The many faces of online learning

Hoeven, D. van der

Author: Hoeven, D. van der
Title: The many faces of online learning
Issue Date: 2021-03-4

The Many Faces of Online Learning

Dirk van der Hoeven

Cover illustration based on an illustration by istockphoto.com/luckytd.

The research in this dissertation was supported by the Netherlands Organization for Scientific Research (NWO grant TOP2EW.15.211).

The Many Faces of Online Learning

Proefschrift

ter verkrijging van
de graad van Doctor aan de Universiteit Leiden,
op gezag van Rector Magnificus Prof. dr. ir. H. Bijl,
volgens besluit van het College voor Promoties
te verdedigen op donderdag 4 maart 2021
klokke 16:15 uur

door

Dirk van der Hoeven
geboren te Leidschendam
in 1992

Promotor:

Prof. dr. P. D. Grünwald (Universiteit Leiden en Centrum Wiskunde & Informatica)

Copromotor:

Dr. T. A. L. van Erven (Universiteit van Amsterdam)

Promotiecommissie:

Prof. dr. N. Cesa-Bianchi (Università degli Studi di Milano)
Dr. W.M. Koolen (Centrum Wiskunde & Informatica)
Dr. S.L. van der Pas
Prof. dr. S.J. Edixhoven
Prof. dr. E.R. Eliel

Parts of this dissertation are based on the following papers and collaborations.

Chapter 2 is based on:

Van der Hoeven, D., Van Erven, T., and Kotłowski, W. (2018). The many faces of exponential weights in online learning. In *Proceedings of the 31st Annual Conference on Learning Theory (COLT)*, pages 2067–2092

Chapter 3 is based on:

Van der Hoeven, D. (2019). User-specified local differential privacy in unconstrained adaptive online learning. In *Advances in Neural Information Processing Systems 32*, pages 14103–14112

Chapter 4 is based on:

Van der Hoeven, D., Cutkosky, A., and Luo, H. (2020). Comparator-adaptive convex bandits. *To Appear in Advances in Neural Information Processing Systems 33*

Chapter 5 is based on:

Van Erven, T., Koolen, W. M., and Van der Hoeven, D. (2020a). Metagrad: Universal adaptation using multiple learning rates in online learning. *Manuscript in preparation*

Chapter 6 is based on:

Van der Hoeven, D. (2020). Exploiting the surrogate gap in online multiclass classification. *To Appear in Advances in Neural Information Processing Systems 33*

Chapter 7 is based on:

Van Erven, T., Van der Hoeven, D., Kotłowski, W., and Koolen, W. M. (2020b). Open problem: Fast and optimal online portfolio selection. In *Proceedings of the 33rd Annual Conference on Learning Theory (COLT)*, pages 3864–3869

Contents

CHAPTER 1

Introduction

Online Learning is a fundamental task in Machine Learning. It is the task of sequentially making predictions given previous feedback and possibly additional information. Online Learning tasks have many interesting theoretical properties and practical applications. Examples of Online Learning tasks include gambling on sport matches (Vovk and Zhdanov, 2009), spam filtering, weather prediction, portfolio selection (Cover, 1991), training machine learning models on very large data, and many more.

Online Learning proceeds in a sequence of rounds $t = 1, \dots, T$, where in each round t the learner has to issue a prediction. In specialized cases of Online Learning such as Online Classification these predictions may come from a very limited set of a finite number of possible labels. In other settings in Online Learning such as Online Convex Optimization predictions may come from any convex set. After the learner has issued his prediction he receives feedback from the environment in the form of a loss function ℓ_t, which tells the learner the error in his prediction. In the full-information setting the learner sees the entire loss function, which means that the learner can also assess the quality of alternative predictions. In the more difficult bandit setting the learner only sees the value of the loss function evaluated at his prediction, which means the learner *cannot* assess the quality of alternative predictions.

As a simple example let us consider predicting whether or not it is going to rain tomorrow. Suppose that we encode "it will not rain" as -1 and "it will rain" as 1. We will denote the prediction of the learner with $\hat{y}_t \in \{-1, 1\}$ and the actual outcome as $y_t \in \{-1, 1\}$. If the learner makes a prediction today he will only get feedback tomorrow, which he may use to make his prediction for the next day. The loss function in this case could be the zero-one loss, i.e. $\ell_t(\hat{y}_t) = \mathbb{1}[\hat{y}_t \neq y_t]$, where $\mathbb{1}$ is the indicator function. This loss function is easy for the learner to come by as he only has to pay attention to the weather throughout the day. The goal here is straightforward, the learner wants to make as few mistakes as possible. As a more involved example let us consider online regression. Suppose that the learner is interested in predicting how much it is going to rain. Before making his prediction the learner might have some additional information, such as barometric pressure, whether or not there are clouds, and all the tweets from the preceding twenty-four hours that contain the word weather. In each round t, the learner could use a hypothesis h_t from hypothesis set $\mathcal{H}$ to map the additional information, which we will denote by $\boldsymbol{x}_t$, to amount of rain. In this example, the learner could be interested in minimizing the quadratic difference between his prediction and the actual amount of rain $y_t \in [0, m]$, i.e. $\ell_t(h_t) = (h_t(\boldsymbol{x}_t) - y_t)^2$, where m is the size of the cup the learner uses to measure the amount of rain with.

In both examples the goal of the learner is to minimize the cumulative loss he suffers, summed over rounds $t = 1, \ldots, T$. The learner could try to use the feedback he received the previous days to continuously improve his predictions. However, the learner could have a neighbour who is accidentally disturbing his measurements by spraying his garden with water. Clearly, this makes the task of the learner harder as the relation between his observations is weakened. Even more problematic, the learner could face an adversarial neighbour who actively tries to disrupt the learner's measurements. This adversarial neighbour makes learning very difficult, as any correlation between previous days and the next is removed.

Unlike in classical statistical learning theory, which makes probabilistic assumptions about the environment, Online Learning is able to provide some guarantees about the cumulative loss of the learner without probabilistic assumptions, even in adversarial environments. The guarantees of Online Learning are about the regret, which measures how "sorry" the learner is for making the predictions he has made. To ensure that the regret can be suitably bounded we often require that the predictions and best fixed prediction strategy come from the same set, which is often bounded and convex. Suppose that the predictions h_t of the learner come from an abstract hypothesis class $\mathcal{H}$. This hypothesis class can be anything from the set of vectors in a unit ball to a set of functions that map additional information to outcomes, to a simple class that only consists of -1 and 1 as in the example above. The regret is defined as the difference between the cumulative loss of the learner and the cumulative loss of the best fixed prediction strategy in hindsight:

$$\mathcal{R}_T = \sum_{t=1}^{T} \ell_t(h_t) - \min_{h \in \mathcal{H}} \sum_{t=1}^{T} \ell_t(h).$$

The goal of the learner is to have small regret. This does not mean that the best strategy is a good strategy; it is merely a benchmark with which the learner compares himself. The learner is satisfied as long as the regret is sublinear, which is to say that the average loss of the learner minus the average loss of the best fixed strategy goes to zero as T grows.

Overview of the dissertation The chapters in this dissertation can be read independently, but for a full appreciation of each chapter it is recommended that the reader starts with the current chapter. The remainder of this chapter serves to provide a gentle introduction for the chapters that follow.

A common theme in several chapters of this dissertation is how to design algorithms that can handle adversarial environments but also exploit benign environments. In Chapter 2 we show how we can design adaptive algorithms for the Prediction with Expert Advice setting, which we introduce in Section 1.1, by using a reduction based on the Exponential Weights algorithm (Vovk, 1990; Littlestone and Warmuth, 1994). In Chapter 3 we derive algorithms that are adaptive to unknown noise in the Online Convex Optimization setting, a setting which we will introduce together with the Bandit Convex Optimization setting in Section 1.2. In Chapter 4 we derive the first algorithms that are adaptive to the norm of the offline optimizer for the Bandit Convex Optimization setting. Finally, in Chapter 5 we describe MetaGrad, which operates in the Online Convex Optimization setting. MetaGrad is adaptive to a large class of loss functions, including exp-concave and various other types of functions.

Since the learner updates his predictions each round it is important that the updates can be performed reasonably fast. For many Online learning settings algorithms with per round running time larger than quadratic in the dimension of the problem are considered impractical. In several chapters we will show how to improve the running time of Online Learning algorithms while maintaining similar or even improved guarantees. In Chapter 6 we consider the full-information and bandit Online Multiclass Classification settings, which we introduce in Section 1.3. We introduce a new approach to Online Multiclass Classification which allows us to use an algorithm that has a per round running time that is linear in the dimension of the problem that guarantees small regret bounds. Interestingly, our algorithm often improves upon the regret bounds of slower algorithms. In Chapter 5 we show how to improve the running time of MetaGrad by using matrix sketching methods at the cost of a slightly larger regret bound. Finally, in Chapter 7 we pose an open problem that asks for a fast and optimal algorithm for online portfolio selection. We propose a fast algorithm and an analysis of this algorithm that shows that in some special cases of online portfolio selection this algorithm indeed obtains the optimal regret bound.

1.1 Prediction with Expert Advice

Our first setting in Online Learning is perhaps the most well-studied setting: the prediction with expert advice setting. In the prediction with expert advice setting the learner has access to d experts. In a given round t, each expert i sends his prediction $\hat{y}_t^i$ to the learner, who may use these expert predictions to form his own prediction. These experts can be anything, for example the learner's neighbours who predict how much rain is going to fall, static experts $i = 1, \ldots, d$ who always say it is going to rain i mm, or arbitrary points in a convex set. To issue his predictions, the learner forms a distribution p_t over the experts. The learner's loss becomes $\hat{\ell}_t = \underset{i \sim p_t}{\mathbb{E}}[\ell_t^i]$, where $\ell_t^i = \ell_t(\hat{y}_t^i)$ is the loss of expert i at time t. Loss $\hat{\ell}_t$ can be motivated in several ways:

(a) If the learner randomly chooses an expert $i \sim p_t$ then this is the expected loss.

(b) If ℓ_t is convex and the learner predicts $\hat{y}_t = \mathbb{E}_{p_t}[\hat{y}_t^i]$ then by Jensen's inequality $\hat{\ell}_t$ is an upper bound on the learner's loss.

The goal of the learner in the prediction with expert advice setting is to predict almost as well as the best expert in hindsight, which is to say that the regret with respect to the best expert in hindsight is sublinear.

A fundamental algorithm in the prediction with Expert Advice setting and Online Learning in general is the Exponential Weights algorithm. With a discrete set of experts, the distribution of Exponential Weights has the following form:

$$p_t(i) \propto \pi(i) e^{-\eta \sum_{s=1}^{t-1} \ell_s^i},$$

where $\eta > 0$ is called the learning rate and $\pi(i)$ is the prior mass on expert i. Unsurprisingly, Exponential Weights gets its name from the exponentially weighted losses of each expert. Somewhat surprisingly, Exponential Weights can be applied in many different settings and several other algorithms are special cases of it. For example, in Chapter 2 we will see that with a continuous set of experts and a Gaussian prior Online Gradient Descent (Zinkevich, 2003) is a special case of Exponential Weights.

For losses such that $\ell_t(\hat{y}_t^i) \in [0, 1]$, Exponential Weights with learning rate $\eta = \sqrt{\frac{8 \ln(d)}{T}}$ provides the following guarantee (see for example Theorem 2.2 by Cesa-

Bianchi and Lugosi (2006)):

$$\sum_{t=1}^{T} \hat{\ell}_t - \min_i \sum_{t=1}^{T} \ell_t^i \leq \sqrt{\frac{\ln(d)T}{2}}.$$

As we can see, as T grows the difference between the average loss of the learner and the average loss of the best expert decreases.

In an adversarial environment the learner can not do better than the above regret bound (see section 3.7 by Cesa-Bianchi and Lugosi (2006)). However, in more benign environments the learner could have had a better guarantee. For example, it could have been clear from the start that the predictions from the neighbour who works at the KNMI[1] are the best predictions. In fact, if the learner only listens to the KNMI neighbour after a few rounds he will no longer suffer any additional regret after these initial rounds. This means that the learner will have to quickly learn to only listen to the KNMI neighbour to get a small regret bound.

With the Exponential Weights algorithm the speed at which the algorithm learns is governed by the learning rate η. In adversarial settings the learning rate is set such that the distribution over the experts p_t does not change drastically between rounds. However, to quickly learn that one expert is clearly the best expert, η would have to be tuned so that p_t quickly converges to a point mass on the best expert. Unfortunately, the learner usually does not know beforehand whether or not his environment is adversarial, benign, or something in between adversarial and benign. This means that the safest thing for the learner to do is to tune his algorithms to deal with an adversarial setting. In Chapter 2 we will show that with a reduction based on Exponential Weights we recover the Squint (Koolen and Van Erven, 2015) and coin betting for experts (Orabona and Pál, 2016) algorithms that adjust their learning rate automatically, which allows these algorithms to adapt to different environments.

1.2 Online Convex Optimization

The Prediction with Expert Advice setting is a special case of the Online Convex Optimization setting. In the Online Convex Optimization setting the predictions of the learner can come from any convex set, for example an L_2 ball or the probability simplex. This setting is called the Online *Convex* Optimization setting because the loss functions are assumed to be convex. Applications of Online Convex

[1] Dutch National Weather Institute

Optimization include training machine learning models on very large data and online classification.

In Online Convex Optimization, in each of $t = 1, \dots, T$ rounds, the learner has to make a prediction $\boldsymbol{w}_t$ in a convex domain $\mathcal{W}$ before observing a convex loss function $\ell_t : \mathcal{W} \to \mathbb{R}$. The goal is to obtain a guaranteed bound on the regret

$$\mathcal{R}_T = \sum_{t=1}^{T} \ell_t(\boldsymbol{w}_t) - \min_{\boldsymbol{w} \in \mathcal{W}} \sum_{t=1}^{T} \ell_t(\boldsymbol{w})$$

that holds for any possible sequence of loss functions ℓ_t. To be able to bound the regret a standard assumption is that the domain is bounded, but in Chapter 3 we consider algorithms that are able to achieve suitable regret bounds with an unbounded domain.

To see how the Prediction with Expert Advice setting is a special case of the Online Convex Optimization setting we set the domain $\mathcal{W} = \{\boldsymbol{w} \in \mathbb{R}_+^d \mid \sum_{i=1}^d w_i = 1\}$ to be the probability simplex and let the losses be linear: $\ell_t(\boldsymbol{w}_t) = \boldsymbol{w}_t^\intercal \boldsymbol{g}_t = \hat{\ell}_t$, where $\boldsymbol{g}_t = (\ell_t^1, \dots, \ell_t^d)$. With this loss the definition of the regret in the Online Convex Optimization setting coincides with the regret of the Prediction with Expert Advice setting.

Another example of an Online Convex Optimization task is online portfolio selection (Cover, 1991). Online portfolio selection corresponds to the special case that the domain is the probability simplex and the loss functions are restricted to be of the form $\ell_t(\boldsymbol{w}) = -\ln(\boldsymbol{w}^\intercal \boldsymbol{x}_t)$ for vectors $\boldsymbol{x}_t \in \mathbb{R}_+^d$. With online portfolio selection the goal of the learner is to distribute his funds over several assets. Online portfolio selection was introduced by Cover (1991) with the interpretation that $x_{t,i}$ represents the factor by which the value of an asset $i \in \{1, \dots, d\}$ grows in round t and $w_{t,i}$ represents the fraction of our capital we re-invest in asset i in round t. The factor by which our initial capital grows over T rounds then becomes $\prod_{t=1}^T \boldsymbol{w}_t^\intercal \boldsymbol{x}_t = e^{-\sum_{t=1}^T \ell_t(\boldsymbol{w}_t)}$.

Cover (1991); Cover and Ordentlich (1996) show that the best possible guarantee on the regret is of order $\mathcal{R}_T = O(d \ln T)$ and that this is achieved by choosing $\boldsymbol{w}_{t+1}$ as the mean of a continuous Exponential Weights distribution $\mathrm{d}P_{t+1}(\boldsymbol{w}) \propto e^{-\sum_{s=1}^t \ell_s(\boldsymbol{w})} \mathrm{d}\pi(\boldsymbol{w})$ with Dirichlet-prior π (and learning rate $\eta = 1$). Unfortunately, this approach has a runtime of order $O(T^d)$, which scales exponentially in the number of assets d, and is therefore computationally infeasible when d exceeds, say, 3. A sampling-based implementation by Kalai and Vempala (2002) greatly improves the runtime to $\tilde{O}(T^4(T + d)d^2)$, but even this is still infeasible already for modest d and T.

A common approach to runtime problems in Online Convex Optimization is instead of optimizing the loss ℓ_t directly, optimizing a linear or quadratic approximation of ℓ_t. With a linear approximation, we make use of the convexity to upper bound the regret

$$\sum_{t=1}^{T} (\ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{u})) \leq \sum_{t=1}^{T} (\boldsymbol{w}_t - \boldsymbol{u})^\intercal \nabla \ell_t(\boldsymbol{w}_t) = \sum_{t=1}^{T} \tilde{\ell}_t(\boldsymbol{w}_t) - \tilde{\ell}_t(\boldsymbol{u}),$$

where $\boldsymbol{u} = \arg\min_{\boldsymbol{w}\in\mathcal{W}} \sum_{t=1}^{T} \ell_t(\boldsymbol{w})$, $\tilde{\ell}_t(\boldsymbol{w}) = \boldsymbol{w}^\intercal \nabla \ell_t(\boldsymbol{w}_t)$, and $\nabla \ell_t(\boldsymbol{w}_t)$ is the gradient of ℓ_t evaluated at $\boldsymbol{w}_t$. Instead of having to optimize the complicated ℓ_t we can now run our algorithms on the linear $\tilde{\ell}_t$. For example, we could now run Online Gradient Descent (Zinkevich, 2003), which has a running time of order $O(dT)$, and obtain a regret bound of $O(G\sqrt{T})$, where G is an upper bound on the L_2 norm of $\nabla \ell_t(\boldsymbol{w}_t)$.

For many loss functions running Online Gradient Descent or a related algorithm often gives satisfying guarantees. However, for online portfolio selection assuming a bound on $\nabla \ell_t(\boldsymbol{w}_t) = -\frac{\boldsymbol{x}_t}{\boldsymbol{w}_t^\intercal \boldsymbol{x}_t}$ involves making assumptions on either $\boldsymbol{x}_t$ or $\boldsymbol{w}_t$. This means that bounding the gradients is very restrictive: we either need to (i) assume that $\boldsymbol{x}_t$ is lower bounded i.e. the asset prices do not fluctuate too rapidly, which defeats the purpose of using adversarial online learning; or (ii) we need to allocate a minimum amount of capital $w_{t,i} \geq \alpha$ to each asset, which means we cannot drop any poorly performing assets from our portfolio. To see how these assumptions affect the gradient suppose that we manage two assets and that up to round t the first asset has been performing poorly compared to the second asset. This means that we would want to put (almost) all of our money on the second asset to maximize revenue. Suppose that in round t the asset prices fluctuate rapidly: in round t the first asset remains constant, i.e. $x_{t,1} = 1$, but the second asset loses all of its value, i.e. $x_{t,2} = 0$. The gradient would be $\frac{1}{w_{t,2}}$, which would ruin the regret bound of for example Online Gradient Descent if $w_{t,2}$ is very small, for example of order $O(\frac{1}{T})$. Even when we are willing to assume a bound on the gradient, running Online Gradient Descent gives unsatisfactory results as the regret is $O(\sqrt{T})$, which is far from optimal. For online portfolio selection and other problems with exp-concave losses Online Newton Step (Hazan et al., 2007) often has better regret at the cost of increased running time, which is of order $O(d^3T)$. However, with Online Newton Step we still need a bound on the gradient as its regret bound is of order $O(Gd\ln(T))$, making the algorithm too restrictive. Because of these issues with standard algorithms, online portfolio selection is a challenging research area, as illustrated by the open problem in Chapter 7.

In Chapter 2 we will provide a unifying view of several algorithms in the Online

Convex Optimization setting by viewing them as special cases of (Continuous) Exponential Weights. This unified view leads to a straightforward analysis of various algorithms in Online Convex Optimization, including Exponentiated Gradient Plus-Minus (Kivinen and Warmuth, 1997), Online Mirror Descent (Beck and Teboulle, 2003), and Online Newton Step (Hazan et al., 2007).

1.2.1 Bandit Convex Optimization

The Bandit Convex Optimization setting (Flaxman et al., 2005; Kleinberg, 2005) is a more difficult version of the Online Convex Optimization setting. In the Bandit Convex Optimization setting, rather than seeing the loss function ℓ_t, the learner only observes the loss function evaluated at his prediction, $\ell_t(\boldsymbol{w}_t)$. This significantly hinders the learner, as there is less information to improve his predictions for the next round. An application of the Bandit Convex Optimization setting is online auctions (Kleinberg and Leighton, 2003). In online auctions the learner has to set a price for products he wants to sell. Unfortunately the price at which people are willing to buy the learner's products is unknown, so he will have to guess a price and infer information based on whether or not people are buying the product at the guessed price. The learner's goal is to maximize his revenue, but the buyers are trying to get the lowest price, which means that learner could be facing an adversarial environment: a perfect place to apply ideas from Online Learning. Other applications of the Bandit Convex Optimization setting include recommendation systems and the online shortest path problem (see for example Hazan et al. (2016)).

The most straightforward instance of the Bandit Convex Optimization setting is when the losses are linear, i.e. $\ell_t(\boldsymbol{w}) = \boldsymbol{w}^\intercal \boldsymbol{g}_t$, where $\boldsymbol{g}_t \in \mathbb{R}^d$ is a loss vector. To still be able to update the predictions the learner will estimate the loss vector, which the learner will do by randomizing his predictions. Since the learner randomizes his predictions the guarantees in the Bandit Convex Optimization setting are about the expected regret, where the expectation is with respect to the randomness of the learner. To see how the learner estimates the loss vectors with the use of randomisation suppose that the learner plays $\tilde{\boldsymbol{w}}_t = \boldsymbol{w}_t + \varepsilon_t$, where ε_t is sampled from a distribution with mean $\mathbb{E}[\varepsilon_t] = \mathbf{0}$ and covariance matrix $\mathbb{E}[\varepsilon_t \varepsilon_t^\intercal] = \Sigma$. This means that the learner observes $\tilde{\boldsymbol{w}}_t^\intercal \boldsymbol{g}_t$, which in expectation is equivalent to $\boldsymbol{w}_t^\intercal \boldsymbol{g}_t$ because $\mathbb{E}[\varepsilon_t] = \mathbf{0}$. To estimate $\boldsymbol{g}_t$ the learner can now use $\hat{\boldsymbol{g}}_t = \Sigma^{-1} \varepsilon_t \tilde{\boldsymbol{w}}_t^\intercal \boldsymbol{g}_t$. If we take the expectation of $\hat{\boldsymbol{g}}_t$ we can see that $\hat{\boldsymbol{g}}_t$ is an unbiased estimate of $\boldsymbol{g}_t$:

$$\mathbb{E}[\hat{\boldsymbol{g}}_t] = \mathbb{E}\left[\Sigma^{-1} \varepsilon_t (\boldsymbol{w}_t + \varepsilon_t)^\intercal \boldsymbol{g}_t\right] = \Sigma^{-1}\, \mathbb{E}\left[\varepsilon_t \varepsilon_t^\intercal\right] \boldsymbol{g}_t = \boldsymbol{g}_t.$$

We can use this to say something about the expected regret because $\mathbb{E}\left[\tilde{\boldsymbol{w}}_t^\intercal \boldsymbol{g}_t\right] = \boldsymbol{w}_t^\intercal \boldsymbol{g}_t = \mathbb{E}\left[\boldsymbol{w}_t^\intercal \hat{\boldsymbol{g}}_t\right]$, where the expected regret is defined as $\mathbb{E}\left[\sum_{t=1}^T \tilde{\boldsymbol{w}}_t^\intercal \boldsymbol{g}_t\right] -$

$\min_{\boldsymbol{u}\in\mathcal{W}} \mathbb{E}\left[\sum_{t=1}^T \boldsymbol{u}^\intercal \boldsymbol{g}_t\right]$. Algorithms in bandit information setting often suffer an additional factor d regret compared to their counterparts in the full-information setting. One of the most well-known algorithm in the Bandit Convex Optimization setting is SCRiBLe (Abernethy et al., 2012), which achieves an $O(d^{3/2}\sqrt{T})$ expected regret bound with linear losses.

In several chapters we will provide new algorithms or analysis of algorithms in the Bandit Convex Optimization setting. In Chapter 2 we use Continuous Exponential Weights to sample $\tilde{\boldsymbol{w}}_t$. As already observed by Bubeck and Eldan (2015), this improves the regret of SCRiBLe by a factor of $d^{1/2}$. As we mentioned before, in Chapter 4 we provide several Bandit Convex Optimization algorithms that are adaptive to the norm of the offline optimizer.

1.3 Online Multiclass Classification

Another setting in Online Learning is the Online Multiclass Classification setting. In each round t in the Online Multiclass Classification setting the learner has to predict the true label y_t out of N possible labels given some extra information $\boldsymbol{x}_t \in \mathbb{R}^d$. An example of this setting is the weather forecasting example. The learner might want to predict whether it is going to rain or not and he probably has some extra information such as humidity or barometric pressure. As in the Online Convex Optimization setting there is a distinction between the full-information setting, in which the learner gets to see the true label, and the bandit setting, in which the learner only sees whether his prediction was correct or not. The loss function in both the full-information and bandit settings is the zero-one loss: $\ell_t(\hat{y}_t) = \mathbb{1}[\hat{y}_t \neq y_t]$, where $\hat{y}_t$ is the prediction of the learner and $\mathbb{1}$ is the indicator function. The goal of both settings is to control the number of mistakes the learner makes i.e. $\sum_{t=1}^T \ell_t(\hat{y}_t)$.

1.3.1 Full-Information

We start by introducing the full-information multiclass classification setting. Since the zero-one loss is a non-convex loss the standard approach is to make use of a surrogate loss, which is a convex upper bound on the zero-one loss. In the case where there are only two possible labels the surrogate loss is a function of $z_t = y_t \boldsymbol{w}_t^\intercal \boldsymbol{x}_t$, where $y_t \in \{-1, 1\}$. The learner predicts with $\hat{y}_t = \operatorname{sign}(\boldsymbol{w}_t^\intercal \boldsymbol{x}_t)$ (if $\boldsymbol{w}_t^\intercal \boldsymbol{x}_t = 0$ the learner can arbitrarily pick -1 or 1). One of the most well-known surrogate losses is the hinge loss: $\tilde{\ell}(z) = \max\{1 - z\}$. Another well known

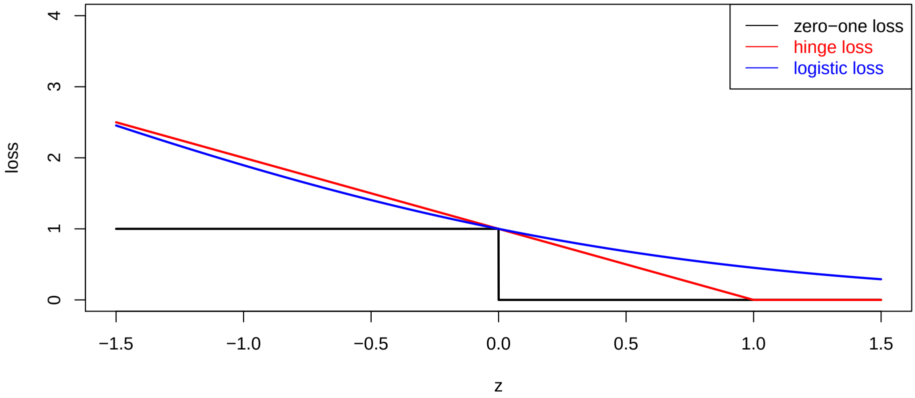


Figure 1.1: A depiction of the zero-one loss and various surrogate losses.

surrogate loss function is the logistic loss, $\tilde{\ell}(z) = \log_2(1 + \exp(z))$,[2] which is used for logistic regression. In Figure 1.1 we can see how the hinge loss and the logistic loss are convex upper bounds for the zero-one loss.

Below we will provide a standard approach to controlling the number of mistakes the learner makes with the use of surrogate losses. The analysis of the well-known Perceptron (Rosenblatt, 1958), which uses the hinge loss as the surrogate loss, as well as many other algorithms follows a similar approach. We start the analysis by bounding the zero-one loss in terms of the surrogate loss:

$$\sum_{t=1}^{T} \ell_t(\hat{y}_t) \leq \sum_{t=1}^{T} \tilde{\ell}(z_t). \tag{1.3.1}$$

Now, we will slightly change notation and write $\tilde{\ell}_t(\boldsymbol{w}) = \tilde{\ell}(y_t \boldsymbol{w}^\intercal \boldsymbol{x}_t)$. In the next step we will add and subtract the surrogate loss again, but now evaluated at the offline optimizer $\boldsymbol{u} \in \arg\min_{\{\boldsymbol{w} \in \mathcal{W}\}} \sum_{t=1}^{T} \tilde{\ell}_t(\boldsymbol{w})$:

$$\sum_{t=1}^{T} \ell_t(\hat{y}_t) \leq \sum_{t=1}^{T} \tilde{\ell}_t(\boldsymbol{w}_t) = \sum_{t=1}^{T} \left(\tilde{\ell}_t(\boldsymbol{w}_t) - \tilde{\ell}_t(\boldsymbol{u}) + \tilde{\ell}_t(\boldsymbol{u})\right).$$

We now almost have a guarantee on the number of mistakes we make, we only need to control $\sum_{t=1}^{T} \left(\tilde{\ell}_t(\boldsymbol{w}_t) - \tilde{\ell}_t(\boldsymbol{u})\right)$. Fortunately we can use various tools from Online Convex Optimization, for example Continuous Exponential Weights

[2] The logarithm has base 2 because with base 2 at $z = 0$ the surrogate loss is equivalent to the zero-one loss.

or Online Gradient Descent, to guarantee that $\sum_{t=1}^T \left(\tilde{\ell}_t(\boldsymbol{w}_t) - \tilde{\ell}_t(\boldsymbol{u})\right)$ is suitably bounded. As we have seen before, each of these Online Convex Optimization algorithms have their own advantages and disadvantages: Continuous Exponential Weights can be slow but has good guarantees and while Online Gradient Descent is very fast it may not have optimal guarantees for some (surrogate) loss functions.

In the end, the guarantee of this type of classifiers is of the form

$$\tilde{\mathcal{R}}_T = \sum_{t=1}^T \ell_t(\hat{y}_t) - \sum_{t=1}^T \tilde{\ell}_t(\boldsymbol{u}).$$

We will refer to $\tilde{\mathcal{R}}_T$ as the surrogate regret. In the worst-case, the Perceptron, which uses Online Gradient Descent to optimize the surrogate loss, has $O(\sqrt{T})$ surrogate regret with respect to the hinge loss. An alternative to the Perceptron is Online Logistic Regression. Foster et al. (2018a) show that if we use continuous Exponential Weights to optimize the logistic loss the surrogate regret is $O(dN \log(T+1))$, with the drawback that continuous Exponential Weights on the logistic loss has running time $O(\max\{dN, T\}^{12}T)$.

Even though more sophisticated versions of the analysis above exist, many algorithms in the Online Multiclass Classification setting roughly follow the same approach. In Chapter 6 we will introduce a new approach that provides a randomized linear time algorithm with $O(K)$ expected surrogate regret, where the expectation is with respect to the learner's randomness. In particular, we will exploit the gap between the zero-one loss and the surrogate loss from equation 1.3.1. As can be seen in Figure 1.1 this upper bound is wasteful for many values of z as the gap between the zero-one loss and the surrogate loss can be quite substantial. By exploiting the aforementioned gap we are able to significantly reduce the impact of $\sum_{t=1}^T \tilde{\ell}_t(\boldsymbol{w}_t) - \tilde{\ell}_t(\boldsymbol{u})$ on the surrogate regret bound, which leads to our new result.

1.3.2 Bandit Information

In the Bandit Multiclass Classification setting (Kakade et al., 2008) the learner only receives $\mathbb{1}[\hat{y}_t \neq y_t]$ as feedback. This means that we can no longer directly use the surrogate loss approach to bound the number of mistakes the learner makes. However, since the learner does know $\mathbb{1}[\hat{y}_t \neq y_t]$ in all rounds, in rounds where $\hat{y}_t = y_t$ the learner also knows y_t. So how does the learner leverage this to guarantee a suitable bound on the number of mistakes?

As in Section 1.2.1 the learner will have to randomize his prediction: $\hat{y}_t \sim q_t$. If we use a technique called importance weighting, which multiplies the surrogate loss by

$\mathbb{1}[\hat{y}_t = y_t]q_t(\hat{y}_t)^{-1}$, the learner can use the weighted surrogate losses to update $\boldsymbol{w}_t$. Note that this means that we only update whenever we have guessed the correct label, which hopefully happens often. In expectation the weighted surrogate losses are equivalent to their full-information counterparts, which means the learner could use the standard techniques from the previous section to provide surrogate regret guarantees.

As in the full-information setting the learner can use algorithms from Online Convex Optimization to optimize the weighted surrogate losses. In the bandit setting the predictions of the learner are randomized so the guarantees are for the expected surrogate regret, where the expectation is with respect to the randomness of the learner. Several authors have proposed polynomial time algorithms that have a $O(N\sqrt{dT\ln(T+1)})$ expected surrogate regret bound (see for example Hazan and Kale (2011); Beygelzimer et al. (2017); Foster et al. (2018a)). In Chapter 6 we will exploit the gap between the surrogate loss and the zero-one loss to provide the first linear time algorithm with $O(N\sqrt{T})$ expected surrogate regret bounds with respect to various surrogate losses. Interestingly, our new algorithm improves upon the expected surrogate regret bound of slower algorithms by a factor of $\sqrt{d\log(T+1)}$, which makes our new algorithm the first answer to the open question of Abernethy and Rakhlin (2009) with an expected surrogate regret bound that does not depend on the dimension of the feature vectors.

1.4 Organisation

The remainder of this dissertation is concerned with various settings and algorithms in Online Learning. In Chapter 2 we show that many algorithms in Online Learning are special cases of Exponential Weights. We also provide a reduction for several adaptive expert algorithms based on Exponential Weights, which recovers Squint (Koolen and Van Erven, 2015), iProd (Koolen and Van Erven, 2015), and Coin Betting for experts (Orabona and Pál, 2016).

Throughout this dissertation we provide several new adaptive algorithms. In Chapter 3 we show how we can adapt to unknown noise in the unconstrained Online Convex Optimization setting, which allows users to choose their privacy requirements without having to disclose them to whoever receives their data. In Chapter 4 we study Bandit Convex Optimization methods that adapt to the norm of the offline optimizer, a topic that has only been studied before for its full-information counterpart. We show that algorithms from the full information setting can be adapted to develop algorithms that adapt to the norm of the offline optimizer for linear bandits. These ideas are then extended to the Bandit Convex Optimization setting by using a new single-point gradient estimator and carefully designed surrogate losses. In Chapter 5 we introduce MetaGrad, which is an algorithm that is adaptive to a broad class of loss functions. We then improve the running time of MetaGrad by applying sketching methods and evaluate the performance of several versions of MetaGrad in numerous experiments.

As we mentioned above, in Chapter 6 we provide a new approach to Online Multiclass Classification based on exploiting the gap between the zero-one loss and a surrogate loss. In the Bandit Multiclass Classification setting we use our new approach to provide the first linear time algorithm with $O(N\sqrt{T})$ surrogate regret. Furthermore, the surrogate regret of this new bandit algorithm is independent of the dimension of the feature vector, contrary to algorithms with similar surrogate regret bounds in the Bandit Multiclass Classification setting.

Finally, in Chapter 7 we pose an open problem which asks for a fast and optimal algorithm for online portfolio selection. We provide an algorithm and the first steps of the analysis which shows that in some special cases this algorithm indeed yields the optimal regret bound.

