_{\min}$ is the minimum cost,

$$\Pr(Z_c = Y) = p(c),$$

and $p(c)$ is assumed to be a strictly increasing, differentiable, and concave function with

$$\lim_{c \to 0} p(c) = 1/2,$$
$$\lim_{c \to \infty} p(c) = 1.$$

Note that this function controls the tradeoff between cost and accuracy. As an example, we can consider a class of exponential functions as

$$p(c) = 1 - \exp(-\alpha c)/2$$

for $\alpha > 0$. Under this setting, suppose that we are given a limited budget $B > 0$, and consider a class of repetition-based strategies that fuse outputs of several unreliable computational units to recover the error-free computation.

For a positive integer n, a repetition-based strategy S_n with a cost budget B distributes the budget across n independent computational units incurring costs

$$\mathbf{c} = (c_1, \ldots, c_n),$$

where $c_i \geq c_{\min}$ for each $i = 1, \ldots, n$, and fuses their outputs to estimate the error-free computation Y. We assume that fusing outputs of n computational units incurs an additional cost $D(n)$, which is a strictly increasing function with $D(1) = 0$. Thus the cost vector $\mathbf{c}$ must satisfy

$$B = \sum_{i=1}^{n} c_i + D(n).$$

The next lemma applies Neyman and Pearson's result [100] on the optimal fusion of n independent Bernoulli random variables with fixed parameters in terms of minimizing the probability of error to the output of the strategy S_n, and characterizes the optimal strategy S_n with budget B, given $Z_{c_1}, \ldots, Z_{c_n}$ incurring costs $c_1, \ldots, c_n$, respectively.

Lemma 5.1.1. *Given independent estimates $Z_{c_1}, \ldots, Z_{c_n}$ of the error-free computation Y with accuracies $p(c_1), \ldots, p(c_n)$, respectively, the probability of error in estimating Y is minimized by*

$$\hat{Y}_n(\mathbf{c}; B) = \begin{cases} 1 & \text{if} \sum_{i=1}^{N} w(c_i) Z_{c_i} \geq \frac{1}{2} \sum_{j=1}^{N} w(c_j) \\ 0 & \text{otherwise,} \end{cases} \tag{5.1}$$

where

$$w(c_i) = \log\left(\frac{p(c_i)}{1 - p(c_i)}\right)$$

for $i = 1, \ldots, n$.

Although this weighted majority voting scheme is optimal in minimizing the probability of error, there are no analytical expressions for this minimum probability of error. However, Berend and Kontorovich [101] prove an upper bound on the probability of error achieved by the weighted majority voting scheme using independent variables with fixed accuracies. The next lemma applies this result to the strategy S_n with budget B based on the fusion rule given in (5.1).

Lemma 5.1.2. *Given a budget $B > 0$, the output $\hat{Y}_n(\mathbf{c}; B)$ of the strategy S_n with a cost vector $\mathbf{c}$, where $c_i \geq c_{\min}$ for $i = 1, \ldots, n$ and*

$$\sum_{i=1}^{n} c_i = B - D(n),$$

satisfies

$$\Pr\Big(\hat{Y}_n(\mathbf{c}; B) \neq Y\Big) \leq \exp(-\psi(\mathbf{c})/2), \tag{5.2}$$

where

$$\psi(\mathbf{c}) = \sum_{i=1}^{n} \phi(c_i)$$

and

$$\phi(c_i) = (p(c_i) - 1/2)w(c_i)$$

for each $i = 1, \ldots, n$.

One potential approach to analyze and optimize the performance of the repetition-based strategies under a cost budget is to minimize the upper bound (5.2) on the probability of error achieved by the output $\hat{Y}_n(\mathbf{c}; B)$ of the strategy S_n with a cost vector $\mathbf{c}$. We can formulate this approach as a constrained optimization problem as follows:

$$\begin{aligned} \underset{\mathbf{c}\in\mathbb{R}^n}{\text{maximize}} \quad & \psi(\mathbf{c}) \\ \text{subject to} \quad & c_i \geq c_{\min},\ i = 1, \ldots, n, \\ & \sum_{i=1}^{n} c_i = B - D(n). \end{aligned} \tag{5.3}$$

We can further define

$$g_i \triangleq \frac{c_i - c_{\min}}{u(n)} \geq 0$$

for $i = 1, \ldots, n$, where

$$u(n) \triangleq B - nc_{\min} - D(n) > 0,$$

and $\mathbf{g} = (g_1, \ldots, g_n)$, so that $g_i \geq 0$ for $i = 1, \ldots, n$, and

$$\sum_{i=1}^{n} g_i = 1.$$

We also define

$$S_n(\mathbf{g}) \triangleq \psi(u(n)\mathbf{g} + \mathbf{a}),$$

where $\mathbf{a} = (c_{\min}, \ldots, c_{\min})$. Hence the optimization problem in (5.3) is equivalent to

$$\underset{\mathbf{g}\in\Delta_n}{\text{maximize}} \quad S_n(\mathbf{g}) \tag{5.4}$$

where

$$\Delta_n \triangleq \left\{ \mathbf{g} \in \mathbb{R}^n \middle| g_i \geq 0,\ i = 1, \ldots, n,\ \text{and } \sum_{i=1}^{n} g_i = 1 \right\}$$

is $n-1$-dimensional standard simplex, which is a convex set.

5.2 M-ary Alphabet Version

Here we describe an M-ary alphabet version of the problem presented in Section 5.1. Suppose that the error-free computation $Y = f(\mathbf{X})$ and outputs of unreliable computational units are limited to a finite set

$$\mathcal{Y} \triangleq \{0, 1, \ldots, M-1\},$$

where $M > 2$ is an integer. In this case, an output of an unreliable computational unit incurring cost $c \geq c_{\text{min}}$ can be written as

$$\Pr(Z_c = i \mid Y = j) = \begin{cases} q(c) & \text{if } i = j \\ \dfrac{1 - q(c)}{M - 1} & \text{otherwise} \end{cases} \tag{5.5}$$

for any $i, j \in \mathcal{Y}$. In general, we can consider a function

$$q : [c_{\text{min}}, \infty) \to \left[\frac{1}{M}, 1\right)$$

that are strictly increasing in c and satisfy

$$q(c_{\text{min}}) = \frac{1}{M},$$
$$\lim_{c \to \infty} q(c) = 1.$$

Note that this function controls how increasing cost translates into improved (decreased) probability of error. We can write the output Z_c of an unreliable computational unit equivalently as

$$Z_c = Y + U_c \mod M,$$

where, for any $i \in \mathcal{Y}$,

$$\Pr(U_c = i) = \begin{cases} q(c) & \text{if } i = 0 \\ \dfrac{1 - q(c)}{M - 1} & \text{if } i = 1, \ldots, M - 1. \end{cases} \tag{5.6}$$

We remark that the uniform noise given in (5.6) can be seen as the worst-case model in terms of information [99]. We note that Zhang and Shanbhag [102] studied probabilistic error models for machine learning kernels implemented on low signal-to-noise (SNR) circuit fabrics where errors arise due to voltage overscaling, process variations, or defects. In particular, they investigate error models that are additive over algebraic fields to predict the performance of machine learning kernels under hardware errors.

We first present repetition-based strategies that distribute a limited cost budget across several unreliable computational units and aggregates their outputs using a fusion rule, which is an extension of the weighted majority voting scheme given in (5.1) to the M-ary case, to make a final decision on the error-free computation Y. We next provide another fusion scheme that, instead of performing a majority voting across unreliable outputs, performs coding across several unreliable outputs to introduce additional error-correction capability. Design and use of such schemes have been considered in distributed decision-making for wireless sensor networks [103, 104] and machine learning [105].

5.2.1 Repetition-Based Strategies

Suppose that there are n independent unreliable unreliable computational outcomes $Z_{c_1}, \ldots, Z_{c_n}$ incurring costs $c_1, \ldots, c_n$, respectively, where

$$B = \sum_{i=1}^{n} c_i + D(n) > 0$$

is a budget, and $D(n)$ is a fusion cost. The weighted majority voting is performed as follows. We first represent each class in $\mathcal{Y}$ with a bit vector of length $m \triangleq \log_2 M$, where we assume that m is an integer, and divides n. We denote the binary representation of each unreliable computational outcome Z_{c_i} with a binary vector $\mathbf{b}_i = [b_{1,i}, \ldots, b_{m,i}] \in \{0, 1\}^m$, for each $i = 1, \ldots, n$.