CHAPTER 2

The Many Faces of Exponential Weights in Online Learning

This chapter is based on: Van der Hoeven, D., Van Erven, T., and Kotłowski, W. (2018). The many faces of exponential weights in online learning. In *Proceedings of the 31st Annual Conference on Learning Theory (COLT)*, pages 2067–2092.[1]

Abstract

A standard introduction to online learning might place Online Gradient Descent at its center and then proceed to develop generalizations and extensions like Online Mirror Descent and second-order methods. Here we explore the alternative approach of putting Exponential Weights (EW) first. We show that many standard methods and their regret bounds then follow as a special case by plugging in suitable surrogate losses and playing the EW posterior mean. For instance, we easily recover Online Gradient Descent by using EW with a Gaussian prior on linearized losses, and, more generally, all instances of Online Mirror Descent based on regular Bregman divergences also correspond to EW with a prior that depends on the mirror map. Furthermore, appropriate quadratic surrogate losses naturally give rise to Online Gradient Descent for strongly convex losses and to Online Newton Step. We further interpret several recent adaptive methods (iProd, Squint, and a variation of Coin Betting for experts) as a series of closely related reductions to exp-concave surrogate losses that are then handled by Exponential Weights. Finally, a benefit of our EW interpretation is that it opens up the possibility of sampling from the EW posterior distribution instead of playing the mean. As already observed by Bubeck and Eldan (2015), this recovers the best-known rate in Online Bandit Linear Optimization.

[1] The author of this dissertation performed the following tasks: co-deriving the theoretical results and co-writing the paper.

2.1 Introduction

Exponential Weights (EW) (Vovk, 1990; Littlestone and Warmuth, 1994) is a method for keeping track of uncertainty about the best action in sequential prediction tasks. It is most commonly considered for a finite number of actions in the prediction with expert advice setting, where each of the actions corresponds to following the advice of one of a finite number of experts, and in this context it is asymptotically minimax optimal (Cesa-Bianchi and Lugosi, 2006, Section 2.2). However, in the present work we mostly consider EW on continuous action spaces in the more general setting of Online Convex Optimization (Hazan et al., 2016), where we show that surprisingly many standard methods turn out to be special cases of EW.

EW keeps track of a probability distribution over actions that is updated in each round of the prediction task by multiplying the probability of each action by a factor that is exponentially decreasing in the action's error or *loss* in that round, and renormalizing. This type of update is quite flexible: by assigning appropriate surrogate losses to the actions, it covers any kind of multiplicative probability updates, including, for instance, those of the Prod algorithm (Cesa-Bianchi et al., 2007). For best performance, losses often need to be scaled by a positive parameter called the learning rate, and the algorithm may also be biased towards particular actions by the choice of its initial distribution, which is called the prior. For continuous sets of actions, efficient implementations of EW are often restricted to conjugate priors for which the EW distribution can be analytically computed, but sampling approximations based on random walks can also provide appealing trade-offs between computational complexity and prediction accuracy, even for a single random walk step per round (Narayanan and Rakhlin, 2017; Kalai and Vempala, 2002).

The usual presentation of Online Convex Optimization would introduce EW as a special case of Mirror Descent (MD) or Follow-the-Regularized-Leader (FTRL) with the Kullback-Leibler divergence as the regularizer. However, here we turn this view on its head and show that all instances of MD based on regular Bregman divergences (Banerjee et al., 2005) in fact correspond to EW on a continuous set of actions (Section 2.3.3). In particular, Gradient Descent (GD) comes from using a Gaussian prior on linearized losses (Section 2.3.2), which is striking because GD has been contrasted with the Exponentiated Gradient Plus-Minus algorithm (Kivinen and Warmuth, 1997) that is readily seen to be an instance of EW (Section 2.3.1). In addition, the unnormalized relative entropy regularizer (Helmbold and Warmuth, 2009), which is normally considered a generalization of EW, turns out to be a special case of EW as well for a multivariate Poisson prior (Section 2.3.3). Furthermore, in Section 2.4 we show that running EW on suitable quadratic approximations of the losses recovers Gradient Descent for strongly convex losses (Hazan et al., 2007) and, as already observed by van Erven and Koolen (2016), Online Newton Step (Hazan et al., 2007). The Vovk-Azoury-Warmuth forecaster would also be an example of running EW on quadratic losses, but we refer to (Vovk, 2001) for its analysis, which requires a generalized proof technique (see also the discussion by Orabona et al. (2015a)). We do consider the recent adaptive iProd, Squint and Coin Betting methods of Koolen and Van Erven (2015); Orabona and Pál (2016), which learn the optimal learning rate for prediction with expert advice, and show that these may also be viewed as running EW after a reduction of the original prediction task to various closely related surrogate tasks in which the learning rate is just one of the parameters that does not need to be treated specially (Section 2.5). Finally, in the context of Bandit Linear Optimization, the SCRiBLe method (Abernethy et al., 2008) may be viewed as an approximation to EW, and an application of EW outlined by Bubeck and Eldan (2015) achieves the best-known rate (we provide the technical details they omit in Section 2.6).

Related Work The diverse applications of EW on a finite number of actions range, for instance, from boosting (Freund and Schapire, 1997) to differential privacy (Dwork and Roth, 2014) to multi-armed bandits (Auer et al., 2002), and many algorithms in computer science can be viewed as special cases of EW (Arora et al., 2012). EW has also been considered for continuous sets of actions, often in the context of universal coding in information theory, where the goal is to sequentially compress a sequence of symbols. In this case, actions parametrize a set of probability distributions and the loss of an action is the logarithmic loss for the corresponding probability distribution on the symbol that is being compressed (Cesa-Bianchi and Lugosi, 2006, Chapter 9). EW (with learning rate 1) then simplifies to Bayesian probability updating. The choice of prior has received much attention in this literature, with Jeffreys' prior being shown to be asymptotically minimax optimal for exponential families with parameters restricted to suitable bounded sets (Grünwald, 2007, Chapter 8). Without parameter restrictions, Jeffreys' prior is still minimax optimal up to constants for the Bernoulli and multinomial models (Krichevsky and Trofimov, 1981; Xie and Barron, 2000). Several applications to other losses are also closely related to the log loss: Online Ridge Regression corresponds to EW on the squared loss, which matches the log loss for Gaussian distributions; and Cover's method for portfolio selection (Cover, 1991), which is EW on Cover's loss, may be interpreted as learning a mixture model under the log loss (Orseau et al., 2017). In general, continuous EW is not restricted to the log loss, however, and has been considered e.g. for general convex losses (Dick et al., 2014)

Input: a convex set of distributions $\mathcal{P}$ over $\boldsymbol{w}$, a prior $P_1 \in \mathcal{P}$ and learning rates $\eta_1 \geq \eta_2 \geq \cdots \geq \eta_T > 0$	
Lazy Exponential Weights	Greedy Exponential Weights
$\tilde{P}_{t+1} = \underset{P}{\arg\min}\ \mathbb{E}_P\left[\sum_{s=1}^t f_s(\boldsymbol{w})\right] + \frac{1}{\eta_t}\operatorname{KL}(P\\|P_1)$ $P_{t+1} = \underset{P\in\mathcal{P}}{\arg\min}\ \operatorname{KL}(P\\|\tilde{P}_{t+1})$	$\tilde{P}_{t+1} = \underset{P}{\arg\min}\ \mathbb{E}_P[f_t(\boldsymbol{w})] + \frac{1}{\eta_t}\operatorname{KL}(P\\|P_t)$ $P_{t+1} = \underset{P\in\mathcal{P}}{\arg\min}\ \operatorname{KL}(P\\|\tilde{P}_{t+1})$

Figure 2.1: The lazy and greedy versions of Exponential Weights

or as a computationally inefficient gold standard for exp-concave losses (Hazan et al., 2007).

2.2 Exponential Weights

In Online Convex Optimization (OCO) (Shalev-Shwartz, 2011; Hazan et al., 2016) a learner repeatedly chooses actions $\boldsymbol{w}_t$ from a convex set $\mathcal{W} \subseteq \mathbb{R}^d$ during rounds $t = 1, \ldots, T$, and suffers losses $f_t(\boldsymbol{w}_t)$, where $f_t : \mathcal{W} \to \mathbb{R}$ is a convex function. The learner's goal is to achieve small *regret* $\mathcal{R}_T(\boldsymbol{u}) = \sum_{t=1}^T f_t(\boldsymbol{w}_t) - \sum_{t=1}^T f_t(\boldsymbol{u})$ with respect to any comparator action $\boldsymbol{u} \in \mathcal{W}$, which measures the difference between the cumulative loss of the learner and the cumulative loss it could have achieved by playing the oracle action $\boldsymbol{u}$ from the start. We will assume the domain of the losses f_t is extended from $\mathcal{W}$ to $\mathbb{R}^d$ with convexity of f_t being preserved. This comes without loss of generality as one can always set $f_t(\boldsymbol{w}) = \infty$ outside $\mathcal{W}$, but we will use more natural and straightforward extensions throughout the chapter (e.g. when the f_t are linear or quadratic functions).

The central topic of this work is the Exponential Weights (EW) algorithm, which keeps track of uncertainty over actions expressed by a distribution P_t and comes in the two flavors shown in Figure 2.1 (our naming follows Zinkevich (2004)), where we let $\operatorname{KL}(P\|Q) = \mathbb{E}_P\left[\ln \frac{\mathrm{d}P}{\mathrm{d}Q}\right]$ denote the Kullback-Leibler (KL) divergence between distributions P and Q. The algorithm gets its name from the distributions $\tilde{P}_t$, whose densities have the following exponential forms:

$$\mathrm{d}\tilde{P}_{t+1}(\boldsymbol{w}) = \frac{e^{-\eta_t \sum_{s=1}^t f_s(\boldsymbol{w})}\,\mathrm{d}P_1(\boldsymbol{w})}{\int e^{-\eta_t \sum_{s=1}^t f_s(\boldsymbol{w})}\,\mathrm{d}P_1(\boldsymbol{w})} \qquad \text{(lazy EW)} \qquad (2.2.1)$$

$$\mathrm{d}\tilde{P}_{t+1}(\boldsymbol{w}) = \frac{e^{-\eta_t f_t(\boldsymbol{w})}\,\mathrm{d}P_t(\boldsymbol{w})}{\int e^{-\eta_t f_t(\boldsymbol{w})}\,\mathrm{d}P_t(\boldsymbol{w})} \qquad \text{(greedy EW).} \qquad (2.2.2)$$

In the case that $\mathcal{P}$ contains all possible distributions over $\mathbb{R}^d$ (for which the projection step becomes void) and the *learning rates* η_t are constant $\eta_1 = \cdots = \eta_T = \eta$, both versions of EW are equivalent. In general they differ, and enjoy the following regret bounds with respect to a potentially randomized comparator drawn from a comparator distribution Q, which follow from a standard MD analysis (Hazan et al., 2016) and a reformulation of the standard FTRL analysis that works for distributions P_t on continuous spaces, which cannot be expressed as the finite-dimensional vectors that are usually assumed (the proof details are in Section 2.8):

Lemma 1 (EW Regret)**.** *Suppose that* $\eta_1 \geq \eta_2 \geq \ldots \geq \eta_T > 0$*, and that the minima that define* $\tilde{P}_t$ *and* P_t *are uniquely achieved. Let* $Q \in \mathcal{P}$ *be any comparator distribution such that* $\mathrm{KL}(Q\|\tilde{P}_t) < \infty$ *for all t, let* $\{\boldsymbol{w}_t \in \mathcal{W}\}_{t=1}^T$ *be the actions of any learner, and define* $\eta_0 \overset{\text{def}}{=} \eta_1$*. Then lazy EW satisfies*

$$\mathbb{E}_{\boldsymbol{u}\sim Q}[\mathcal{R}(\boldsymbol{u})] \leq \frac{1}{\eta_T}\mathrm{KL}(Q\|P_1) + \sum_{t=1}^{T}\Big\{ \underbrace{f_t(\boldsymbol{w}_t) + \frac{1}{\eta_{t-1}}\ln \mathbb{E}_{P_t(\boldsymbol{w})}\left[e^{-\eta_{t-1}f_t(\boldsymbol{w})}\right]}_{\text{“mixability gap”}} \Big\} \qquad (2.2.3)$$

and greedy EW satisfies

$$\mathbb{E}_{\boldsymbol{u}\sim Q}[\mathcal{R}(\boldsymbol{u})] \leq \frac{1}{\eta_1}\mathrm{KL}(Q\|P_1) + \left(\frac{1}{\eta_T} - \frac{1}{\eta_1}\right)\max_{t=2,\ldots,T}\mathrm{KL}(Q\|P_t) + \sum_{t=1}^{T}\Big\{ \underbrace{f_t(\boldsymbol{w}_t) + \frac{1}{\eta_t}\ln \mathbb{E}_{P_t(\boldsymbol{w})}\left[e^{-\eta_t f_t(\boldsymbol{w})}\right]}_{\text{“mixability gap”}} \Big\}. \qquad (2.2.4)$$

While the predictions $\boldsymbol{w}_t$ in Lemma 1 are arbitrary actions from $\mathcal{W}$, one always chooses $\boldsymbol{w}_t$ to be some function of P_t. A general mapping from P_t to $\boldsymbol{w}_t$ is called a *substitution function* (Vovk, 2001) and is usually designed to give the best bound on the mixability gap in trial t. Throughout the chapter, we will use the mean $\boldsymbol{w}_t = \mathbb{E}_{P_t}[\boldsymbol{w}]$ as our substitution function, which is a typical choice, although alternatives may be better in specific cases (Vovk, 2001). To ensure that $\boldsymbol{w}_t \in \mathcal{W}$, we will also generally assume that $\mathcal{P} = \{P : \mathbb{E}_P[\boldsymbol{w}] \in \mathcal{W}\}$, which is convex.

Bounding the mixability gap is a crucial part of the regret analysis of EW (Vovk, 2001; De Rooij et al., 2014). In the special case that the losses are *α-exp-concave* for $\alpha > 0$ (i.e. if $e^{-\alpha f(\boldsymbol{w})}$ is concave), the mixability gap for $\eta_t \leq \alpha$ is at most 0. This happens in the following example.

Example 1 (The Krichevsky-Trofimov Estimator)**.** *Let* $\mathcal{W} = [0, 1]$ *and let the loss function be the log loss:* $f_t(w) = -x_t \ln(w) - (1 - x_t)\ln(1 - w)$*, where*

$x_t \in \{0,1\}$. A standard algorithm in this case is the Krichevsky-Trofimov forecaster $w_t = (\sum_{s=1}^{t-1} x_s + \frac{1}{2})/t$ (Cesa-Bianchi and Lugosi, 2006, Chapter 9), which is is well known to be the mean $w_t = \mathbb{E}_{P_t}[w]$ of non-projected EW with a $\beta(\frac{1}{2},\frac{1}{2})$ prior and a fixed learning rate $\eta_t = 1$. For the log loss, the mixability gap is 0. To bound the remaining terms in Lemma 1, we choose $Q = P_{T+1}$, which gives:

$$\begin{aligned}
\sum_{t=1}^T f_t(w_t) &\le \mathbb{E}_{P_{T+1}(w)}\left[\sum_{t=1}^T f_t(w)\right] + \mathrm{KL}(P_{T+1}\|P_1)\\
&= -\ln \mathbb{E}_{P_1(w)}[w^{\sum_{t=1}^T x_t}(1-w)^{T-\sum_{t=1}^T x_t}]\\
&\le -\ln \max_w \left\{w^{\sum_{t=1}^T x_t}(1-w)^{T-\sum_{t=1}^T x_t}\right\} + \ln(2\sqrt{T})\\
&= \min_w \sum_{t=1}^T f_t(w) + \ln(2\sqrt{T}),
\end{aligned}$$

where the last inequality holds by (Cesa-Bianchi and Lugosi, 2006, Lemma 9.3).

For most regret bounds derived from Lemma 1 the structure of the proof remains the same: we need both a bound on the mixability gap, and a choice for Q for which the expected loss under Q together with $\mathrm{KL}(Q\|P_1)$ can be related to the loss of a deterministic comparator.

2.3 Linearized Losses

A standard approach in OCO is to lower-bound the convex losses f_t by their tangent at $\boldsymbol{w}_t$, which leads to the following upper bound on the regret in terms of the linearized surrogate losses $\ell_t(\boldsymbol{w}) = \langle \boldsymbol{w}, \boldsymbol{g}_t\rangle$, where $\boldsymbol{g}_t = \nabla f_t(\boldsymbol{w}_t) = (g_{t,1},\ldots,g_{t,d})^\intercal$ is the gradient at $\boldsymbol{w}_t$:

$$\sum_{t=1}^T (f_t(\boldsymbol{w}_t) - f_t(\boldsymbol{u})) \le \sum_{t=1}^T \big(\ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{u})\big). \tag{2.3.1}$$

2.3.1 Exponentiated Gradient Plus-Minus as Exponential Weights

The Exponentiated Gradient Plus-Minus (EG$^\pm$) algorithm (Kivinen and Warmuth, 1997) starts with weight vectors $\boldsymbol{w}_t^- = \boldsymbol{w}_t^+ = (1/d,\ldots,1/d) \in \mathbb{R}^d$, which are updated according to

$$
\begin{aligned}
w^+_{t+1,i} &= \frac{w^+_{t,i} e^{-\eta_t \langle \boldsymbol{e}_i, \boldsymbol{g}_t\rangle}}{\sum_{j=1}^d \big(w^+_{t,j} e^{-\eta_t \langle \boldsymbol{e}_j, \boldsymbol{g}_t\rangle} + w^-_{t,j} e^{\eta_t \langle \boldsymbol{e}_j, \boldsymbol{g}_t\rangle}\big)},\\
w^-_{t+1,i} &= \frac{w^-_{t,i} e^{\eta_t \langle \boldsymbol{e}_i, \boldsymbol{g}_t\rangle}}{\sum_{j=1}^d \big(w^+_{t,j} e^{-\eta_t \langle \boldsymbol{e}_j \boldsymbol{g}_t\rangle} + w^-_{t,j} e^{\eta_t \langle \boldsymbol{e}_j, \boldsymbol{g}_t\rangle}\big)},
\end{aligned}
$$

and predicts by $\boldsymbol{w}_t \in \{\boldsymbol{w} : \|\boldsymbol{w}\|_1 \leq 1\}$ with components $w_{t,i} = w^+_{t,i} - w^-_{t,i}$.

This is readily seen to be the mean $\boldsymbol{w}_t = \mathbb{E}_{P_t}[\boldsymbol{w}]$ of EW (without projections) on the linearized losses (2.3.1) with a discrete uniform prior P_1 on the standard basis vectors $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d$, which form the corners of the probability simplex, and their negations $-\boldsymbol{e}_1, \ldots, -\boldsymbol{e}_d$. The regular Exponentiated Gradient algorithm is recovered by initializing $\boldsymbol{w}^-_1 = (0, \ldots, 0)$, which corresponds to placing prior mass only on $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d$. Kivinen and Warmuth (1997) also extend the algorithm to scale up the domain by a factor $M > 0$, which corresponds to a discrete prior on $M\boldsymbol{e}_1, \ldots, M\boldsymbol{e}_d$ for EG and also on $-M\boldsymbol{e}_1, \ldots, -M\boldsymbol{e}_d$ for EG$^\pm$. Hence we may analyze these methods using Lemma 1, which leads to the following regret bound for EG$^\pm$ (see Section 2.9):

Theorem 1 (EG$^\pm$ as EW)**.** *Suppose $\|\boldsymbol{g}_t\|_\infty \leq G$ for all t. Then the regret of EG$^\pm$ for scale factor $M > 0$ and constant learning rate $\eta_t = \sqrt{\frac{2\ln(2d)}{TM^2G^2}}$ satisfies*

$$\mathcal{R}_T(\boldsymbol{u}) \leq GM\sqrt{2T\ln(2d)} \qquad \textit{for all } \boldsymbol{u} \textit{ such that } \|\boldsymbol{u}\|_1 \leq M.$$

2.3.2 Gradient Descent as Exponential Weights

The prior of EG$^\pm$ is adapted to comparators $\boldsymbol{u}$ with small L_1-norm. How do we change the prior to favor comparators with small L_2-norm? A natural and computationally efficient choice is to use a Gaussian prior $P_1 = \mathcal{N}(\boldsymbol{w}_1, \sigma^2\boldsymbol{I})$, where $\boldsymbol{I}$ is the identity matrix. Then it turns out that all EW distributions P_t are Gaussian with the Gradient Descent (GD) predictions as their means:

Theorem 2 (Gradient Descent as EW)**.** *Let $\mathcal{P} = \{P : \mathbb{E}_P[\boldsymbol{w}] \in \mathcal{W}\}$. Then, for Gaussian prior $P_1(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}_1, \sigma^2\boldsymbol{I})$, lazy and greedy EW with learning rates η_t on the linearized losses* (2.3.1) *yield Gaussian distributions $\tilde{P}_t = \mathcal{N}(\tilde{\boldsymbol{w}}_t, \sigma^2\boldsymbol{I})$ and $P_t = \mathcal{N}(\boldsymbol{w}_t, \sigma^2\boldsymbol{I})$ with the same covariance as the prior. The means $\tilde{\boldsymbol{w}}_t$ and $\boldsymbol{w}_t$ coincide with lazy and greedy GD (Figure 2.2), except that the learning rates in GD*

Input: Convex set $\mathcal{W}$, and learning rates $\eta_1 \geq \eta_2 \geq \ldots \geq \eta_T > 0$	
Lazy Gradient Descent	Greedy Gradient Descent
$\tilde{\boldsymbol{w}}_{t+1} = \boldsymbol{w}_1 - \eta_t \sum_{s=1}^{t} \boldsymbol{g}_s$ $\boldsymbol{w}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{w} \in \mathcal{W}} \frac{1}{2}\|\boldsymbol{w} - \tilde{\boldsymbol{w}}_{t+1}\|_2^2$	$\tilde{\boldsymbol{w}}_{t+1} = \boldsymbol{w}_t - \eta_t \boldsymbol{g}_t$ $\boldsymbol{w}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{w} \in \mathcal{W}} \frac{1}{2}\|\boldsymbol{w} - \tilde{\boldsymbol{w}}_{t+1}\|_2^2$

Figure 2.2: The lazy and greedy versions of Gradient Descent

are scaled to $\sigma^2\eta_t$ by the prior variance σ^2. Moreover, Lemma 1 directly implies:

$$\mathcal{R}_T(\boldsymbol{u}) \;\leq\; \frac{\|\boldsymbol{u} - \boldsymbol{w}_1\|_2^2}{2\sigma^2\eta_T} + \frac{\sigma^2}{2}\sum_{t=1}^{T} \eta_{t-1}\|\boldsymbol{g}_t\|_2^2 \qquad \textit{(lazy GD)}$$

$$\mathcal{R}_T(\boldsymbol{u}) \leq \frac{\max_t \|\boldsymbol{u} - \boldsymbol{w}_t\|_2^2}{2\sigma^2\eta_T} + \frac{\sigma^2}{2}\sum_{t=1}^{T} \eta_t\|\boldsymbol{g}_t\|_2^2 \qquad \textit{(greedy GD).}$$

We note that in this case the parametrization of EW is redundant, because changing the prior variance σ^2 has the same effect on the predictions $\boldsymbol{w}_t$ and the regret bounds as scaling all η_t.

Proof. $\tilde{P}_t = \mathcal{N}(\tilde{\boldsymbol{w}}_t, \sigma^2\boldsymbol{I})$ may be verified analytically from (2.2.1) and (2.2.2). The fact that the projections P_t onto $\mathcal{P}$ preserve Gaussianity with the same covariance matrix is a property of projecting a member of an exponential family onto a set of distributions defined by a convex constraint on their means. (This follows from Lemma 3 in Section 2.10 or see (van Erven and Koolen, 2016, Lemma 9) for the Gaussian case.) The regret bounds follow by taking $Q = \mathcal{N}(\boldsymbol{u}, \sigma^2\boldsymbol{I})$, for which $\mathrm{KL}(Q\|P_t) = \frac{1}{2\sigma^2}\|\boldsymbol{u} - \boldsymbol{w}_t\|_2^2$, and evaluating the mixability gap in closed form. □

2.3.3 Mirror Descent and FTRL as EW

The fact that Gradient Descent is an instance of EW raises the question of whether other instances of MD or FTRL are special cases of EW as well. Let $F^*(\boldsymbol{w}) = \sup_{\boldsymbol{\theta}}\langle \boldsymbol{w}, \boldsymbol{\theta}\rangle - F(\boldsymbol{\theta})$ denote the convex conjugate of F, and let $B_{F^*}(\boldsymbol{u}\|\boldsymbol{w}) = F^*(\boldsymbol{u}) - F^*(\boldsymbol{w}) - \nabla F^*(\boldsymbol{w})^\intercal(\boldsymbol{u} - \boldsymbol{w})$ denote the corresponding Bregman divergence. Then MD and FTRL are defined in Figure 2.3 for Legendre functions $F(\boldsymbol{\theta})$ on $\mathbb{R}^d$ (Cesa-Bianchi and Lugosi, 2006). We consider exponential families that take the form $\mathcal{E} = \{P_{\boldsymbol{\theta}} \mid \mathrm{d}P_{\boldsymbol{\theta}}(\boldsymbol{w}) = e^{\langle \boldsymbol{\theta}, \boldsymbol{w}\rangle - F(\boldsymbol{\theta})}\mathrm{d}K(\boldsymbol{w}), \boldsymbol{\theta} \in \Theta\}$ for a nonnegative *carrier measure* K, cumulant generating function $F(\boldsymbol{\theta}) = \ln \int e^{\langle \boldsymbol{\theta}, \boldsymbol{w}\rangle}\mathrm{d}K(\boldsymbol{w})$ and parameter space $\Theta = \{\boldsymbol{\theta} \mid F(\boldsymbol{\theta}) < \infty\} \subset \mathbb{R}^d$. These are called *regular* if Θ is an

<table>
<tr><td colspan="2"><b>Input:</b> Legendre function F, convex set $\mathcal{W}$, and learning rates $\eta_1 \geq \eta_2 \geq \ldots \geq \eta_T > 0$</td></tr>
<tr><td>FTRL / Lazy Mirror Descent</td><td>Greedy Mirror Descent</td></tr>
<tr><td>$\tilde{\boldsymbol{w}}_{t+1} = \arg\min_{\boldsymbol{w}} \sum_{s=1}^{t} \langle \boldsymbol{w}, \boldsymbol{g}_s \rangle + \frac{1}{\eta_t} B_{F^*}(\boldsymbol{w} \| \boldsymbol{w}_1)$
$\boldsymbol{w}_{t+1} = \arg\min_{\boldsymbol{w} \in \mathcal{W}} B_{F^*}(\boldsymbol{w} \| \tilde{\boldsymbol{w}}_{t+1})$</td><td>$\tilde{\boldsymbol{w}}_{t+1} = \arg\min_{\boldsymbol{w}} \langle \boldsymbol{w}, \boldsymbol{g}_t \rangle + \frac{1}{\eta_t} B_{F^*}(\boldsymbol{w} \| \boldsymbol{w}_t)$
$\boldsymbol{w}_{t+1} = \arg\min_{\boldsymbol{w} \in \mathcal{W}} B_{F^*}(\boldsymbol{w} \| \tilde{\boldsymbol{w}}_{t+1})$</td></tr>
</table>

Figure 2.3: The lazy and greedy versions of Mirror Descent. Lazy MD is usually called FTRL.

open set. We then start with the following relation between MD and EW, which is proved in Section 2.10:

Theorem 3 (Mirror Descent as EW)**.** *Suppose F is the cumulant generating function of a regular exponential family $\mathcal{E}$. Then the lazy and greedy versions of MD predict with the means $\boldsymbol{w}_t = \mathbb{E}_{P_t}[\boldsymbol{w}]$ of lazy and greedy EW on the linearized losses* (2.3.1) *with the same η_t, prior $P_{\boldsymbol{\theta}_1}$ for $\boldsymbol{\theta}_1 = \nabla F^*(\boldsymbol{w}_1)$ and $\mathcal{P} = \{P : \mathbb{E}_P[\boldsymbol{w}] \in \mathcal{W}\}$.*

To answer our question, we therefore need to know whether, for any Legendre function F^*, the convex conjugate $(F^*)^* = F$ corresponds to the cumulant generating function of some exponential family, which means we need to find a corresponding carrier K. Nonconstructive existence of such K has been studied by Banerjee et al. (2005, Theorem 6), who show that there is in fact a bijection between *regular* Bregman divergences and regular exponential families, where regular Bregman divergences based on F^* are defined to be those for which $e^{F(\boldsymbol{\theta})}$ is a continuous, exponentially convex[2] function such that $\Theta = \{\boldsymbol{\theta} \mid F(\boldsymbol{\theta}) < \infty\}$ is open and F is strictly convex.

There is no easy general procedure to construct the corresponding carrier K for a given Legendre function F^*. However, for the Gradient Descent example from Section 2.3.2 we see that $F^*(\boldsymbol{w}) = \frac{1}{2\sigma^2}\|\boldsymbol{w}\|_2^2$ is the convex conjugate of the cumulant generating function for $K(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$. We also give another example:

Example 2 (Unnormalized Relative Entropy)**.** *Consider MD with regularization based on the* unnormalized relative entropy $B_{F^*}(\boldsymbol{w}\|\boldsymbol{u}) = \sum_{i=1}^d (w_i \ln \frac{w_i}{u_i} - w_i + u_i)$ *for $\boldsymbol{w}, \boldsymbol{u} \in \mathbb{R}_+^d$, which is the Bregman divergence generated by $F^*(\boldsymbol{w}) = \sum_{i=1}^d w_i(\ln(w_i) - 1)$ (Cesa-Bianchi and Lugosi, 2006). We have $F(\boldsymbol{\theta}) = \sum_{i=1}^d e^{\theta_i}$. Interestingly, the exponential family with this cumulant generating function is the*

[2] Exponentially convex in the sense of Banerjee et al. (2005, Definition 7).

Input: Convex set $\mathcal{W}$ and learning rate $\eta > 0$	
Lazy EW Gaussian prior quadratic loss	Greedy EW Gaussian prior quadratic loss
$\Sigma_{t+1}^{-1} = \Sigma_t^{-1} + \eta \boldsymbol{M}_t$ $\tilde{\boldsymbol{w}}_{t+1} = \tilde{\boldsymbol{w}}_t - \eta \Sigma_{t+1} \boldsymbol{g}_t$ $\boldsymbol{w}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{w} \in \mathcal{W}} (\boldsymbol{w} - \tilde{\boldsymbol{w}}_{t+1})^\intercal \Sigma_{t+1}^{-1} (\boldsymbol{w} - \tilde{\boldsymbol{w}}_{t+1})$	$\Sigma_{t+1}^{-1} = \Sigma_t^{-1} + \eta \boldsymbol{M}_t$ $\tilde{\boldsymbol{w}}_{t+1} = \boldsymbol{w}_t - \eta \Sigma_{t+1} \boldsymbol{g}_t$ $\boldsymbol{w}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{w} \in \mathcal{W}} (\boldsymbol{w} - \tilde{\boldsymbol{w}}_{t+1})^\intercal \Sigma_{t+1}^{-1} (\boldsymbol{w} - \tilde{\boldsymbol{w}}_{t+1})$

Figure 2.4: The means and covariances of both versions of Exponential Weights with a multivariate normal prior and a constant learning rate η run on the quadratic surrogate loss (2.4.1)

set of Poisson distributions, extended i.i.d. to d dimensions. To see this for $d = 1$, note that if we start with the usual parametrization of Poisson, we have

$$P_\lambda(w) = e^{-\lambda} \frac{\lambda^w}{w!} = \frac{1}{w!} e^{-\lambda + w \ln \lambda} \qquad \text{on } w \in \{0, 1, 2, \ldots\},$$

for which the natural parameter is $\theta = \ln \lambda$ and we see that the cumulant generating function is $F(\theta) = \lambda = e^\theta$. Thus, EW with the product prior $P_1(\boldsymbol{w}) = \prod_{i=1}^d P_{\lambda_i}(w_i)$ corresponds to MD with unnormalized relative entropy, where we need to set $(\lambda_1, \ldots, \lambda_d) = \exp(\boldsymbol{\theta}_1) = \exp(\nabla F^*(\boldsymbol{w}_1)) = \boldsymbol{w}_1$ to match the starting point of MD: $\mathbb{E}_{P_1}[\boldsymbol{w}] = \boldsymbol{w}_1$. Note that in this case the EW distributions P_t are discrete.

2.4 Quadratic Losses

In this section we assume that the losses f_t satisfy quadratic lower bounds:

$$f_t(\boldsymbol{w}) - f_t(\boldsymbol{w}_t) \geq \langle \boldsymbol{w} - \boldsymbol{w}_t, \boldsymbol{g}_t \rangle + \frac{1}{2} (\boldsymbol{w} - \boldsymbol{w}_t)^\intercal \boldsymbol{M}_t (\boldsymbol{w} - \boldsymbol{w}_t) =: \ell_t(\boldsymbol{w}), \tag{2.4.1}$$

where $\boldsymbol{M}_t$ is a positive semi-definite matrix. Generalizing the results from Section 2.3, EW with Gaussian prior on the surrogate loss ℓ_t yields explicitly computable Gaussian distributions P_t (see also van Erven and Koolen, 2016; Koolen, 2016):

Theorem 4. *Let $P_1 = \mathcal{N}(\boldsymbol{w}_1, \Sigma_1)$. Both versions of the Exponential Weights algorithm, run on ℓ_t with learning rate η and $\mathcal{P} = \{P : \mathbb{E}_P[\boldsymbol{w}] \in \mathcal{W}\}$, yield a multivariate normal distribution $P_{t+1} = \mathcal{N}(\boldsymbol{w}_{t+1}, \Sigma_{t+1})$ with mean and covariance matrix given in Figure 2.4. Furthermore, Lemma 1 implies that for all $\boldsymbol{u} \in \mathcal{W}$ both versions of EW satisfy:*

$$\mathcal{R}_T(\boldsymbol{u}) \leq \frac{1}{2\eta} (\boldsymbol{w}_1 - \boldsymbol{u})^\intercal \Sigma_1^{-1} (\boldsymbol{w}_1 - \boldsymbol{u}) + \frac{\eta}{2} \sum_{t=1}^T \boldsymbol{g}_t^\intercal \Sigma_{t+1} \boldsymbol{g}_t. \tag{2.4.2}$$

The proof of Theorem 4 in Section 2.11.1 is a straightforward generalization of Theorem 2 for constant learning rate $\eta_t = \eta$, which is recovered with $\boldsymbol{M}_t = \boldsymbol{0}$. Like in Theorem 2, the parametrization by η and σ^2 is redundant in that only the product $\eta\sigma^2$ affects the predictions $\boldsymbol{w}_t$ or the bound (2.4.2).

2.4.1 Gradient Descent: Quadratic Approximation of Strongly Convex Losses

For α-strongly convex loss functions, (2.4.1) holds with $\boldsymbol{M}_t = \alpha \boldsymbol{I}$. The standard approach for these loss functions is to use greedy Gradient Descent with a time-varying learning rate $\eta_t = 1/(\alpha t)$ (Hazan et al., 2007). Interestingly, greedy GD with the closely related choice $\eta_t = 1/(\frac{1}{\eta\sigma^2} + \alpha t)$ turns out to be a special case of greedy EW with *fixed* learning rate η and prior $P_1 = \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$. Applying Theorem 4 results in the following corollary, proved in Section 2.11.2:

Corollary 4.1. *Suppose* $\|\boldsymbol{u}\|_2 \le D$ *and* $\|\boldsymbol{g}_t\|_2 \le G$*. Then the regret of both versions of the Exponential Weights algorithm with prior* $\mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$ *and constant learning rate* η*, run on the surrogate loss* (2.4.1) *with* $\boldsymbol{M}_t = \alpha \boldsymbol{I}$*, satisfies:*

$$\mathcal{R}_T(\boldsymbol{u}) \le \frac{G^2}{2\alpha} \ln\left(\frac{\frac{1}{\eta\sigma^2} + \alpha T}{\frac{1}{\eta\sigma^2} + \alpha}\right) + \frac{G^2}{\frac{2}{\eta\sigma^2} + 2\alpha} + \frac{D^2}{2\eta\sigma^2}.$$

The standard learning rate and corresponding regret bound for GD (Hazan et al., 2007) correspond to the limiting case $\eta\sigma^2 \to \infty$. Formally speaking, this case is not covered here, but for $\eta \to \infty$ EW reduces to Follow-the-Leader (on the surrogate loss (2.4.1)), and taking $\sigma^2 \to \infty$ would lead to EW with an *improper prior*, which becomes a proper EW posterior P_2 after one round.

2.4.2 Online Newton Step: Quadratic Approximation of Exp-concave Losses

For α-exp-concave loss functions, (2.4.1) holds with $\boldsymbol{M}_t = \beta \boldsymbol{g}_t \boldsymbol{g}_t^\intercal$, where $\beta = \frac{1}{2}\min\{\frac{1}{4GB}, \alpha\}$, assuming $\|\boldsymbol{g}_t\|_2 \le G$ and $B = \max_{\boldsymbol{w},\boldsymbol{u} \in \mathcal{W}} \|\boldsymbol{w} - \boldsymbol{u}\|_2$ (Hazan et al., 2007, Lemma 3). Running Exponential Weights on $\ell_t(\boldsymbol{w})$ with prior $\mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$ leads to the Online Newton Step algorithm (Hazan et al., 2007) with the following regret bound, shown in Section 2.11.3:

Corollary 4.2. *Suppose* $\|\boldsymbol{u}\|_2 \le D$ *and* $\|\boldsymbol{g}_t\|_2 \le G$*. Then the regret of both versions of the Exponential Weights algorithm with prior* $\mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$ *and learning* *rate η, run on the surrogate loss* (2.4.1) *with $\boldsymbol{M}_t = \beta \boldsymbol{g}_t \boldsymbol{g}_t^\intercal$, satisfies:*

$$\mathcal{R}_T(\boldsymbol{u}) \leq \frac{d}{2\beta} \ln\left(1 + \frac{\eta\sigma^2\beta G^2 T}{d}\right) + \frac{D^2}{2\eta\sigma^2}. \tag{2.4.3}$$

The results of Hazan et al. (2007) correspond to setting $\eta\sigma^2 = \beta D^2$, together with some simplifying upper bounds on (2.4.3).

2.5 Adaptivity by Reduction to Exponential Weights

In this section we show how several recent adaptive methods in the prediction with experts setting – namely iProd (Koolen and Van Erven, 2015), Squint (Koolen and Van Erven, 2015) and a variation of Coin Betting for experts (Orabona and Pál, 2016) –, whose original analyses seem unrelated at first sight, can all be viewed as applying exponential weights after reductions of the original OCO task to various closely related surrogate OCO tasks. The known regret bounds for these methods are also recovered from the reductions upon plugging in regret bounds for EW in the surrogate tasks.

2.5.1 Reduction for iProd

The experts setting consists of linear losses $f_t(\boldsymbol{w}) = \langle \boldsymbol{w}, \boldsymbol{g}_t \rangle$ over the simplex $\mathcal{W} = \{\boldsymbol{w} : w_i \geq 0, \sum_{i=1}^d w_i = 1\}$, with $g_{t,i} \in [0,1]$. The instantaneous regret in round t with respect to expert i is $r_t(i) = f_t(\boldsymbol{w}_t) - f_t(\boldsymbol{e}_i)$ and $\mathcal{R}_T(i) = \sum_{t=1}^T r_t(i)$ is the total regret. iProd achieves a second-order regret bound in terms of the data-dependent quantity $\boldsymbol{V}_T(i) = \sum_{t=1}^T r_t(i)^2$, which is much smaller than the worst-case regret in many common cases (Koolen et al., 2016).

In the surrogate OCO task for iProd, predictions take the form of joint distributions P_t on (η, i) for $\eta \in [0,1]$ and $i \in \{1, \ldots, d\}$. These map back to predictions in the original task via

$$\boldsymbol{w}_t = \frac{\mathbb{E}_{P_t}[\eta \boldsymbol{e}_i]}{\mathbb{E}_{P_t}[\eta]}, \tag{2.5.1}$$

which is like the marginal mean of P_t on experts, except that it is *tilted* to favor larger η. The surrogate loss in the surrogate task is

$$\ell_t(\eta, i) = -\ln\left(1 + \eta r_t(i)\right), \tag{2.5.2}$$

and our aim will be to achieve small *mix-regret* with respect to any comparator distribution Q on (η, i), which we define as $S(Q) = \sum_{t=1}^T - \ln \mathbb{E}_{P_t}\left[e^{-\ell_t(\eta,i)}\right] -$

$\mathbb{E}_Q\left[\sum_{t=1}^T \ell_t(\eta, i)\right]$. The mix-regret allows exponential mixing of predictions according to P_t just like for exp-concave losses, so there is no mixability gap to pay. Exponential weights with constant learning rate 1 on the losses ℓ_t therefore achieves $S(Q) \leq \mathrm{KL}(Q\|P_1)$ for any Q.[3] The resulting predictions $\boldsymbol{w}_t$ are those of the iProd algorithm. As shown in Section 2.12.1, they achieve the following regret bound, which depends on the surrogate regret of EW:

Theorem 5 (iProd Reduction to EW)**.** *Restrict the domain for η to $[0, \frac{1}{2}]$. Then any choice of P_t in the surrogate OCO task defined above induces regret bounded by*

$$\mathbb{E}_Q[\eta] \sum_{t=1}^T f_t(\boldsymbol{w}_t) - \mathbb{E}_Q\left[\eta \sum_{t=1}^T f_t(\boldsymbol{e}_i)\right] \leq \mathbb{E}_Q\left[\eta^2 \boldsymbol{V}_T(i)\right] + S(Q) \qquad (2.5.3)$$

for any Q on (η, i) in the original prediction with expert advice task.

In particular, if we use EW in the surrogate OCO task with learning rate 1 and any product prior $P_1 = \gamma \times \pi$ for γ a distribution on $\eta \in [0, \frac{1}{2}]$ and π a distribution on i, and we take as comparator $Q = \gamma(\eta \mid \eta \in [\hat{\eta}/2, \hat{\eta}]) \times \hat{\pi}$ for any $\hat{\eta} \in [0, \frac{1}{2}]$ and distribution $\hat{\pi}$ on i that can both depend on all the losses, then

$$\mathbb{E}_{\hat{\pi}}\left[\mathcal{R}_T(i)\right] \leq 2\hat{\eta}\, \mathbb{E}_{\hat{\pi}}[\boldsymbol{V}_T(i)] + \frac{2}{\hat{\eta}}\Big(\mathrm{KL}(\hat{\pi}\|\pi) - \ln \gamma([\hat{\eta}/2, \hat{\eta}])\Big). \qquad (2.5.4)$$

Crucially, the algorithm does not need to know $\hat{\eta}$ in advance, but (2.5.4) still holds for all $\hat{\eta}$ simultaneously. To minimize (2.5.4) in $\hat{\eta}$ we can restrict ourselves to $\hat{\eta} \geq 1/\sqrt{T}$ without loss of generality, so that a prior density $\mathrm{d}\gamma(\eta)/\mathrm{d}\eta \propto 1/\eta$ on $[1/\sqrt{T}, 1/2]$ achieves $-\ln \gamma([\hat{\eta}/2, \hat{\eta}]) = O(\ln \ln T)$. After optimizing $\hat{\eta}$, this leads to an adaptive regret bound of

$$\mathbb{E}_{\hat{\pi}}\left[\mathcal{R}_T(i)\right] = O\left(\sqrt{\mathbb{E}_{\hat{\pi}}[\boldsymbol{V}_T(i)]\Big(\mathrm{KL}(\hat{\pi}\|\pi) + \ln \ln T\Big)}\right) \qquad \text{for all } \hat{\pi}, \quad (2.5.5)$$

which recovers the results of Koolen and Van Erven (2015) (see also (Koolen, 2015)).

2.5.2 Reduction for Squint

Running EW with a continuous prior on η for the iProd surrogate losses from (2.5.2) requires evaluating a t-degree polynomial in η in every round, and therefore leads to

[3] This follows e.g. from Lemma 1 by subtracting $\sum_t f_t(\boldsymbol{w}_t)$ on both sides of (2.2.3) and rearranging.

$O(T^2)$ total running time. This may be reduced to $O(T \ln T)$ by using a prior γ on an exponentially spaced grid of η (as in MetaGrad (van Erven and Koolen, 2016)), but in the experts setting even the extra $\ln T$ factor in run time can be avoided. This is possible by moving the 'prod bound' that occurs in the proof of Theorem 5, from the analysis into the algorithm by replacing the surrogate loss from (2.5.2) by the slightly larger surrogate loss

$$\ell_t(\eta, i) = -\eta r_t(i) + \eta^2 r_t(i)^2, \tag{2.5.6}$$

which turns iProd into Squint. Because this surrogate is quadratic in η, it becomes possible to run EW in the resulting surrogate OCO task and evaluate the resulting integrals over η in closed form for suitable choices of the prior on η, so that Squint has $O(T)$ run time (see Koolen and Van Erven (2015) for a detailed discussion of the choice of prior). Moreover, as shown in Section 2.12.2, it satisfies exactly the same guarantees as iProd.

2.5.3 Reduction for Coin Betting

If we are willing to give up on second-order bounds, but still want to learn η, then there is another way to obtain an algorithm with $O(T)$ run time by bounding the iProd surrogate loss, which leads to a variant of the Coin Betting algorithm for experts of Orabona and Pál (2016). Our presentation and analysis are very different from (Orabona and Pál, 2016), but we obtain exactly the same regret bound for essentially the same algorithm, and we can explain some design choices that required clever insights by Orabona and Pál (2016), as natural consequences of running EW in the surrogate OCO task that we end up with.

The idea is to split the learning of $\eta \in [0, 1]$ and i into separate steps: for each i, we restrict $P_t(\eta \mid i)$ to be a point mass on some η_t^i, and we will choose η_t^i to achieve small regret for the surrogate loss

$$\ell_t^i(\eta) = -\frac{1 + r_t(i)}{2} \ln \frac{1+\eta}{2} - \frac{1 - r_t(i)}{2} \ln \frac{1-\eta}{2} - \ln 2,$$

which upper bounds (2.5.2) by convexity of the negative logarithm. We then plug in the choices of η_t^i in (2.5.2) and learn i for the resulting surrogate losses $\tilde{\ell}_t(i) = -\ln(1 + \eta_t^i r_t(i))$. For $\eta \in [0, 1]$ and $\hat{\pi}$ a distribution on i, let

$$S_T^i(\eta) = \sum_{t=1}^T \ell_t^i(\eta_t^i) - \sum_{t=1}^T \ell_t^i(\eta),$$

$$\tilde{S}_T(\hat{\pi}) = \sum_{t=1}^T - \ln \mathbb{E}_{i \sim P_t} \left[e^{-\tilde{\ell}_t(i)} \right] - \mathbb{E}_{\hat{\pi}} \left[\sum_{t=1}^T \tilde{\ell}_t(i) \right]$$

be the mix-regret in the two surrogate OCO tasks. (Notice that in S_T^i the mix-regret has collapsed to the ordinary regret, because we are restricting ourselves to play point masses on η.) Also let $\mathcal{R}_T^+(i) = \max\{\mathcal{R}_T(i), 0\}$ be the nonnegative part of the regret, and define $\mathrm{B}(x\|y) = x \ln \frac{x}{y} + (1-x) \ln \frac{1-x}{1-y}$ to be the Kullback-Leibler divergence between two Bernoulli distributions, which satisfies $\mathrm{B}(x\|y) \geq 2(x-y)^2$ by Pinsker's inequality. Then this reduction gives the following regret bound, proved in Section 2.12.3:

Theorem 6 (Coin Betting Reduction to EW)**.** *Any choice of distributions P_t on i and learning rates η_t^i in the surrogate OCO task defined above induces regret bounded by*

$$\mathbb{E}_{\hat{\pi}}\left[\mathrm{B}\left(\tfrac{1}{2} + \tfrac{\mathcal{R}_T^+(i)}{2T} \| \tfrac{1}{2}\right)\right] \leq \tfrac{1}{T}\left(\mathbb{E}_{\hat{\pi}}\left[S_T^i\left(\tfrac{\mathcal{R}_T^+(i)}{T}\right)\right] + \tilde{S}_T(\hat{\pi})\right) \qquad \textit{for any } \hat{\pi} \textit{ on } i \tag{2.5.7}$$

in the original prediction with expert advice task.

In particular, if we use EW with learning rate 1 and prior π on i for the losses $\tilde{\ell}_t$, and for the losses ℓ_t^i we let η_t^i be the mean of lazy EW with learning rate 1 and with prior on $\eta \in [-1,+1]$ such that $\frac{1+\eta}{2}$ has a beta-distribution $\beta(a,a)$ with $a = \frac{T}{4} + \frac{1}{2}$ and with projections onto $\mathcal{P} = \{P \mid \mathbb{E}_P[\eta] \in [0,1]\}$, then

$$\mathbb{E}_{\hat{\pi}}\left[\mathcal{R}_T(i)\right] \leq \sqrt{3T\left(\mathrm{KL}(\hat{\pi}\|\pi) + 3\right)} \qquad \textit{for any } \hat{\pi} \textit{ on } i. \tag{2.5.8}$$

Compared to (2.5.5), (2.5.8) avoids a $\ln\ln T$ term, but it has lost the benefits of the second-order factor $\mathbb{E}_{\hat{\pi}}[\boldsymbol{V}_T(i)] \leq T$. This may be explained by its upper bound $\ell_t^i(\eta) \geq \ell_t(\eta, i)$, which is tight only in the extreme case that $r_t(i) \in \{-1,+1\}$.

The Resulting Coin Betting Algorithm EW on the losses ℓ_t^i with the (conjugate) $\beta(a,a)$ prior is a generalization of the Krichevsky-Trofimov estimator (see Example 1) and its mean has the closed form $\frac{\mathcal{R}_{t-1}(i)}{t-1+2a}$. Lazily projecting onto $\mathcal{P}$ then simply amounts to clipping at 0 (by convexity of KL-divergence in its first argument, which implies that the constraint $\mathbb{E}_P[\eta] \geq 0$ will be satisfied with equality when we project from a distribution with negative mean). This means that $\eta_t^i = \max\left\{\frac{\mathcal{R}_{t-1}(i)}{t-1+2a}, 0\right\}$. By (2.5.1) the Coin Betting algorithm from the theorem predicts with weights $w_{t,i}$ obtained by normalizing the unnormalized weights $\tilde{w}_{t,i} = \tilde{p}_t(i)\eta_t^i$, where $\tilde{p}_t(i)$ is the unnormalized probability $P_t(i)$ of EW on the losses $\tilde{\ell}_t$, which recursively satisfies

$$\begin{aligned}\tilde{p}_t(i) :=& \pi(i)\prod_{s=1}^{t-1}(1+\eta_s^i r_s(i)) = \tilde{p}_{t-1}(i) + \tilde{w}_{t-1,i} r_{t-1}(i) = \dots \\ =& \pi(i) + \sum_{s=1}^{t-1} \tilde{w}_{s,i} r_s(i).\end{aligned}$$

Interestingly, Orabona and Pál (2016) interpret the unnormalized EW probabilities $\tilde{p}_t(i)$ as the *Wealth* for expert i that is achieved by a gambler.

The interpretation in Theorem 6 explains three design choices by Orabona and Pál (2016): first, their choice of potential function, which naturally arises in our proof when we bound the regret $S_T^i(\mathcal{R}_T^+(i)/T)$ for EW using Lemma 1. Second, the choice for a, which in the original analysis comes from defining a shifted potential function, is simply specifying a prior with most mass in a region of order $1/\sqrt{T}$ around $\eta = 0$. And, third, the clipping of the unnormalized weights $\tilde{w}_{t,i}$ to 0 when $\mathcal{R}_{t-1}(i) < 0$, which in our presentation happens automatically because the learning rate η_t^i is projected to be 0 if it would otherwise become negative. Defining a prior on positive learning rates directly would be possible in theory, but not with a conjugate prior, so the computational efficiency of the algorithm is made possible by the projections.

There is one slight difference between the algorithm we obtain here and the original Coin Betting algorithm of Orabona and Pál (2016): in the original method the instantaneous regrets are clipped to $\max\{r_t(i), 0\}$ when $\mathcal{R}_{t-1}(i) < 0$, which our method does not do. Apparently there is some amount of freedom in the design of this type of algorithm.

2.6 Online Linear Optimization with Bandit Feedback

A benefit of the EW interpretation of MD is that it opens up the possibility of sampling from the EW posterior distribution instead of playing the mean. Here we show how this option can be leveraged to obtain an algorithm for online linear optimization with bandit feedback (Dani et al., 2008; Abernethy et al., 2008), which recovers the best known rate $O(d\sqrt{T}\ln T)$. A proof of this fact has already been outlined by Bubeck and Eldan (2015), but here we fill in the technical details.

The linear bandit setting consists of linear losses $f_t(\boldsymbol{w}) = \langle \boldsymbol{w}, \boldsymbol{g}_t \rangle \in [-1, +1]$, but instead of seeing the vectors $\boldsymbol{g}_t$ we only observe $f_t(\boldsymbol{w}_t)$ for the algorithm's choice $\boldsymbol{w}_t$. The algorithm can randomize its choice $\boldsymbol{w}_t$, and $\boldsymbol{g}_t$ is fixed before the outcome of this randomization. The goal is to minimize the expected regret $\mathbb{E}[\mathcal{R}_T(\boldsymbol{u})]$, where the expectation is with respect to the algorithm's randomness.

We consider the EW algorithm with fixed learning rate η and uniform prior distribution P_1 over $\mathcal{W}$. In each round t, after observing $f_t(\boldsymbol{w}_t) = \langle \boldsymbol{w}_t, \boldsymbol{g}_t \rangle$, the algorithm constructs a random, unbiased estimate $\tilde{\boldsymbol{g}}_t$ of the loss vector $\boldsymbol{g}_t$ and uses this estimate to update P_t to P_{t+1}. It is easy to verify that, for each t, P_t is a member of the exponential family with cumulant generating function $F(\boldsymbol{\theta}) = \ln \int_{\mathcal{W}} e^{\langle \boldsymbol{w}, \boldsymbol{\theta} \rangle} \, \mathrm{d}\boldsymbol{w}$. At trial t, the algorithm samples $\boldsymbol{w}_t \sim Q_t$, where $Q_t = (1-\gamma)P_t + \gamma R$ is a mixture of the EW distribution P_t and a fixed "exploration" distribution R, chosen to be *John's exploration* (Bubeck et al., 2012). Using that the convex conjugate of F is a universal $O(d)$-self concordant barrier on $\mathcal{W}$ (Bubeck and Eldan, 2015), it can be shown that, when η and γ are appropriately chosen, this algorithm achieves expected regret of order $O(d\sqrt{T \ln T})$ (see Section 2.13).

It is interesting to compare with the *SCRiBLe* algorithm (Abernethy et al., 2012), which replaces EW by MD. By the results of Section 2.3.3, this is an essentially equivalent approach, except that SCRiBLe employs a sampling strategy based on the spectrum of the Hessian of F^*, without reference to the EW distribution, and achieves a regret bound that is suboptimal in d. This shows that the EW interpretation of MD is clearly beneficial in the bandit setting.

2.7 Discussion

We conclude with several remarks: first, we point out that there may be computational reasons to avoid defining the prior directly on the domain $\mathcal{W}$ of interest: as shown for instance in Sections 2.3.2 and 2.4, defining a Gaussian prior on all of $\mathbb{R}^d$ and then projecting the mean onto $\mathcal{W}$ can be computationally more efficient. In the context of sampling from the EW distribution, discussed in Section 2.6, this might also make sense if we project onto the alternative (smaller) set of distributions $\mathcal{P} = \{P \mid P(\mathcal{W}) = 1\} \subset \{P \mid \mathbb{E}_P[\boldsymbol{w}] \in \mathcal{W}\}$ that are supported on $\mathcal{W}$, which amounts to conditioning on $\mathcal{W}$. Second, there seems to be a discrepancy between the body of work for the log loss cited in the introduction, which strongly suggests using Jeffreys' prior, and the uniform prior suggested in Section 2.6 in the context of the universal barrier.

2.8 Proof of Lemma 1 from Section 2.2

Proof. In the following we make use of the generalized Pythagorean inequality for Kullback-Leibler divergence (Csiszár, 1975): for $P_t = \arg\min_{P \in \mathcal{P}} \mathrm{KL}(P \| \tilde{P}_t)$

and any $Q \in \mathcal{P}$:

$$\mathrm{KL}(Q\|\tilde{P}_t) \geq \mathrm{KL}(Q\|P_t) + \mathrm{KL}(P_t\|\tilde{P}_t). \tag{2.8.1}$$

For greedy EW we have

$$\begin{aligned}
&\frac{1}{\eta_t}\big(\mathrm{KL}(Q\|P_t) - \mathrm{KL}(Q\|P_{t+1})\big) \\
&\geq \frac{1}{\eta_t}\left(\mathrm{KL}(Q\|P_t) - \mathrm{KL}(Q\|\tilde{P}_{t+1})\right) && \text{(from (2.8.1))} \\
&= -\mathbb{E}_Q[f_t(\boldsymbol{w})] - \frac{1}{\eta_t}\ln \mathbb{E}_{P_t}\left[e^{-\eta_t f_t(\boldsymbol{w})}\right] && \text{(from (2.2.2))}
\end{aligned}$$

in any trial t. Summing over trials gives:

$$\begin{aligned}
\sum_{t=1}^{T} -\mathbb{E}_Q[f_t(\boldsymbol{w})] - \frac{1}{\eta_t}\ln \mathbb{E}_{P_t}\left[e^{-\eta_t f_t(\boldsymbol{w})}\right] &\leq \sum_{t=1}^{T}\frac{1}{\eta_t}\big(\mathrm{KL}(Q\|P_t) - \mathrm{KL}(Q\|P_{t+1})\big) \\
&= \frac{1}{\eta_1}\mathrm{KL}(Q\|P_1) - \frac{1}{\eta_T}\mathrm{KL}(Q\|P_{T+1}) \\
&\quad + \sum_{t=2}^{T}\mathrm{KL}(Q\|P_t)\left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}\right) \\
&\leq \frac{1}{\eta_1}\mathrm{KL}(Q\|P_1) + \max_{t=2,\dots,T}\mathrm{KL}(Q\|P_t)\left(\frac{1}{\eta_T} - \frac{1}{\eta_1}\right).
\end{aligned}$$

Rearranging the terms and adding $\sum_{t=1}^{T} f_t(\boldsymbol{w}_t)$ on both sides results in (2.2.4).

We now proceed with the proof of lazy EW, starting from:

$$\begin{aligned}
&-\frac{1}{\eta_{t-1}}\ln \mathbb{E}_{P_t}[e^{-\eta_{t-1}f_t(\boldsymbol{w})}] && (2.8.2) \\
&= \min_P \left\{\mathbb{E}_P[f_t(\boldsymbol{w})] + \frac{1}{\eta_{t-1}}\mathrm{KL}(P\|P_t)\right\} \\
&\leq \mathbb{E}_{P_{t+1}}[f_t(\boldsymbol{w})] + \frac{1}{\eta_{t-1}}\mathrm{KL}(P_{t+1}\|P_t) \\
&\leq \mathbb{E}_{P_{t+1}}[f_t(\boldsymbol{w})] + \frac{1}{\eta_{t-1}}\mathrm{KL}(P_{t+1}\|\tilde{P}_t) - \frac{1}{\eta_{t-1}}\mathrm{KL}(P_t\|\tilde{P}_t), && (2.8.3)
\end{aligned}$$

where the last inequality is from the Pythagorean inequality (2.8.1) applied with $Q = P_{t+1}$. By (2.2.1):

$$\ln\frac{\mathrm{d}\tilde{P}_t(\boldsymbol{w})}{\mathrm{d}P_1(\boldsymbol{w})} = -\eta_{t-1}\sum_{s=1}^{t-1} f_s(\boldsymbol{w}) - \ln \mathbb{E}_{P_1}\left[e^{-\eta_{t-1}\sum_{s=1}^{t-1} f_s(\boldsymbol{w})}\right],$$

which gives:

$$\begin{aligned}
&\frac{1}{\eta_{t-1}} \mathrm{KL}(P_{t+1}\|\tilde{P}_t) - \frac{1}{\eta_{t-1}} \mathrm{KL}(P_t\|\tilde{P}_t)\\
&= \frac{1}{\eta_{t-1}} \mathrm{KL}(P_{t+1}\|P_1) - \frac{1}{\eta_{t-1}} \mathrm{KL}(P_t\|P_1)\\
&+ \mathbb{E}_{P_{t+1}}\left[\sum_{s=1}^{t-1} f_s(\boldsymbol{w})\right] - \mathbb{E}_{P_t}\left[\sum_{s=1}^{t-1} f_s(\boldsymbol{w})\right].
\end{aligned}$$

Plugging this into (2.8.3) and using $\eta_t \le \eta_{t-1}$ results in:

$$\begin{aligned}
-\frac{1}{\eta_{t-1}} \ln \mathbb{E}_{P_t}[e^{-\eta_{t-1} f_t(\boldsymbol{w})}] \le \frac{1}{\eta_t} \mathrm{KL}(P_{t+1}\|P_1) - \frac{1}{\eta_{t-1}} \mathrm{KL}(P_t\|P_1)&\\
+ \mathbb{E}_{P_{t+1}}\left[\sum_{s=1}^{t} f_s(\boldsymbol{w})\right] - \mathbb{E}_{P_t}\left[\sum_{s=1}^{t-1} f_s(\boldsymbol{w})\right]&.
\end{aligned}$$

Summing over trials makes the terms on the right-hand side telescope and gives:

$$\begin{aligned}
\sum_{t=1}^{T} -\frac{1}{\eta_{t-1}} \ln \mathbb{E}_{P_t}[e^{-\eta_{t-1} f_t(\boldsymbol{w})}] &\le \frac{1}{\eta_T} \mathrm{KL}(P_{T+1}\|P_1) + \mathbb{E}_{P_{T+1}}\left[\sum_{t=1}^{T} f_t(\boldsymbol{w})\right]\\
&= \min_{P\in\mathcal{P}} \left\{ \mathbb{E}_P\left[\sum_{t=1}^{T} f_t(\boldsymbol{w})\right] + \frac{1}{\eta_T} \mathrm{KL}(P\|P_1) \right\}\\
&\le \mathbb{E}_Q\left[\sum_{t=1}^{T} f_t(\boldsymbol{w})\right] + \frac{1}{\eta_T} \mathrm{KL}(Q\|P_1),
\end{aligned}$$

where the equality expresses an equivalent way to define lazy EW. Rearranging the terms and adding $\sum_{t=1}^{T} f_t(\boldsymbol{w}_t)$ on both sides results in (2.2.3). □

2.9 Proof of Theorem 1

Proof. Rather than scaling canonical vectors $\boldsymbol{e}_i$, $i = 1, \ldots, d$ and the comparator $\boldsymbol{u}$ by M, we scale the loss vectors by defining $\boldsymbol{g}'_t = M\boldsymbol{g}_t$, so that the losses remain the same: $\langle \boldsymbol{e}_i, \boldsymbol{g}'_t\rangle = \langle M\boldsymbol{e}_i, \boldsymbol{g}_t\rangle$ for all i and all t. Let $\boldsymbol{w}_1 = (\boldsymbol{w}_1^+, \boldsymbol{w}_1^-)$, and let $\boldsymbol{w}_t^+$, $\boldsymbol{w}_t^-$ be the result of running EG plus-minus on $\boldsymbol{g}'_t$. For any $\boldsymbol{u}$ with $\sum_{i=1}^{2d} u_i = 1$ and $u_i \ge 0$ invoking Lemma 1 gives:

$$\begin{aligned}
\sum_{t=1}^{T} \langle \boldsymbol{w}_t - \boldsymbol{u}, \boldsymbol{g}'_t\rangle \le \frac{1}{\eta} \mathrm{KL}(\boldsymbol{u}\|\boldsymbol{w}_1) + \sum_{t=1}^{T} \langle \boldsymbol{w}_t^+, \boldsymbol{g}'_t\rangle - \langle \boldsymbol{w}_t^-, \boldsymbol{g}'_t\rangle&\\
+ \frac{1}{\eta} \ln\Big(\sum_{i=1}^{d} (w_{t,i}^+ e^{-\eta_t\langle \boldsymbol{e}_i, \boldsymbol{g}'_t\rangle} + w_{t,i}^- e^{\eta_t\langle \boldsymbol{e}_i, \boldsymbol{g}'_t\rangle})\Big)&. \qquad (2.9.1)
\end{aligned}$$

The first term on the right-hand side of (2.9.1) can be bounded by:

$$\max_{\boldsymbol{u}:\sum_{i=1}^{2d} u_i=1,\ u_i\geq 0} \mathrm{KL}(\boldsymbol{u}\|\boldsymbol{w}_1) = \ln(2d).$$

To bound the second term on the right-hand side of (2.9.1), we make use of Hoeffding's Lemma (Cesa-Bianchi and Lugosi, 2006, Lemma A.1), which together with $|\langle \boldsymbol{e}_i, \boldsymbol{g}_t'\rangle| \leq MG$ gives:

$$\sum_{t=1}^{T}\langle \boldsymbol{w}_t^+, \boldsymbol{g}_t'\rangle - \langle \boldsymbol{w}_t^-, \boldsymbol{g}_t'\rangle + \frac{1}{\eta}\ln\Big(\sum_{i=1}^{d}(w_{t,i}^+ e^{-\eta_t\langle \boldsymbol{e}_i,\boldsymbol{g}_t'\rangle} + w_{t,i}^- e^{\eta_t\langle \boldsymbol{e}_i,\boldsymbol{g}_t'\rangle})\Big) \leq \frac{\eta M^2 G^2}{2}.$$

Summing over trials results in a bound on the regret:

$$\sum_{t=1}^{T}\langle \boldsymbol{w}_t - \boldsymbol{u}, \boldsymbol{g}_t'\rangle \leq \frac{\ln(2d)}{\eta} + \eta\frac{TM^2G^2}{2}.$$

Plugging in the optimal $\eta = \sqrt{\frac{2\ln(2d)}{TM^2G^2}}$ yields the desired result. □

2.10 Proof of Theorem 3

Before proving the theorem, we need two lemmas:

Lemma 2 (Banerjee et al. (2005); Nielsen and Nock (2010))**.** *The KL divergence between two members, P and Q, of the same regular exponential family $\mathcal{E}$ with cumulant generating function F can be expressed by the Bregman divergence between their natural parameters, $\boldsymbol{\theta}_P$ and $\boldsymbol{\theta}_Q$, or their expectation parameters, $\boldsymbol{\mu}_P$ and $\boldsymbol{\mu}_Q$. The first Bregman divergence is generated by the cumulant generating function F and the second Bregman divergence is generated by the convex conjugate of the cumulant generating function F^*:*

$$\mathrm{KL}(P\|Q) = B_F(\boldsymbol{\theta}_Q\|\boldsymbol{\theta}_P) = B_{F^*}(\boldsymbol{\mu}_P\|\boldsymbol{\mu}_Q).$$

Lemma 3. *(Ihara, 1993, Theorem 3.1.4) Let $\boldsymbol{\mu}$ be arbitrary and define $\mathcal{P} = \{P : \mathbb{E}_P[\boldsymbol{w}] = \boldsymbol{\mu}\}$. Then, for any member Q of an exponential family $\mathcal{E}$,*

$$\min_{P\in\mathcal{P}} \mathrm{KL}(P\|Q)$$

is achieved by $P \in \mathcal{E}$ such that $\mathbb{E}_P[\boldsymbol{w}] = \boldsymbol{\mu}$, provided such a P exists.