The fusion rule uses the weighted majority vote of the ith bit of $\mathbf{b}_1, \ldots, \mathbf{b}_n$ for each $i = 1, \ldots, m$ to decide on each bit location separately, and then concatenates the results to yield the final result $\hat{Y}_n(\mathbf{c}; B) \in \mathcal{Y}$. More precisely, if we denote the binary representation of $\hat{Y}_n(\mathbf{c}; B)$ with

$$\hat{\mathbf{y}} = [\hat{y}_1, \ldots, \hat{y}_m],$$

then we have, for each $i = 1, \ldots, m$,

$$\hat{y}_i = \begin{cases} 1 & \text{if} \sum_{=1}^{n} w_j b_{i,j} \geq \frac{1}{2} \sum_{k=1}^{n} w_k \\ 0 & \text{otherwise} \end{cases}, \tag{5.7}$$

where

$$w_i \triangleq \log\left(\frac{s_i}{1 - s_i}\right), \quad i = 1, \ldots, n, \tag{5.8}$$

and

$$s_i \triangleq \frac{M(1 - q(c_i))}{2(M - 1)} \tag{5.9}$$

is the probability of bit error in any given location for the unreliable computational unit incurring cost c_i, for each $i = 1, \ldots, n$.

The next lemma presents an upper bound on the probability of error attained by the weighted majority voting scheme given in (5.7) in terms of estimating Y. Note that the final output is correct if and only if the decision on each bit location is correct. This result is a straightforward application of the upper bound proven in Lemma 5.1.2 for the binary case.

Lemma 5.2.1. *The probability of error of a majority-based fusion scheme in* (5.7) *satisfies*

$$\Pr\left(\hat{Y}_n(\mathbf{c}; B) \neq Y\right) \leq 1 - \left(1 - \exp\left(-\frac{1}{2}\Phi\right)\right)^m,$$

where

$$\Phi \triangleq \sum_{i=1}^{N} \left(s_i - \frac{1}{2}\right) w_i,$$

and w_i and s_i are defined in (5.8) *and* (5.9)*, respectively.*

We next present a coding-based fusion scheme that aggregates outputs of several unreliable computational units by incorporating additional error-correction capability.

5.2.2 Coding-Based Strategies

Suppose that we have n independent unreliable computational units incurring costs $c_1, \ldots, c_n$, respectively. We represent the fusion scheme with an $M \times n$ binary code matrix $\mathbf{A}$. The rows of $\mathbf{A}$ are denoted by $\mathbf{r}_0, \ldots, \mathbf{r}_{M-1}$, where $\mathbf{r}_i \in \{0,1\}^n$ is a codeword assigned to the hypothesis $i \in \mathcal{Y}$. The columns of $\mathbf{A}$ are denoted by $\mathbf{c}_1, \ldots, \mathbf{c}_N$, where $\mathbf{c}_j \in \{0,1\}^M$ represents the decision rule corresponding to the jth unreliable computational unit for each $j = 1, \ldots, n$. Since the matrix $\mathbf{A}$ is binary, each column is designed to discriminate between only two classes.

We make a local binary decision, $v_i \in \{0,1\}$, for each noisy computational outcome Z_{c_i} based on the ith column of the code matrix $\mathbf{A}$ for each $i = 1, \ldots, n$. Then the fusion rule receives the n-bit vector

$$\mathbf{v} = [v_1, \ldots, v_n],$$

a single bit from each unreliable computational outcome, and employs the minimum Hamming distance criterion to give its final computation:

$$\hat{Y}_n^{\mathrm{c}}(\mathbf{c}; B) = \arg \min_{0 \le j \le M-1} d(\mathbf{v}, \mathbf{r}_j),$$

where, for any $\mathbf{a}, \mathbf{b} \in \{0,1\}^n$,

$$d(\mathbf{a}, \mathbf{b}) \triangleq \sum_{i=1}^{n} |a_i - b_i|$$

is the Hamming distance between $\mathbf{a}$ and $\mathbf{b}$. Note that in general the quality of an error-correcting code can be measured by the minimum Hamming distance between any pair of code words, which in this case is given by

$$d_{\min} \triangleq \min_{i \neq j} d(\mathbf{r}_i, \mathbf{r}_j).$$

For any $i \in \mathcal{Y}$, the decision region D_i of the codeword $\mathbf{r}_i$ is defined as

$$D_i \triangleq \{\mathbf{s} \in \{0,1\}^n \mid d(\mathbf{s}, \mathbf{r}_i) \leq d(\mathbf{s}, \mathbf{r}_j) \text{ for any } j \in \mathcal{Y}\}.$$

We note that the coding matrix based fusion operation introduces a cost overhead. In particular, we can define the fusion cost as $F(M, n)$, where

$$F : \mathbb{Z} \times \mathbb{Z} \to \mathbb{R}_+.$$

Intuitively, the fusion cost in our framework plays a similar role to the decoder complexity in communications theory, which increases significantly when trying to achieve arbitrarily low bit error probabilities using only finite transmit power [106].

By constraining the local decisions through the code matrix $\mathbf{A}$, binary local decisions are sufficient for an M-ary hypothesis testing problem without losing information regarding the hypotheses [103]. We remark that the code matrix $\mathbf{A}$ is employed for both local decision rules and the final fusion rule so that its design plays a crucial rule in the overall performance.

Note that Vempaty, *et al.* [99] considered a multi-class labeling problem in crowdsourcing framework where the workers are unreliable. They employ an error-correcting code based approach to improve the final decision performance. The main distinction is that in their framework the workers are anonymous, and their misclassification probabilities are assumed to be randomly drawn from a probability distribution. In our framework, however, we assume that the probability of error achieved by each unreliable computational unit is determined and controlled by the cost incurred to obtain it in a deterministic fashion.

CHAPTER 6

CONCLUSIONS

In this dissertation we focused on decision-making systems and algorithms under uncertain environments from an abstract point of view. We focused on robustness and reliability, which are significant concerns in a wide range of applications including machine learning and optimization, circuits and systems, neuroscience, crowdsourcing, communications, investment, and wireless sensor networks. One particular motivation behind this thesis work is the implementation of information processing tasks on modern circuit fabrics such as nanoscale CMOS or spintronics, which are stochastic in nature. We proposed and investigated two generic frameworks to study the robustness and reliability issues that critically impact the design and analysis of systems and algorithms based on unreliable components, respectively. We first presented the framework of online optimization, and used this framework to study robustness of online decision-making systems and algorithms under worst-case adversarial perturbations. We next proposed a cost-fidelity framework to study the performance of repetition-based approaches in decision-making systems and algorithms based on unreliable components. We finally investigated a partial information version of the cost-fidelity framework, where the cost-fidelity function is unknown, and applied our results to the problem of stochastic gradient descent.

We first formulated a game-theoretic framework of online optimization under adversarial perturbations to study robustness of decision-making systems and algorithms. This framework includes a large class of machine learning problems as special cases. More specifically, we introduced and investigated an adversarial worst-case perturbation framework for online optimization, where an online player's strategy is subject to perturbations by an adversary. We cast this problem as a new repeated game, where a randomized player is pitted against two opponents, namely, Nature and a strategy-perturbing adversary. We introduced a robust randomized algorithm and presented an upper bound on its worst-case expected regret under our worst-case model. In particular, we proved that this algorithm is Hannan-consistent even under adversarial perturbations, when certain regularity conditions are satisfied. We presented some numerical experiments to illustrate our theoretical results.

We turned our attention to the reliability issue, and considered the problem of fusing several unreliable computational units performing the same task in parallel subject to cost and fidelity constraints. We proposed a new framework, where any unreliable computational unit has a certain level of fidelity and an associated cost that is a function of that fidelity. In particular, we formalized the relation between cost and fidelity of an unreliable computational unit using different classes of cost functions, and investigated the limits of achievable performance by using repetition-based strategies. We showed that a single and more reliable computational unit incurs less than a fusion of several less costly and less reliable computational units while achieving the same performance under concave and linear costs. When the cost function is convex, we demonstrated that fusing several cheaper but unreliable computational units may yield a better cost-performance tradeoff than an expensive and reliable unit under certain conditions.

We next proposed and investigated an application of our cost-fidelity framework to a stochastic gradient descent problem, where the underlying cost-fidelity function is assumed to be unknown. We considered a class of mini-batch oracles, which distributes a limited budget across a number of stochastic gradients and aggregates them to produce a final stochastic gradient, which is used to estimate the true gradient. Since the optimal mini-batch oracle depends on the unknown cost-fidelity function, we have propose an algorithm that explores the performance of mini-batch oracles and exploits the current knowledge to estimate the best mini-batch oracle in an online manner. We demonstrated performance guarantees for this algorithm with respect to the optimal mini-batch oracle, and illustrated our results for strongly convex objectives with Lipschitz continuous gradients.

We finally provided some extensions of the problems considered in this thesis, as well as some open problems for future research.

APPENDIX A

PROOFS OF CHAPTER 2

A.1 Proof of Lemma 2.4.1

At any time $t = 1, \dots, T$, we have

$$\begin{aligned}
\eta\, \mathbb{E}[\ell_t(X_t)] &= \eta \int_{\mathcal{X}} d\mathrm{M}_t(x) \ell_t(x) \\
&= \int_{\mathcal{X}} d\mathrm{M}_t(x) \ln(\exp(\eta \ell_t(x))) \\
&= \int_{\mathcal{X}} d\mathrm{M}_t(x) \ln\biggl(\frac{\mu_t(x)}{Z_t w_{t+1}(x)}\biggr) \\
&= \int_{\mathcal{X}} d\mathrm{M}_t(x) \ln\biggl(\frac{d\mathrm{M}_t(x)}{dW_{t+1}(x)}\biggr) - \ln(Z_t).
\end{aligned}$$

Using the similar lines as in the proof of Lemma 2.2.1, we get

$$\begin{aligned}
&\int_{\mathcal{X}} d\mathrm{M}_t(x) \ln\biggl(\frac{d\mathrm{M}_t(x)}{dW_{t+1}(x)}\biggr) = \eta\, \mathbb{E}[\ell_t(X_t)] + \ln(Z_t) \\
&= \ln\bigl(\mathbb{E}\bigl[e^{-\eta(\ell_t(X_t) - \mathbb{E}[\ell_t(X_t)])}\bigr]\bigr) \le \frac{(\eta C \operatorname{diam}(\mathcal{X}))^2}{8}.
\end{aligned}$$

Hence, we conclude that

$$\mathbb{E}[\ell_t(X_t)] \le -\frac{1}{\eta} \ln(Z_t) + \frac{\eta (C \operatorname{diam}(\mathcal{X}))^2}{8}.$$

A.2 Proof of Corollary 2.4.1

To prove (2.44), we first note that $\ln(1+x) \leq 1 + \ln(x)$ holds for any $x \geq 1/(e-1)$. Then, we can write

$$\begin{aligned}
&\ln\left(\Gamma_{\mathrm{u}} + \exp(\eta C \operatorname{diam}(\mathcal{X}))\frac{1-\Gamma_{\mathrm{l}}}{\Gamma_{\mathrm{l}}}\right) \\
&\leq 1 + \ln\left(\exp(\eta C \operatorname{diam}(\mathcal{X}))\frac{1-\Gamma_{\mathrm{l}}}{\Gamma_{\mathrm{l}}}\right) \qquad \text{(A.1)} \\
&= 1 + \ln\left(\frac{1-\Gamma_{\mathrm{l}}}{\Gamma_{\mathrm{l}}}\right) + \eta C \operatorname{diam}(\mathcal{X}),
\end{aligned}$$

where (A.1) holds since $\exp(\eta C \operatorname{diam}(\mathcal{X})) \geq 1$ and $\Gamma_{\mathrm{l}} \leq 1 - \exp(-1)$, yielding

$$\exp(\eta C \operatorname{diam}(\mathcal{X}))\frac{1-\Gamma_{\mathrm{l}}}{\Gamma_{\mathrm{l}}} \geq \frac{1}{e-1}.$$

Hence we get

$$F_\eta(\Gamma_{\mathrm{l}}, \Gamma_{\mathrm{u}}) \leq 2C \operatorname{diam}(\mathcal{X}) + \frac{1}{\eta}\ln\left(\frac{e(1-\Gamma_{\mathrm{l}})}{\Gamma_{\mathrm{l}}(1-\Gamma_{\mathrm{u}})}\right).$$

Therefore, by Theorem 2.4.1, we get

$$\begin{aligned}
&\mathbb{E}\left[R_T^{(\mathrm{o,p})}\left(X_{\mathrm{r}}^T; \mathcal{L}, \mathcal{P}, \mathcal{A}_k^{\mathrm{w}}\right)\right] \\
&\leq \frac{1}{\eta}\left(r + k\ln\left(\frac{e(1-\Gamma_{\mathrm{l}})}{\Gamma_{\mathrm{l}}(1-\Gamma_{\mathrm{u}})}\right)\right) + \frac{T\eta(C \operatorname{diam}(\mathcal{X}))^2}{8} \\
&+ 2kC \operatorname{diam}(\mathcal{X}).
\end{aligned}$$

If we set $\eta = \eta_{\mathrm{o}}$, then we get

$$\begin{aligned}
&\mathbb{E}\left[R_T^{(\mathrm{o,p})}\left(X_{\mathrm{r}}^T; \mathcal{L}, \mathcal{P}, \mathcal{A}_k^{\mathrm{w}}\right)\right] \\
&\leq \sqrt{T}(\mathrm{A}_1(r) + k\mathrm{A}_2(r, \Gamma_{\mathrm{l}}, \Gamma_{\mathrm{u}})) + 2kC \operatorname{diam}(\mathcal{X}).
\end{aligned}$$

Moreover, if $k = o\left(\sqrt{T}\right)$, then it follows that

$$\mathbb{E}\left[R_T^{(\mathrm{o,p})}\left(X_{\mathrm{r}}^T; \mathcal{L}, \mathcal{P}, \mathcal{A}_k^{\mathrm{w}}\right)\right] = o(T). \qquad \text{(A.2)}$$

In general, when the learning rate satisfies $\eta = O\left(1/\sqrt{T}\right)$ and $k = o\left(\sqrt{T}\right)$, the result (A.2) holds.

APPENDIX B

PROOFS OF CHAPTER 3

B.1 Proof of Lemma 3.4.1

We solve this optimization problem using the method of Lagrange multipliers, where we first form the Lagrangian

$$J(\theta_1, \ldots, \theta_N, \lambda) \triangleq \sum_{i=1}^{N} C(\theta_i) + \lambda\left(\tau^{-1} - \sum_{i=1}^{N} \theta_i\right).$$

Then, we set the derivative of the Lagrangian with respect to θ_j to 0, which is given by

$$\frac{\partial J}{\partial \theta_j}(\theta_1, \ldots, \theta_N, \lambda) = C'(\theta_j) - \lambda = 0,$$

for each $j = 1, \ldots, N$. Hence the necessary conditions for optimality are given by $\lambda = C'(\theta_j)$ for $j = 1, \ldots, N$.

Here, we note that the cost function $C(\theta)$ is convex and strictly increasing in θ, and its derivative $C'(\theta)$ is nondecreasing. This implies that it is invertible, so we can write

$$\theta_j = (C')^{-1}(\lambda),$$

for each $j = 1, \ldots, N$, where $(C')^{-1}$ is the inverse of the function C'. That is, $\theta_1 = \cdots = \theta_N$. Moreover, by imposing the MSE constraint, we get $\theta_j = (\tau N)^{-1}$ for any $j = 1, \ldots, N$, which yields the desired result.

B.2 Proof of Lemma 3.4.2

We first differentiate the total cost function as

$$\frac{\partial \text{Cost}_\tau(a)}{\partial a} = G\left(\frac{1}{\tau a}\right) - \frac{1}{\tau a} G'\left(\frac{1}{\tau a}\right) + c_{\min} + D'(a).$$

We next find its second derivative as

$$\begin{aligned}
&\frac{\partial^2 \text{Cost}_\tau(a)}{\partial a^2} \\
&= -\frac{1}{\tau a^2} G'\left(\frac{1}{\tau a}\right) + \frac{1}{\tau a^2} G'\left(\frac{1}{\tau a}\right) + \frac{1}{\tau^2 a^3} G''\left(\frac{1}{\tau a}\right) + D''(a) \\
&= \frac{1}{\tau^2 a^3} G''\left(\frac{1}{\tau a}\right) + D''(a),
\end{aligned}$$

which is non-negative since the incremental cost function $G(\cdot)$ and the fusion cost function $D(\cdot)$ are both convex and $a > 0$.