Proof of Theorem 3. Let $\boldsymbol{w}_t$ be the weights produced by the greedy version of MD. Then

$$\begin{aligned}
&\min_{P\in\mathcal{P}}\left\{\mathbb{E}_P[\langle\boldsymbol{w},\boldsymbol{g}_t\rangle]+\frac{1}{\eta_t}\operatorname{KL}(P\|P_t)\right\}\\
&=\min_{\boldsymbol{\mu}\in\mathcal{W}}\ \min_{P\,:\,\mathbb{E}_P[\boldsymbol{w}]=\boldsymbol{\mu}}\left\{\mathbb{E}_P[\langle\boldsymbol{w},\boldsymbol{g}_t\rangle]+\frac{1}{\eta_t}\operatorname{KL}(P\|P_t)\right\}\\
&=\min_{\boldsymbol{\mu}\in\mathcal{W}}\ \min_{P\in\mathcal{E}\,:\,\mathbb{E}_P[\boldsymbol{w}]=\boldsymbol{\mu}}\left\{\langle\boldsymbol{\mu},\boldsymbol{g}_t\rangle+\frac{1}{\eta_t}\operatorname{KL}(P\|P_t)\right\},
\end{aligned}$$

where in the second step we can restrict to minimization over $\mathcal{E}$ by Lemma 3. Introducing the short-hand notation $\boldsymbol{\mu}_P=\mathbb{E}_P[\boldsymbol{w}]$, we thus get for the greedy version of EW:

$$\begin{aligned}
P_{t+1}&=\operatorname*{arg\,min}_{P\in\mathcal{E}:\boldsymbol{\mu}_P\in\mathcal{W}}\left\{\langle\boldsymbol{\mu}_P,\boldsymbol{g}_t\rangle+\frac{1}{\eta_t}\operatorname{KL}(P\|P_t)\right\}\\
&=\operatorname*{arg\,min}_{P\in\mathcal{E}:\boldsymbol{\mu}_P\in\mathcal{W}}\left\{\langle\boldsymbol{\mu}_P,\boldsymbol{g}_t\rangle+\frac{1}{\eta_t}B_{F^*}(\boldsymbol{\mu}_P\|\boldsymbol{\mu}_{P_t})\right\},
\end{aligned}$$

where we used Lemma 2. But the last expression coincides with the definition of the greedy MD weight update, and since it applies to all t, we have $\boldsymbol{\mu}_{P_{t+1}}=\boldsymbol{w}_{t+1}$ for all t, provided $\boldsymbol{\mu}_{P_1}=\boldsymbol{w}_1$ (which holds by assumption). An analogous argument can be made to show the equivalence of the lazy versions of MD and EW. □

2.11 Proofs for Section 2.4

2.11.1 Proof of Theorem 4

Proof. $\tilde{P}_t=\mathcal{N}(\tilde{\boldsymbol{w}}_t,\Sigma_t)$ may be verified analytically from (2.2.1) and (2.2.2). The fact that projections P_t onto $\mathcal{P}$ preserve Gaussianity with the same covariance matrix follows from Lemma 9 in van Erven and Koolen (2016). Lemma 1 gives a bound on the regret w.r.t. randomized forecaster $Q=\mathcal{N}(\boldsymbol{u},\Sigma_Q)$:

$$\sum_{t=1}^T\ell_t(\boldsymbol{w}_t)-\sum_{t=1}^T\mathbb{E}_Q[\ell_t(\boldsymbol{w})]\le\frac{1}{\eta}\operatorname{KL}(Q\|P_1)+\sum_{t=1}^T\ell_t(\boldsymbol{w}_t)+\frac{1}{\eta}\ln\mathbb{E}_{P_t}\left[e^{-\eta\ell_t(\boldsymbol{w})}\right].$$

The KL divergence between two Gaussians is given by (Ihara, 1993, Theorem 1.8.2):

$$\operatorname{KL}(Q\|P_1)=\frac{1}{2}(\ln\left(\frac{\det(\Sigma_1)}{\det(\Sigma_Q)}\right)+\operatorname{Tr}(\Sigma_Q\Sigma_1^{-1})+(\boldsymbol{u}-\boldsymbol{w}_1)^\intercal\Sigma_1^{-1}(\boldsymbol{u}-\boldsymbol{w}_1)-d).$$

The mixability gap can be evaluated in closed form by calculating the Gaussian integral:

$$\ln \mathbb{E}_{P_t}\left[e^{\eta(\ell_t(\boldsymbol{w}_t)-\ell_t(\boldsymbol{w}))}\right] = \frac{\eta^2}{2}\boldsymbol{g}_t^\intercal \Sigma_{t+1}\boldsymbol{g}_t - \frac{1}{2}\ln\left(\frac{\det(\Sigma_t)}{\det(\Sigma_{t+1})}\right).$$

Also, the expectation of the instantaneous regret can be computed exactly:

$$\ell_t(\boldsymbol{w}_t) - \mathbb{E}_Q[\ell_t(\boldsymbol{w})] = \ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{u}) - \frac{1}{2}\operatorname{Tr}(\Sigma_Q \boldsymbol{M}_t).$$

Summing the above over the trials, we get the following upper bound on the regret:

$$\sum_{t=1}^T \ell_t(\boldsymbol{w}_t) - \sum_{t=1}^T \ell_t(\boldsymbol{u}) \leq \eta \sum_{t=1}^T \boldsymbol{g}_t^\intercal \Sigma_{t+1}\boldsymbol{g}_t + \frac{\ln\left(\frac{\det(\Sigma_{T+1})}{\det(\Sigma_Q)}\right) + \operatorname{Tr}(\Sigma_Q\Sigma_{T+1}^{-1}) - d + (\boldsymbol{w}_1-\boldsymbol{u})^\intercal \Sigma_1^{-1}(\boldsymbol{w}_1-\boldsymbol{u})}{2\eta},$$

which holds for all Σ_Q. By plugging in the optimal value $\Sigma_Q = \Sigma_{T+1}$, the bound simplifies to:

$$\sum_{t=1}^T \ell_t(\boldsymbol{w}_t) - \sum_{t=1}^T \ell_t(\boldsymbol{u}) \leq \frac{1}{2\eta}(\boldsymbol{w}_1-\boldsymbol{u})^\intercal \Sigma_1^{-1}(\boldsymbol{w}_1-\boldsymbol{u}) + \frac{\eta}{2}\sum_{t=1}^T \boldsymbol{g}_t^\intercal \Sigma_{t+1}\boldsymbol{g}_t,$$

which concludes the proof after using (2.4.1). □

2.11.2 Proof of Corollary 4.1

Proof. Using Theorem 4 gives:

$$\begin{aligned}
&\sum_{t=1}^T f_t(\boldsymbol{w}_t) - \sum_{t=1}^T f_t(\boldsymbol{u}) \\
&\leq \frac{1}{2\eta\sigma^2}\|\boldsymbol{u}\|_2^2 + \frac{\eta}{2}\sum_{t=1}^T \frac{1}{\frac{1}{\sigma^2}+\alpha\eta t}\|\boldsymbol{g}_t\|_2^2 \\
&\leq \frac{1}{2\eta\sigma^2}D^2 + \frac{\eta}{2}G^2\sum_{t=1}^T \frac{1}{\frac{1}{\sigma^2}+\alpha\eta t} \\
&\leq \frac{1}{2\eta\sigma^2}D^2 + \frac{\eta G^2}{2(\frac{1}{\sigma^2}+\alpha\eta)} + \frac{\eta}{2}G^2\int_1^T \frac{1}{\frac{1}{\sigma^2}+\alpha\eta t}dt \\
&= \frac{1}{2\eta\sigma^2}D^2 + \frac{G^2}{2(\frac{1}{\eta\sigma^2}+\alpha)} + \frac{G^2}{2\alpha}\big(\ln(\tfrac{1}{\eta\sigma^2}+\alpha T) - \ln(\tfrac{1}{\eta\sigma^2}+\alpha)\big),
\end{aligned}$$

which was to be shown. □

2.11.3 Proof of Corollary 4.2

Proof. Using Theorem 4 gives:

$$\mathcal{R}_T(\boldsymbol{u}) \leq \frac{D^2}{2\eta\sigma^2} + \frac{\eta}{2}\sum_{t=1}^{T} \boldsymbol{g}_t^\intercal \Sigma_{t+1}\boldsymbol{g}_t. \tag{2.11.1}$$

We start by bounding the second term on the right-hand side of (2.11.1). Using Lemma 12 from Hazan et al. (2007) we bound:

$$\eta\beta\boldsymbol{g}_t^\intercal\Sigma_{t+1}\boldsymbol{g}_t = \mathrm{Tr}(\Sigma_{t+1}(\Sigma_{t+1}^{-1} - \Sigma_t^{-1})) \leq \ln\frac{\det(\Sigma_{t+1}^{-1})}{\det(\Sigma_t^{-1})},$$

which after summing over trials gives:

$$\begin{aligned}\sum_{t=1}^{T}\eta\beta\boldsymbol{g}_t^\intercal\Sigma_{t+1}\boldsymbol{g}_t \leq \ln\frac{\det(\Sigma_{T+1}^{-1})}{\det(\Sigma_1^{-1})} &= \ln\det\left(\boldsymbol{I} + \eta\sigma^2\beta\sum_{t=1}^{T}\boldsymbol{g}_t\boldsymbol{g}_t^\intercal\right) \\ &= \sum_{i=1}^{d}\ln(1+\lambda_i) \leq d\ln\left(1 + \frac{\eta\sigma^2\beta G^2 T}{d}\right),\end{aligned}$$

where $\lambda_1, \ldots, \lambda_d$ are the eigenvalues of $\eta\sigma^2\beta\sum_{t=1}^{T}\boldsymbol{g}_t\boldsymbol{g}_t^\intercal$, and the last inequality follows by maximizing under the constraint that $\sum_i \lambda_i = \mathrm{Tr}(\eta\sigma^2\beta\sum_{t=1}^{T}\boldsymbol{g}_t\boldsymbol{g}_t^\intercal) \leq \sigma^2\eta\beta G^2 T$. As discussed by Cesa-Bianchi and Lugosi (2006, proof and discussion of Theorem 11.7), the maximum is achieved when $\lambda_i = \sigma^2\eta\beta G^2 T/d$ for all i.

All together we find:

$$\mathcal{R}_T(\boldsymbol{u}) \leq \frac{D^2}{2\eta\sigma^2} + \frac{d}{2\beta}\ln\left(1 + \frac{\eta\sigma^2\beta G^2 T}{d}\right),$$

which was to be shown.

□

2.12 Proofs for Section 2.5

2.12.1 Proof of Theorem 5

Abbreviate $m_t(P) = -\ln\mathbb{E}_P\left[e^{-\ell_t(\eta,i)}\right]$ and define the potential $\Phi_T = e^{-\sum_{t=1}^{T} m_t(P_t)}$. Then $\Phi_T = \Phi_{T-1} = \cdots = \Phi_0 = 1$ since

$$\Phi_T - \Phi_{T-1} = e^{-\sum_{t=1}^{T-1} m_t(P_t)}\,\mathbb{E}_{P_T}\left[\eta r_T(i)\right] = 0,$$

where the last identity holds for any loss vector $\boldsymbol{g}_t$ by the definition of $\boldsymbol{w}_T$. For any comparator Q on (η, i), it follows that

$$0 = \sum_{t=1}^{T} m_t(P_t) = \sum_{t=1}^{T} \mathbb{E}_Q[\ell_t(\eta, i)] + S(Q) \leq \sum_{t=1}^{T} \mathbb{E}_Q[-\eta r_t(i) + \eta^2 r_t(i)^2] + S(Q),$$

where the last inequality is an application of the 'prod-bound' $-\ln(1+x) \leq -x + x^2$ with $x = \eta r_t(i)$, which holds for any $x \geq -\frac{1}{2}$ (Cesa-Bianchi et al., 2007, Lemma 1). The result (2.5.3) is a direct consequence, and (2.5.4) follows upon bounding $\mathbb{E}_Q[\eta] \geq \hat{\eta}/2$ and $\mathbb{E}_Q[\eta^2] \leq \hat{\eta}^2$ and plugging in that $S(Q) \leq \mathrm{KL}(Q\|P_1) = \mathrm{KL}(\hat{\pi}\|\pi) - \ln \gamma([\hat{\eta}/2, \hat{\eta}])$ for EW.

2.12.2 Proof of Theorem 7

Theorem 7 (Squint Reduction to EW)**.** *The exact same statement as in Theorem 5 also holds when we replace the surrogate loss* (2.5.2) *by* (2.5.6).

Thus (2.5.5) also holds, and we recover the results of (Koolen and Van Erven, 2015) for Squint.

Remark 8. *The Metagrad algorithm (van Erven and Koolen, 2016) is similar to Squint on a continuous set of experts indexed by $\boldsymbol{w} \in \mathbb{R}^d$ with losses $f_t(\boldsymbol{w}) = \boldsymbol{w}^\intercal \boldsymbol{g}_t$, and the analysis of Theorem 7 can be extended to handle this case.*

Proof. Let $m_t(P)$ and Φ_T be as in the proof of Theorem 5, but for the new surrogate loss (2.5.6). Then $\Phi_T \leq \Phi_{T-1} \leq \ldots \leq \Phi_0 = 1$, because

$$\begin{aligned}\Phi_T - \Phi_{T-1} =& e^{-\sum_{t=1}^{T-1} m_t(P_t)} \left(\mathbb{E}_{P_T}\left[e^{-f_t(\eta,i)}\right] - 1\right) \\ \leq& e^{-\sum_{t=1}^{T-1} m_t(P_t)} \, \mathbb{E}_{P_T}\left[\eta r_T(i)\right] = 0,\end{aligned}$$

where the inequality follows from the 'prod bound' (see the proof of Theorem 5) and the final equality is again by definition of $\boldsymbol{w}_T$. For any Q, it follows that

$$0 \leq \sum_{t=1}^{T} m_t(P_t) = \sum_{t=1}^{T} \mathbb{E}_Q[\ell_t(\eta, i)] + S(Q) = \sum_{t=1}^{T} \mathbb{E}_Q[-\eta r_t(i) + \eta^2 r_t(i)^2] + S(Q),$$

which implies that (2.5.3) also holds for Squint. Since (2.5.4) is a corollary, it also follows directly. □

2.12.3 Proof of Theorem 6

The proof of Theorem 6 follows the same general steps as the proofs for Theorems 5 and 7. However, bounding the mix-regret $S_T^i(\eta)$ using a similar analysis as for the Krichevsky-Trofimov estimator from Example 1 would lead to an extra $\ln T$ factor in the regret. This is avoided using a more delicate analysis that holds specifically for the regret with respect to $\eta = \mathcal{R}_T^+(i)/T$, which requires a technical analytic inequality by Orabona and Pál (2016, Lemma 16).

Proof. For ℓ_t as in (2.5.2), let $m_t = -\ln \mathbb{E}_{i\sim P_t}\left[e^{-\ell_t(\eta_t^i, i)}\right]$. Then, by the same argument as in the proof of Theorem 5, $\Phi_T = e^{-\sum_{t=1}^T m_t} = 1$. For any distribution $\hat\pi$ on i and any $\hat\eta^i \in [0,1]$, we therefore have

$$\begin{aligned} 0 = \sum_{t=1}^T m_t &= \mathbb{E}_{\hat\pi}\left[\sum_{t=1}^T \ell_t(\eta_t^i, i)\right] + \tilde{S}_T(\hat\pi) \le \mathbb{E}_{\hat\pi}\left[\sum_{t=1}^T \ell_t^i(\eta_t^i)\right] + \tilde{S}_T(\hat\pi) \\ &= \mathbb{E}_{\hat\pi}\left[\sum_{t=1}^T \ell_t^i(\hat\eta^i) + S_T^i(\hat\eta^i)\right] + \tilde{S}_T(\hat\pi). \end{aligned} \tag{2.12.1}$$

The minimizer of $\sum_{t=1}^T \ell_t^i(\eta)$ over $\eta \in [0,1]$ is $\hat\eta^i = \mathcal{R}_T^+(i)/T$. Plugging this in, we find that

$$\sum_{t=1}^T \ell_t^i(\hat\eta^i) = -T\,\mathrm{B}\left(\tfrac{1}{2} + \tfrac{\mathcal{R}_T^+(i)}{2T} \| \tfrac{1}{2}\right). \tag{2.12.2}$$

Substituting (2.12.2) in (2.12.1) and reorganizing we obtain (2.5.7).

If we specialize to EW, then $\tilde{S}_T(\hat\pi) \le \mathrm{KL}(\hat\pi \| \pi)$ by the same argument as for iProd. In addition, to bound $S_T^i(\hat\eta^i)$, let $\tilde\beta(x,y)$ be the distribution on $\eta \in [-1,+1]$ such that $(1+\eta)/2$ has a $\beta(x,y)$ distribution. Then Lemma 1 and the observation that the mixability gap is at most 0 because ℓ_t^i is 1-exp-concave, together imply that

$$S_T^i(\hat\eta^i) \le \min_{Q\in\mathcal{P}} \underbrace{\left\{ \mathbb{E}_{\eta\sim Q}\left[\sum_{t=1}^T \ell_t^i(\eta)\right] + \mathrm{KL}(Q \| \tilde\beta(a,a)) \right\}}_{A(Q,i)} - \underbrace{\sum_{t=1}^T \ell_t^i(\hat\eta^i)}_{B(i)}.$$

We first rewrite $B(i)$ using (2.12.2). Then it remains to bound the term with $A(Q,i)$ in expectation under $\hat\pi$. To this end we may assume that $\mathcal{R}_T(\hat\pi) := \mathbb{E}_{\hat\pi}[\mathcal{R}_T(i)] \ge 0$

without loss of generality (otherwise (2.5.8) holds trivially). Hence

$$
\begin{aligned}
\mathbb{E}_{i\sim\hat\pi}\big[\min_{Q\in\mathcal{P}} A(Q,i)\big] &\le \min_{Q\in\mathcal{P}} \mathbb{E}_{i\sim\hat\pi}\left[A(Q,i)\right]\\
&= \min_{Q\in\mathcal{P}}\Bigg\{\mathbb{E}_{\eta\sim Q}\Big[-\frac{T+\mathcal{R}_T(\hat\pi)}{2}\ln\frac{1+\eta}{2}-\frac{T-\mathcal{R}_T(\hat\pi)}{2}\ln\frac{1-\eta}{2}-T\ln 2\Big]\\
&\qquad +\mathrm{KL}(Q\|\tilde\beta(a,a))\Bigg\}\\
&= -\ln\Big(2^T\,\mathbb{E}_{X\sim\beta(a,a)}\Big[X^{\frac{T+\mathcal{R}_T(\hat\pi)}{2}}(1-X)^{\frac{T-\mathcal{R}_T(\hat\pi)}{2}}\Big]\Big)\\
&= -\ln\left(\frac{2^T\Gamma(2a)\Gamma\big(\frac{T+\mathcal{R}_T(\hat\pi)}{2}+a\big)\Gamma\big(\frac{T-\mathcal{R}_T(\hat\pi)}{2}+a\big)}{\Gamma(a)^2\Gamma(T+2a)}\right)\\
&\le \frac{-\mathcal{R}_T(\hat\pi)^2}{2T+4a-2}+\tfrac{1}{2}\ln\frac{T+2a-1}{2a}+\ln(e\sqrt{\pi}),
\end{aligned}
$$

where we have plugged in the minimizing $Q=\tilde\beta(\frac{T+\mathcal{R}_T(\hat\pi)}{2}+a,\frac{T-\mathcal{R}_T(\hat\pi)}{2}+a)$, which has nonnegative mean under our assumption that $\mathcal{R}_T(\hat\pi)\ge 0$, and where the last inequality holds by (Orabona and Pál, 2016, Lemma 16), which applies for $a\ge 1/2$, $\mathcal{R}_T(\hat\pi)\in[-T,T]$ and $T\ge 1$.

With these regret bounds for EW, (2.5.7) specializes to

$$
\mathcal{R}_T(\hat\pi)\le\sqrt{(2T+4a-2)\left(\tfrac{1}{2}\ln\frac{T+2a-1}{2a}+\ln(e\sqrt{\pi})+\mathrm{KL}(\hat\pi\|\pi)\right)}.
$$

The result so far holds for any $a\ge\frac{1}{2}$. Plugging in the choice $a=\frac{T}{4}+\frac{1}{2}$, suggested by Orabona and Pál (2016), and using $\frac{1}{2}\ln\frac{3T}{T+2}+\ln(e\sqrt{\pi})\le 3$ completes the proof. □

2.13 Analysis of the Algorithm from Section 2.6

Let $\mathcal{W}\subset\mathbb{R}^d$ be a compact convex set. Following Bubeck et al. (2012), we assume without loss of generality that $\mathcal{W}$ is full rank, meaning that the linear combinations of $\mathcal{W}$ span $\mathbb{R}^d$ (otherwise we can express the elements of $\mathcal{W}$ in a lower dimensional space).

At trials $t=1,2,\dots,T$, the algorithm plays with a randomized choice $\boldsymbol{w}_t\in\mathcal{W}$, the adversary chooses an unobserved loss vector $\boldsymbol{g}_t$, which is not allowed to depend on the realization of $\boldsymbol{w}_t$, and the learner suffers and observes bounded loss $\langle\boldsymbol{w}_t,\boldsymbol{g}_t\rangle$. The goal is to minimize the expected regret: $\mathbb{E}[\mathcal{R}_T(\boldsymbol{u})]=\mathbb{E}\left[\sum_{t=1}^T\langle\boldsymbol{w}_t-\boldsymbol{u},\boldsymbol{g}_t\rangle\right]$ for any choice of the comparator $\boldsymbol{u}\in\mathcal{W}$. We consider EW with a fixed learning rate η and a prior distribution P_1 that is uniform over $\mathcal{W}$. At each trial t, after observing the loss $\langle \boldsymbol{w}_t, \boldsymbol{g}_t \rangle$, the algorithm constructs a random, unbiased estimate $\tilde{\boldsymbol{g}}_t$ of the loss vector $\boldsymbol{g}_t$ (described below), and uses this estimate to update the posterior. Since the projection step can be dropped (as P_1 is supported on $\mathcal{W}$), the greedy and lazy versions of EW coincide and the posterior is given by $\mathrm{d}P_t(\boldsymbol{w}) \propto \exp(-\eta \sum_{s=1}^{t-1} \langle \boldsymbol{w}, \tilde{\boldsymbol{g}}_s \rangle) \mathrm{d}\boldsymbol{w}$ for all $\boldsymbol{w} \in \mathcal{W}$. Defining $\boldsymbol{\theta}_t = -\eta \sum_{s=1}^{t-1} \tilde{\boldsymbol{g}}_s$ (with $\boldsymbol{\theta}_1 = \boldsymbol{0}$), we can concisely write:

$$\mathrm{d}P_{t+1}(\boldsymbol{w}) = e^{\langle \boldsymbol{w}, \boldsymbol{\theta}_t \rangle - F(\boldsymbol{\theta}_t)} \mathrm{d}\boldsymbol{w} \quad \forall \boldsymbol{w} \in \mathcal{W}, \qquad \text{where } F(\boldsymbol{\theta}) = \ln \int_{\mathcal{W}} e^{\langle \boldsymbol{w}, \boldsymbol{\theta} \rangle} \, \mathrm{d}\boldsymbol{w}$$

is the cumulant generating function. At trial t, the EW algorithm samples $\boldsymbol{w}_t \sim Q_t$, where $Q_t = (1-\gamma)P_t + \gamma R$ for $\gamma \in (0,1)$ is a mixture of the posterior P_t and a fixed "exploration" distribution R. The exploration distribution is chosen to be *John's exploration*, defined as follows (Bubeck et al., 2012). Let $\mathcal{K}$ be the ellipsoid of minimal volume *enclosing* $\mathcal{W}$:

$$\mathcal{K} = \{\boldsymbol{w} \in \mathbb{R}^d \colon (\boldsymbol{w} - \boldsymbol{w}_0)^\intercal \boldsymbol{H}^{-1} (\boldsymbol{w} - \boldsymbol{w}_0) \leq 1\} \tag{2.13.1}$$

for some positive definite matrix $\boldsymbol{H}$ and $\boldsymbol{w}_0 \in \mathbb{R}^d$. In what follows we assume without loss of generality that $\mathcal{W}$ is centered in the sense that $\boldsymbol{w}_0 = \boldsymbol{0}$ (otherwise all $\boldsymbol{w} \in \mathcal{W}$ need to be shifted by $\boldsymbol{w}_0$). Bubeck et al. (2012) show that one can choose $M \leq d(d+1)/2 + 1$ contact points $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_M \in \mathcal{K} \cap \mathcal{W}$, and a distribution R over these points that satisfies:

$$\mathbb{E}_{\boldsymbol{w} \sim R}[\boldsymbol{w}\boldsymbol{w}^\intercal] = \frac{1}{d}\boldsymbol{H}. \tag{2.13.2}$$

The estimate $\tilde{\boldsymbol{g}}_t$ is constructed based on the observed loss $\langle \boldsymbol{w}_t, \boldsymbol{x}_t \rangle$, by:

$$\tilde{\boldsymbol{g}}_t = \langle \boldsymbol{w}_t, \boldsymbol{g}_t \rangle \left(\mathbb{E}_{Q_t}[\boldsymbol{w}\boldsymbol{w}^\intercal]\right)^{-1} \boldsymbol{w}_t.$$

We now show the following regret bound for the resulting algorithm:

Theorem 9. *Assume the losses are bounded: $|\langle \boldsymbol{w}, \boldsymbol{g}_t \rangle| \leq 1$ for all $\boldsymbol{w} \in \mathcal{W}$ and all t. Let $\eta = \sqrt{\frac{\nu \ln T}{3dT}}$, where $\nu = O(d)$ is the self-concordant barrier parameter of F^*, and let $\gamma = \eta d$. Then the expected regret for the EW algorithm described above is bounded by*

$$\mathbb{E}[\mathcal{R}_T(\boldsymbol{u})] \leq 2\sqrt{3\nu d T \ln T} + 2 = O(d\sqrt{T \ln T}).$$

Proof. We first verify that the estimate $\tilde{\boldsymbol{g}}_t$ of $\boldsymbol{g}_t$ is unbiased:

$$\begin{aligned} \mathbb{E}_{\boldsymbol{w}_t \sim Q_t}[\tilde{\boldsymbol{g}}_t] &= \mathbb{E}_{\boldsymbol{w}_t \sim Q_t}\left[\left(\mathbb{E}_{\boldsymbol{w} \sim Q_t}[\boldsymbol{w}\boldsymbol{w}^\intercal]\right)^{-1} \boldsymbol{w}_t \langle \boldsymbol{w}_t, \boldsymbol{g}_t \rangle\right] \\ &= \left(\mathbb{E}_{\boldsymbol{w} \sim Q_t}[\boldsymbol{w}\boldsymbol{w}^\intercal]\right)^{-1} \mathbb{E}_{\boldsymbol{w}_t \sim Q_t}\left[\boldsymbol{w}_t \boldsymbol{w}_t^\intercal\right] \boldsymbol{g}_t = \boldsymbol{g}_t. \end{aligned}$$

Furthermore, due to the inclusion of the exploration distribution R, we have:

$$\mathbb{E}_{\boldsymbol{w}\sim Q_t}[\boldsymbol{w}\boldsymbol{w}^\intercal] = (1-\gamma)\,\mathbb{E}_{\boldsymbol{w}\sim P_t}[\boldsymbol{w}\boldsymbol{w}^\intercal] + \gamma\,\mathbb{E}_{\boldsymbol{w}\sim R}[\boldsymbol{w}\boldsymbol{w}^\intercal] \succeq \frac{\gamma}{d}\boldsymbol{H},$$

(where $\boldsymbol{A} \succeq \boldsymbol{B}$ means $\boldsymbol{A}-\boldsymbol{B}$ is positive semidefinite), and hence for any $\boldsymbol{u} \in \mathcal{W}$:

$$\left\langle \boldsymbol{u}, \left(\mathbb{E}_{\boldsymbol{w}\sim Q_t}[\boldsymbol{w}\boldsymbol{w}^\intercal]\right)^{-1}\boldsymbol{u}\right\rangle \leq \left\langle \boldsymbol{u}, \frac{d}{\gamma}\boldsymbol{H}^{-1}\boldsymbol{u}\right\rangle \leq \frac{d}{\gamma}, \tag{2.13.3}$$

where the last inequality is from the fact that $\mathcal{W} \subseteq \mathcal{K}$ and from the definition of $\mathcal{K}$ in (2.13.1). This, however, implies that the linear losses induced by $\tilde{\boldsymbol{g}}_t$ are bounded for any $\boldsymbol{u} \in \mathcal{W}$:

$$\begin{aligned}
&\langle \boldsymbol{u}, \tilde{\boldsymbol{g}}_t\rangle \\
&= \langle \boldsymbol{w}_t, \boldsymbol{g}_t\rangle \left\langle \boldsymbol{u}, \left(\mathbb{E}_{\boldsymbol{w}\sim Q_t}[\boldsymbol{w}\boldsymbol{w}^\intercal]\right)^{-1}\boldsymbol{w}_t\right\rangle \\
&\leq |\langle \boldsymbol{w}_t, \boldsymbol{g}_t\rangle| \left\langle \boldsymbol{w}_t, \left(\mathbb{E}_{\boldsymbol{w}\sim Q_t}[\boldsymbol{w}\boldsymbol{w}^\intercal]\right)^{-1}\boldsymbol{w}_t\right\rangle^{1/2} \left\langle \boldsymbol{u}, \left(\mathbb{E}_{\boldsymbol{w}\sim Q_t}[\boldsymbol{w}\boldsymbol{w}^\intercal]\right)^{-1}\boldsymbol{u}\right\rangle^{1/2} \\
&\leq \frac{d}{\gamma},
\end{aligned} \tag{2.13.4}$$

where the first inequality is from the Cauchy-Schwarz inequality (for positive semidefinite $\boldsymbol{A}$, $\boldsymbol{x}^\intercal \boldsymbol{A}\boldsymbol{y} \leq (\boldsymbol{x}^\intercal\boldsymbol{A}\boldsymbol{x})^{1/2}(\boldsymbol{y}^\intercal\boldsymbol{A}\boldsymbol{y})^{1/2}$), while the second inequality is due to assumption $|\langle \boldsymbol{w}, \boldsymbol{g}_t\rangle| \leq 1$ and due to (2.13.3) applied twice (first to $\boldsymbol{u}$ and then to $\boldsymbol{w}_t$).

Let $\boldsymbol{\mu}_t$ be the mean value of P_t: $\boldsymbol{\mu}_t = \mathbb{E}_{P_t}[\boldsymbol{w}]$. As a general property of exponential families or as a consequence of Theorem 3, we have $\boldsymbol{\mu}_t = \nabla F(\boldsymbol{\theta}_t)$, and $\boldsymbol{\mu}_t$ and $\boldsymbol{\theta}_t$ are conjugate parameters of the exponential family. Let us fix a comparator $\boldsymbol{u} \in \mathcal{W}$ and define $P_{\boldsymbol{u}}$ to be the member of the exponential family with cumulant generating function F that has mean value $\boldsymbol{u}$: $\mathbb{E}_{\boldsymbol{w}\sim P_{\boldsymbol{u}}}[\boldsymbol{w}] = \boldsymbol{u}$. We now apply Lemma 1 for the EW algorithm on the sequence of linear losses induced by $\tilde{\boldsymbol{g}}_1, \ldots, \tilde{\boldsymbol{g}}_T$ to get:

$$\begin{aligned}
\sum_{t=1}^T \langle \boldsymbol{\mu}_t - \boldsymbol{u}, \tilde{\boldsymbol{g}}_t\rangle &= \sum_{t=1}^T \langle \boldsymbol{\mu}_t, \tilde{\boldsymbol{g}}_t\rangle - \sum_{t=1}^T \mathbb{E}_{\boldsymbol{w}\sim P_{\boldsymbol{u}}}[\langle \boldsymbol{w}, \tilde{\boldsymbol{g}}_t\rangle] \\
&\leq \frac{1}{\eta}\mathrm{KL}(P_{\boldsymbol{u}}\|P_1) + \sum_{t=1}^T \langle \boldsymbol{\mu}_t, \tilde{\boldsymbol{g}}_t\rangle + \frac{1}{\eta}\ln \mathbb{E}_{\boldsymbol{w}\sim P_t}\left[e^{-\eta\langle \boldsymbol{w}, \tilde{\boldsymbol{g}}_t\rangle}\right]
\end{aligned}$$

(note that in this section we use $\boldsymbol{\mu}_t$ to denote the mean of P_t, while $\boldsymbol{w}_t$ is reserved for the randomized action at trial t sampled from Q_t). Since $P_{\boldsymbol{u}}$ and P_1 are members of the same exponential family, the KL-term can be re-expressed using Lemma 2:

$$\begin{aligned}\mathrm{KL}(P_{\boldsymbol{u}}\|P_1) =& D_{F^*}(\boldsymbol{u}\|\boldsymbol{\mu}_1)\\ =& F^*(\boldsymbol{u}) - F^*(\boldsymbol{\mu}_1) - \underbrace{\nabla F^*(\boldsymbol{\mu}_1)^\intercal}_{\boldsymbol{0}}(\boldsymbol{\mu} - \boldsymbol{\mu}_1)\\ =& F^*(\boldsymbol{u}) - F^*(\boldsymbol{\mu}_1),\end{aligned}$$

where we used the fact that $\boldsymbol{\mu}_1$ has conjugate parameter $\boldsymbol{\theta}_1 = \boldsymbol{0}$, and thus $\nabla F^*(\boldsymbol{\mu}_1) = \boldsymbol{\theta}_1 = \boldsymbol{0}$. To bound the mixability gap, we will now use that by assumption $\eta = \frac{\gamma}{d}$, so that by (2.13.4) we have $|\eta\langle \boldsymbol{w}, \tilde{\boldsymbol{g}}_t\rangle| \leq 1$ for any $\boldsymbol{w} \in \mathcal{W}$. Using the fact that $e^{-s} \leq 1 - s + s^2$ holds for $s \geq -1$, and combining with $\ln(1+x) \leq x$ gives:

$$\begin{aligned}&\langle \boldsymbol{\mu}_t, \tilde{\boldsymbol{g}}_t\rangle + \frac{1}{\eta}\ln \mathbb{E}_{\boldsymbol{w}\sim P_t}\left[e^{-\eta\langle \boldsymbol{w},\tilde{\boldsymbol{g}}_t\rangle}\right]\\ &\leq \langle \boldsymbol{\mu}_t, \tilde{\boldsymbol{g}}_t\rangle + \frac{1}{\eta}\ln\left(1 + \mathbb{E}_{\boldsymbol{w}\sim P_t}\left[-\eta\langle \boldsymbol{w}, \tilde{\boldsymbol{g}}_t\rangle + \eta^2\langle \boldsymbol{w}, \tilde{\boldsymbol{g}}_t\rangle^2\right]\right)\\ &\leq \underbrace{\langle \boldsymbol{\mu}_t, \tilde{\boldsymbol{g}}_t\rangle - \mathbb{E}_{\boldsymbol{w}\sim P_t}[\langle \boldsymbol{w}, \tilde{\boldsymbol{g}}_t\rangle]}_{=0} + \eta\,\mathbb{E}_{\boldsymbol{w}\sim P_t}\left[\langle \boldsymbol{w}, \tilde{\boldsymbol{g}}_t\rangle^2\right]\\ &= \eta\tilde{\boldsymbol{g}}_t^\intercal\,\mathbb{E}_{\boldsymbol{w}\sim P_t}\left[\boldsymbol{w}\boldsymbol{w}^\intercal\right]\tilde{\boldsymbol{g}}_t.\end{aligned}$$

Combining the bounds on the KL-term and the mixability gap gives:

$$\sum_{t=1}^T \langle \boldsymbol{\mu}_t - \boldsymbol{u}, \tilde{\boldsymbol{g}}_t\rangle \leq \frac{F^*(\boldsymbol{u}) - F^*(\boldsymbol{\mu}_1)}{\eta} + \eta\sum_{t=1}^T \tilde{\boldsymbol{g}}_t^\intercal\,\mathbb{E}_{\boldsymbol{w}\sim P_t}\left[\boldsymbol{w}\boldsymbol{w}^\intercal\right]\tilde{\boldsymbol{g}}_t. \qquad (2.13.5)$$

We can use this result to bound the regret of the original algorithm in the following way. First, note that:

$$\begin{aligned}\mathbb{E}_{\boldsymbol{w}_t\sim Q_t}\left[\langle \boldsymbol{w}_t - \boldsymbol{u}, \boldsymbol{g}_t\rangle\right] &= \gamma\langle \mathbb{E}_{\boldsymbol{w}_t\sim R}[\boldsymbol{w}_t] - \boldsymbol{u}, \boldsymbol{g}_t\rangle + (1-\gamma)\langle \mathbb{E}_{\boldsymbol{w}_t\sim P_t}[\boldsymbol{w}_t] - \boldsymbol{u}, \boldsymbol{g}_t\rangle\\ &\leq 2\gamma + (1-\gamma)\langle \boldsymbol{\mu}_t - \boldsymbol{u}, \boldsymbol{g}_t\rangle\\ &= 2\gamma + (1-\gamma)\,\mathbb{E}_{\boldsymbol{w}_t\sim Q_t}\left[\langle \boldsymbol{\mu}_t - \boldsymbol{u}, \tilde{\boldsymbol{g}}_t\rangle\right],\end{aligned}$$

where the random quantity in the last expectation is $\tilde{\boldsymbol{g}}_t$, because it depends on $\boldsymbol{w}_t$.