B.3 Proof of Lemma 3.4.4

Suppose that $\lambda \in [0, 1]$. Since f is concave, we have

$$\begin{aligned}
f(\lambda x) &= f(\lambda x + (1 - \lambda)0) \\
&\geq \lambda f(x) + (1 - \lambda) f(0) \geq \lambda f(x).
\end{aligned}$$

Then, for any $x, y > 0$, we can write

$$\begin{aligned}
f(x) + f(y) &= f\left((x + y)\frac{x}{x + y}\right) + f\left((x + y)\frac{y}{x + y}\right). \\
&\geq \frac{x}{x + y} f(x + y) + \frac{y}{x + y} f(x + y) \\
&= f(x + y),
\end{aligned}$$

where we use $x/(x + y), y/(x + y) \in [0, 1]$.

APPENDIX C

PROOFS OF CHAPTER 4

C.1 Trace of the Sample Covariance Matrix as a Quadratic Form

Lemma C.1.1. *On each round t, the trace of the sample covariance matrix $\mathbf{Cov}_t(n)$ can be written as*

$$V_t(n) = \mathbf{s}_{t,n}^T \mathbf{A}_{t,n} \mathbf{s}_{t,n}, \quad n = 1, \ldots, N,$$

where

$$\mathbf{s}_{t,n} = \left(\mathbf{G}_1(\mathbf{w}, n)^T, \ldots, \mathbf{G}_{\gamma_t(n)}(\mathbf{w}, n)^T\right)^T,$$
$$\mathbf{A}_{t,n} = (\gamma_t(n) - 1)^{-1}\big(\mathbf{I} - \gamma_t(n)^{-1}\mathbf{E}\big),$$

$\mathbf{I} \in \mathbb{R}^{d\gamma_t(n) \times d\gamma_t(n)}$ is an identity matrix, and $\mathbf{E} \in \mathbb{R}^{d\gamma_t(n) \times d\gamma_t(n)}$ is a block matrix with $d \times d$ identity blocks.

Proof. Note that

$$\begin{aligned}
V_t(n) &= \mathrm{Tr}(\mathbf{Cov}_t(n)) \\
&= \frac{1}{\gamma_t(n) - 1} \sum_{i=1}^{\gamma_t(n)} (\mathbf{G}_i(\mathbf{w}, n) - \mathbf{m}_t(n))^T (\mathbf{G}_i(\mathbf{w}, n) - \mathbf{m}_t(n)) \\
&= \frac{1}{\gamma_t(n) - 1} \left(\sum_{i=1}^{\gamma_t(n)} \mathbf{G}_i(\mathbf{w}, n)^T \mathbf{G}_i(\mathbf{w}, n) - \gamma_t(n) \mathbf{m}_t(n)^T \mathbf{m}_t(n) \right),
\end{aligned}$$

where

$$\mathbf{m}_t(n)^T \mathbf{m}_t(n) = \frac{1}{\gamma_t(n)^2} \left(\sum_{i=1}^{\gamma_t(n)} \mathbf{G}_i(\mathbf{w}, n)^T \right) \left(\sum_{j=1}^{\gamma_t(n)} \mathbf{G}_j(\mathbf{w}, n) \right)$$

$$
\begin{aligned}
&= \frac{1}{\gamma_t(n)^2} \sum_{i=1}^{\gamma_t(n)} \sum_{j=1}^{\gamma_t(n)} \mathbf{G}_i(\mathbf{w}, n)^T \mathbf{G}_j(\mathbf{w}, n) \\
&= \frac{1}{\gamma_t(n)^2} \mathbf{s}_{t,n}^T \mathbf{E} \mathbf{s}_{t,n}.
\end{aligned}
$$

Noting

$$
\sum_{i=1}^{\gamma_t(n)} \mathbf{G}_i(\mathbf{w}, n)^T \mathbf{G}_i(\mathbf{w}, n) = \mathbf{s}_{t,n}^T \mathbf{s}_{t,n},
$$

we get

$$
\begin{aligned}
V_t(n) &= \frac{1}{\gamma_t(n) - 1} \left(\mathbf{s}_{t,n}^T \mathbf{s}_{t,n} - \frac{1}{\gamma_t(n)} \mathbf{s}_{t,n}^T \mathbf{E} \mathbf{s}_{t,n} \right) \\
&= \mathbf{s}_{t,n}^T \mathbf{A}_{t,n} \mathbf{s}_{t,n}.
\end{aligned}
$$

This concludes the proof. □

C.2 Hanson-Wright Inequality

Lemma C.2.1. *Suppose that for $m > 1$,*

$$
\mathbf{W} = [W_1, \ldots, W_m]^T \in \mathbb{R}^m,
$$

where W_i are zero-mean sub-Gaussian with a parameter $\sigma^2 > 0$. Then, given an arbitrary matrix $\mathbf{A} \in \mathbb{R}^{m \times m}$, we have, for any $\varepsilon > 0$,

$$
\Pr\left(\mathbf{W}^T \mathbf{A} \mathbf{W} - \mathbb{E}_{\mathbf{w}}\left[\mathbf{W}^T \mathbf{A} \mathbf{W}\right] > \varepsilon\right) \le \exp\left(-c \min\left(\frac{\varepsilon^2}{\sigma^4 \|\mathbf{A}\|_{\mathrm{F}}^2}, \frac{\varepsilon}{\sigma^2 \|\mathbf{A}\|}\right)\right),
$$

where $\|\mathbf{A}\|_{\mathrm{F}}$ and $\|\mathbf{A}\|$ are Frobenius and operator norms of $\mathbf{A}$, and $c > 0$ is an absolute constant.

C.3 Concentration Result on the Trace of the Sample Covariance Matrices

Lemma C.3.1. *Suppose that $\gamma_t(n) > 1$. Then the tail probability of $V_t(n)$ satisfies, for any $\varepsilon > 0$,*

$$\Pr\bigl(V_t(n) - \sigma_n^2 S(\mathbf{w}) > \varepsilon\bigr) \le \exp(-(\gamma_t(n) - 1)\phi(\varepsilon)),$$

where

$$\phi(\varepsilon) \triangleq \frac{c\varepsilon}{\beta P} \min\biggl(1, \frac{\varepsilon/d}{\beta P}\biggr),$$

for $n = 1, \ldots, N$, where $c > 0$ is an absolute constant.

Proof. Note that

$$\mathbf{I} - \frac{1}{\gamma_t(n)}\mathbf{E}$$

is a $d\gamma_t(n) \times d\gamma_t(n)$ block matrix with $d \times d$ blocks, where the diagonal and non-diagonal matrices are given by

$$\frac{(\gamma_t(n) - 1)}{\gamma_t(n)}\mathbf{I},$$

and

$$-\frac{1}{\gamma_t(n)}\mathbf{I},$$

respectively, and

$$\|\mathbf{I}\|_{\mathrm{F}}^2 = d.$$

This implies that

$$\begin{aligned}\|\mathbf{A}_{t,n}\|_{\mathrm{F}}^2 &= \frac{1}{(\gamma_t(n) - 1)^2}\Biggl(\gamma_t(n)\biggl(\frac{\gamma_t(n) - 1}{\gamma_t(n)}\biggr)^2 \|\mathbf{I}\|_{\mathrm{F}}^2 + (\gamma_t(n) - 1)\gamma_t(n)\frac{1}{\gamma_t(n)^2}\|\mathbf{I}\|_{\mathrm{F}}^2\Biggr) \\ &= \frac{d}{\gamma_t(n) - 1}.\end{aligned}$$

Next suppose that

$$\mathbf{v} = \bigl(\mathbf{v}_1^T, \ldots, \mathbf{v}_{\gamma_t(n)}^T\bigr)^T \in \mathbb{R}^{d\gamma_t(n)}$$

such that $\mathbf{v}_i \in \mathbb{R}^d$ and $\|\mathbf{v}\|_2 = 1$. Then we write

$$\begin{aligned}
\|\mathbf{A}_{t,n}\mathbf{v}\|_2^2 &= \frac{1}{(\gamma_t(n)-1)^2}\left(\|\mathbf{v}\|_2^2 + \frac{1}{\gamma_t(n)^2}\|\mathbf{E}\mathbf{v}\|_2^2 - \frac{2}{\gamma_t(n)}\mathbf{v}^T\mathbf{E}\mathbf{v}\right) \\
&= \frac{1}{(\gamma_t(n)-1)^2}\left(1 - \frac{1}{\gamma_t(n)}\left\|\sum_{i=1}^{\gamma_t(n)}\mathbf{v}_i\right\|_2^2\right) \\
&\leq \frac{1}{(\gamma_t(n)-1)^2},
\end{aligned}$$

where equality is achieved by $\mathbf{v} = \left(\mathbf{v}_1^T, \ldots, \mathbf{v}_{\gamma_t(n)}^T\right)^T$ such that

$$\begin{aligned}
\mathbf{v}_1 &= \left(\frac{1}{\sqrt{2}}, 0, \ldots, 0\right), \\
\mathbf{v}_2 &= -\mathbf{v}_1,
\end{aligned}$$

and $\mathbf{v}_i = (0, \ldots, 0)$ for $i = 3, \ldots, \gamma_t(n)$. This yields

$$\begin{aligned}
\|\mathbf{A}_{t,n}\| &= \sup_{\|\mathbf{v}\|_2=1} \|\mathbf{A}_{t,n}\mathbf{v}\|_2 \\
&= (\gamma_t(n)-1)^{-1}.
\end{aligned}$$

We finally note that the trace of the sample covariance matrix can be written as

$$\begin{aligned}
V_t(n) &= \frac{1}{\gamma_t(n)-1}\sum_{i=1}^{\gamma_t(n)}(\mathbf{G}_i(\mathbf{w},n) - \mathbf{m}_t(n))^T(\mathbf{G}_i(\mathbf{w},n) - \mathbf{m}_t(n)) \\
&= \frac{1}{\gamma_t(n)-1}\sum_{i=1}^{\gamma_t(n)}(\mathbf{Q}_i(n) - \mathbf{q}_t(n))^T(\mathbf{Q}_i(n) - \mathbf{q}_t(n)),
\end{aligned}$$

where

$$\mathbf{Q}_i(n) \triangleq \mathbf{G}_i(\mathbf{w},n) - \nabla F(\mathbf{w})$$

for $i \in [\gamma_t(n)]$, and

$$\mathbf{q}_t(n) = (1/\gamma_t(n))\sum_{i=1}^{\gamma_t(n)}\mathbf{Q}_i(n).$$

This implies the same expression holds for the mean-removed versions of

$\mathbf{G}_i(\mathbf{w}, n)$s. Hence, we can assume that

$$\mathbb{E}_{\mathbf{w}}[\mathbf{G}_i(\mathbf{w}, n)] = 0.$$

We apply Lemma C.2.1 to $V_t(n)$ by using Lemma C.1.1 to get, for any $\varepsilon > 0$,

$$\Pr\bigl(V_t(n) - \sigma_n^2 S(\mathbf{w}) > \varepsilon\bigr) \leq \exp(-(\gamma_t(n) - 1)\phi_n(\varepsilon)),$$

where

$$\phi_n(\varepsilon) \triangleq \frac{c\varepsilon}{\sigma_n^2 S(\mathbf{w})} \min\left(1, \frac{\varepsilon/d}{\sigma_n^2 S(\mathbf{w})}\right),$$

which is strictly increasing in ε, for $n \in [N]$, where $c > 0$ is an absolute constant. Finally, we note

$$\phi_n(\varepsilon) \geq \phi(\varepsilon),$$

since we assumed

$$\max_{n=1,\ldots,N} \sigma_n^2 \leq \beta,$$

and

$$S(\mathbf{w}) \leq P.$$

This concluses the proof. □

C.4 Pseudo-Regret Bound

Lemma C.4.1. *For any $\alpha > 2$, the pseudo-regret term in* (4.11) *satisfies, for any T,*

$$\mathbb{E}_{\mathbf{w}}\left[\sum_{n=1}^{N} \Delta_n \gamma_T(n)\right] S(\mathbf{w}) \leq (\mathrm{C}_1(\mathbf{w}) \ln(T) + \mathrm{C}_2) S(\mathbf{w}),$$

where

$$\mathrm{C}_1(\mathbf{w}) \triangleq \sum_{n:\Delta_n > 0} \frac{\alpha \Delta_n}{\phi(\Delta_n S(\mathbf{w})/2)},$$
$$\mathrm{C}_2 \triangleq \left(\sum_{n=1}^{N} \Delta_n\right) \frac{2(\alpha - 1)}{\alpha - 2}. \tag{C.1}$$

Proof. We follow along similar steps to the proof of Theorem 2.1 in [25]. Suppose that $n_t = n$, and consider the events

$$\begin{aligned}
E_{t,1} &\triangleq \left\{ V_t(n_*) - f\left(\frac{\alpha \ln(t)}{\gamma_t(n_*) - 1}\right) \geq \sigma_*^2 S(\mathbf{w}) \right\}, \\
E_{t,2} &\triangleq \left\{ V_t(n) < \sigma_n^2 S(\mathbf{w}) - f\left(\frac{\alpha \ln(t)}{\gamma_t(n) - 1}\right) \right\}, \\
E_{t,3} &\triangleq \left\{ \gamma_t(n) < 1 + \frac{\alpha \ln(T)}{\phi(\Delta_n S(\mathbf{w})/2)} \right\}.
\end{aligned}$$

We claim that

$$E_{t,1} \cup E_{t,2} \cup E_{t,3}$$

must occur. Assume, by contradiction, that $E_{t,i}$ are all false. We obtain

$$\begin{aligned}
V_t(n_*) - f\left(\frac{\alpha \ln(t)}{\gamma_t(n_*) - 1}\right) &< \sigma_*^2 S(\mathbf{w}) \\
&= \sigma_n^2 S(\mathbf{w}) - \Delta_n S(\mathbf{w}) \\
&\leq V_t(n) + f\left(\frac{\alpha \ln(t)}{\gamma_t(n) - 1}\right) - \Delta_n S(\mathbf{w}). \qquad \text{(C.2)}
\end{aligned}$$

By assumption $E_{t,3}$ is false, and we have

$$\gamma_t(n) - 1 \geq \alpha \ln(T)/\phi(\Delta_n S(\mathbf{w})/2),$$

which is equivalent to

$$\Delta_n S(\mathbf{w}) \geq 2f\left(\frac{\alpha \ln(T)}{\gamma_t(n) - 1}\right), \qquad \text{(C.3)}$$

If we use (C.3) in (C.2), then we obtain the following result, which contradicts the rule in (4.4):

$$V_t(n_*) - f\left(\frac{\alpha \ln(t)}{\gamma_t(n_*) - 1}\right) < V_t(n) - f\left(\frac{\alpha \ln(t)}{\gamma_t(n) - 1}\right).$$

For all n such that $\Delta_n > 0$, we define

$$M_n \triangleq \left\lceil \frac{\alpha \ln(T)}{\phi(\Delta_n S(\mathbf{w})/2)} \right\rceil.$$

We next upper bound $\mathbb{E}_{\mathbf{w}}[\gamma_T(n)]$ as

$$\begin{aligned}\mathbb{E}_{\mathbf{w}}[\gamma_T(n)] &= \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{n_t = n\}\right] \\ &= \mathbb{E}_{\mathbf{w}}\left[\sum_{t=1}^{T} \mathbb{1}\{n_t = n \text{ and } \gamma_t(n) < M_n\}\right] \\ &+ \mathbb{E}_{\mathbf{w}}\left[\sum_{t=1}^{T} \mathbb{1}\{n_t = n \text{ and } \gamma_t(n) \geq M_n\}\right] \\ &\leq M_n + \mathbb{E}_{\mathbf{w}}\left[\sum_{t=M_n+1}^{T} \mathbb{1}\{n_t = n \text{ and } \gamma_t(n) \geq M_n\}\right]. \qquad \text{(C.4)}\end{aligned}$$