Therefore:

$$
\begin{aligned}
&\sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{w}_t \sim Q_t}[\langle \boldsymbol{w}_t - \boldsymbol{u}, \boldsymbol{g}_t \rangle] \\
&\le 2\gamma T + (1-\gamma) \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{w}_t \sim Q_t}\left[\langle \boldsymbol{\mu}_t - \boldsymbol{u}, \tilde{\boldsymbol{g}}_t \rangle\right] \\
&\le 2\gamma T + \frac{F^*(\boldsymbol{u}) - F^*(\boldsymbol{\mu}_1)}{\eta} + \eta(1-\gamma) \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{w}_t \sim Q_t}\left[\tilde{\boldsymbol{g}}_t^\intercal \, \mathbb{E}_{\boldsymbol{w} \sim P_t}\left[\boldsymbol{w}\boldsymbol{w}^\intercal\right] \tilde{\boldsymbol{g}}_t\right] \\
&\le 2\gamma T + \frac{F^*(\boldsymbol{u}) - F^*(\boldsymbol{\mu}_1)}{\eta} + \eta \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{w}_t \sim Q_t}\left[\tilde{\boldsymbol{g}}_t^\intercal \, \mathbb{E}_{\boldsymbol{w} \sim Q_t}\left[\boldsymbol{w}\boldsymbol{w}^\intercal\right] \tilde{\boldsymbol{g}}_t\right], \qquad (2.13.6)
\end{aligned}
$$

where the second inequality is from (2.13.5), while the last inequality is due to:

$$
\mathbb{E}_{\boldsymbol{w} \sim Q_t}[\boldsymbol{w}\boldsymbol{w}^\intercal] = (1-\gamma)\,\mathbb{E}_{\boldsymbol{w} \sim P_t}[\boldsymbol{w}\boldsymbol{w}^\intercal] + \gamma\,\mathbb{E}_{\boldsymbol{w} \sim R}[\boldsymbol{w}\boldsymbol{w}^\intercal] \succeq (1-\gamma)\,\mathbb{E}_{\boldsymbol{w} \sim P_t}[\boldsymbol{w}\boldsymbol{w}^\intercal].
$$

Using the definition of $\tilde{\boldsymbol{g}}_t$ and $\langle \boldsymbol{w}_t, \boldsymbol{g}_t \rangle^2 \le 1$, we further bound:

$$
\begin{aligned}
&\mathbb{E}_{\boldsymbol{w}_t \sim Q_t}\left[\tilde{\boldsymbol{g}}_t^\intercal \, \mathbb{E}_{\boldsymbol{w} \sim Q_t}\left[\boldsymbol{w}\boldsymbol{w}^\intercal\right] \tilde{\boldsymbol{g}}_t\right] \\
&\le \mathbb{E}_{\boldsymbol{w}_t \sim Q_t}\left[\boldsymbol{w}_t^\intercal \left(\mathbb{E}_{\boldsymbol{w} \sim Q_t}[\boldsymbol{w}\boldsymbol{w}^\intercal]\right)^{-1} \mathbb{E}_{\boldsymbol{w} \sim Q_t}\left[\boldsymbol{w}\boldsymbol{w}^\intercal\right] \left(\mathbb{E}_{\boldsymbol{w} \sim Q_t}[\boldsymbol{w}\boldsymbol{w}^\intercal]\right)^{-1} \boldsymbol{w}_t\right] \\
&= \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{w}_t \sim Q_t}\left[\mathrm{Tr}\left(\left(\mathbb{E}_{\boldsymbol{w} \sim Q_t}[\boldsymbol{w}\boldsymbol{w}^\intercal]\right)^{-1} \boldsymbol{w}_t \boldsymbol{w}_t^\intercal\right)\right] \\
&= \sum_{t=1}^{T} \mathrm{Tr}\left(\boldsymbol{I}\right) = Td.
\end{aligned}
$$

Plugging the above into (2.13.6) and taking expectation with respect to the randomness of the algorithm results in the following bound on the expected regret:

$$
\mathbb{E}[\mathcal{R}_T(\boldsymbol{u})] = \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{w}_t \sim Q_t}[\langle \boldsymbol{w}_t - \boldsymbol{u}, \boldsymbol{g}_t \rangle]\right] \le 2\gamma T + \frac{F^*(\boldsymbol{u}) - F^*(\boldsymbol{\mu}_1)}{\eta} + \eta T d.
$$

What is left to bound is $F^*(\boldsymbol{u}) - F^*(\boldsymbol{\mu}_1)$. To this end, define the Minkowski function (Abernethy et al., 2012) on $\mathcal{W}$ as:

$$
\pi_{\boldsymbol{\mu}}(\boldsymbol{w}) = \inf\{t \ge 0 \colon \boldsymbol{\mu} + t^{-1}(\boldsymbol{w} - \boldsymbol{\mu}) \in \mathcal{W}\}.
$$

Bubeck and Eldan (2015) show that F^* is a ν-self concordant barrier on $\mathcal{W}$ with $\nu = O(d)$. Using this property and Theorem 2.2 from Abernethy et al. (2012) we get:

$$
F^*(\boldsymbol{u}) - F^*(\boldsymbol{\mu}_1) \le \nu \ln\left(\frac{1}{1 - \pi_{\boldsymbol{\mu}_1}(\boldsymbol{u})}\right).
$$

If $\boldsymbol{u}$ is such that $\pi_{\boldsymbol{\mu}_1}(\boldsymbol{u}) \leq 1 - \frac{1}{T}$, then $F^*(\boldsymbol{u}) - F^*(\boldsymbol{\mu}_1) \leq \nu \ln T$. On the other hand, if $\pi_{\boldsymbol{\mu}_1}(\boldsymbol{u}) \leq 1 - \frac{1}{T}$, we define a new comparator $\boldsymbol{u}' = (1 - \frac{1}{T})\boldsymbol{u} + \frac{1}{T}\boldsymbol{\mu}_1$, for which $\pi_{\boldsymbol{\mu}_1}(\boldsymbol{u}') \leq 1 - \frac{1}{T}$ (Abernethy et al., 2012), and use the regret bound above for $\boldsymbol{u}'$ to get:

$$\begin{aligned}\mathbb{E}[\mathcal{R}_T(\boldsymbol{u})] &= \mathbb{E}[\mathcal{R}_T(\boldsymbol{u}')] + \sum_{t=1}^{T} \langle \boldsymbol{u}' - \boldsymbol{u}, \boldsymbol{g}_t \rangle = \mathbb{E}[\mathcal{R}_T(\boldsymbol{u}')] + \frac{1}{T}\sum_{t=1}^{T} \langle \boldsymbol{\mu}_1 - \boldsymbol{u}, \boldsymbol{g}_t \rangle \\ &\leq 2\gamma T + \frac{F^*(\boldsymbol{u}') - F^*(\boldsymbol{\mu}_1)}{\eta} + \eta T d + 2 \leq 2\gamma T + \frac{\nu \ln T}{\eta} + \eta T d + 2.\end{aligned}$$

Recalling that $\gamma = \eta d$ and tuning $\eta = \sqrt{\frac{\nu \ln T}{3dT}}$ gives the claimed bound. □

CHAPTER 3

User-Specified Local Differential Privacy in Unconstrained Adaptive Online Learning

This chapter is based on Van der Hoeven, D. (2019). User-specified local differential privacy in unconstrained adaptive online learning. In *Advances in Neural Information Processing Systems 32*, pages 14103–14112.

Abstract

Local differential privacy is a strong notion of privacy in which the provider of the data guarantees privacy by perturbing the data with random noise. In the standard application of local differential privacy the distribution of the noise is constant and known by the learner. In this chapter we generalize this approach by allowing the provider of the data to choose the distribution of the noise without disclosing any parameters of the distribution to the learner, under the constraint that the distribution is symmetrical. We consider this problem in the unconstrained Online Convex Optimization setting with noisy feedback. In this setting the learner receives the subgradient of a loss function, perturbed by noise, and aims to achieve sublinear regret with respect to some competitor, without constraints on the norm of the competitor. We derive the first algorithms that have adaptive regret bounds in this setting, i.e. our algorithms adapt to the unknown competitor norm, unknown noise, and unknown sum of the norms of the subgradients, matching state of the art bounds in all cases.

3.1 Introduction

In learning, a natural tension exists between learners and the providers of data. The learner aims to make optimal use of the data, perhaps even at the cost of the privacy of the providers. To nevertheless ensure sufficient privacy the provider can add random noise to the data that he sends to the learner. This idea is called ϵ-local differential privacy (Wasserman and Zhou, 2010; Duchi et al., 2014) and the standard implementation has constant ϵ for all providers. However, not all providers care equivalently about their privacy (Song et al., 2015). Some providers may wish to aid the learner in making optimal use of their data, while other providers value their privacy over helping the learner. For instance, celebrities might care more for their privacy than others because they want to preserve the privacy they have left. To complicate things further, the providers of the data may not wish to reveal how much they care about their privacy, because when privacy levels differ between providers these privacy levels become privacy sensitive themselves. Furthermore, not all parts of the data are equally privacy sensitive. For example, tweets are already publicly available, but browsing history may contain sensitive information that should be kept private. To capture these varying privacy constraints we allow each provider to choose how much noise is added for each dimension of the data.

In this chapter, we consider these problems in the Online Convex Optimization (OCO) setting (Hazan et al., 2016) with local differential privacy guarantees. The OCO framework is a popular and successful framework to design and analyse many algorithms used to train machine learning models. The OCO setting proceeds in rounds $t = 1, \dots, T$. In a given round t the learner is to provide a prediction $\boldsymbol{w}_t \in \mathbb{R}^d$. An adversary then chooses a convex loss function ℓ_t and sends a subgradient $\boldsymbol{g}_t \in \partial \ell_t(\boldsymbol{w}_t)$ to the learner. We work with an unconstrained domain for $\boldsymbol{w}$, which has recently grown in popularity (see McMahan and Orabona (2014); Foster et al. (2015); Orabona and Pál (2016); Foster et al. (2017); Cutkosky and Boahen (2017); Kotłowski (2017); Cutkosky and Orabona (2018); Foster et al. (2018b); Jun and Orabona (2019)). We aim to develop online learning methods that make the best use of data providers who wish to help the learner while at the same time guaranteeing the desired level of privacy for providers that care about their privacy, without knowing how much each provider cares for their privacy.

We consider the local differential privacy model with varying levels of privacy unknown to the learner. Differential privacy (Dwork and Roth, 2014) is a privacy model that is used in many recent machine-learning applications. The local differential privacy model is a variant of differential privacy in which the learner can only access the data of the provider via noisy estimates (Wasserman and Zhou, 2010;

Duchi et al., 2014). The local differential privacy model with varying levels of privacy appeared before in Song et al. (2015), but with known levels of noise and only two levels of noise.

Learning in our setting is modelled by the OCO framework with noisy estimates of the subgradient (see also Jun and Orabona (2019)). To ensure local differential privacy the provider adds zero-mean noise $\boldsymbol{\xi}_t \in \mathbb{R}^d$ to the subgradient $\boldsymbol{g}_t$. The learner then receives the perturbed subgradient $\tilde{\boldsymbol{g}}_t = \boldsymbol{g}_t + \boldsymbol{\xi}_t$. We allow each $\boldsymbol{\xi}_t$ to follow a different distribution each round to satisfy different privacy guarantees. In the standard OCO framework the goal of the learner is to minimize the *regret* with respect to some parameter $\boldsymbol{u} \in \mathbb{R}^d$:

$$\mathcal{R}_T(\boldsymbol{u}) = \sum_{t=1}^{T} \left(\ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{u})\right).$$

However, since the learner receives perturbed subgradients we consider the expected regret $\mathbb{E}[\mathcal{R}(\boldsymbol{u})]$, where the expectation is over the randomness in $\boldsymbol{w}_t$ due to the noisy subgradients. The setting will be formally introduced in Section 3.2. Because $\tilde{\boldsymbol{g}}_t \in \mathbb{R}^d$, standard algorithms for unconstrained domains do not work since they require bounded $\tilde{\boldsymbol{g}}_t$. Initial work in this setting by Jun and Orabona (2019) was motivated by a lower bound of Cutkosky and Boahen (2017), which shows that one can suffer an exponential penalty when both the domain and subgradients are unbounded. They replace the boundedness assumption on $\tilde{\boldsymbol{g}}_t$ by a boundedness assumption on $\mathbb{E}[\tilde{\boldsymbol{g}}_t]$ and an assumption on the tails of the noise distribution. Jun and Orabona (2019) achieved expected regret guarantees of $O(\|\boldsymbol{u}\|\sqrt{(G^2+\sigma^2)T\ln(1+\|\boldsymbol{u}\|T)})$, where σ^2 is a uniform upper bound on $\mathbb{E}[\|\boldsymbol{\xi}_t\|_\star^2]$, G^2 is a uniform upper bound on $\|\boldsymbol{g}_t\|_\star^2$, and $\|\cdot\|$ and $\|\cdot\|_\star$ are dual norms. This bound is useful when the distribution of the noise is constant and known and an adversary selects $\boldsymbol{g}_t$. We derive an algorithm that satisfies

$$E[\mathcal{R}_T(\boldsymbol{u})] = O\left(\|\boldsymbol{u}\|\sqrt{(G^2T + \sum_{t=1}^{T}\sigma_t^2)\ln(1+\|\boldsymbol{u}\|T))}\right), \tag{3.1.1}$$

where $\sigma_t^2 = \mathbb{E}[\|\boldsymbol{\xi}_t\|_\star^2]$. This bound can be smaller in cases where only a few σ_t are large but most are small, for example when only few providers have privacy requirements. In fact, we will prove something stronger than (3.1.1):

$$\mathbb{E}[\mathcal{R}_T(\boldsymbol{u})] = O\left(\mathbb{E}[\|\boldsymbol{u}\|\sqrt{\sum_{t=1}^{T}\|\tilde{\boldsymbol{g}}_t\|_\star^2\ln(1+\|\boldsymbol{u}\|T))]}\right), \tag{3.1.2}$$

which implies (3.1.1) via Jensen's inequality and $\mathbb{E}[\|\tilde{\boldsymbol{g}}_t\|_\star^2] \leq 3\,\mathbb{E}[\|\boldsymbol{\xi}_t\|_\star^2] + 3\,\mathbb{E}[\|\boldsymbol{g}_t\|_\star^2]$. This bound was motivated by work in the noiseless setting, where $O(\|\boldsymbol{u}\|\sqrt{\sum_{t=1}^T \|\boldsymbol{g}_t\|_\star^2 \ln(1+\|\boldsymbol{u}\|T)})$ bounds are possible (Cutkosky and Orabona, 2018). With these type of bounds, when the sum of the squared norms of the subgradients is small the regret is also small. To achieve (3.1.2) we require two assumptions: bounded $\|\boldsymbol{g}_t\|_\star$ and zero-mean symmetrical noise $\boldsymbol{\xi}_t$. The assumption on $\boldsymbol{g}_t$ is common in standard OCO. The symmetrical noise assumption is satisfied for common mechanisms to ensure local differential privacy. The dependence on $\mathbb{E}[\|\boldsymbol{\xi}_t\|_\star^2]$ and $\mathbb{E}[\|\boldsymbol{g}_t\|_\star^2]$ is unimprovable, which is shown by the lower bound for this setting by Jun and Orabona (2019).

The algorithms in this chapter are built using the recently developed wealth-regret duality approach (Mcmahan and Streeter, 2012). We provide two algorithms. The first achieves the bound in (3.1.2). The second algorithm satisfies (3.1.2) for each dimension separately. This second algorithm can exploit sparse privacy structures, which combined with sparse subgradients yields low expected regret bounds.

Contributions We extend the known results in several directions. Many common local differential privacy applications use symmetric additive noise (Laplace mechanism, normal mechanism). We use the symmetry of the noise to adapt to unknown levels of privacy and achieve adaptive expected regret bounds. We also adapt to dimension specific privacy requirements, again without requiring knowledge of the structure of the noise other than symmetry in each dimension. Our algorithms interpolate between no noise and maximum noise, matching state of the art bounds in both cases. This can reduce the cost of privacy in some cases, outlined in Section 3.4. Our work partially answers two problems left open by Jun and Orabona (2019). The first question asks whether or not data-dependent bounds are possible in the noisy OCO setting, which we answer affirmatively. The second question is how to adapt to different levels of noise without using extra parameters compared to the noiseless setting, which we do for symmetric noise.

Related work There has been significant work on unconstrained and adaptive methods in OCO with noiseless subgradients $\boldsymbol{g}_t$ (Foster et al., 2015; Orabona and Pál, 2016; Foster et al., 2017; Cutkosky and Boahen, 2017; Kotłowski, 2017; Cutkosky and Orabona, 2018; Foster et al., 2018b). However, these results do not extend to the setting with noisy unbounded subgradients $\tilde{\boldsymbol{g}}_t$, which is possible with our work. For bounded domains regret bounds of $O(D\sqrt{\sum_{t=1}^T \|\tilde{\boldsymbol{g}}_t\|_\star^2})$ are possible without knowledge of the noise (Duchi et al., 2011; Orabona and Pál, 2018), where D is an upper bound on $\|\boldsymbol{u}\|$. However, these bounds do not adapt to unknown $\|\boldsymbol{u}\|$, which may be costly for large D but small $\|\boldsymbol{u}\|$. We provide an algorithm that both scales with $\|\boldsymbol{u}\|$ instead of D and does not require knowledge of the noise.

There is a body of literature in the differential privacy setting with online feedback (Jain et al., 2012; Jain and Thakurta, 2014; Thakurta and Smith, 2013; Agarwal and Singh, 2017; Abernethy et al., 2019). In this chapter we consider *local* differential privacy (Wasserman and Zhou, 2010; Duchi et al., 2014), which is a stronger notion of privacy than differential privacy. Duchi et al. (2014) provide an algorithm with constant local differential privacy that learns by using SGD. (Song et al., 2015) derive how to use knowledge of several levels of local differential privacy for SGD, but only with two different levels of noise. Jun and Orabona (2019) consider local privacy with an unbounded domain and constant noise. With knowledge of the noise it is possible to extend the results of Jun and Orabona (2019) to achieve (3.1.1), but not (3.1.2).

Outline In Section 3.2 we introduce our problem formally and introduce the key techniques. In Section 3.3 we derive a one-dimensional algorithm that achieves our goals, which we use in a black-box reduction in Section 3.3.1 and we apply it coordinate-wise in Section 3.3.2. Section 3.4 contains two scenarios in which our new algorithm achieves improvements compared to current algorithms. Finally, in Section 3.5 we present our conclusions.

3.2 Problem Formulation and Preliminaries

In this Section we describe our notation, introduce the version of local differential privacy we use, briefly introduce the OCO setting with noisy subgradients, and provide some background to the reward-regret duality paradigm.

Notation. A random variable $\boldsymbol{x}$ is called symmetric if the density function ρ of the random variable $\boldsymbol{z} = \boldsymbol{x} - \mathbb{E}[\boldsymbol{x}]$ satisfies $\rho(\boldsymbol{z}) = \rho(-\boldsymbol{z})$. The inner product between vectors $\boldsymbol{g} \in \mathbb{R}^d$ and $\boldsymbol{w} \in \mathbb{R}^d$ is denoted by $\langle \boldsymbol{w}, \boldsymbol{g} \rangle$. The Fenchel conjugate of a convex function F, $F^\star$ is defined as $F^\star(\boldsymbol{w}) = \sup_{\boldsymbol{g}} \langle \boldsymbol{w}, \boldsymbol{g} \rangle - F(\boldsymbol{g})$. $\|\cdot\|$ denotes a norm and $\|\boldsymbol{g}\|_\star = \sup_{\boldsymbol{w}:\|\boldsymbol{w}\|\leq 1} \langle \boldsymbol{w}, \boldsymbol{g} \rangle$ denotes the dual norm. $g_{t,j}$ indicates the j^{th} component of vector $\boldsymbol{g}_t$.

3.2.1 User-Specified Local Differential Privacy

In the local differential privacy setting each datum is kept private from the learner. The standard definition of local privacy requires a randomiser R that perturbs $\boldsymbol{g}_t$ with random noise $\boldsymbol{\xi}_t$, where $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_T$ are independently distributed (Wasserman and Zhou, 2010; Kasiviswanathan et al., 2011; Duchi et al., 2014). The amount of perturbation is controlled by ϵ, where smaller ϵ means more privacy. We allow the provider to specify his desired level of privacy, so in a given round t we have ϵ_t-local differential privacy.

Definition 1. *[Duchi et al. (2014)] Let $A = (X_1, \ldots, X_T)$ be a sensitive dataset where each $X_t \in A$ corresponds to data about individual t. A randomiser R which outputs a disguised version of $S = (U_1, \ldots, U_T)$ of A is said to provide ϵ-local differential privacy to individual t, if for all $x, x' \in A$ and for all $S \subseteq \mathcal{S}$,*

$$\Pr(U_t \in S | X_t = x) \leq \exp(\epsilon) \Pr(U_t \in S | X_t = x').$$

In this chapter we make use of randomisers of the form $R_t(\boldsymbol{g}_t) = \boldsymbol{g}_t + \boldsymbol{\xi}_t$, where $\boldsymbol{\xi}_t$ is generated by a zero-mean symmetrical distribution ρ_t. A common choice for ρ_t is $\rho_t(\boldsymbol{z}) \propto \exp(-\frac{\epsilon_t}{2}\|\boldsymbol{z}\|)$ (Song et al., 2015). This randomiser is ϵ_t-local differentially private for $\|\boldsymbol{g}_t\| \leq 1$ (Song et al., 2015, Theorem 1). We use a small variation of this randomiser, which we call the local Laplace randomiser: $\rho_t(\boldsymbol{z}) \propto \exp(-\sum_{j=1}^d \frac{\tau_{t,j}}{2}|\boldsymbol{z}_j|)$, where $\sum_{j=1}^d \tau_{t,j} = \epsilon_t$, $\tau_{t,j} \geq 0$. The following result shows that the local Laplace randomiser preserves ϵ_t-local differential privacy.

Lemma 4. *Suppose $|g_{t,j}| \leq 1$, then the local Laplace randomiser is ϵ_t-local differentially private, where $\epsilon_t = \sum_{j=1}^d \tau_{t,j}$.*

The proof follows from applying Theorem 1 of Song et al. (2015) to each dimension and summing the $\tau_{t,j}$. For completeness the proof is provided in Section 3.6. This randomiser is the Laplace randomiser (Dwork and Roth, 2014) applied to each dimension with a possibly different ϵ per dimension. The local Laplace randomiser gives the user more control over the details of the privacy guarantees: with the local Laplace randomiser each dimension j is $\tau_{t,j}$-local differentially private. This can also lead to lower regret in some cases, of which we give an example in Section 3.4.

3.2.2 Online Convex Optimization with Noisy Subgradients

The analysis of many efficient online learning tools has been influenced by the Online Convex Optimization framework. As mentioned in the introduction, the OCO setting with noisy subgradients proceeds in rounds $t = 1, \ldots, T$. In each round t

1. The learner sends $\boldsymbol{w}_t \in \mathbb{R}^d$ to the provider of the t^{th} subgradient.

2. The provider samples $\boldsymbol{\xi}_t$ from zero-mean and symmetrical ρ_t and computes subgradient $\boldsymbol{g}_t \in \partial \ell_t(\boldsymbol{w}_t)$, where $\|\boldsymbol{g}_t\|_\star \leq G$.

3. The provider sends $\tilde{\boldsymbol{g}}_t = \boldsymbol{g}_t + \boldsymbol{\xi}_t \in \mathbb{R}^d$ to the learner.

This protocol is a slight adaptation of the protocol of Duchi et al. (2014), where we allow a different ρ_t in each round t instead of using a constant ρ. In each round the provider only sends $\tilde{\boldsymbol{g}}_t$ to the learner. The learner has no information about ρ_t other than that ρ_t is symmetrical and zero-mean. Also note that ρ_t is allowed to change with each round t, complicating things even further. Since the feedback the learner receives is random we are interested in the expected regret. To bound the expected regret we upper bound the losses by their tangents:

$$\mathbb{E}[\mathcal{R}_T(\boldsymbol{u})] \leq \mathbb{E}[\sum_{t=1}^{T} \langle \boldsymbol{w}_t - \boldsymbol{u}, \boldsymbol{g}_t \rangle] = \mathbb{E}[\sum_{t=1}^{T} \langle \boldsymbol{w}_t - \boldsymbol{u}, \tilde{\boldsymbol{g}}_t \rangle], \tag{3.2.1}$$

where the equality holds because of the law of total expectation. The analysis focusses on bounding the r.h.s of (3.2.1), which is a standard approach in OCO. In the following we introduce a recently popularized method to control the regret when $\boldsymbol{w}_t$ and $\boldsymbol{u}$ are unbounded.

3.2.3 Reward Regret Duality

In this Section we introduce the main technical workhorse in this chapter: the reward regret duality (McMahan and Orabona, 2014, Theorem 1). Informally, for noiseless $\boldsymbol{g}_t$, suppose we are able to guarantee $-\sum_{t=1}^T \langle \boldsymbol{w}_t, \boldsymbol{g}_t \rangle \geq F_T(-\sum_{t=1}^T \boldsymbol{g}_t) - c_T$ for a convex F_T and $c_T \in \mathbb{R}$. We will refer to F_T as the potential function. Here, $-\sum_{t=1}^T \langle \boldsymbol{w}_t, \boldsymbol{g}_t \rangle$ is seen as the reward. By Fenchel's inequality we have $F_T(-\sum_{t=1}^T \boldsymbol{g}_t) \geq -F_T^\star(\boldsymbol{u}) - \sum_{t=1}^T \langle \boldsymbol{u}, \boldsymbol{g}_t \rangle$, which gives us a bound on the regret after using that $-\sum_{t=1}^T \langle \boldsymbol{w}_t, \boldsymbol{g}_t \rangle \geq F_T(-\sum_{t=1}^T \boldsymbol{g}_t) - c_T$ and reordering the terms. For noisy $\tilde{\boldsymbol{g}}_t$, the formal result is found in the following lemma (see also Theorem 3 of Jun and Orabona (2019)).

Lemma 5. *If $-\mathbb{E}[\sum_{t=1}^T \langle \boldsymbol{w}_t, \boldsymbol{g}_t \rangle] \geq \mathbb{E}[F_T(-\sum_{t=1}^T \tilde{\boldsymbol{g}}_t) - c_T]$ for some convex function F_T and $c_T \in \mathbb{R}$, then $\mathbb{E}[\mathcal{R}_T(\boldsymbol{u})] \leq E[c_T] + F_T^\star(\boldsymbol{u})$.*

Proof. From the definition of Fenchel conjugates we have $\mathbb{E}[F_T(-\sum_{t=1}^T \tilde{\boldsymbol{g}}_t)] \geq \mathbb{E}[-F_T^\star(\boldsymbol{u}) - \sum_{t=1}^T \langle \boldsymbol{u}, \tilde{\boldsymbol{g}}_t \rangle] = -F_T^\star(\boldsymbol{u}) - \sum_{t=1}^T \langle \boldsymbol{u}, \boldsymbol{g}_t \rangle$. Using $-\mathbb{E}[\sum_{t=1}^T \langle \boldsymbol{w}_t, \boldsymbol{g}_t \rangle] \geq \mathbb{E}[F_T(-\sum_{t=1}^T \tilde{\boldsymbol{g}}_t) - c_T]$ and reordering the terms completes the proof. □

The difficulty lies in finding a suitable F_T and c_T. For example, we could use gradient descent with learning rate η to find $F_T(-\sum_{t=1}^T \tilde{\boldsymbol{g}}_t) = \frac{\eta}{2}\|\sum_{t=1}^T \tilde{\boldsymbol{g}}_t\|_2^2$ and $c_T = \sum_{t=1}^T \frac{\eta}{2}\|\tilde{\boldsymbol{g}}_t\|_2^2$. However, it would be impossible to tune η optimally due to the dependence on the unknown $\boldsymbol{u}$ in $F_T^*(\boldsymbol{u}) = \frac{1}{2\eta}\|\boldsymbol{u}\|_2^2$. For noiseless subgradients $\boldsymbol{g}_t$ (Cutkosky and Orabona, 2018) provide a route to find a suitable F_T, with a constant c_T. Jun and Orabona (2019) extend this idea to noisy subgradients $\tilde{\boldsymbol{g}}_t$: one needs to find an F_t, F_{t-1}, and $\boldsymbol{w}_t$ that satisfy $F_{t-1}(\boldsymbol{x}) - \langle \boldsymbol{w}_t, \boldsymbol{g}_t \rangle \geq \mathbb{E}_{\tilde{\boldsymbol{g}}_t}[F_t(\boldsymbol{x} - \tilde{\boldsymbol{g}}_t)]$. By assuming that $-\mathbb{E}[\sum_{s=1}^{t} \langle \boldsymbol{w}_s, \boldsymbol{g}_s \rangle] \geq \mathbb{E}[F_t(-\sum_{s=1}^{t} \tilde{\boldsymbol{g}}_s)]$ holds one can show that if F_t and F_{t-1} satisfy $F_{t-1}(\boldsymbol{x}) - \langle \boldsymbol{w}_t, \boldsymbol{g}_t \rangle \geq \mathbb{E}_{\tilde{\boldsymbol{g}}_t}[F_t(\boldsymbol{x} - \tilde{\boldsymbol{g}}_t)]$, then $-\mathbb{E}[\sum_{t=1}^{T} \langle \boldsymbol{w}_t, \boldsymbol{g}_t \rangle] \geq \mathbb{E}[F_T(-\sum_{t=1}^{T} \tilde{\boldsymbol{g}}_t)]$ holds by induction. The result is given in the following lemma, of which the proof can be found in Section 3.6.

Lemma 6. *Suppose that* $F_{t-1}(\boldsymbol{x}) - \langle \boldsymbol{w}_t, \boldsymbol{g}_t \rangle \geq \mathbb{E}_{\tilde{\boldsymbol{g}}_t}[F_t(\boldsymbol{x} - \tilde{\boldsymbol{g}}_t)]$ *holds for all t, then*

$$-\mathbb{E}[\sum_{t=1}^{T} \langle \boldsymbol{w}_t, \boldsymbol{g}_t \rangle] \geq \mathbb{E}[F_T(-\sum_{t=1}^{T} \tilde{\boldsymbol{g}}_t)].$$

3.3 One-Dimensional Private Adaptive Potential Function

Algorithm 1 Local Differentially Private Adaptive Potential Function

```
Input: $G$ such that $|\mathbb{E}[\tilde{g}_t]| \leq G$ and prior $P$ on $v \in [-\frac{1}{5G}, \frac{1}{5G}]$.
for $t = 1, \ldots, T$ do
    Play $w_t = \mathbb{E}_{v \sim P}[v \exp(-\sum_{s=1}^{t-1} (v\tilde{g}_s + (v\tilde{g}_s)^2))]$.
    Receive symmetric $\tilde{g}_t \in \mathbb{R}$.
end for
```

In this Section we derive a suitable potential function for a one-dimensional problem. In the remainder of this chapter we use this one-dimensional potential to derive new algorithms. To derive our one-dimensional potential function we we rely on a property of symmetric random variables with bounded means. The following Lemma is key deriving our potential function F_T.

Lemma 7. *Suppose* $\boldsymbol{x}$ *is a symmetrical random variable with* $|\mathbb{E}[\langle \boldsymbol{v}, \boldsymbol{x} \rangle]| \leq \frac{1}{5}$ *for some* $\boldsymbol{v}$. *Then* $\mathbb{E}[\exp(\langle \boldsymbol{v}, \boldsymbol{x} \rangle - \langle \boldsymbol{v}, \boldsymbol{x} \rangle^2)] \leq 1 + \mathbb{E}[\langle \boldsymbol{v}, \boldsymbol{x} \rangle]$.

The proof of Lemma 7 can be found in Section 3.7. We can now use Lemma 7 to derive a one-dimensional potential function. Suppose $\tilde{g}_t \in \mathbb{R}$ is a symmetrical random variable with $|\mathbb{E}[\tilde{g}_t]| \leq G$. Then $v\tilde{g}_t$ with $v \in [-\frac{1}{5G}, \frac{1}{5G}]$ satisfies the assumptions in Lemma 7. Multiplying the lower bound of Lemma 7 for $1 - \mathbb{E}[v\tilde{g}_t]$, for $t = 1, \ldots, T$, yields a potential function via Lemma 6. The potential we find is

$$F_t(-\sum_{s=1}^{t} \tilde{g}_s) = \mathbb{E}_{v\sim P}[\exp(-\sum_{s=1}^{t} \big(v\tilde{g}_s + (v\tilde{g}_s)^2\big)) - 1], \tag{3.3.1}$$

where P is an (improper) prior on $v \in [-\frac{1}{5G}, \frac{1}{5G}]$, the first expectation is over $\tilde{g}_1, \ldots, \tilde{g}_t$, and $F_0(0) = 0$. This kind of potential function has been used before by Chernov and Vovk (2010); Koolen and Van Erven (2015); Jun and Orabona (2019). The novelty in this particular potential function is that it allows for the incorporation of the symmetrical noise in the analysis. The $\sum_{s=1}^{t}(v\tilde{g}_s)^2$ term is unique to our potential function and allows us to derive adaptive regret bounds for unconstrained u. Note that the $c_T = 1$ term has moved inside the definition of F_T. While this does not influence the analysis for proper priors it does influence the analysis for improper priors. The corresponding prediction strategy is given by

$$w_t = \mathbb{E}_{v\sim P}[v \exp(-\sum_{s=1}^{t-1} \big(v\tilde{g}_s + (v\tilde{g}_s)^2\big)]. \tag{3.3.2}$$

Algorithm 1 summarizes the strategy. Note that Algorithm 1 does not require any extra parameters compared to the setting with noiseless subgradients.

The following result shows that F_T defined by (3.3.1) and w_t defined by (3.3.2) satisfy our assumptions.

Lemma 8. *Suppose $\tilde{g}_t$ is a symmetrical random variable with $|\mathbb{E}[\tilde{g}_t]| \leq G$. Then F_t defined by* (3.3.1) *and w_t defined by* (3.3.2) *satisfy $\mathbb{E}_{\tilde{g}_t}[F_t(-\sum_{s=1}^{t} \tilde{g}_s)] \leq F_{t-1}(-\sum_{s=1}^{t-1} \tilde{g}_s) - w_t \mathbb{E}[\tilde{g}_t]$.*

The proof follows from an application of Lemma 7 and can be found in Section 3.7. We consider two types of priors. The first type are proper priors that are of the form:

$$\frac{dP(v)}{dv} = \frac{\nu(v)\exp(-bv^2)}{Z}, \tag{3.3.3}$$

Where $b \geq 0$, $\nu : [-\frac{1}{5G}, \frac{1}{5G}] \mapsto \mathbb{R}_+$, and $Z = \int_{-\frac{1}{5G}}^{\frac{1}{5G}} \nu(v)e^{-bv^2}dv$ is a normalizing constant. This captures several priors used in literature, including the conjugate prior $\frac{dP}{dv} = \frac{\exp(-bv^2)}{Z}$ (Koolen and Van Erven, 2015), a variant of the *CV* prior $\frac{dP}{dv} = \frac{1}{Z|v|\ln(|v|)^2}$ (for $G > \frac{1}{5}$), (Chernov and Vovk, 2010; Koolen and Van Erven, 2015), and the uniform prior on $[-\frac{1}{5G}, \frac{1}{5G}]$ (Jun and Orabona, 2019).

The second type of prior is an improper prior: $\frac{dP}{dv} = \frac{1}{|v|}$. A variant of this prior was previously used by (Koolen and Van Erven, 2015). For all priors we derive a regret bound by computing an upper bound on the convex conjugate of F_T, $F_T^\star$. For conciseness we only present the regret bound for the conjugate prior in the main text. In Section 3.8 we present the analysis of the regret of the improper prior, for which a slightly different analysis is required compared to the proper priors. The analysis for all priors can be seen as performing a Laplace approximation of the integral over v to show that the prior places sufficient mass in a neighbourhood of the optimal v.

Abbreviating $B_t = b + \sum_{s=1}^{t-1} \tilde{g}_s^2$, $L_t = -\sum_{s=1}^{t-1} \tilde{g}_s$, and $C = \frac{1}{5G}$, the predictions (3.3.2) with the conjugate prior are given by:

$$\begin{aligned} w_t =& \frac{\sqrt{b} L_t \exp\left(\frac{(L+2CB_t)^2}{4B_t}\right)\left(\operatorname{erf}\left(\frac{L_t - 2CB_t}{2\sqrt{B_t}}\right) - \operatorname{erf}\left(\frac{L_t + 2CB_t}{2\sqrt{B_t}}\right)\right)}{\operatorname{erf}(C\sqrt{b}) \exp(C\,(L_t + CB_t)) 4 B_t^{\frac{3}{2}}} \\ &+ \frac{2\sqrt{B_t}\,(\exp(2CL_t) - 1)}{\operatorname{erf}(C\sqrt{b}) \exp(C\,(L_t + CB_t)) 4 B_t^{\frac{3}{2}}}. \end{aligned} \tag{3.3.4}$$

These w_t can be computed efficiently, but see Koolen and Van Erven (2015) for numerically stable evaluation. With the conjugate prior we find the following result:

Theorem 10. *Suppose $\tilde{g}_t$ is a symmetrical random variable with $|\mathbb{E}[\tilde{g}_t]| \leq G$ for all t. The predictions* (3.3.4) *satisfy:*

$$\begin{aligned} \mathbb{E}[\mathcal{R}_T(u)] \leq& 1 + |u| \max\Bigg\{ 11G\left(\ln(|u| 11G) - 1 + \ln\left(\frac{\sqrt{5}G\sqrt{\pi}}{4\sqrt{b}}\right)\right), \\ & \mathbb{E}\left[\sqrt{8\left(b + \sum_{t=1}^{T} \tilde{g}_t^2\right) \ln(16|u|^2 \left(b + \sum_{t=1}^{T} \tilde{g}_t^2\right)^{\frac{3}{2}} \frac{\sqrt{\pi}}{\sqrt{b}} + 1)}\right] \Bigg\}. \end{aligned}$$

The proof of Theorem 10 can be found in Section 3.7.1 and follows from computing the Fenchel conjugate of the potential function. For noisy subgradients this is the first bound that is adaptive to the sum of the squares of the noisy subgradients. Compared to the expected regret bound for the improper prior (see Theorem 12 in Section 3.8) this bound has worse constants. However, with the conjugate prior all non-constant terms scale with $|u|$, which is not the case with the improper prior. For all proper priors of the form (3.3.3) a similar regret bound can be computed. This can be seen from Lemma 11 in Section 3.7.1, which shows that the convex conjugate of the potential function for these priors is $O(\mathbb{E}[|u|\sqrt{\sum_{t=1}^{T} \tilde{g}_t^2 \ln(|u|T + 1)}])$.