In (C.4), we observe that

$$\gamma_t(n) \geq M_n$$

is equivalent to $E_{t,3}$ being false, which is further equivalent to

$$E_{t,1} \cup E_{t,2}$$

being true, i.e., $E_{t,1}$ or $E_{t,2}$ must occur. Therefore we can further upper bound (C.4) as

$$\begin{aligned}\mathbb{E}_{\mathbf{w}}[\gamma_T(n)] &\leq M_n + \mathbb{E}_{\mathbf{w}}\left[\sum_{t=M_n+1}^{T} \mathbb{1}\{E_{t,1} \text{ or } E_{t,2} \text{ is true}\}\right] \\ &= M_n + \sum_{t=M_n+1}^{T} \Pr(E_{t,1} \cup E_{t,2} \text{ is true}) \\ &\leq M_n + \sum_{t=M_n+1}^{T} \Pr(E_{t,1}) + \sum_{t=M_n+1}^{T} \Pr(E_{t,2}). \qquad \text{(C.5)}\end{aligned}$$

where we used the union bound. We upper bound $\Pr(E_{t,1})$ for each $t = M_n + 1, \ldots, T$. Note that

$$\Pr(E_{t,1} = 1) = \Pr\left(V_t(n_*) - f\left(\frac{\alpha \ln(t)}{\gamma_t(n_*) - 1}\right) \geq \sigma_*^2 S(\mathbf{w})\right), \qquad \text{(C.6)}$$

where $\gamma_t(n_*)$ can take values in $\{2, \ldots, t\}$. Hence we apply the union bound in (C.6), which yields

$$\begin{aligned}
\Pr(E_{t,1} = 1) &\leq \sum_{s=1}^{t} \Pr\left(V_s'(n_*) - f\left(\frac{\alpha \ln(t)}{s}\right) \geq \sigma_*^2 S(\mathbf{w})\right) \\
&\leq \sum_{s=1}^{t} \frac{1}{t^\alpha} \\
&= t^{1-\alpha}, \qquad \text{(C.7)}
\end{aligned}$$

where (C.7) follows from (4.13). Here, $V_s'(n_*)$ is the trace of a sample covariance matrix given $s+2$ independent random vectors with sub-Gaussian components with the parameter $\sigma_*^2 S(\mathbf{w})$. Hence we obtain

$$\begin{aligned}
\sum_{t=M_n+1}^{T} \Pr(E_{t,1} = 1) &\leq \sum_{t=M_n+1}^{T} t^{1-\alpha} \\
&\leq \sum_{t=1}^{\infty} t^{1-\alpha} \\
&\leq 1 + \int_1^\infty t^{1-\alpha} dt \\
&= \frac{\alpha - 1}{\alpha - 2}. \qquad \text{(C.8)}
\end{aligned}$$

The same upper bound holds for $\Pr(E_{t,2} = 1)$ so that

$$\sum_{t=M_n+1}^{T} \Pr(E_{t,2} = 1) \leq \frac{\alpha - 1}{\alpha - 2}.$$

By incorporating these upper bounds into (C.5) we obtain

$$\mathbb{E}_{\mathbf{w}}[\gamma_t(n)] \leq M_n + 2(\alpha - 1)/(\alpha - 2).$$

Finally we use this result to get

$$\mathbb{E}_{\mathbf{w}}\left[\sum_{n=1}^{N} \Delta_n \gamma_T(n)\right] S(\mathbf{w}) \leq (\mathrm{C}_1(\mathbf{w}) \ln(T) + \mathrm{C}_2) S(\mathbf{w}),$$

where $\mathrm{C}_1(\mathbf{w})$ and C_2 are defined in (C.1). □

REFERENCES

[1] M. Zinkevich, “Online convex programming and generalized infinitesimal gradient ascent,” in *Proceedings of the 20th International Conference on Machine Learning (ICML 2003)*, Aug. 2003, pp. 928–936.

[2] A. Kalai and S. Vempala, “Efficient algorithms for online optimization,” *Journal of Computer and System Sciences*, vol. 71, no. 3, pp. 291–307, 2005.

[3] E. Hazan, “Efficient algorithms for online convex optimization and their applications,” Ph.D. dissertation, Princeton University, Princeton, NY, 2006.

[4] E. Hazan, A. Kalai, S. Kale, and A. Agarwal, “Logarithmic regret algorithms for online convex optimization,” in *Proceedings of the Nineteenth Annual Conference on Learning Theory (COLT)*, June 2006, pp. 499–513.

[5] E. Hazan, S. Kale, and A. Agarwal, “Logarithmic regret algorithms for online convex optimization,” *Machine Learning*, vol. 69, no. 2-3, pp. 169–192, Dec. 2007.

[6] S. Shalev-Shwartz and Y. Singer, “Convex repeated games and fenchel duality,” in *Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems (NIPS)*, Dec. 2007, pp. 1265–1272.

[7] J. Abernethy, P. Bartlett, A. Rakhlin, and A. Tewari, “Optimal strategies and minimax lower bounds for online convex games,” in *Proceedings of the Twenty-First Annual Conference on Learning Theory (COLT)*, July 2008, pp. 415–424.

[8] J. Abernethy, E. Hazan, and A. Rakhlin, “Competing in the dark: An efficient algorithm for bandit linear optimization,” in *Proceedings of the Twenty-First Annual Conference on Learning Theory (COLT)*, July 2008, pp. 263–274.

[9] N. Littlestone, “Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm,” *Machine Learning*, vol. 2, no. 4, pp. 285–318, Apr. 1988.

[10] V. Vovk, “Aggregating strategies,” in *Proceedings of the Third Annual Workshop on Computational Learning Theory (COLT)*, Aug. 1990, pp. 371–383.

[11] N. Cesa-Bianchi, Y. Freund, D. Haussler, D. Helmbold, R. Schapire, and M. Warmuth, “How to use expert advice,” *Journal of the ACM*, vol. 44, no. 3, pp. 427–485, 1997.

[12] D. Haussler, J. Kivinen, and M. Warmuth, “Sequential prediction of individual sequences under general loss functions,” *IEEE Transactions on Information Theory*, vol. 44, no. 2, pp. 1906–1925, 1998.

[13] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games.* New York, NY, USA: Cambridge University Press, 2006.

[14] T. Cover, “Universal portfolios,” *Mathematical Finance*, vol. 1, no. 1, pp. 1–29, Jan. 1991.

[15] A. Blum and A. Kalai, “Universal portfolios with and without transaction costs,” in *Proceedings of the Tenth Annual Conference on Computational Learning Theory (COLT)*, July 1997, pp. 309–313.

[16] A. Kalai and S. Vempala, “Efficient algorithms for universal portfolios,” *Journal of Machine Learning Research*, vol. 3, pp. 423–440, 2003.

[17] N. Merhav and M. Feder, “Universal schemes for sequential decision from individual data sequences,” *IEEE Transactions on Information Theory*, vol. 39, no. 4, pp. 1280–1292, 1993.

[18] M. Weinberger, N. Merhav, and M. Feder, “Optimal sequential probability assignment for individual sequences,” *IEEE Transactions on Information Theory*, vol. 40, no. 2, pp. 384–396, Mar. 1994.

[19] T. Weissman and N. Merhav, “Universal prediction of individual binary sequences in the presence of noise,” *IEEE Transactions on Information Theory*, vol. 47, no. 6, pp. 2151–2173, sep 2001.

[20] D. Foster, “Regret in the on-line decision problem,” *Games and Economic Behavior*, vol. 29, pp. 7–36, 1999.

[21] W. H. Butler, T. Mewes, C. K. A. Mewes, P. B. Visscher, W. H. Rippard, S. E. Russek, and R. Heindl, “Switching distributions for perpendicular spin-torque devices within the macrospin approximation,” *IEEE Transactions on Magnetics*, vol. 48, no. 12, pp. 4684–4700, Dec. 2012.

[22] A. Patil, S. Manipatruni, D. E. Nikonov, I. Young, and N. Shanbhag, “Enabling spin logic via Shannon-inspired statistical computing,” in *Thirteenth Joint MMM-Intermag Conference*, Jan. 2016.

[23] L. R. Varshney, P. J. Sjöström, and D. B. Chklovskii, “Optimal information storage in noisy synapses under resource constraints,” *Neuron*, vol. 52, no. 3, pp. 409–423, Nov. 2006.

[24] P. Auer, “Using confidence bounds for exploitation-exploration tradeoffs,” *Journal of Machine Learning Research*, vol. 3, no. 1, pp. 397–422, Nov. 2002.

[25] S. Bubeck and N. Cesa-Bianchi, “Regret analysis of stochastic and nonstochastic multi-armed bandit problems,” *Foundations and Trends in Machine Learning*, vol. 5, no. 1, pp. 1–122, Dec. 2012.

[26] E. Hazan, “The convex optimization approach to regret minimization,” in *Optimization for Machine Learning*, S. Sra, S. Nowozin, and S. Wright, Eds. Cambridge, MA: MIT Press, 2011, pp. 287–303.

[27] S. Shalev-Shwartz, “Online learning and online convex optimization,” *Foundations and Trends in Machine Learning*, vol. 4, no. 2, pp. 107–194, 2011.

[28] J. Duchi, E. Hazan, and Y. Singer, “Adaptive subgradient methods for online learning and stochastic optimization,” *Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, July 2011.

[29] E. Takimoto and M. Warmth, “The minimax strategy for gaussian density estimation,” in *Proceedings of the Thirteenth Annual Conference on Computational Learning Theory (COLT)*, 2000, pp. 100–106.

[30] H. Narayanan and A. Rakhlin, “Random walk approach to regret minimization,” in *Proceedings of the Twenty-Third Annual Conference on Neural Information Processing Systems (NIPS)*, Dec. 2010, pp. 1777–1785.

[31] H. Narayanan and A. Rakhlin, “Efficient sampling from time-varying log-concave distributions,” *arXiv preprint*, 2013. [Online]. Available: http://arxiv.org/pdf/1309.5977v1.pdf

[32] T. Weissman, “How to filter an individual sequence with feedback,” *IEEE Transactions on Information Theory*, vol. 54, no. 8, pp. 3831–3841, Aug. 2008.

[33] R. Arora, O. Dekel, and A. Tewari, “Online bandit learning against an adaptive adversary: From regret to policy regret,” in *Proceedings of the 29th International Conference on Machine Learning (ICML12)*, 2012, pp. 1503–1510.

[34] N. Cesa-Bianchi, O. Dekel, and O. Shamir, "Online learning with switching costs and other adaptive adversaries," in *Proceedings of the Twenty-Seventh Annual Conference on Neural Information Processing Systems (NIPS)*, Dec. 2013, pp. 1160–1168.

[35] D. A. McQuarrie, *Statistical Mechanics*. Herndon, VA, USA: University Science Books, 2000.

[36] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. Cambridge, MA, USA: MIT Press, 2001.

[37] J. Hannan, "Approximation to Bayes risk in repeated play," in *Contributions to the Theory of Games*, M. Dresher, A. Tucker, and P. Wolfe, Eds. Princeton, NJ: Princeton University Press, 1957, vol. 3, pp. 97–139.

[38] H. P. Wong, D. J. Frank, P. M. Solomon, C. H. J. Wann, and J. J. Welser, "Nanoscale CMOS," *Proceedings of the IEEE*, vol. 87, no. 4, pp. 537–570, Apr. 1999.

[39] L. Wang and N. R. Shanbhag, "Low-power filtering via adaptive error-cancellation," *IEEE Transactions on Signal Processing*, vol. 51, no. 2, pp. 575–583, Feb. 2003.

[40] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*. New York, NY, USA: Springer, 1999.

[41] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Upper Saddle River, NJ, USA: Prentice Hall, 2010.

[42] J. von Neumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," *Automata Studies*, vol. 34, pp. 43–98, 1956.

[43] J. G. Tryon, "Quadded logic," in *Redundancy Techniques for Computing Systems*, R. H. Wilcox and W. C. Mann, Eds. Washington: Spartan Books, 1962, pp. 205–228.

[44] S. Winograd and J. D. Cowan, *Reliable Computation in the Presence of Noise*. Boston, MA, USA: MIT Press, 1963.

[45] N. Pippenger, "Reliable computation by formulas in the presence of noise," *IEEE Transactions on Information Theory*, vol. 34, no. 2, pp. 194–197, Mar. 1988.

[46] B. Hajek and T. Weller, "On the maximum tolerable noise for reliable computation by formulas," *IEEE Transactions on Information Theory*, vol. 37, no. 2, pp. 388–391, Mar. 1991.

[47] W. Evans and N. Pippenger, “On the maximum tolerable noise for reliable computation by formulas,” *IEEE Transactions on Information Theory*, vol. 44, no. 3, pp. 1299–1305, May 1998.

[48] Y. Yang, P. Grover, and S. Kar, “Computing linear transforms with unreliable components,” in *Proceedings of the 2016 IEEE International Symposium on Information Theory*, July 2016, pp. 1934–1938.

[49] R. Viswanathan and P. K. Varshney, “Distributed detection with multiple sensors Part I. Fundamentals,” *Proceedings of the IEEE*, vol. 85, no. 1, pp. 54–63, Jan. 1997.

[50] P. Ishwar, R. Puri, K. Ramchandran, and S. S. Pradhan, “On rate-constrained distributed estimation in unreliable sensor networks,” *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 765–775, Apr. 2005.

[51] A. Ribeiro and G. B. Giannakis, “Bandwidth-constrained distributed estimation for wireless sensor networks—Part I: Gaussian case,” *IEEE Transactions on Signal Processing*, vol. 54, no. 3, pp. 1131–1143, Mar. 2006.

[52] A. Ribeiro and G. B. Giannakis, “Bandwidth-constrained distributed estimation for wireless sensor networks—Part II: Unknown probability density function,” *IEEE Transactions on Signal Processing*, vol. 54, no. 7, pp. 1131–1143, July 2006.

[53] S. Barbarossa, S. Sardellitti, and P. D. Lorenzo, “Distributed detection and estimation in wireless sensor networks,” *arXiv preprint*, 2013. [Online]. Available: https://arxiv.org/pdf/1307.1448.pdf

[54] S. A. Aldosari and J. M. F. Moura, “Fusion in sensor networks with communication constraints,” in *Proceedings of the Third International Symposium on Information Processing in Sensor Networks (IPSN’04)*, Apr. 2004, pp. 108–115.

[55] D. Marco and D. L. Neuhoff, “Reliability vs. efficiency in distributed source coding for field-gathering sensor networks,” in *Proceedings of the Third International Symposium on Information Processing in Sensor Networks (IPSN’04)*, Apr. 2004, pp. 161–168.

[56] S. Boucheron, G. Lugosi, and P. Massart, *Concentration Inequalities.* Oxford, UK: Oxford University Press, 2013.

[57] Y. Mishchenko, T. Hu, J. Spacek, J. Mendenhall, K. M. Harris, and D. B. Chklovskii, “Ultrastructural analysis of hippocampal neuropil from the connectomics perspective,” *Neuron*, vol. 67, no. 6, pp. 1009–1020, Sep. 2010.

[58] C. Shannon, “A mathematical theory of communication,” *The Bell System Technical Journal*, vol. 27, pp. 379–423, July 1948.

[59] N. Brunel, V. Hakim, P. Isope, J.-P. Nadal, and B. Barbour, “Optimal information storage and distribution of synaptic weights: Perceptron versus Purkinje cells,” *Neuron*, vol. 43, no. 5, pp. 745–757, Sep. 2004.

[60] S. B. Laughlin, R. R. de Ruyter van Steveninck, and J. C. Anderson, “The metabolic cost of neural information,” *Nature Neuroscience*, vol. 1, no. 1, pp. 36–41, May 1998.

[61] A. Zador, “Impact of synaptic unreliability on the information transmitted by spiking neurons,” *Journal of Neurophysiology*, vol. 79, no. 3, pp. 1219–1229, Mar. 1998.

[62] A. Manwani and C. Koch, “Detecting and estimating signals over noisy and unreliable synapses: Information-theoretic analysis,” *Neural Computation*, vol. 13, no. 1, pp. 1–33, Jan. 2001.

[63] W. B. Levy and R. A. Baxter, “Energy-efficient neuronal computation via quantal synaptic failures,” *The Journal of Neuroscience*, vol. 22, no. 11, pp. 4746–4755, June 2002.

[64] M. S. Goldman, “Enhancement of information transmission efficiency by synaptic failures,” *Neural Computation*, vol. 16, no. 6, pp. 1137–1162, June 2004.

[65] S. B. Laughlin and T. J. Sejnowski, “Communication in neuronal networks,” *Science*, vol. 301, no. 5641, pp. 1870–1874, Sep. 2003.

[66] J. Abrevaya and W. Jiang, “A nonparametric approach to measuring and testing curvature,” *Journal of Business & Economic Statistics*, vol. 23, no. 1, pp. 1–19, Sep. 2005.

[67] D. K. Morest, “The collateral system of the medial nucleus of the trapezoid body of the cat, its neuronal architecture and relation to the olivocochlear bundle,” *Brain Research*, vol. 9, no. 2, pp. 288–311, July 1968.

[68] P. H. Smith, P. X. Joris, L. H. Carney, and T. C. Yin, “Projections of physiologically characterized globular bushy cell axons from the cochlear nucleus of the cat,” *The Journal of Comparative Neurology*, vol. 304, no. 3, pp. 387–407, Feb. 1991.