Algorithm 2 Black-Box Reduction

```
Input: $G$ such that $\|\mathbb{E}[\tilde{\boldsymbol{g}}_t]\|_\star \le G$ and Algorithm $\mathcal{A}_{\mathcal{Z}}$ with domain $\mathcal{Z} = \{\boldsymbol{z} : \|\boldsymbol{z}\| \le 1\}$
for $t = 1, \ldots, T$ do
    Get $\boldsymbol{z}_t \in \mathcal{Z}$ from $\mathcal{A}_{\mathcal{Z}}$
    Get $v_t \in \mathbb{R}$ from Algorithm 1
    Play $\boldsymbol{w}_t = v_t \boldsymbol{z}_t$, receive symmetrical $\tilde{\boldsymbol{g}}_t$ such that $\|\mathbb{E}[\tilde{\boldsymbol{g}}_t]\|_\star \le G$
    Send $\tilde{\boldsymbol{g}}_t$ to $\mathcal{A}_{\mathcal{Z}}$
    Send $\langle \boldsymbol{z}_t, \tilde{\boldsymbol{g}}_t \rangle$ to Algorithm 1
end for
```

3.3.1 Black-Box Reductions

In this Section we use our potential function in a black-box reduction: we take a constrained noisy OCO algorithm $\mathcal{A}_{\mathcal{Z}}$ and turn it into an unconstrained algorithm using our potential function. The same reduction is used by Cutkosky and Orabona (2018) and Jun and Orabona (2019). The algorithm can be found in Figure 2. The potential function and the OCO algorithm each have their task: the potential function is to learn the norm of $\boldsymbol{u}$ and the constrained OCO algorithm is to learn the direction of $\boldsymbol{u}$. In each round t we play $\boldsymbol{w}_t = v_t \boldsymbol{z}_t$, where $\boldsymbol{z}_t \in \mathcal{Z}$, $\mathcal{Z} = \{\boldsymbol{z} : \|\boldsymbol{z}\| \le 1\}$, is the prediction of the OCO algorithm and v_t is the prediction of Algorithm 1. We feed $\tilde{\boldsymbol{g}}_t$ as feedback to $\mathcal{A}_{\mathcal{Z}}$ and $\langle \boldsymbol{z}_t, \tilde{\boldsymbol{g}}_t \rangle$ as feedback to Algorithm 1. Since $\tilde{\boldsymbol{g}}_t$ is a symmetrical random variable and $\mathbb{E}[\langle \boldsymbol{z}_t, \tilde{\boldsymbol{g}}_t \rangle] \le G$, $\langle \boldsymbol{z}_t, \tilde{\boldsymbol{g}}_t \rangle$ satisfies the assumptions in Lemma 7. This allows us to control the regret for learning the norm of $\boldsymbol{u}$ using Theorem 10.

As outlined by Cutkosky and Orabona (2018) the expected regret of Algorithm 2 decomposes into two parts. The first part of the regret is for learning the norm of $\boldsymbol{u}$, and is controlled by Algorithm 1. The second part of the regret for learning the direction of $\boldsymbol{u}$ and is controlled by $\mathcal{A}_{\mathcal{Z}}$. The proof is given by Cutkosky and Orabona (2018), but for completeness we provide the proof in Section 3.7.2.

Lemma 9. *Suppose $\tilde{\boldsymbol{g}}_t$ is a symmetrical random variable with $\|\mathbb{E}[\tilde{\boldsymbol{g}}_t]\|_\star \le G$ for all t. Let $\mathcal{R}_T^{\mathcal{V}}(\|\boldsymbol{u}\|) = \mathbb{E}[\sum_{t=1}^T (v_t - \|\boldsymbol{u}\|)\langle \boldsymbol{z}_t, \tilde{\boldsymbol{g}}_t \rangle]$ be the regret for learning $\|\boldsymbol{u}\|$ by Algorithm 1 and let $\mathcal{R}_T^{\mathcal{Z}}(\frac{\boldsymbol{u}}{\|\boldsymbol{u}\|}) = \mathbb{E}[\sum_{t=1}^T \langle \boldsymbol{z}_t - \frac{\boldsymbol{u}}{\|\boldsymbol{u}\|}, \tilde{\boldsymbol{g}}_t \rangle]$ be the regret for learning $\frac{\boldsymbol{u}}{\|\boldsymbol{u}\|}$ by $\mathcal{A}_{\mathcal{Z}}$. Then Algorithm 2 satisfies $\mathbb{E}[\mathcal{R}_T(\boldsymbol{u})] = \mathcal{R}_T^{\mathcal{V}}(\|\boldsymbol{u}\|) + \|\boldsymbol{u}\| \mathcal{R}_T^{\mathcal{Z}}\left(\frac{\boldsymbol{u}}{\|\boldsymbol{u}\|}\right)$.*

Orabona and Pál (2018) show that Mirror Descent with learning rates $\eta_t = (\sqrt{\sum_{s=1}^t \|\tilde{\boldsymbol{g}}_s\|_\star^2})^{-1}$ yields $\mathcal{R}_T^{\mathcal{Z}}(\frac{\boldsymbol{u}}{\|\boldsymbol{u}\|}) = O(\mathbb{E}[\sqrt{\sum_{t=1}^T \|\tilde{\boldsymbol{g}}_t\|_\star^2}])$. Since Algorithm 1 satisfies $\mathcal{R}^{\mathcal{V}}_T(\|\boldsymbol{u}\|) = O(\mathbb{E}[\|\boldsymbol{u}\|\sqrt{\sum_{t=1}^T \|\tilde{\boldsymbol{g}}_t\|_\star^2 \ln(\|\boldsymbol{u}\| \sum_{t=1}^T \|\tilde{\boldsymbol{g}}_t\|_\star^2 + 1)}])$ the total regret of Algorithm 2 is

$$\mathbb{E}[\mathcal{R}_T(\boldsymbol{u})] = O\left(\|\boldsymbol{u}\|\,\mathbb{E}\left[\sqrt{\sum_{t=1}^T \|\tilde{\boldsymbol{g}}_t\|_\star^2 \ln(\|\boldsymbol{u}\| \sum_{t=1}^T \|\tilde{\boldsymbol{g}}_t\|_\star^2 + 1)}\right]\right). \tag{3.3.5}$$

This bound matches state of the art bounds for for noiseless subgradients and is never worse than the bound of Jun and Orabona (2019) for noisy subgradients, but can be substantially better.

3.3.2 Private Unconstrained Adaptive Sparse Gradient Descent

Algorithm 3 Private Unconstrained Adaptive Sparse Gradient Descent

```
Input: $G$ such that $|\mathbb{E}[\tilde{g}_{t,j}]|_\star \leq G$.
for $t = 1, \dots, T$ do
    Play $\boldsymbol{w}_t$
    for $j = 1, \dots, d$ do
        Receive symmetrical $\tilde{g}_{t,j}$ such that $|\tilde{g}_{t,j}| \leq G$
        Send $\tilde{g}_{t,j}$ to the $j$-th instance of Algorithm 1
        Receive $v_{t+1,j} \in \mathbb{R}$ from the $j$-th instance of Algorithm 1 with the conjugate prior
        Set $\boldsymbol{w}_{t+1,j} = v_{t+1,j}$
    end for
end for
```

In this Section we propose a noisy unconstrained OCO algorithm that can exploit sparse subgradients. The algorithm is summarized in Algorithm 3. Algorithm 3 runs a copy of Algorithm 1 with the conjugate prior coordinate-wise. A similar strategy is used by Orabona and Tommasi (2017). This strategy can exploit sparse privacy structures, which, combined with sparse subgradients, may yield low regret (see Section 3.4). Its expected regret bound is given below. The proof follows from applying Theorem 10 per dimension.

Theorem 11. *Suppose $\tilde{g}_{t,j}$ is a symmetric random variable with $|\mathbb{E}[\tilde{g}_{t,j}]| \leq G$ for* *all t and j. Then the expected regret of Algorithm 3 satisfies*

$$\mathbb{E}[\mathcal{R}_T(u)] \leq d + \sum_{j=1}^{d} |\boldsymbol{u}_j| \max \left\{ 11G \left(\ln(|\boldsymbol{u}_j| 11G) - 1 + \ln\left(\frac{\sqrt{5}G\sqrt{\pi}}{4\sqrt{b_j}} \right) \right), \right.$$

$$\left. \mathbb{E}\left[\sqrt{8 \left(b_j + \sum_{t=1}^{T} \tilde{g}_{t,j}^2 \right) \ln(16 |\boldsymbol{u}_j|^2 \left(b_j + \sum_{t=1}^{T} \tilde{g}_{t,j}^2 \right)^{\frac{3}{2}} \frac{\sqrt{\pi}}{\sqrt{b_j}} + 1))} \right] \right\}.$$

3.4 Motivating Examples

In this Section we present two scenarios in which our algorithms provide better expected regret guarantees than standard algorithms. The first scenario concerns a case where many providers do not care for their privacy (so they do not perturb the subgradients) and few providers care substantially for their privacy. Suppose that the providers who care for their privacy are $\lceil \ln(T) \rceil$ of the total number of providers T. Suppose that $\|\boldsymbol{g}_t\|_2^2 \leq 1$ and that the providers who care for their privacy use $\rho(\boldsymbol{z}) \propto \exp(-\frac{\epsilon}{2}\|\boldsymbol{z}\|_2)$, then $\mathbb{E}[\|\boldsymbol{\xi}_t\|_2^2] \leq 4 + 4\frac{d^2+d}{\epsilon^2}$ (Song et al., 2015, Theorem 1). Using Algorithm 2, Jensen's inequality, and the fact that the square root is subadditive we see from (3.3.5) that the expected regret is upper bounded by $O(\|\boldsymbol{u}\|_2 \sqrt{\sum_{t=1}^{T} \|\boldsymbol{g}_t\|_2^2 \ln(1 + \|\boldsymbol{u}\|_2 T)} + \|\boldsymbol{u}\|_2 \frac{d}{\epsilon} \ln(\|\boldsymbol{u}\|_2 T + T))$ instead of $O(\|\boldsymbol{u}\|_2 \frac{d}{\epsilon} \sqrt{T \ln(1 + \|\boldsymbol{u}\|_2 T)})$ had we used the maximum privacy guarantee for all providers instead of letting the providers choose their desired level of privacy.

In the second scenario the providers use the local Laplace randomiser. Suppose that $\boldsymbol{g}_t$ is sparse. A standard algorithm that has good performance for sparse $\boldsymbol{g}_t$ is AdaGrad (Duchi et al., 2011). AdaGrad achieves $O(\mathbb{E}[D \sum_{j=1}^{d} \sqrt{\sum_{t=1}^{T} \tilde{g}_{t,j}^2}])$ expected regret, where $\max_j |u_j| \leq D$, and D has to be guessed prior to running AdaGrad. Using Jensen's inequality and the fact that the square root is subadditive the expected regret can be upper bounded by $O(D \sum_{j=1}^{d} (\sqrt{3 \sum_{t=1}^{T} g_{t,j}^2} + \sqrt{\sum_{t=1}^{T} 3\, \mathbb{E}[\xi_{t,j}^2]}))$. Algorithm 3 achieves $O(\sum_{j=1}^{d} |u_j| (\sqrt{3 \sum_{t=1}^{T} g_{t,j}^2 \ln(|u_j| T + 1)} + \sqrt{3 \sum_{t=1}^{T} \mathbb{E}[\xi_{t,j}^2] \ln(|u_j| T + 1)}))$ regret, which can be significantly smaller than the bound of AdaGrad if D is much larger than all u_j or if $\boldsymbol{u}$ is sparse. Furthermore, since we allow the provider of the data to choose $\tau_{t,j}$, the parameter of the Laplace randomiser for dimension j, $\boldsymbol{\xi}_t$ can be sparse as well. While this does not give local differential privacy guarantees for all attributes it does give local differential privacy guarantees for attributes with $\tau_j < \infty$.

3.5 Conclusions

In this chapter, we extended the local differential privacy framework in unconstrained Online Convex Optimization by allowing the provider of the data to choose their privacy guarantees. Standard algorithms do not yield satisfactory regret bounds in this setting, either due to dependence on the unknown parameters of the noise or due to dependence on bounded subgradients. Hence, we proposed two new algorithms that match state of the art regret algorithms in both the noisy and noiseless setting, without requiring knowledge of the noise other than symmetry. Our algorithms do not require parameters other than a bound on the norm of the expectation of the subgradients, which allows the privacy requirements of all providers to be private itself. The new algorithms are a step towards practically useful algorithms with local differential privacy guarantees that have sound theoretical guarantees. Furthermore, our algorithms are the first adaptive unconstrained algorithms in the noisy OCO setting without requiring extra parameters compared to the standard OCO setting, solving two problems left open by Jun and Orabona (2019).

3.6 Details from Section 3.2

Proof. (of Lemma 4) Evaluating and rewriting Definition 1 gives

$$\begin{aligned}\prod_{j=1}^{d} \frac{\exp(-\frac{\tau_{t,j}}{2}|\tilde{g}_{t,j} - g_{t,j'}|)}{\exp(-\frac{\tau_{t,j}}{2}|\tilde{g}_{t,j} - g_{t,j'}|)} &\leq \prod_{j=1}^{d} \exp(\frac{\tau_{t,j}}{2}(|g_{t,j}| + |g_{t,j'}|)) \\ &\leq \prod_{j=1}^{d} \exp(\tau_{t,j}) = \exp(\epsilon_t),\end{aligned}$$

where the first inequality follows from applying the triangle inequality for each j and the second inequality follows from the assumption that $|\boldsymbol{g}_{t,j}| \leq 1$. □

Proof. (of Lemma 6) We will prove the result by induction. In a given round t assume that $-\mathbb{E}[\sum_{s=1}^{t}\langle \boldsymbol{w}_s, \boldsymbol{g}_s\rangle] \geq \mathbb{E}[F_t(-\sum_{s=1}^{t}\tilde{\boldsymbol{g}}_s)]$ holds. Now,

$$\begin{aligned}-\mathbb{E}[\sum_{s=1}^{t+1}\langle \boldsymbol{w}_s, \boldsymbol{g}_s\rangle] =& \mathbb{E}[-\langle \boldsymbol{w}_{t+1}, \boldsymbol{g}_{t+1}\rangle - \sum_{s=1}^{t}\langle \boldsymbol{w}_s, \boldsymbol{g}_s\rangle] \\ \geq& \mathbb{E}[F_t(-\sum_{s=1}^{t}\tilde{\boldsymbol{g}}_s) - \langle \boldsymbol{w}_{t+1}, \boldsymbol{g}_{t+1}\rangle] \\ \geq& \mathbb{E}[F_{t+1}(-\sum_{s=1}^{t+1}\tilde{\boldsymbol{g}}_s)],\end{aligned}$$

where the first inequality comes from the inductive hypothesis and the second inequality is by the assumption that $F_{t-1}(\boldsymbol{x}) - \langle \boldsymbol{w}_t, \boldsymbol{g}_t \rangle \geq \mathbb{E}_{\tilde{\boldsymbol{g}}_t}[F_t(\boldsymbol{x} - \tilde{\boldsymbol{g}}_t)]$ for all t. Now, by induction $-\mathbb{E}[\sum_{t=1}^T \langle \boldsymbol{w}_t, \boldsymbol{g}_t \rangle] \geq \mathbb{E}[F_T(-\sum_{t=1}^T \tilde{\boldsymbol{g}}_t)]$. □

3.7 Details from Section 3.3

Proof. (of Lemma 7) We start by rewriting the l.h.s.:

$$\begin{aligned}&\mathbb{E}[\exp(\langle \boldsymbol{v}, \boldsymbol{x} \rangle - \langle \boldsymbol{v}, \boldsymbol{x} \rangle^2)]\\ &= \mathbb{E}[\exp(y\langle \boldsymbol{v}, \boldsymbol{z} \rangle - \langle \boldsymbol{v}, \boldsymbol{z} \rangle^2)] \exp(\mathbb{E}[\langle \boldsymbol{v}, \boldsymbol{x} \rangle] - \mathbb{E}[\langle \boldsymbol{v}, \boldsymbol{x} \rangle]^2).\end{aligned}$$

where $\boldsymbol{z} = \boldsymbol{x} - \mathbb{E}[\boldsymbol{x}]$ and $y = 1 - 2\,\mathbb{E}[\langle \boldsymbol{v}, \boldsymbol{x} \rangle]$. $\boldsymbol{z}$ is a random variable with mean $\mathbf{0}$ and $|y| \leq 1.4$ due to the restrictions on $\mathbb{E}[\langle \boldsymbol{v}, \boldsymbol{x} \rangle]$. By Lemma 10, $\mathbb{E}[\exp(y\langle \boldsymbol{v}, \boldsymbol{z} \rangle - \langle \boldsymbol{v}, \boldsymbol{z} \rangle^2)] \leq 1$. It remains to show that $\exp(\mathbb{E}[\langle \boldsymbol{v}, \boldsymbol{x} \rangle] - \mathbb{E}[\langle \boldsymbol{v}, \boldsymbol{x} \rangle]^2) \leq 1 + \mathbb{E}[\langle \boldsymbol{v}, \boldsymbol{x} \rangle]$, which holds for $\mathbb{E}[\langle \boldsymbol{v}, \boldsymbol{x} \rangle] \geq -\frac{1}{2}$ (Cesa-Bianchi and Lugosi, 2006, Lemma 2.4). □

Lemma 10. *Let $\boldsymbol{z} \in \mathbb{R}^d$ be a zero-mean symmetrical random variable. Then for $|y| \leq 1.4$ and arbitrary $\boldsymbol{v} \in \mathbb{R}^d$*

$$\mathbb{E}[\exp(y\langle \boldsymbol{v}, \boldsymbol{z} \rangle - \langle \boldsymbol{v}, \boldsymbol{z} \rangle^2)] \leq 1.$$

Proof. Due to symmetry of $\boldsymbol{z}$ we can write

$$\begin{aligned}&\mathbb{E}[\exp(y\langle \boldsymbol{v}, \boldsymbol{z} \rangle - \langle \boldsymbol{v}, \boldsymbol{z} \rangle^2)]\\ &= \mathbb{E}[\frac{1}{2}\exp(-y\langle \boldsymbol{v}, \boldsymbol{z} \rangle - \langle \boldsymbol{v}, \boldsymbol{z} \rangle^2) + \frac{1}{2}\exp(y\langle \boldsymbol{v}, \boldsymbol{z} \rangle - \langle \boldsymbol{v}, \boldsymbol{z} \rangle^2)].\end{aligned}$$

We continue by showing that the expression inside the expectation is smaller than 1:

$$\begin{aligned}\frac{1}{2}\exp(-y\langle \boldsymbol{v}, \boldsymbol{z} \rangle - \langle \boldsymbol{v}, \boldsymbol{z} \rangle^2) + \frac{1}{2}\exp(y\langle \boldsymbol{v}, \boldsymbol{z} \rangle - \langle \boldsymbol{v}, \boldsymbol{z} \rangle^2) \leq& 1\\ \ln(\cosh(y\langle \boldsymbol{v}, \boldsymbol{z} \rangle)) - \langle \boldsymbol{v}, \boldsymbol{z} \rangle^2 \leq& 0.\end{aligned}$$

which holds because for $|y| \leq 1.4$ $f(x) = \ln(\cosh(yx)) - x^2$ is concave and maximized at $x = 0$, which gives $f(0) = 0$. □

Proof. (of Lemma 8) Let $\ell_t(v) = v\tilde{g}_t + (v\tilde{g}_t)^2$

$$\begin{aligned}\mathbb{E}_{\tilde{g}_t}[F_t(-\sum_{s=1}^{t}\tilde{g}_s)] =& \mathbb{E}_v[\mathbb{E}_{\tilde{g}_t}[\exp(-\ell_t(v) - \sum_{s=1}^{t-1}\ell_t(v)) - 1]] \\ \leq& \mathbb{E}_v[(1 - v\,\mathbb{E}[\tilde{g}_t])\exp(-\sum_{s=1}^{t-1}\ell_t(v)) - 1]] \\ =& F_{t-1}(-\sum_{s=1}^{t-1}\tilde{g}_s) - w_t\,\mathbb{E}[\tilde{g}_t]\end{aligned}$$

where the first equality is due to Tonelli's theorem and the inequality is due to Lemma 7, which applies due to the restrictions on v and $\mathbb{E}[\tilde{g}_t]$. Since $F_0(x) = 0$ the proof is complete. □

3.7.1 Regret Analysis for Proper Priors

Proof. (of Theorem 10). By Lemma 5, Lemma 6, and Lemma 8 we only have to compute the convex conjugate of the potential function. We do the analysis for $-\sum_{t=1}^{T}\tilde{g}_t \geq 0$. The analysis for $-\sum_{t=1}^{T}\tilde{g}_t \leq 0$ is analogous. We have $-\sum_{t=1}^{T}w_t\tilde{g}_t \geq F_T(-\sum_{t=1}^{T}\tilde{g}_t) \geq -1$. Suppose $\sum_{t=1}^{T}\tilde{g}_t \leq \sqrt{2(\sum_{t=1}^{T}\tilde{g}_t^2 + b)}$, then $\mathbb{E}[\mathcal{R}_T(u)] = \mathbb{E}[\sum_{t=1}^{T}w_t\tilde{g}_t - u\tilde{g}_t] \leq \mathbb{E}[\sum_{t=1}^{T}|u||\sum_{t=1}^{T}\tilde{g}_t|] + 1 \leq |u|\,\mathbb{E}[\sqrt{2(\sum_{t=1}^{T}\tilde{g}_t^2 + b)}] + 1$, which implies the result.

Now, suppose $\sum_{t=1}^{T}\tilde{g}_t \geq \sqrt{2(\sum_{t=1}^{T}\tilde{g}_t^2 + b)}$. For the conjugate prior $\nu([\eta, \mu]) = \eta - \mu$ and $Z \leq \frac{\sqrt{\pi}}{\sqrt{b}}$. In the case where $-\sum_{t=1}^{T}\tilde{g}_t \leq \frac{2}{5G}(\sum_{t=1}^{T}\tilde{g}_t^2 + b)$ set $\mu = \frac{-\sum_{t=1}^{T}\tilde{g}_t}{2(\sum_{t=1}^{T}\tilde{g}_t^2+b)}$. Using Lemma 11 we obtain:

$$\begin{aligned}&F_T^\star(u) \\ &\leq \sqrt{8|u|^2\left(\sum_{t=1}^{T}\tilde{g}_t^2 + b\right)\ln(16|u|^2\left(\sum_{t=1}^{T}\tilde{g}_t^2 + b\right)\sqrt{\pi}\frac{\sqrt{\sum_{t=1}^{T}\tilde{g}_t^2 + b}}{\sqrt{b}} + 1)} + 1.\end{aligned} \tag{3.7.1}$$

In the case where $-\sum_{t=1}^{T}\tilde{g}_t \geq \frac{2}{5G}(\sum_{t=1}^{T}\tilde{g}_t^2 + b)$ set $\eta = \frac{5-\sqrt{5}}{50G}$ and $\mu = \frac{1}{2}$ to obtain:

$$F_T^\star(u) \leq 11G|u|(\ln(|u|11G) - 1 + \ln\left(\frac{\sqrt{5}G\sqrt{\pi}}{4\sqrt{b}}\right)) + 1. \tag{3.7.2}$$

Combining the expectations of (3.7.1) and (3.7.2) completes the proof. □

Lemma 11. *Suppose $L > \sqrt{2(V+b)}$. Let $F_T(L) = \mathbb{E}_{v \sim P}[\exp(vL - v^2 V) - 1]$ with P as in* (3.3.3). *If $L \leq \frac{2}{5G}(V+b)$ then*

$$F_T^\star(u) \leq \sqrt{8|u|^2(V+b)\ln(16|u|^2(V+b)S_t([\eta_1, \mu_1]) + 1)} + 1,$$

where $S_t([\eta, \mu]) = \frac{Z}{\nu([\eta,\mu])}$, $\eta_1 = \frac{L}{2(V+b)} - \frac{1}{\sqrt{2(V+b)}}$, $|\mu_1| \in [\eta_1, \frac{1}{5G}]$ such that $\mu_1 \leq \frac{L}{2(V+b)}$, and $\nu([\eta, \mu]) = \int_\eta^\mu \nu(v)dv$. If $L \geq \frac{2}{5G}(V+b)$ then

$$F_T^\star(u) \leq \frac{|u|}{\eta - \eta^2\frac{5}{2}G}(\ln\left(\frac{|u|}{\eta_2 - \eta_2^2\frac{5}{2}G}\right) - 1 + \ln(S_T([\eta_2, \mu_2]))) + 1,$$

where $[\eta_2, \mu_2] \subseteq [-\frac{1}{5G}, \frac{1}{5G}]$ such that $\mu_2 \leq \frac{L}{2(V+b)}$.

Proof. The initial part of the analysis is parallel to the analysis of Theorem 3 by Koolen and Van Erven (2015). Denote by $B = V + b$. For $v \leq \hat{\eta} = \frac{L}{2B}$, $vL - v^2 B$ is non-decreasing in v. Therefore, for $[\eta, \mu] \subseteq [-\frac{1}{5G}, \frac{1}{5G}]$ such that $\mu \leq \hat{\eta}$:

$$\begin{aligned} F_T(-\sum_{t=1}^T x_t) =& \frac{1}{Z}\int_{-\frac{1}{5G}}^{\frac{1}{5G}} \nu(v)\exp(vL - v^2 B)dv - 1 \\ \geq& \frac{1}{Z}\nu([\eta, \mu])\exp(\eta L - \eta^2 B) - 1, \end{aligned}$$

where $\nu([\eta, \mu]) = \int_\eta^\mu \nu(v)dv$. First suppose that $\hat{\eta} \leq \frac{1}{5G}$. Take $\eta = \hat{\eta} - \frac{1}{\sqrt{2B}}$, which yields

$$F_T(L) \geq \frac{\nu([\eta, \mu])}{Z}\exp\left(\frac{L^2}{4B} - \frac{1}{2}\right) - 1 = g(m(L)) - 1$$

where $g(x) = \exp(x - \frac{1}{2} - \ln\left(\frac{Z}{\nu([\eta,\mu])}\right))$ and $m(x) = \frac{x^2}{4B}$. By Hiriart-Urruty (2006, Theorem 2) we have

$$\begin{aligned} F_T^\star(u) \leq (g(m(u)))^\star &= \inf_{\gamma \geq 0} g^\star(\gamma) + \gamma m^\star(\frac{u}{\gamma}) \\ &= \inf_{\gamma \geq 0} \gamma\ln(\gamma) + \gamma(\ln(\frac{Z}{\nu([\eta, \mu])}) - \frac{1}{2}) + \frac{1}{\gamma}4|u|^2 B + 1. \end{aligned} \tag{3.7.3}$$

Denote by $S = \ln(\frac{Z}{\nu([\eta,\mu])})$ and $H = 4|u|^2 B$. Setting the derivative to 0 we find that $\hat{\gamma} = \sqrt{\frac{2H}{W(2H\exp(S_T^a + \frac{1}{2}))}}$ minimizes (3.7.3), where W is the Lambert function.

Plugging $\hat{\gamma}$ in (3.7.3) gives

$$F_T^\star(u) \leq \frac{H(2W(2H\exp(S+\frac{1}{2}))-1)}{\sqrt{2H(W(2H\exp(S+\frac{1}{2}))}} + 1 \leq \sqrt{2H(W(2H\exp(S+\frac{1}{2}))} + 1.$$

Using $W(x) \leq \ln(x+1)$ (Orabona and Pál, 2016, Lemma 17) we obtain

$$F_T^\star(u) \leq \sqrt{2H\ln(2H\exp(S+\frac{1}{2})+1)} \leq \sqrt{8|u|^2B\ln(16|u|^2B\exp(S)+1)}+1.$$

Now suppose that $\hat{\eta} > \frac{1}{5G}$, which is equivalent to $\frac{5}{2}GL > B$. Then

$$F_T(L) \geq \frac{\nu([\eta,\mu])}{Z}\exp((\eta-\eta^2\frac{5}{2}G)L)-1.$$

The convex conjugate of this lower bound is well known and is an upper bound on $F_T^\star$:

$$F_T^\star(u) \leq \frac{|u|}{\eta-\eta^2\frac{5}{2}G}(\ln\left(\frac{|u|}{\eta-\eta^2\frac{5}{2}G}\right) - 1 + \ln\left(\frac{Z}{\nu([\eta,\mu])}\right)) + 1,$$

which concludes the proof. □

3.7.2 Details From section 3.3.1

Proof. (of Lemma 9) We have

$$\begin{aligned}
\mathbb{E}[\mathcal{R}_u(\boldsymbol{u})] =& \mathbb{E}\left[\sum_{t=1}^{T}\langle \boldsymbol{w}_t - \boldsymbol{u}, \tilde{\boldsymbol{g}}_t\rangle\right] \\
=& \mathbb{E}\left[\sum_{t=1}^{T}\langle \boldsymbol{z}_t, \tilde{\boldsymbol{g}}_t\rangle(v_t - \|\boldsymbol{u}\|)\right] + \|\boldsymbol{u}\|\,\mathbb{E}\left[\sum_{t=1}^{T}\langle \boldsymbol{z}_t - \frac{\boldsymbol{u}}{\|\boldsymbol{u}\|}, \tilde{g}_t\rangle\right] \\
=& \mathcal{R}_T^{\mathcal{V}}(\|\boldsymbol{u}\|) + \|\boldsymbol{u}\|\mathcal{R}_T^{\mathcal{Z}}\left(\frac{\boldsymbol{u}}{\|\boldsymbol{u}\|}\right)
\end{aligned}$$

□

3.8 Regret Analysis for the Improper Prior

Abbreviating $B_t = \sum_{s=1}^{t-1}\tilde{g}_s^2$, $L_t = -\sum_{s=1}^{t-1}\tilde{g}_s$, and $C = \frac{1}{5G}$, the predictions (3.3.2) with the improper prior are given by:

$$\frac{\sqrt{\pi}\exp(\frac{L^2}{4B})\left(2\operatorname{erf}\left(\frac{L}{2\sqrt{B}}\right) - \operatorname{erf}\left(\frac{L+2CB}{2\sqrt{B}}\right) - \operatorname{erf}\left(\frac{L-2CB}{2\sqrt{B}}\right)\right)}{2\sqrt{B}}. \qquad (3.8.1)$$

With the predictions in (3.8.1) we can show the following result.

Theorem 12. *Suppose $\tilde{g}_t$ is a symmetrical random variable with $|\mathbb{E}[\tilde{g}_t]| \leq G$ for all t. The the expected regret of algorithm 1 with the improper prior $\frac{dP}{dv} = \frac{1}{|v|}$ satisfies*

$$\begin{aligned}\mathbb{E}[\mathcal{R}_T(u)] \leq \max\Bigg\{ & |u|\,\mathbb{E}\left[\sqrt{8\sum_{t=1}^T \tilde{g}_t^2}\left(\sqrt{\ln(8|u|^2\sum_{t=1}^T \tilde{g}_t^2 + 1)} + 1\right)\right], \\ & |u|11G(\ln(|u|11G\ln(2)) - 1) + \ln(2), \\ & |u|\,\mathbb{E}[\sqrt{2\sum_{t=1}^T \tilde{g}_t^2}] + 1 + \mathbb{E}\left[\ln\left(1 + 2\sqrt{2\sum_{t=1}^T \tilde{g}_t^2}\right)\right]\Bigg\}.\end{aligned} \tag{3.8.2}$$

Proof. By Lemma 5, Lemma 6, and Lemma 8 we only have to compute the convex conjugate of the potential function. The initial part of the analysis is parallel to the analysis of Theorem 4 by Koolen and Van Erven (2015). Denote by $L = -\sum_{t=1}^T \tilde{g}_t$ and by $V = \sum_{t=1}^T \tilde{g}_t^2$. We do the analysis for $L \geq 0$. The analysis for $L \leq 0$ is analogous. We start by considering the case where $L \leq \sqrt{2V}$. We have

$$\begin{aligned}F_T(L) \geq & \int_0^\epsilon \frac{1}{v}(\exp(-vL - v^2V) - 1) + \int_\epsilon^{\frac{1}{5G}} \frac{1}{v}(\exp(-vL - v^2V) - 1) \\ \geq & -\epsilon L - \epsilon^2 V + \ln(5G\epsilon),\end{aligned}$$

where we used $\exp(x) \geq 1 + x$. Choosing $\epsilon = \frac{1}{5G + 2\sqrt{2V}}$ gives $-\mathbb{E}[\sum_{t=1}^T w_t\tilde{g}_t] \geq \mathbb{E}[F_T(L)] \geq -1 - \mathbb{E}[\ln\left(1 + 2\sqrt{2V}\right)]$. Now, $\mathbb{E}[\mathcal{R}_T(u)] = \mathbb{E}[\sum_{t=1}^T w_t\tilde{g}_t - u\tilde{g}_t] \leq \mathbb{E}[\sum_{t=1}^T |u||L|] + 1 + \mathbb{E}[\ln\left(1 + 2\sqrt{2V}\right)] \leq |u|\,\mathbb{E}[\sqrt{2V}] + 1 + \mathbb{E}[\ln\left(1 + 2\sqrt{2V}\right)]$.

Now consider the case where $L > \sqrt{2V}$. For $v \leq \hat{\eta} = \frac{L}{2V}$, $vL - v^2V$ is non-decreasing in v. Therefore, for $[\eta, \mu] \subseteq [0, \frac{1}{5G}]$ such that $\mu \leq \hat{\eta}$, we have:

$$\begin{aligned}F_T(L) = & \int_{-\frac{1}{5G}}^{\frac{1}{5G}} \frac{1}{|v|}(\exp(vL - v^2V) - 1)dv \\ \geq & (\exp(\eta L - \eta^2 V) - 1)\int_\eta^\mu \frac{1}{v}dv - \int_\mu^{\frac{1}{5G}} \frac{1}{v}dv \\ = & (\exp(\eta L - \eta^2 V) - 1)\ln\left(\frac{\mu}{\eta}\right) + \ln(5G\mu).\end{aligned}$$

First, suppose that $\hat{\eta} \leq \frac{1}{5G}$. Set $\mu = \hat{\eta}$ and $\eta = \hat{\eta} - \frac{1}{\sqrt{2V}}$ and use $L \geq 2\sqrt{V}$ to obtain

$$\begin{aligned}
F_T(L) \geq & \exp\left(\frac{L^2}{4V} - \frac{1}{2}\right) \ln\left(\frac{1}{1 - \frac{\sqrt{2V}}{L}}\right) + \ln\left(\frac{L}{V}\right) \\
\geq & \exp\left(\frac{L^2}{4V} - \frac{1}{2}\right) \ln\left(\frac{1}{1 - \frac{\sqrt{2V}}{L}}\right) - \frac{1}{2}\ln\left(\frac{V}{4}\right) \\
\geq & \exp\left(\frac{1}{2}\left(\frac{L}{\sqrt{2V}} - 1\right)^2\right) - 1,
\end{aligned}$$

where the last inequality follows by using $\exp\left(\frac{1}{2}(x^2 - 1)\right) \geq \exp\left(\frac{1}{2}(x-1)^2\right) x, -1 \geq -\frac{L}{\sqrt{2V}}$, and $-\ln(1-x) \geq x$. Write $\exp\left(\frac{1}{2}\left(\frac{L}{\sqrt{2V}} - 1\right)^2\right) - 1 = g(m(x))$, where $g(x) = \exp(x) - 1$ and $m(x) = \left(\frac{x}{\sqrt{2V}} - 1\right)^2$. By Hiriart-Urruty (2006, Theorem 2) we have

$$\begin{aligned}
F_T^\star(u) \leq (g(m(u)))^\star &= \inf_{\gamma \geq 0} g^\star(\gamma) + \gamma m^\star(\frac{u}{\gamma}) \\
&= \inf_{\gamma \geq 0} \gamma \ln(\gamma) - \gamma + \frac{1}{\gamma} 4|u|^2 V + 2|u|\sqrt{2V}.
\end{aligned} \tag{3.8.3}$$

Setting the derivative to 0 we find that $\hat{\gamma} = \exp\left(\frac{1}{2}W(8|u^2|V)\right)$ minimizes (3.8.3), where W is the Lambert function. Plugging $\hat{\gamma}$ in (3.8.3) gives

$$F_T^\star(u) \leq |u|\sqrt{8VW(8|u^2|V)} - \hat{\gamma} + 2|u|\sqrt{2V}.$$

Using $W(x) \leq \ln(x+1)$ (Orabona and Pál, 2016, Lemma 17) and dropping the negative term we obtain

$$F_T^\star(u) \leq |u|\sqrt{8V}\left(\sqrt{\ln(8|u|^2 V + 1)} + 1\right).$$

Now suppose that $\hat{\eta} > \frac{1}{5G}$. Using that $\frac{5G}{2}L \geq V$, choosing $\mu = \frac{1}{5G}$, and $\eta = \frac{5 - \sqrt{5}}{50G}$ we obtain

$$\begin{aligned}
F_T(L) \geq & (\exp(\left(\frac{2(\sqrt{5} - 1)}{25G}\right) L) - 1) \ln\left(\frac{1}{1 - \frac{1}{\sqrt{5}}}\right) \\
\geq & (\exp(\left(\frac{1}{11G}\right) L) - 1) \ln(2).
\end{aligned} \tag{3.8.4}$$

The convex conjugate of the last expression in (3.8.4) is well known and given by

$$F_T^\star(u) \leq |u| 11G(\ln(|u| 11G \ln(2)) - 1) + \ln(2).$$

Combining the above completes the proof. □

CHAPTER 4

Comparator-Adaptive Convex Bandits

This chapter is based on: Van der Hoeven, D., Cutkosky, A., and Luo, H. (2020). Comparator-adaptive convex bandits. *To Appear in Advances in Neural Information Processing Systems 33*.[1]

Abstract

We study bandit convex optimization methods that adapt to the norm of the comparator, a topic that has only been studied before for its full-information counterpart. Specifically, we develop convex bandit algorithms with regret bounds that are small whenever the norm of the comparator is small. We first use techniques from the full-information setting to develop comparator-adaptive algorithms for linear bandits. Then, we extend the ideas to convex bandits with Lipschitz or smooth loss functions, using a new single-point gradient estimator and carefully designed surrogate losses.