[69] J. Hermann, “Information processing at the calyx of Held synapse under natural conditions,” Ph.D. dissertation, Ludwig Maximilian University of Munich, Munich, Germany, 2008.

[70] K. M. Spangler, W. B. Warr, and C. K. Henkel, “The projections of principal cells of the medial nucleus of the trapezoid body in the cat,” *The Journal of Comparative Neurology*, vol. 283, no. 3, pp. 249–262, Aug. 1985.

[71] C. Tsuchitani, “Input from the medial nucleus of trapezoid body to an interaural level detector,” *Hearing Research*, vol. 105, no. 1-2, pp. 211–224, Mar. 1997.

[72] K. Futai, M. Okada, K. Matsuyama, and T. Takahashi, “High-fidelity transmission acquired via a developmental decrease in NMDA receptor expression at an auditory synapse,” *The Journal of Neuroscience*, vol. 21, no. 10, pp. 3342–3349, May 2001.

[73] J. W. Choi, B. Shim, A. C. Singer, and N. I. Cho, “Low-power filtering via minimum power soft error cancellation,” *IEEE Transactions on Signal Processing*, vol. 55, no. 10, pp. 5084–5096, Oct. 2007.

[74] R. A. Abdallah and N. R. Shanbhag, “An energy-efficient ECG processor in 45-nm CMOS using statistical error compensation,” *IEEE Journal of Solid-State Circuits*, vol. 48, no. 11, pp. 2882–2893, Nov. 2013.

[75] J. Han, E. Leung, L. Liu, and F. Lombardi, “A fault-tolerant technique using quadded logic and quadded transistors,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 8, pp. 1562–1566, Aug. 2015.

[76] C. N. Hadjicostis, “Coding approaches to fault tolerance in dynamic systems,” Ph.D. dissertation, EECS Department, Massachusetts Institute of Technology, Cambridge, MA, 1999.

[77] R. Hegde and N. R. Shanbhag, “Soft digital signal processing,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 9, no. 6, pp. 813–823, Dec. 2001.

[78] E. P. Kim and N. R. Shanbhag, “Statistical analysis of algorithmic noise tolerance,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2013)*, May 2013, pp. 2731–2735.

[79] D. Seo and L. R. Varshney, “Information-theoretic limits of algorithmic noise tolerance,” in *Proceedings of the IEEE International Conference on Rebooting Computing (ICRC)*, Nov. 2016, pp. 1–4.

[80] M. Kang, M. S. Keel, N. R. Shanbhag, S. Eilert, and K. Curewitz, "An energy-efficient VLSI architecture for pattern recognition via deep embedding of computation in SRAM," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2014)*, May 2014, pp. 8323–8330.

[81] Y. Hu, W. Rieutort-Louis, J. Sanz-Robinson, K. Song, J. C. Sturm, S. Wagner, and N. Verma, "High-resolution sensing sheet for structural-health monitoring via scalable interfacing of flexible electronics with high-performance ICs," in *Proceedings of the 2012 Symposium on VLSI Circuits (VLSIC)*, June 2012, pp. 120–121.

[82] N. R. Shanbhag, "Energy-efficient machine learning in silicon: A communications-inspired approach," *arXiv preprint*, 2016. [Online]. Available: https://arxiv.org/abs/1611.03109

[83] S. Jianhan, V. W. I. Phua, and L. R. Varshney, "Distributed estimation via paid crowd work," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2016)*, Mar. 2016, pp. 6200–6204.

[84] F. Lahouti and B. Hassibi, "Fundamental limits of budget-fidelity trade-off in label crowdsourcing," in *Proceedings of the Twenty-Ninth Annual Conference on Neural Information Processing Systems (NIPS)*, Dec. 2016, pp. 5058–5066.

[85] D. R. Karger, S. Oh, and D. Shah, "Budget-optimal task allocation for reliable crowdsourcing systems," *Operations Research*, vol. 62, no. 1, pp. 1–24, Feb. 2014.

[86] A. Khetan and S. Oh, "Achieving budget-optimality with adaptive schemes in crowdsourcing," in *Proceedings of the Twenty-Ninth Annual Conference on Neural Information Processing Systems (NIPS)*, Dec. 2016, pp. 4844–4852.

[87] C.-J. Ho, A. Slivkins, S. Suri, and J. W. Vaughan, "Incentivizing high quality crowdwork," in *Proceedings of the 24nd International Conference on World Wide Web (WWW'15)*, May 2015, pp. 419–429.

[88] L. Devroye, M. Lerasle, G. Lugosi, and R. I. Oliveira, "Sub-Gaussian mean estimators," *arXiv preprint*, 2015. [Online]. Available: https://arxiv.org/abs/1509.05845

[89] A. Xu and M. Raginsky, "Information-theoretic lower bounds for distributed function computation," *arXiv preprint*, 2015. [Online]. Available: https://arxiv.org/abs/1509.00514

[90] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," arXiv preprint 1606.04838, 2016. [Online]. Available: https://arxiv.org/abs/1606.04838

[91] C. Wang, X. Chen, A. Smola, and E. P. Xing, "Variance reduction for stochastic gradient optimization," in *Proceedings of the Twenty-Seventh Annual Conference on Neural Information Processing Systems (NIPS)*, Dec. 2013, pp. 181–189.

[92] M. Li, T. Zhang, Y. Chen, and A. J. Smola, "Efficient mini-batch training for stochastic optimization," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2014, pp. 661–670.

[93] M. A. Zinkevich, M. Weimer, A. Smola, and L. Li, "Parallelized stochastic gradient descent," in *Proceedings of the Twenty-Third Annual Conference on Neural Information Processing Systems (NIPS)*, Dec. 2010, pp. 2595–2603.

[94] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," arXiv preprint 1602.05629, 2016. [Online]. Available: https://arxiv.org/abs/1602.05629

[95] H. Robbins, "Some aspects of the sequential design of experiments," *Bulletin of the American Mathematical Society*, vol. 58, no. 5, pp. 527–535, 1952.

[96] P. Auer and R. Ortner, "UCB revisited: Improved regret bounds for the stochastic multi-armed bandit problem," *Periodica Mathematica Hungarica*, vol. 61, no. 1, pp. 55–65, Sep. 2010.

[97] D. L. Hanson and F. T. Wright, "A bound on tail probabilities for quadratic forms in independent random variables," *The Annals of Mathematical Statistics*, vol. 41, pp. 1079–1083, 1971.

[98] M. Rudelson and R. Vershynin, "Hanson-Wright inequality and sub-Gaussian concentration," *Electron. Commun. Probab.*, vol. 18, no. 82, pp. 1–9, 2013.

[99] A. Vempaty, L. R. Varshney, and P. K. Varshney, "Reliable crowdsourcing for multi-class labeling using coding theory," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 4, pp. 667–679, Aug. 2014.

[100] J. Neyman and E. S. Pearson, "On the problem of the most efficient tests of statistical hypotheses," *Philosophical Transactions of the Royal Society. Series A*, vol. 231, no. 694-706, pp. 289–337, 1933.

[101] D. Berend and A. Kontorovich, "A finite sample analysis of the naive Bayes classifier," *Journal of Machine Learning Research*, vol. 16, pp. 1519–1545, 2015.

[102] S. Zhang and N. R. Shanbhag, "Probabilistic error models for machine learning kernels implemented on stochastic nanoscale fabrics," in *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE16)*, Apr. 2016, pp. 481–486.

[103] T.-Y. Wang, Y. S. Han, P. K. Varshney, and P.-N. Chen, "Distributed fault-tolerant classification in wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 724–734, Apr. 2005.

[104] C. Yao, P.-N. Chen, T.-Y. Wang, Y. S. Han, and P. K. Varshney, "Performance analysis and code design for minimum Hamming distance fusion in wireless sensor networks," *IEEE Transactions on Information Theory*, vol. 53, no. 5, pp. 1716–1734, Apr. 2007.

[105] T. G. Dietterich and G. Bakiri, "Solving multiclass learning problems via error-correcting output codes," *Journal of Artificial Intelligence Research*, vol. 2, no. 1, pp. 263–286, Aug. 1994.

[106] A. Sahai and P. Grover, "The price of certainty: 'Waterslide curves' and the gap to capacity," *arXiv preprint*, 2008. [Online]. Available: https://arxiv.org/abs/0801.0352