[1] The author of this dissertation performed the following tasks: co-deriving the theoretical results and co-writing the paper. Part of the work on this Chapter was done while the author of this dissertation was visiting Haipeng Luo at the University of Southern California.

4.1 Introduction

In many situations, information is readily available. For example, if a gambler were to bet on the outcome of a football game, he can observe the outcome of the game regardless of what bet he made. In other situations, information is scarce. For example, the gambler could be deciding what to eat for dinner: should I eat a salad, a pizza, a sandwich, or not at all? These actions will result in different and unknown outcomes, but the gambler will only see the outcome of the action he actually takes, with one notable exception: not eating results in a predetermined outcome of being very hungry.

These two situations are instantiations of two different settings in online convex optimization: the full information setting and the bandit setting. More formally, both settings are sequential decision making problems where in each round $t = 1, \dots, T$, a learner has to make a prediction $\boldsymbol{w}_t \in \mathcal{W} \subseteq \mathbb{R}^d$ and an adversary provides a convex loss function $\ell_t : \mathcal{W} \to \mathbb{R}$. Afterwards, in the full information setting (Zinkevich, 2003) the learner has access to the loss function ℓ_t, while in the bandit setting (Kleinberg, 2005; Flaxman et al., 2005) the learner only receives the loss evaluated at the prediction, that is, $\ell_t(\boldsymbol{w}_t)$. In both settings the goal is to minimize the regret with respect to some benchmark point $\boldsymbol{u}$ in hindsight, referred to as the *comparator*. More specifically, the regret against $\boldsymbol{u}$ is the difference between the total loss incurred by the predictions of the learner and that of the comparator:

$$\mathcal{R}_T(\boldsymbol{u}) = \sum_{t=1}^{T} \left(\ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{u})\right).$$

When the learner's strategy is randomized, we measure the performance by the expected regret $\mathbb{E}\left[\mathcal{R}_T(\boldsymbol{u})\right]$.

Standard algorithms in both the full information setting and the bandit setting assume that the learner's decision space $\mathcal{W}$ is a convex *compact* set and achieve sublinear regret against the optimal comparator in this set: $\boldsymbol{u} = \arg\min_{\boldsymbol{u}^* \in \mathcal{W}} \sum_{t=1}^{T} \ell_t(\boldsymbol{u}^*)$. To tune these standard algorithms optimally, however, one requires knowledge of the norm of the comparator $\|\boldsymbol{u}\|$, which is unknown. A common work-around is to simply tune the algorithms in terms of the worst-case norm: $\max_{\boldsymbol{u} \in \mathcal{W}} \|\boldsymbol{u}\|$, assumed to be 1 without loss of generality. This results in worst-case bounds that do not take advantage of the case when $\|\boldsymbol{u}\|$ is small. For example, when the loss functions are L-Lipschitz, classic Online Gradient Descent (Zinkevich, 2003) guarantees $\mathcal{R}_T(\boldsymbol{u}) = O(L\sqrt{T})$ in the full information setting, while the algorithm of (Flaxman et al., 2005) guarantees

$\mathbb{E}\left[\mathcal{R}_T(\boldsymbol{u})\right] = O(d\sqrt{L}T^{3/4})$ in the bandit setting, both of which are independent of $\|\boldsymbol{u}\|$.

Recently, there has been a series of works in the full information setting that addresses this problem by developing *comparator-adaptive* algorithms, whose regret against $\boldsymbol{u}$ depends on $\|\boldsymbol{u}\|$ for *all* $\boldsymbol{u} \in \mathcal{W}$ *simultaneously* (see for example McMahan and Orabona (2014); Orabona and Pál (2016); Foster et al. (2017); Cutkosky and Boahen (2017); Kotłowski (2017); Cutkosky and Orabona (2018); Foster et al. (2018b); Jun and Orabona (2019); Van der Hoeven (2019)). These bounds are often not worse than the standard worst-case bounds, but could be much smaller in the case when there exists a comparator with small norm and reasonably small total loss. Moreover, most of these results also hold for the so-called *unconstrained* setting where $\mathcal{W} = \mathbb{R}^d$, that is, both the learner's predictions and the comparator can be any point in $\mathbb{R}^d$. For example, Cutkosky and Orabona (2018) achieve $\mathcal{R}_T(\boldsymbol{u}) = \widetilde{O}(\|\boldsymbol{u}\|L\sqrt{T})$ for all $\boldsymbol{u}$, in both the constrained and unconstrained settings, under full information feedback.[2]

While developing comparator-adaptive algorithms is relatively well-understood at this point in the full information setting, to the best of our knowledge, this has not been studied at all for the more challenging bandit setting. In this work, we take the first attempt in this direction and develop comparator-adaptive algorithms for several situations, including learning with linear losses, general convex losses, and convex and smooth losses, for both the constrained and unconstrained settings. Our results are summarized in Table 4.1. Ignoring other parameters for simplicity, for the linear case, we achieve $\widetilde{O}(\|\boldsymbol{u}\|\sqrt{T})$ regret (Section 4.3.2); for the general convex case, we achieve $\widetilde{O}(\|\boldsymbol{u}\|T^{\frac{3}{4}})$ regret in both the constrained and unconstrained setting (Sections 4.4.1 and 4.4.2); and for the convex and smooth case, we achieve $\widetilde{O}\left(\max\{\|\boldsymbol{u}\|^2, \|\boldsymbol{u}\|\}\beta(dLT)^{\frac{2}{3}}\right)$ regret in the unconstrained setting (Section 4.4.1).

In order to achieve our results for the convex case, we require an assumption on the loss, namely that the value of $\ell_t(\mathbf{0})$ is known for all t.[3] While restrictive at first sight, we believe that there are abundant applications where this assumption holds. As one instance, in control or reinforcement learning problems, $\mathbf{0}$ may represent some nominal action which has a known outcome: not eating results in hunger, or buying zero inventory will result in zero revenue. Another application is a classification problem where the features are not revealed to the learner. For example, end-users of a prediction service may not feel comfortable revealing their information to the

[2] Throughout the chapter, the notation $\widetilde{O}$ hides logarithmic dependence on parameters T,L, and $\|\boldsymbol{u}\|$.

[3] For the linear case, this clearly holds since $\ell_t(\mathbf{0}) = 0$.

Table 4.1: Summary of main results. Regret is measured with respect to the total loss of an arbitrary point $\boldsymbol{u} \in \mathbb{R}^d$ in the unconstrained setting, or an arbitrary point $\boldsymbol{u} \in \mathcal{W}$ in the constrained setting with a decision space $\mathcal{W}$ contained in the unit ball. T is the total number of rounds and $1/c$ is radius of the largest ball contained by $\mathcal{W}$. c is bounded by $O(d)$.

Loss functions (L-Lipschitz)	Regret for unconstrained settings	Regret for constrained settings
Linear (Section 4.3.2)	$\widetilde{O}\left(\|\boldsymbol{u}\|dL\sqrt{T}\right)$	$\widetilde{O}\left(\|\boldsymbol{u}\|cdL\sqrt{T}\right)$
Convex (Section 4.4.1 and 4.4.2)	$\widetilde{O}\left(\|\boldsymbol{u}\|L\sqrt{d}T^{\frac{3}{4}}\right)$	$\widetilde{O}\left(\|\boldsymbol{u}\|cL\sqrt{d}T^{\frac{3}{4}}\right)$
Convex and β-smooth (Section 4.4.2)	$\widetilde{O}\left(\max\{\|\boldsymbol{u}\|^2, \|\boldsymbol{u}\|\}\beta(dLT)^{\frac{2}{3}}\right)$	-

service. Instead, they may be willing to do some local computation and report the loss of the service's model. Most classification models (e.g. logistic regression) have the property that the loss of the **0** parameter is a known constant regardless of the data, and so this situation would also fit into our framework. Common loss functions that satisfy this assumption are linear loss, logistic loss, and hinge loss.

Techniques Our algorithms are based on sophisticated extensions of the black-box reduction introduced by Cutkosky and Orabona (2018), which separately learns the magnitude and the direction of the prediction. To make the reduction work in the bandit setting, however, new ideas are required, including designing an appropriate surrogate loss function and a new one-point gradient estimator with time-varying parameters. Note that (Cutkosky and Orabona, 2018) also proposes a method to convert any unconstrained algorithm to a constrained one in the full information setting, but this does not work in the bandit setting for technical reasons. Instead, we take a different approach by constraining the magnitude of the prediction directly.

Related work As mentioned, there has been a line of recent works on comparator-adaptive algorithms for the full information setting. Most of them do not transfer to the bandit setting, except for the approach of Cutkosky and Orabona (2018) from which we draw heavy inspiration. To the best of our knowledge, comparator-adaptive bandit algorithms have not been studied before. Achieving "adaptivity" in a broader sense is generally hard for problems with bandit feedback; see negative results such as (Daniely et al., 2015; Lattimore, 2015) as well as recent progress such as (Chen et al., 2019; Foster et al., 2019).

In terms of worst-case (non-adaptive) regret, the seminal work of (Abernethy et al., 2008) is the first to achieve $O(\sqrt{T})$ regret for bandit with linear losses, and (Kleinberg, 2005; Flaxman et al., 2005) are the first to achieve sublinear regret for general convex case. Over the past decade, the latter result has been improved in many different ways (Agarwal et al., 2010; Saha and Tewari, 2011; Agarwal et al., 2011; Hazan and Levy, 2014), and regret of order $O(\sqrt{T})$ under no extra assumptions was recently achieved (Bubeck et al., 2015; Bubeck and Eldan, 2016; Bubeck et al., 2017). However, these $O(\sqrt{T})$ bounds are achieved by very complicated algorithms that incur a huge dependence on the dimension d. Our algorithms are more aligned with the simpler ones with milder dimension-dependence (Abernethy et al., 2008; Flaxman et al., 2005; Saha and Tewari, 2011) and achieve the same dependence on T in different cases. How to achieve comparator-adaptive regret of order $O(\sqrt{T})$ for the general convex case is an important future direction.

4.2 Preliminaries

In this section, we describe our notation, state the definitions we use, and introduce the bandit convex optimization setting formally. We also describe the black-box reduction of Cutkosky and Orabona (2018) we will use throughout the chapter.

Notation and definitions The inner product between vectors $\boldsymbol{g} \in \mathbb{R}^d$ and $\boldsymbol{w} \in \mathbb{R}^d$ is denoted by $\langle \boldsymbol{w}, \boldsymbol{g} \rangle$. $\mathbb{R}_+$ denotes the set of positive numbers. The Fenchel conjugate $F^\star$ of a convex function F is defined as $F^\star(\boldsymbol{w}) = \sup_{\boldsymbol{g}} \langle \boldsymbol{w}, \boldsymbol{g} \rangle - F(\boldsymbol{g})$. $\|\cdot\|$ denotes a norm and $\|\boldsymbol{g}\|_\star = \sup_{\boldsymbol{w}:\|\boldsymbol{w}\| \leq 1} \langle \boldsymbol{w}, \boldsymbol{g} \rangle$ denotes the dual norm of $\boldsymbol{g}$. The Bregman divergence associated with convex function F between points $\boldsymbol{x}$ and $\boldsymbol{y}$ is denoted by $B_F(\boldsymbol{x}\|\boldsymbol{y}) = F(\boldsymbol{x}) - F(\boldsymbol{y}) - \langle \nabla F(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \rangle$, where $\nabla F(\boldsymbol{x})$ denotes the gradient of F evaluated at $\boldsymbol{x}$. The unit ball equipped with norm $\|\cdot\|$ is denoted by $\mathcal{B} = \{\boldsymbol{w} : \|\boldsymbol{w}\| \leq 1\}$. The unit sphere with norm $\|\cdot\|$ is denoted by $\mathcal{S} = \{\boldsymbol{w} : \|\boldsymbol{w}\| = 1\}$. The unit ball and sphere with norm $\|\cdot\|_2$ are denoted by $\mathbb{B}$ and $\mathbb{S}$ respectively. $\boldsymbol{x} \sim U(\mathcal{Z})$ denotes that $\boldsymbol{x}$ follows the uniform distribution over $\mathcal{Z}$. We say a function f is β-smooth over the set $\mathcal{W}$ if the following holds:

$$f(\boldsymbol{y}) \leq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{\beta}{2} \|\boldsymbol{x} - \boldsymbol{y}\|_2^2, \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{W}.$$

We say a function f is L-Lipschitz over the set $\mathcal{W}$ if the following holds:

$$|f(\boldsymbol{y}) - f(\boldsymbol{x})| \leq L \|\boldsymbol{y} - \boldsymbol{x}\|_2, \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{W}.$$

Throughout the chapter we will assume that $\beta, L \geq 1$. Also, by mild abuse of notation, we use $\partial f(x)$ to indicate an arbitrary subgradient of a convex function f at x.

All of our algorithms are reductions that use prior algorithms in disparate ways to obtain our new results. In order for these reductions to work, we need some

Algorithm 4 Black-Box Reduction with Full Information

```
Input: "Direction" algorithm $\mathcal{A}_{\mathcal{Z}}$ and "scaling" algorithm $\mathcal{A}_{\mathcal{V}}$
for $t = 1 \dots T$ do
    Get $\boldsymbol{z}_t \in \mathcal{Z}$ from $\mathcal{A}_{\mathcal{Z}}$
    Get $v_t \in \mathbb{R}$ from algorithm $\mathcal{A}_{\mathcal{V}}$
    Play $\boldsymbol{w}_t = v_t \boldsymbol{z}_t$, receive $\boldsymbol{g}_t$
    Send $\boldsymbol{g}_t$ to algorithm $\mathcal{A}_{\mathcal{Z}}$ as the $t$-th loss vector
    Send $\langle \boldsymbol{z}_t, \boldsymbol{g}_t \rangle$ to algorithm $\mathcal{A}_{\mathcal{V}}$ as the $t$-th loss value
end for
```

assumptions on the base algorithms. We will encapsulate these assumptions in *interfaces* that describe inputs, outputs, and guarantees described by an algorithm rather than its actual operation (see Interfaces 6 and 7 for examples). We can use specific algorithms from the literature to implement these interfaces, but our results depend only on the properties described in the interfaces.

4.2.1 Bandit Convex Optimization

The bandit convex optimization protocol proceeds in rounds $t = 1, \dots, T$. In each round t the learner plays $\boldsymbol{w}_t \in \mathcal{W} \subseteq \mathbb{R}^d$. Simultaneously, the environment picks an L-Lipschitz convex loss function $\ell_t : \mathcal{W} \to \mathbb{R}$, after which the learner observes $\ell_t(\boldsymbol{w}_t)$. Importantly, the learner only observes the loss function evaluated at $\boldsymbol{w}_t$, not the function itself. This forces the learner to play random points and estimate the feedback he wants to use to update $\boldsymbol{w}_t$. Therefore, in the bandit feedback setting, the goal is to bound the *expected* regret $\mathbb{E}\left[\mathcal{R}_T(\boldsymbol{u})\right]$, where the expectation is with respect to randomisation of the learner.

We make a distinction between linear bandits, where $\ell_t(\boldsymbol{w}) = \langle \boldsymbol{w}, \boldsymbol{g}_t \rangle$, and convex bandits, where ℓ_t can be any L-Lipschitz convex function. Throughout the chapter, if $\mathcal{W} \neq \mathbb{R}^d$ we assume that $\mathcal{W}$ is compact, has a non-empty interior, and contains $\mathbf{0}$. Without loss of generality we assume that $\frac{1}{c}\mathbb{B} \subseteq \mathcal{W} \subseteq \mathbb{B}$ for some $c \geq 1$. Some of our bounds depend on c, which, without loss of generality, is always bounded by d, due to a reshaping trick discussed in (Flaxman et al., 2005).

4.2.2 Black-Box Reductions with Full Information

Our algorithms are based on a black-box reduction from (Cutkosky and Orabona, 2018) for the full information setting (see Algorithm 4). The reduction works as follows. In each round t the algorithms plays $\boldsymbol{w}_t = v_t \boldsymbol{z}_t$, where $\boldsymbol{z}_t \in \mathcal{Z}$ for some domain $\mathcal{Z}$, is the prediction of a constrained algorithm $\mathcal{A}_{\mathcal{Z}}$, and v_t is the prediction of a one-dimensional algorithm $\mathcal{A}_{\mathcal{V}}$. The goal of $\mathcal{A}_{\mathcal{Z}}$ is to learn the directions of the comparator while the goal of $\mathcal{A}_{\mathcal{V}}$ is to learn the norm of the comparator. Let $\boldsymbol{g}_t$ be the gradient of ℓ_t at $\boldsymbol{w}_t$, which is known to the algorithm in the full information setting. We feed $\boldsymbol{g}_t$ as feedback to $\mathcal{A}_{\mathcal{Z}}$ and $\langle \boldsymbol{z}_t, \boldsymbol{g}_t \rangle$ as feedback to $\mathcal{A}_{\mathcal{V}}$. Although the original presentation considers only $\mathcal{Z} = \mathcal{B}$, we will need to extend the analysis to more general domains.

As outlined by Cutkosky and Orabona (2018), the regret of Algorithm 4 decomposes into two parts. The first part of the regret is for learning the norm of $\boldsymbol{u}$, and is controlled by Algorithm $\mathcal{A}_{\mathcal{V}}$. The second part of the regret is for learning the direction of $\boldsymbol{u}$ and is controlled by $\mathcal{A}_{\mathcal{Z}}$. The proof is provided in Section 4.6 for completeness.

Lemma 12. *Let $\mathcal{R}_T^{\mathcal{V}}(\|\boldsymbol{u}\|) = \sum_{t=1}^T (v_t - \|\boldsymbol{u}\|)\langle \boldsymbol{z}_t, \boldsymbol{g}_t \rangle$ be the regret for learning $\|\boldsymbol{u}\|$ by Algorithm $\mathcal{A}_{\mathcal{V}}$ and let $\mathcal{R}_T^{\mathcal{Z}}\left(\frac{\boldsymbol{u}}{\|\boldsymbol{u}\|}\right) = \sum_{t=1}^T \langle \boldsymbol{z}_t - \frac{\boldsymbol{u}}{\|\boldsymbol{u}\|}, \boldsymbol{g}_t \rangle$ be the regret for learning $\frac{\boldsymbol{u}}{\|\boldsymbol{u}\|}$ by $\mathcal{A}_{\mathcal{Z}}$. Then Algorithm 4 satisfies*

$$\mathcal{R}_T(\boldsymbol{u}) = \mathcal{R}_T^{\mathcal{V}}(\|\boldsymbol{u}\|) + \|\boldsymbol{u}\|\mathcal{R}_T^{\mathcal{Z}}\left(\frac{\boldsymbol{u}}{\|\boldsymbol{u}\|}\right). \tag{4.2.1}$$

Cutkosky and Orabona (2018) provide an algorithm to ensure $\mathcal{R}_T^{\mathcal{V}}(\|\boldsymbol{u}\|) \leq 1 + \|\boldsymbol{u}\| 8L\sqrt{T\log(\|\boldsymbol{u}\|T+1)}$, given that $\|\boldsymbol{g}_t\|_\star \leq L$. This algorithm satisfies the requirements described later in Interface 6, and will be used throughout this chapter.

4.3 Comparator-Adaptive Linear Bandits

Now, we apply the reduction of section 4.2.2 to develop comparator-adaptive algorithms for linear bandits. We will see that in the unconstrained case, the reduction works almost without modification. In the constrained case we will need to be more careful to enforce the constraints.

4.3.1 Unconstrained Linear Bandits

We begin by discussing the unconstrained linear bandit setting, which turns out to be the easiest setting we consider. Following Algorithm 4, we will still play $\boldsymbol{w}_t = v_t \boldsymbol{z}_t$. However, instead of taking a fixed $\boldsymbol{z}_t$ from a full-information algorithm, we take a random $\boldsymbol{z}_t$ from a *bandit* algorithm. Importantly, we can recover $\langle \boldsymbol{z}_t, \boldsymbol{g}_t \rangle$ exactly since $\langle \boldsymbol{w}_t, \boldsymbol{g}_t \rangle \frac{1}{v_t} = \langle \boldsymbol{z}_t, \boldsymbol{g}_t \rangle$. This means that we have enough information to send appropriate feedback to both $\mathcal{A}_{\mathcal{V}}$ and $\mathcal{A}_{\mathcal{Z}}$ and apply the argument of Lemma 12. Interestingly, we use a full-information one-dimensional algorithm for $\mathcal{A}_{\mathcal{V}}$, and

Algorithm 5 Black-Box Reduction for Linear Bandits

```
Input: Constrained Linear Bandit Algorithm $\mathcal{A}_{\mathcal{Z}}$ and unconstrained 1-d Algorithm $\mathcal{A}_{\mathcal{V}}$
for $t = 1 \dots T$ do
    Get $\boldsymbol{z}_t \in \mathcal{Z}$ from $\mathcal{A}_{\mathcal{Z}}$
    Get $v_t \in \mathbb{R}$ from $\mathcal{A}_{\mathcal{V}}$
    Play $\boldsymbol{w}_t = v_t \boldsymbol{z}_t$
    Receive loss $\langle \boldsymbol{w}_t, \boldsymbol{g}_t \rangle$
    Compute $\mathcal{L}_t = \frac{1}{v_t}\langle \boldsymbol{w}_t, \boldsymbol{g}_t \rangle = \langle \boldsymbol{z}_t, \boldsymbol{g}_t \rangle$.
    Send $\mathcal{L}_t$ to Algorithm $\mathcal{A}_{\mathcal{Z}}$ as $t$-th loss value.
    Send $\mathcal{L}_t$ to Algorithm $\mathcal{A}_{\mathcal{V}}$ as $t$-th loss value.
end for
```

only need $\mathcal{A}_{\mathcal{Z}}$ to take bandit input. This is because $\mathcal{A}_{\mathcal{V}}$ gets full information in the form of $\langle \boldsymbol{z}_t, \boldsymbol{g}_t \rangle$.

The algorithm $\mathcal{A}_{\mathcal{Z}}$ for learning the direction, on the other hand, now must be a bandit algorithm because intuitively we do not immediately get the full direction information $\boldsymbol{g}_t$ from the value of the loss alone. We will need this algorithm to fulfill the requirements described by Interface 7. An important requirement is that the expected regret of the direction learning algorithm is bounded by $dL\tau\sqrt{T\log(T+1)}$, where τ is a constant. One such algorithm is given by continuous Exponential Weights on a constrained set (see Van der Hoeven et al. (2018, section 6) for details).

Our unconstrained linear bandit algorithm then is constructed from Algorithm 5 by choosing an algorithm that implements Interface 7 as $\mathcal{A}_{\mathcal{Z}}$ and Interface 6 with $l = \mathbb{R}$ as $\mathcal{A}_{\mathcal{V}}$. Plugging in the guarantees of the individual algorithms and taking the expectation of (4.2.1), the total expected regret is $\widetilde{O}(1 + \|\boldsymbol{u}\| dL\sqrt{T})$. Compared to the full information setting we have lost a factor d in the regret bound, which is unavoidable given the bandit feedback (Dani et al., 2008). The formal result is below.

Theorem 13. *Suppose $\mathcal{A}_{\mathcal{Z}}$ implements Interface 7 with domain $\mathcal{Z} = \mathcal{B}$ and $\mathcal{A}_{\mathcal{V}}$ implements Interface 6 with $l = \mathbb{R}_+$. Then Algorithm 5 satisfies for all $\boldsymbol{u} \in \mathbb{R}^d$:*

$$\mathbb{E}[\mathcal{R}(\boldsymbol{u})] = 1 + \|\boldsymbol{u}\|(dL\tau\sqrt{T\log(T+1)} + L\sqrt{TB(\hat{v})}).$$

4.3.2 Constrained Linear Bandits

The algorithm in the previous section only works for $\mathcal{W} = \mathbb{R}^d$. In this section, we consider a compact set $\mathcal{W} \subset \mathbb{R}^d$.

Interface 6 Scale Learning Interface (see example implementation in Cutkosky and Orabona (2018))

```
Input: A line segment $l \subseteq \mathbb{R}$
for $t = 1 \dots T$ do
    Play $v_t \in l$
    Receive loss value $g_t$ such that $|g_t| \leq L_\mathcal{V}$
end for
Ensure: for all $\hat{v} \in l$, $\sum_{t=1}^{T}(v_t - \hat{v})g_t \leq 1 + |\hat{v}|L_\mathcal{V}\sqrt{TB(\hat{v})}$
```

Interface 7 Direction Learning Interface for Linear Bandits (see example implementation in Van der Hoeven et al. (2018))

```
Input: Domain $\mathcal{Z}$
for $t = 1 \dots T$ do
    Play $z_t \in \mathcal{Z}$
    Receive loss value $\langle \boldsymbol{z}_t, \boldsymbol{g}_t \rangle$ such that $|\langle \boldsymbol{z}_t, \boldsymbol{g}_t \rangle| \leq L$
end for
Ensure: for all $\boldsymbol{u} \in \mathcal{Z}$, $\mathbb{E}\left[\sum_{t=1}^{T}\langle \boldsymbol{z}_t - \boldsymbol{u}, \boldsymbol{g}_t \rangle\right] \leq dL\tau\sqrt{T\log(T+1)}$
```

In the full-information setting, Cutkosky and Orabona (2018) provide a projection technique for producing constrained algorithms from unconstrained ones. Unfortunately, this technique does not translate directly to the bandit setting, and we must be more careful in designing our constrained linear bandit algorithm. The key idea is to constrain the internal scaling algorithm $\mathcal{A}_\mathcal{V}$, rather than attempting to constrain the final predictions $\boldsymbol{w}_t$. Enforcing constraints on the scaling algorithm's outputs v_t will naturally translate into a constraint on the final predictions $\boldsymbol{w}_t$.

To produce a constrained linear bandit algorithm, we again use Algorithm 5, but now we instantiate $\mathcal{A}_\mathcal{V}$ implementing Interface 6 with $l = [0, 1]$ rather than $l = \mathbb{R}_+$, and instantiate $\mathcal{A}_\mathcal{Z}$ implementing Interface 7 with $\mathcal{Z} = \mathcal{W}$ rather than $\mathcal{Z} = \mathcal{B}$. As in the unconstrained setting, this allows us to feed full information feedback to $\mathcal{A}_\mathcal{V}$. At the same time, restricting Interface 6 to $l = [0, 1]$ also guarantees that $\boldsymbol{w}_t \in \mathcal{W}$. The regret bound of this algorithm is given in Theorem 14. The proof follows from combining Lemma 12 with the guarantees of Interfaces 6 and 7 and can be found in Section 4.7.

Theorem 14. *Suppose $\mathcal{A}_\mathcal{Z}$ implements 7 with domain $\mathcal{Z} = \mathcal{W}$ and $\mathcal{A}_\mathcal{V}$ implements Interface 6 with $l = [0, 1]$. Then Algorithm 5 satisfies for all $\boldsymbol{u} \in \mathcal{W}$,*

$$\mathbb{E}[\mathcal{R}_T(\boldsymbol{u})] \leq 1 + \|\boldsymbol{u}\|cL\left(d\tau\sqrt{T\log(T+1)} + \sqrt{TB(c\|\boldsymbol{u}\|)}\right).$$

If $\mathcal{W}$ is a unit ball, then $c = 1$. For other shapes of $\mathcal{W}$, recall that c is at most d, which leads to a expected regret bound of $\widetilde{O}\left(1 + \|\boldsymbol{u}\|d^2L\sqrt{T}\right)$.

4.4 Comparator-Adaptive Convex Bandits

In the general convex bandit problem, it is not clear how to use the single evaluation point feedback $\ell_t(\boldsymbol{w}_t)$ to derive any useful information about ℓ_t. Fortunately, Flaxman et al. (2005) solved this problem by using randomness to extract the gradients of a smoothed version of ℓ_t. To adapt to the norm of the comparator, we employ the following tweaked version of smoothing used by Flaxman et al. (2005):

$$\ell_t^v(\boldsymbol{w}) = \mathbb{E}_{\boldsymbol{b}\sim U(\mathbb{B})}[\ell_t(\boldsymbol{w} + v\delta\boldsymbol{b})], \tag{4.4.1}$$

where $v, \delta > 0$. In contrast to prior work using this framework, our smoothing now depends on the scaling parameter v. Lemma 13 gives the gradient of $\ell_t^v(\boldsymbol{w})$ and is a straightforward adaptation of Lemma 2.1 by Flaxman et al. (2005).

Lemma 13. *For $\delta \in (0, 1]$, $v > 0$:*

$$\nabla\ell_t^v(\boldsymbol{w}) = \frac{d}{v\delta}\,\mathbb{E}_{\boldsymbol{s}\sim U(\mathbb{S})}[\ell_t(\boldsymbol{w} + v\delta\boldsymbol{s})\boldsymbol{s}]. \tag{4.4.2}$$

With this lemma, we can estimate the gradient of the smoothed version of ℓ_t by evaluating ℓ_t at a random point, essentially converting the convex problem to a linear problem, except that one also needs to control the bias introduced by smoothing. Note that this estimate scales with $\frac{1}{v}$, which can be problematic if v is small. To deal with this issue, we require one extra assumption: the value of $\ell_t(\boldsymbol{0})$ is known to the learner. As discussed in section 4.1, this assumption holds for several applications, including some control or reinforcement learning problems, where $\boldsymbol{0}$ represents a nominal action with a known outcome. Furthermore, certain loss functions satisfy the second assumption by default, such as linear loss, logistic loss, and hinge loss. Without loss of generality we assume that $\ell_t(\boldsymbol{0}) = 0$, as we can always shift ℓ_t without changing the regret.

Our general algorithm template is provided in Algorithm 8. It incorporates the ideas of Algorithm 5, but adds new smoothing and regularization elements in order to deal with the present more general situation. More specifically, it again makes use of subroutine $\mathcal{A}_\mathcal{V}$, which learns the scaling. The direction is learned by Online Gradient Descent (Zinkevich, 2003), as was also done by Flaxman et al. (2005). Given $\boldsymbol{z}_t$ and v_t, our algorithm plays the point $\boldsymbol{w}_t = v_t(\boldsymbol{z}_t + \delta\boldsymbol{s}_t)$ for some parameter δ and $\boldsymbol{s}_t$ drawn uniformly at random from $\mathbb{S}$. By equation (4.4.2), we have

$$\mathbb{E}\left[\frac{d}{v_t\delta}\ell_t(\boldsymbol{w}_t)s_t\right] = \nabla\ell_t^{v_t}(v_t\boldsymbol{z}_t). \tag{4.4.3}$$

This means that we can use $\hat{\boldsymbol{g}}_t = \frac{d}{v_t\delta}\ell_t(\boldsymbol{w}_t)s_t$ as an approximate gradient estimate, and we send this $\hat{\boldsymbol{g}}_t$ to Online Gradient Descent as the feedback. In other words, Online Gradient Descent itself is essentially dealing with a full-information problem with gradient feedback and is required to ensure a regret bound $\mathbb{E}[\sum_{t=1}^{T}\langle\boldsymbol{z}_t - \boldsymbol{u}, \hat{\boldsymbol{g}}_t\rangle] = \widetilde{O}(\frac{dL}{\delta}\sqrt{T})$ for all $\boldsymbol{u}$ in some domain $\mathcal{Z}$. For technical reasons, we will also need to enforce $\boldsymbol{z}_t \in (1-\alpha)\mathcal{Z}$ for some $\alpha \in [0,1]$. This restriction will be necessary in the constrained setting to ensure $v_t(\boldsymbol{z}_t + \delta\boldsymbol{s}_t) \in \mathcal{W}$.

Next, to specify the feedback to the scaling learning black-box $\mathcal{A}_{\mathcal{V}}$, we define a surrogate loss function $\bar{\ell}_t(v)$ which contains a linear term $v\langle\boldsymbol{z}_t, \hat{\boldsymbol{g}}_t\rangle$ and also a regularization term (see Algorithm 8 for the exact definition). The feedback to $\mathcal{A}_{\mathcal{V}}$ is then $\partial\bar{\ell}_t(v_t)$. Therefore, $\mathcal{A}_{\mathcal{V}}$ is essentially learning these surrogate losses, also with full gradient information. The regularization term is added to deal with the bias introduced by smoothing. This term does not appear in prior work on convex bandits, and it is one of the key components needed to ensure that the final regret is in terms of the unknown $\|\boldsymbol{u}\|$.

Algorithm 8 should be seen as the analogue of the black-box reduction of Algorithm 4, but for bandit feedback instead of full information. The expected regret guarantee of Algorithm 8 is shown below, and the proof can be found in Section 4.8.

Lemma 14. *Suppose $\mathcal{A}_{\mathcal{V}}$ implements Interface 6 with $l \subseteq \mathbb{R}_+$. Suppose $\boldsymbol{w}_t \in \mathcal{W}$ for all t, $\ell_t(\boldsymbol{0}) = 0$, and let $L_{\mathcal{V}} = \max_t \partial\bar{\ell}_t(v_t)$. Then Algorithm 8 with $\delta, \alpha \in (0,1]$ and $\eta = \sqrt{\frac{\delta^2}{4(dL)^2T}}$ satisfies for all $\|\boldsymbol{u}\| \in l$ and $r > 0$ with $\frac{\boldsymbol{u}r}{\|\boldsymbol{u}\|} \in \mathcal{Z}$,*

$$\begin{aligned}\mathbb{E}\left[\mathcal{R}_T(\boldsymbol{u})\right] \leq & 1 + 2T\delta L\frac{\|\boldsymbol{u}\|}{r} + \frac{\|\boldsymbol{u}\|}{r}L_{\mathcal{V}}\sqrt{TB\left(\frac{\|\boldsymbol{u}\|}{r}\right)} \\ & + \frac{2\|\boldsymbol{u}\|dL}{r\delta}\sqrt{T} + \alpha\|\boldsymbol{u}\|_2 TL.\end{aligned}$$

In addition, if ℓ_t is also β-smooth for all t, then we have

$$\begin{aligned}\mathbb{E}\left[\mathcal{R}_T(\boldsymbol{u})\right] \leq & 1 + T\beta\delta^2\left(\frac{\|\boldsymbol{u}\|}{r}\right)^2 + \frac{\|\boldsymbol{u}\|}{r}L_{\mathcal{V}}\sqrt{TB\left(\frac{\|\boldsymbol{u}\|}{r}\right)} \\ & + \frac{2\|\boldsymbol{u}\|dL}{r\delta}\sqrt{T} + \alpha\|\boldsymbol{u}\|_2 TL.\end{aligned}$$

Algorithm 8 Black-Box Comparator-Adaptive Convex Bandit Algorithm

```
Input: Scaling algorithm $\mathcal{A}_\mathcal{V}$, $\delta \in (0,1]$, $\alpha \in [0,1]$, domain $\mathcal{Z} \subseteq \mathbb{B}$, and learning rate $\eta$
Set $\boldsymbol{z}_1 = \mathbf{0}$
for $t = 1 \dots T$ do
    Get $v_t$ from $\mathcal{A}_\mathcal{V}$
    Sample $\boldsymbol{s}_t \sim U(\mathbb{S})$
    Set $\boldsymbol{w}_t = v_t(\boldsymbol{z}_t + \delta \boldsymbol{s}_t)$
    Play $\boldsymbol{w}_t$
    Receive $\ell_t(\boldsymbol{w}_t)$
    Set $\hat{\boldsymbol{g}}_t = \frac{d}{v_t \delta} \ell_t(\boldsymbol{w}_t) s_t$
    if $\ell_t$ is $\beta$-smooth then
        Set $\bar{\ell}_t(v) = v\langle \boldsymbol{z}_t, \hat{\boldsymbol{g}}_t \rangle + \beta\delta^2 v^2$
    else
        Set $\bar{\ell}_t(v) = v\langle \boldsymbol{z}_t, \hat{\boldsymbol{g}}_t \rangle + 2\delta L|v|$
    end if
    Send $\partial\bar{\ell}_t(v_t)$ to algorithm $\mathcal{A}_\mathcal{V}$ as the $t$-th loss value
    Update $\boldsymbol{z}_{t+1} = \arg\min_{\boldsymbol{z} \in (1-\alpha)\mathcal{Z}} \eta\langle \boldsymbol{z}, \hat{\boldsymbol{g}}_t \rangle + \|\boldsymbol{z}_t - \boldsymbol{z}\|_2^2$
end for
```

This bound has two main points not obviously under our direct control: the assumption that the $\boldsymbol{w}_t$ lie in $\mathcal{W}$, and the value of $L_\mathcal{V}$, which is a bound on $|\partial\bar{\ell}_t(v_t)|$. In the remainder of this section we will specify the various settings of Algorithm 8 that guarantee that $w_t \in \mathcal{W}$ and that $L_\mathcal{V}$ is suitably bounded: two setting for the unconstrained setting and one for the constrained setting. The $\alpha\|\boldsymbol{u}\|TL$ term due to $\boldsymbol{z}_t \in (1-\alpha)\mathcal{Z}$ rather than $\boldsymbol{z}_t \in \mathcal{Z}$, which induces a small amount of bias. The r in Lemma 14 is to ensure that we satisfy the requirements for Online Gradient Descent to have a suitable regret bound. For unconstrained convex bandits $r = 1$. For constrained convex bandits we will find that $\frac{1}{r} = c$ (recall that we assume that $\frac{1}{c}\mathbb{B} \subseteq \mathcal{W} \subseteq \mathbb{B}$).

4.4.1 Unconstrained Convex Bandits

In this section we instantiate Algorithm 8 and derive regret bounds for either general convex losses or convex and smooth losses. We start with general convex losses. Since $\mathcal{W} = \mathbb{R}^d$, we do not need to ensure that $\boldsymbol{z}_t + \delta\boldsymbol{s}_t \in \mathcal{W}$ and we can safely set $\alpha = 0$. This choice guarantees that $\boldsymbol{z}_t + \delta\boldsymbol{s}_t \in 2\mathbb{B}$ and that $|\partial\bar{\ell}_t(v_t)| \leq \frac{2dL}{\delta} + 2\delta L$. Then, Lemma 14 directly leads to Theorem 15 (the proof is deferred to Section 4.8.1).

Theorem 15. *Supppose $\mathcal{A}_{\mathcal{V}}$ implements Interface 6 with $l = \mathbb{R}_+$ and that $\ell_t(\mathbf{0}) = 0$. Then Algorithm 8 with $\delta = \min\{1, \sqrt{d}T^{-\frac{1}{4}}\}$, $\mathcal{Z} = \mathbb{B}$, $\alpha = 0$, and $\eta = \sqrt{\frac{\delta^2}{4(dL)^2 T}}$ satisfies for all $\boldsymbol{u} \in \mathbb{R}^d$,*

$$\mathbb{E}\left[\mathcal{R}_T(\boldsymbol{u})\right] = 2 + \left(4\|\boldsymbol{u}\|Ld\sqrt{T} + 5\|\boldsymbol{u}\|L\sqrt{d}T^{\frac{3}{4}}\right)(1 + \sqrt{B(\|\boldsymbol{u}\|)}).$$

For unconstrained smooth bandits, we face an extra challenge. To bound the regret of Algorithm 8, $|\partial\bar{\ell}_t(v_t)| = |\langle \boldsymbol{z}_t, \hat{\boldsymbol{g}}_t\rangle + \beta 2\delta^2 v_t|$ must be bounded. Now in contrast to the linear or Lipschitz cases, in the smooth case $\bar{\ell}_t(v_t)$ is not Lipschitz over $\mathbb{R}_+$. We will address this by artificially constraining v_t. Specifically, we ensure that $v_t \leq \frac{1}{\delta^3}$, which implies $|\delta^2 v_t| = O\left(\frac{1}{\delta}\right)$. This makes the Lipschitz constant of $\bar{\ell}_t$ to be dominated by the gradient estimate $\hat{\boldsymbol{g}}_t$ rather than the regularization. To see how this affects the regret bound, consider two cases, $\|\boldsymbol{u}\|_2 \leq \frac{1}{\delta^3}$ and $\|\boldsymbol{u}\|_2 > \frac{1}{\delta^3}$. If $\|\boldsymbol{u}\|_2 \leq \frac{1}{\delta^3}$ then we have not hurt anything by constraining v_t since $\|\boldsymbol{u}\|_2$ satisfies the same constraint. If instead $\|\boldsymbol{u}\|_2 > \frac{1}{\delta^3}$ then the consequences for the regret bound are not immediately clear. However, following a similar technique in Cutkosky (2019), we use the fact that the regret against $\mathbf{0}$ is $O(1)$ and the Lipschitz assumption to show that we have added a penalty of only $O(\|\boldsymbol{u}\|_2 LT)$:

$$\mathbb{E}[\mathcal{R}_T(\boldsymbol{u})] = \mathbb{E}[\mathcal{R}_T(\mathbf{0})] + \sum_{t=1}^{T} \mathbb{E}[\ell_t(\mathbf{0}) - \ell_t(\boldsymbol{u})] = O(1 + \|\boldsymbol{u}\|_2 LT).$$

Since $\|\boldsymbol{u}\|_2 > \frac{1}{\delta^3}$ the penalty for constraining v_t is $O(\|\boldsymbol{u}\|_2 LT) = O(\|\boldsymbol{u}\|_2^2 L\delta^3 T)$, which is $O(\|\boldsymbol{u}\|_2^2 L\sqrt{T})$ if we set $\delta = O(T^{-1/6})$. The formal result can be found below and its proof can be found in Section 4.8.1.

Theorem 16. *Suppose $\mathcal{A}_{\mathcal{V}}$ implements Interface 6 with $l = (0, \frac{1}{\delta^3}]$, that $\ell_t(\mathbf{0}) = 0$, and that ℓ_t is β-smooth for all t. Then Algorithm 8 with $\delta = \min\{1, (dL)^{1/3}T^{-1/6}\}$, $\mathcal{Z} = \mathbb{B}$, $\alpha = 0$, and $\eta = \sqrt{\frac{\delta^2}{4(dL)^2 T}}$ satisfies for all $\boldsymbol{u} \in \mathbb{R}^d$,*

$$\begin{aligned}&\mathbb{E}\left[\sum_{t=1}^{T} \ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{u})\right]\\ &\leq 2 + 2\left(\|\boldsymbol{u}\|_2^2 dL\sqrt{T} + \|\boldsymbol{u}\|^2 (dLT)^{2/3}\beta\right)\\ &\quad + 6\|\boldsymbol{u}\|\left(\beta\sqrt{B(\|\boldsymbol{u}\|)} + 1\right)\left((dL+1)\sqrt{T} + ((dL)^{2/3} + (dL)^{-1/3})T^{2/3}\right).\end{aligned}$$

4.4.2 Constrained convex bandits

For the constrained setting we will set $\mathcal{Z} = \mathcal{W}$ and $\alpha = \delta$. This ensures that $v_t(\boldsymbol{z}_t + \delta \boldsymbol{s}_t) \in \mathcal{W}$ and we can apply Lemma 14 to find the regret bound in Theorem

17 below. Compared to the unconstrained setting, the regret bound now scales with c, which is due to the reshaping trick discussed in Flaxman et al. (2005).

Theorem 17. *Suppose $\mathcal{A}_{\mathcal{V}}$ implements Interface 6 with $l = (0, 1]$ and that $\ell_t(\mathbf{0}) = 0$. Then Algorithm 8 with $\mathcal{Z} = \mathcal{W}$, $\alpha = \delta = \min\{1, \sqrt{d}T^{-1/4}\}$, and $\eta = \sqrt{\frac{\delta^2}{4(dL)^2T}}$ satisfies for all $\boldsymbol{u} \in \mathcal{W}$,*

$$\begin{aligned}\mathbb{E}\left[\mathcal{R}_T(\boldsymbol{u})\right] =& 2 + \left(3\|\boldsymbol{u}\|Ld\sqrt{T} + 4\|\boldsymbol{u}\|L\sqrt{d}T^{\frac{3}{4}}\right)(1 + \sqrt{B(\|\boldsymbol{u}\|)})\\ &+ \|\boldsymbol{u}\|_2 dLT^{3/4}.\end{aligned}$$

4.5 Conclusion

In this chapter, we develop the first algorithms that have comparator-adaptive regret bounds for various bandit convex optimization problems. The regret bounds of our algorithms scale with $\|\boldsymbol{u}\|$, which may yield smaller regret in favourable settings.

For future research, there are a number of interesting open questions. First, our current results do not encompass improved rates for smooth losses on constrained domains. At first blush, one might feel this is relatively straightforward via methods based on self-concordance (Saha and Tewari, 2011), but it turns out that while such techniques provide good direction-learning algorithms, they may cause the gradients provided to the *scaling* algorithm to blow-up. Secondly, there is an important class of loss functions for which we did not obtain norm adaptive regret bounds: smooth and strongly convex losses. It is known that in this case an expected regret bound of $O(d\sqrt{T})$ can be efficiently achieved (Hazan and Levy, 2014). However, to achieve this regret bound the algorithm of Hazan and Levy (2014) uses a clever exploration scheme, which unfortunately leads to sub-optimal regret bounds for our algorithms.

4.6 Details from section 4.2

Proof of Lemma 12. By definition we have

$$\begin{aligned}\mathcal{R}_T(\boldsymbol{u}) =& \sum_{t=1}^T \langle \boldsymbol{w}_t - \boldsymbol{u}, \boldsymbol{g}_t\rangle = \sum_{t=1}^T \langle \boldsymbol{z}_t, \boldsymbol{g}_t\rangle(v_t - \|\boldsymbol{u}\|) + \|\boldsymbol{u}\| \sum_{t=1}^T \langle \boldsymbol{z}_t - \frac{\boldsymbol{u}}{\|\boldsymbol{u}\|}, \boldsymbol{g}_t\rangle\\ =& \mathcal{R}_T^{\mathcal{V}}(\|\boldsymbol{u}\|) + \|\boldsymbol{u}\|\mathcal{R}_T^{\mathcal{Z}}\left(\frac{\boldsymbol{u}}{\|\boldsymbol{u}\|}\right).\end{aligned}$$

□

4.7 Details from section 4.3

Proof of Theorem 14. For any fixed $\boldsymbol{u} \in \mathcal{W}$, let $r = \max_{\frac{r'\boldsymbol{u}}{\|\boldsymbol{u}\|} \in \mathcal{W}} r'$. Note that by definition we have $\frac{\|\boldsymbol{u}\|}{r} \in [0, 1]$ and $\frac{r\boldsymbol{u}}{\|\boldsymbol{u}\|} \in \mathcal{W}$. Therefore, similar to the proof of Lemma 12, we decompose the regret against $\boldsymbol{u}$ as:

$$\begin{aligned}\mathcal{R}_T(\boldsymbol{u}) &= \sum_{t=1}^{T} \langle \boldsymbol{w}_t - \boldsymbol{u}, \boldsymbol{g}_t \rangle \\ &= \sum_{t=1}^{T} \langle \boldsymbol{z}_t, \boldsymbol{g}_t \rangle \left(v_t - \frac{\|\boldsymbol{u}\|}{r} \right) + \frac{\|\boldsymbol{u}\|}{r} \sum_{t=1}^{T} \langle \boldsymbol{z}_t - \frac{r\boldsymbol{u}}{\|\boldsymbol{u}\|}, \boldsymbol{g}_t \rangle,\end{aligned}$$

which, by the guarantees of $\mathcal{A}_{\mathcal{V}}$ and $\mathcal{A}_{\mathcal{Z}}$,[4] is bounded in expectation by

$$\frac{\|\boldsymbol{u}\|}{r} L \sqrt{TB\left(\frac{\|\boldsymbol{u}\|}{r} \right)} + \frac{\|\boldsymbol{u}\|}{r} dL \sqrt{T \log(T+1)}.$$

Finally noticing $\frac{1}{c} \leq r$ by the definition of c finishes the proof. □

4.8 Details from section 4.4

Proof of Lemma 14. Denote by $\tilde{\boldsymbol{w}}_t = v_t \boldsymbol{z}_t$. By Jensen's inequality we have

$$\begin{aligned}\sum_{t=1}^{T} \mathbb{E}\left[\ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{u})\right] &= \sum_{t=1}^{T} \mathbb{E}\left[\ell_t^{v_t}(\boldsymbol{w}_t) - \ell_t(\boldsymbol{u})\right] + \sum_{t=1}^{T} \mathbb{E}\left[\ell_t(\boldsymbol{w}_t) - \ell_t^{v_t}(\boldsymbol{w}_t)\right] \\ &\leq \sum_{t=1}^{T} \mathbb{E}\left[\ell_t^{v_t}(\boldsymbol{w}_t) - \ell_t(\boldsymbol{u})\right].\end{aligned} \tag{4.8.1}$$

We now continue under the assumption that ℓ_t is L-Lipschitz. After completing the proof of the first equation of Lemma 14 we use the β-smoothness assumption to prove the second equation of Lemma 14.

[4] Note that the condition $|\langle z_t, g_t \rangle| \leq 1$ in Algorithm 7 indeed holds in this case since $\mathcal{Z} = \mathcal{W} \subseteq \mathbb{B}$ and $\|g_t\|_2 \leq L$ by the Lipschitzness condition.

Using the L-Lipschitz assumption we proceed:

$$\begin{aligned}
&\sum_{t=1}^{T} \mathbb{E}\left[\ell_t^{v_t}(\boldsymbol{w}_t) - \ell_t(\boldsymbol{u})\right] \\
&\leq \sum_{t=1}^{T} \mathbb{E}\left[\ell_t^{v_t}(\boldsymbol{w}_t) - \ell_t^{v_t}(\boldsymbol{u})\right] + \sum_{t=1}^{T} \mathbb{E}\left[\ell_t^{v_t}(\boldsymbol{u}) - \ell_t(\boldsymbol{u})\right] \\
&\leq \sum_{t=1}^{T} \mathbb{E}\left[\ell_t^{v_t}(\boldsymbol{w}_t) - \ell_t^{v_t}(\boldsymbol{u})\right] + \mathbb{E}[L|v_t|\|\delta \boldsymbol{s}_t\|_2] \\
&\leq \sum_{t=1}^{T} \mathbb{E}\left[\ell_t^{v_t}(\boldsymbol{w}_t) - \ell_t^{v_t}(\boldsymbol{u})\right] + \mathbb{E}[\delta L|v_t|] \\
&= \sum_{t=1}^{T} \mathbb{E}\left[\ell_t^{v_t}(\tilde{\boldsymbol{w}}_t) - \ell_t^{v_t}(\boldsymbol{u})\right] + \mathbb{E}[\delta L|v_t|] \\
&\quad + \sum_{t=1}^{T} \mathbb{E}\left[\ell_t^{v_t}(\boldsymbol{w}_t) - \ell_t^{v_t}(\tilde{\boldsymbol{w}}_t)\right] \\
&\leq \sum_{t=1}^{T} \mathbb{E}\left[\ell_t^{v_t}(\tilde{\boldsymbol{w}}_t) - \ell_t^{v_t}(\boldsymbol{u})\right] + 2\,\mathbb{E}[\delta L|v_t|].
\end{aligned}$$

Now, by using the L-Lipschitz assumption once more we find that

$$\sum_{t=1}^{T} \mathbb{E}[\ell_t^{v_t}((1-\alpha)\boldsymbol{u}) - \ell_t^{v_t}(\boldsymbol{u})] \leq \alpha\|\boldsymbol{u}\|_2 TL \tag{4.8.2}$$

By using equation (4.8.2), the convexity of $\ell_t^{v_t}$, and Lemma 13 we continue with:

$$\begin{aligned}
&\sum_{t=1}^{T} \mathbb{E}\left[\ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{u})\right] \\
&\leq \sum_{t=1}^{T} \mathbb{E}\left[\langle \tilde{\boldsymbol{w}}_t - (1-\alpha)\boldsymbol{u}, \hat{\boldsymbol{g}}_t\rangle\right] + 2\,\mathbb{E}[\delta L|v_t|] + \alpha\|\boldsymbol{u}\|_2 TL \\
&= \sum_{t=1}^{T} \mathbb{E}\left[\left(v_t - \frac{\|\boldsymbol{u}\|}{r}\right)\langle \boldsymbol{z}_t, \hat{\boldsymbol{g}}_t\rangle\right] + \mathbb{E}\left[\frac{\|\boldsymbol{u}\|}{r}\langle \boldsymbol{z}_t - \tilde{\boldsymbol{u}}, \hat{\boldsymbol{g}}_t\rangle\right] \\
&\quad + \sum_{t=1}^{T} 2\,\mathbb{E}[\delta L|v_t|] + \alpha\|\boldsymbol{u}\|_2 TL \\
&= \sum_{t=1}^{T} \mathbb{E}\left[\bar{\ell}_t(v_t) - \bar{\ell}_t\left(\frac{\|\boldsymbol{u}\|}{r}\right)\right] + \sum_{t=1}^{T} \frac{\|\boldsymbol{u}\|}{r}\,\mathbb{E}\left[\langle \boldsymbol{z}_t - \tilde{\boldsymbol{u}}, \hat{\boldsymbol{g}}_t\rangle\right] \\
&\quad + 2T\delta L\frac{\|\boldsymbol{u}\|}{r} + \alpha\|\boldsymbol{u}\|_2 TL
\end{aligned}$$

where $\bar{\ell}_t(v) = v\langle \boldsymbol{z}_t, \hat{\boldsymbol{g}}_t\rangle + 2\delta L|v|$ as defined in Algorithm 8, $\tilde{\boldsymbol{u}} = \frac{r}{\|\boldsymbol{u}\|}(1-\alpha)\boldsymbol{u}$, and $r > 0$ is such that $\frac{\boldsymbol{u}r}{\|\boldsymbol{u}\|} \in \mathcal{Z}$.

Finally, by using the convexity of $\bar{\ell}_t$, plugging in the guarantee of $\mathcal{A}_\mathcal{V}$, and using Theorem 18 we conclude the proof of the first equation of Lemma 14:

$$\begin{aligned}
&\sum_{t=1}^{T} \mathbb{E}\left[\ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{u})\right] \\
&\leq 2T\delta L\frac{\|\boldsymbol{u}\|}{r} + \mathbb{E}\left[\sum_{t=1}^{T}\left(v_t - \frac{\|\boldsymbol{u}\|}{r}\right)\partial\bar{\ell}_t(v_t)\right] \\
&\quad + \frac{\|\boldsymbol{u}\|}{r}\,\mathbb{E}\left[\sum_{t=1}^{T}\langle \boldsymbol{z}_t - \tilde{\boldsymbol{u}}, \hat{\boldsymbol{g}}_t\rangle\right] + \alpha\|\boldsymbol{u}\|_2 TL \\
&\leq 1 + 2T\delta L\frac{\|\boldsymbol{u}\|}{r} + \frac{\|\boldsymbol{u}\|}{r}L_\mathcal{V}\sqrt{TB\left(\frac{\|\boldsymbol{u}\|}{r}\right)} + \frac{2\|\boldsymbol{u}\|dL}{r\delta}\sqrt{T} + \alpha\|\boldsymbol{u}\|_2 TL.
\end{aligned}$$

Next, we continue from equation (4.8.1) under the smoothness condition. Using the definition of smoothness we find

$$\begin{aligned}
&\sum_{t=1}^{T} \mathbb{E}\left[\ell_t^{v_t}(\boldsymbol{w}_t) - \ell_t(\boldsymbol{u})\right] \\
&\leq \sum_{t=1}^{T} \mathbb{E}\left[\ell_t^{v_t}(\boldsymbol{w}_t) - \ell_t^{v_t}(\boldsymbol{u})\right] + \sum_{t=1}^{T} \mathbb{E}\left[\ell_t^{v_t}(\boldsymbol{u}) - \ell_t(\boldsymbol{u})\right] \\
&\leq \sum_{t=1}^{T} \mathbb{E}\left[\ell_t^{v_t}(\boldsymbol{w}_t) - \ell_t^{v_t}(\boldsymbol{u})\right] + \mathbb{E}\left[\tfrac{1}{2}\beta|v_t|^2\|\delta \boldsymbol{s}_t\|_2^2\right] \\
&= \sum_{t=1}^{T} \mathbb{E}\left[\ell_t^{v_t}(\boldsymbol{w}_t) - \ell_t^{v_t}(\boldsymbol{u})\right] + \mathbb{E}\left[\tfrac{1}{2}\delta^2|v_t|^2\beta\right] \\
&= \sum_{t=1}^{T} \mathbb{E}\left[\ell_t^{v_t}(\tilde{\boldsymbol{w}}_t) - \ell_t^{v_t}(\boldsymbol{u})\right] + \mathbb{E}\left[\tfrac{1}{2}\delta^2|v_t|^2\beta\right] \\
&\quad + \sum_{t=1}^{T} \mathbb{E}\left[\ell_t^{v_t}(\boldsymbol{w}_t) - \ell_t^{v_t}(\tilde{\boldsymbol{w}}_t)\right] \\
&\leq \sum_{t=1}^{T} \mathbb{E}\left[\ell_t^{v_t}(\tilde{\boldsymbol{w}}_t) - \ell_t^{v_t}(\boldsymbol{u})\right] + \mathbb{E}\left[\beta\delta^2|v_t|^2\right].
\end{aligned}$$

Using equation (4.8.2), the convexity of $\ell_t^{v_t}$, and Lemma 13 we continue with:

$$\begin{aligned}
&\sum_{t=1}^{T} \mathbb{E}\left[\ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{u})\right] \\
&\leq \sum_{t=1}^{T} \mathbb{E}\left[\langle \tilde{\boldsymbol{w}}_t - (1-\alpha)\boldsymbol{u}, \hat{\boldsymbol{g}}_t\rangle\right] + \mathbb{E}\left[\beta\delta^2|v_t|^2\right] + \alpha\|\boldsymbol{u}\|_2 TL \\
&= \sum_{t=1}^{T} \mathbb{E}\left[\left(v_t - \frac{\|\boldsymbol{u}\|}{r}\right)\langle \boldsymbol{z}_t, \hat{\boldsymbol{g}}_t\rangle\right] + \mathbb{E}\left[\beta\delta^2|v_t|^2\right] \\
&\quad + \sum_{t=1}^{T} \frac{\|\boldsymbol{u}\|}{r}\mathbb{E}\left[\langle \boldsymbol{z}_t - \tilde{\boldsymbol{u}}, \hat{\boldsymbol{g}}_t\rangle\right] + \alpha\|\boldsymbol{u}\|_2 TL \\
&= T\beta\delta^2\left(\frac{\|\boldsymbol{u}\|}{r}\right)^2 + \sum_{t=1}^{T} \mathbb{E}\left[\bar{\ell}_t(v_t) - \bar{\ell}_t\left(\frac{\|\boldsymbol{u}\|}{r}\right)\right] \\
&\quad + \sum_{t=1}^{T} \frac{\|\boldsymbol{u}\|}{r}\mathbb{E}\left[\langle \boldsymbol{z}_t - \tilde{\boldsymbol{u}}, \hat{\boldsymbol{g}}_t\rangle\right] + \alpha\|\boldsymbol{u}\|_2 TL,
\end{aligned}$$

where $\bar{\ell}_t(v) = v\langle \boldsymbol{z}_t, \hat{\boldsymbol{g}}_t\rangle + \beta\delta^2 v^2$ as defined in Algorithm 8. Finally, by using the convexity of $\bar{\ell}_t$, plugging in the guarantee of $\mathcal{A}_{\mathcal{V}}$, and using Theorem 18 we conclude the proof:

$$\begin{aligned}
&\sum_{t=1}^{T} \mathbb{E}\left[\ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{u})\right] \\
&\leq T\beta\delta^2 \left(\frac{\|\boldsymbol{u}\|}{r}\right)^2 + \mathbb{E}\left[\sum_{t=1}^{T}\left(v_t - \frac{\|\boldsymbol{u}\|}{r}\right)\partial\bar{\ell}_t(v_t)\right] \\
&\quad + \frac{\|\boldsymbol{u}\|}{r}\mathbb{E}\left[\sum_{t=1}^{T}\langle \boldsymbol{z}_t - \tilde{\boldsymbol{u}}, \hat{\boldsymbol{g}}_t\rangle\right] + \alpha\|\boldsymbol{u}\|_2 TL \\
&\leq 1 + T\beta\delta^2\left(\frac{\|\boldsymbol{u}\|}{r}\right)^2 + \frac{\|\boldsymbol{u}\|}{r}L_{\mathcal{V}}\sqrt{TB\left(\frac{\|\boldsymbol{u}\|}{r}\right)} + \frac{2\|\boldsymbol{u}\|dL}{r\delta}\sqrt{T} + \alpha\|\boldsymbol{u}\|_2 TL.
\end{aligned}$$

□

Theorem 18. *Suppose that $\ell_t(\mathbf{0}) = 0$, that ℓ_t is L-Lipschitz for all t, and that $\mathcal{Z} \subseteq \mathbb{B}$. For $\boldsymbol{u} \in (1-\alpha)\mathcal{Z}$, Online Gradient Descent on $(1-\alpha)\mathcal{Z}$ with learning rate $\eta = \sqrt{\frac{\delta^2}{(dL)^2 4T}}$ satisfies*

$$\mathbb{E}\left[\sum_{t=1}^{T}\langle \boldsymbol{z}_t - \boldsymbol{u}, \hat{\boldsymbol{g}}_t\rangle\right] \leq 2\frac{dL}{\delta}\sqrt{T}.$$

Proof. The proof essentially follows from the work of Zinkevich (2003); Flaxman et al. (2005) and using the assumptions that $\ell_t(\mathbf{0}) = 0$ and that ℓ_t is L-Lipschitz. We start by bounding the norm of the gradient estimate:

$$\begin{aligned}
\|\hat{\boldsymbol{g}}_t\|_2 =& \frac{d}{v_t\delta}|\ell_t(\boldsymbol{w}_t)|\|\boldsymbol{s}_t\|_2 \\
=& \frac{d}{v_t\delta}|\ell_t(v_t(\boldsymbol{z}_t + \delta\boldsymbol{s}_t)) - \ell_t(\mathbf{0})| \\
\leq& \frac{dL\|\boldsymbol{z}_t + \delta\boldsymbol{s}_t\|_2}{\delta} \leq \frac{dL(1-\alpha+\delta)}{\delta}
\end{aligned} \tag{4.8.3}$$

By using equation (4.8.3) and the regret bound of Online Gradient Descent

(Zinkevich, 2003) we find that

$$\begin{aligned}\sum_{t=1}^{T}\langle \boldsymbol{z}_t, \hat{\boldsymbol{g}}_t\rangle - \min_{\boldsymbol{z}\in(1-\alpha)\mathcal{Z}}\sum_{t=1}^{T}\langle \boldsymbol{z}, \hat{\boldsymbol{g}}_t\rangle \leq & \frac{(1-\alpha)}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\|\hat{\boldsymbol{g}}_t\|_2^2 \\ \leq & \frac{(1-\alpha)}{2\eta} + \frac{\eta}{2}\left(\frac{dL(1-\alpha+\delta)}{\delta}\right)^2 T \\ \leq & \frac{1}{2\eta} + 2\eta\left(\frac{dL}{\delta}\right)^2 T\end{aligned}$$

Plugging in $\eta = \sqrt{\frac{\delta^2}{(dL)^2 4T}}$ completes the proof. □

4.8.1 Details of section 4.4.1

Proof of Theorem 15. First, since $\ell_t(\mathbf{0}) = 0$, ℓ_t is L-Lipschitz, and $\boldsymbol{z}_t \in (1-\alpha)\mathcal{Z} = (1-\alpha)\mathbb{B}$ we have that

$$\langle \boldsymbol{z}_t, \hat{\boldsymbol{g}}_t\rangle \leq \|\boldsymbol{z}_t\|_2\|\hat{\boldsymbol{g}}_t\|_2 \leq (1-\alpha)\frac{dL(1-\alpha+\delta)}{\delta} \leq \frac{2dL}{\delta}, \tag{4.8.4}$$

where the first inequality is the Cauchy-Schwarz inequality and the second is due to equation (4.8.3). Since $|\partial\bar{\ell}_t(v_t)| \leq |\langle \boldsymbol{z}_t, \hat{\boldsymbol{g}}_t\rangle| + 2\delta L \leq \frac{4dL}{\delta} = L_{\mathcal{V}}$ we can use Lemma 14 to find

$$\begin{aligned}\mathbb{E}\left[\mathcal{R}_T(\boldsymbol{u})\right] \leq & 1 + 2T\delta L\frac{\|\boldsymbol{u}\|}{r} + \frac{\|\boldsymbol{u}\|}{r}\frac{4dL}{\delta}\sqrt{TB\left(\frac{\|\boldsymbol{u}\|}{r}\right)} \\ & + \frac{2\|\boldsymbol{u}\|dL}{r\delta}\sqrt{T} + \alpha\|\boldsymbol{u}\|_2 TL.\end{aligned}$$

Plugging in $\alpha = 0$ and $\delta = \min\{1, \sqrt{d}T^{-\frac{1}{4}}\}$ completes the proof. □

Proof of Theorem 16. By equation (4.8.4) $|\langle \boldsymbol{z}_t, \hat{\boldsymbol{g}}_t\rangle| \leq \frac{2dL}{\delta}$. Since $v_t \leq \frac{1}{\delta^3}$ we have that

$$|\partial\bar{\ell}_t(v_t)| \leq \frac{2dL}{\delta} + 2|v_t|\beta\delta^2 \leq 2\frac{dL+\beta}{\delta} \leq \frac{2\beta(dL+1)}{\delta}$$

If $\|\boldsymbol{u}\|_2 \leq \frac{1}{\delta^3}$ applying Lemma 14 with $\alpha = 0$ gives us

$$\begin{aligned}&\sum_{t=1}^{T}\mathbb{E}\left[\ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{u})\right] \\ &\leq 1 + T\beta\delta^2\|\boldsymbol{u}\|^2 + 2\|\boldsymbol{u}\|\frac{(dL+1)\beta}{\delta}\sqrt{TB(\|\boldsymbol{u}\|)} + \frac{2\|\boldsymbol{u}\|dL}{\delta}\sqrt{T}.\end{aligned} \tag{4.8.5}$$

If $\|\boldsymbol{u}\|_2 > \frac{1}{\delta^3}$ then using the Lipschitz assumption on ℓ_t and equation (4.8.5) with $\boldsymbol{u} = \mathbf{0}$ gives us

$$\begin{aligned}\sum_{t=1}^{T} \mathbb{E}\left[\ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{u})\right] &= \sum_{t=1}^{T} \mathbb{E}\left[\ell_t(\boldsymbol{w}_t) - \ell_t(\mathbf{0}) + \ell_t(\mathbf{0}) - \ell_t(\boldsymbol{u})\right] \\ &\leq 1 + \|\boldsymbol{u}\|_2 LT \\ &\leq 1 + \|\boldsymbol{u}\|_2^2 \delta^3 LT,\end{aligned} \tag{4.8.6}$$

where we used that $\|\boldsymbol{u}\|_2 \geq \frac{1}{\delta^3}$. Adding equations (4.8.5) and (4.8.6) gives

$$\begin{aligned}&\sum_{t=1}^{T} \mathbb{E}\left[\ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{u})\right] \\ &\leq 2 + \|\boldsymbol{u}\|_2^2 \delta^3 LT + T\beta\delta^2 \|\boldsymbol{u}\|^2 + 2\|\boldsymbol{u}\| \frac{(dL+1)\beta}{\delta} \sqrt{TB(\|\boldsymbol{u}\|)} + \frac{2\|\boldsymbol{u}\| dL}{\delta} \sqrt{T}\end{aligned}$$

Setting $\delta = \min\{1, (dL)^{1/3} T^{-1/6}\}$ gives us

$$\begin{aligned}&\sum_{t=1}^{T} \mathbb{E}\left[\ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{u})\right] \\ &\leq 2 + 2\left(\|\boldsymbol{u}\|_2^2 dL\sqrt{T} + \|\boldsymbol{u}\|^2 (dLT)^{2/3}\beta\right) \\ &\quad + 6\|\boldsymbol{u}\| \left(\beta\sqrt{B(\|\boldsymbol{u}\|)} + 1\right)\left((dL+1)\sqrt{T} + ((dL)^{2/3} + (dL)^{-1/3})T^{2/3}\right)\end{aligned}$$

□

4.8.2 Details of section 4.4.2

Proof of Theorem 17. First, to see that $\boldsymbol{z}_t + \delta\boldsymbol{s}_t \in \mathcal{W}$ recall that by assumption $\mathcal{W} \subseteq \mathbb{B}$. Since $\alpha = \delta$ we have that $\boldsymbol{z}_t + \delta\boldsymbol{s}_t \in (1-\alpha)\mathcal{W} + \delta\mathbb{S} \subseteq (1-\delta)\mathcal{W} + \delta\mathcal{W} = \mathcal{W}$. For any fixed $\boldsymbol{u} \in \mathcal{W}$, let $r = \max_{\frac{r'\boldsymbol{u}}{\|\boldsymbol{u}\|} \in \mathcal{W}} r'$. Note that by definition we have $\frac{\|\boldsymbol{u}\|}{r} \in [0,1]$ and $\frac{r\boldsymbol{u}}{\|\boldsymbol{u}\|} \in \mathcal{W}$. By using equation (4.8.4) we can see that $|\partial\bar{\ell}_t(v_t)| \leq \frac{dL}{\delta} + 2\delta L$. By definition, $\frac{1}{r} \leq c$. This implies that the regret of $\mathcal{A}_{\mathcal{V}}$ is $\widetilde{O}\left(1 + \frac{\|\boldsymbol{u}\|}{r}\frac{dL}{\delta}\sqrt{T}\right)$. Applying Lemma 14 with the parameters above we find

$$\begin{aligned}&\sum_{t=1}^{T} \mathbb{E}\left[\ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{u})\right] \\ &= 1 + 2T\delta L\frac{\|\boldsymbol{u}\|}{r} + \frac{3dL\|\boldsymbol{u}\|}{r}\sqrt{TB\left(\frac{\|\boldsymbol{u}\|}{r}\right)} + \frac{2\|\boldsymbol{u}\| dL}{r\delta}\sqrt{T} + \alpha\|\boldsymbol{u}\|_2 TL.\end{aligned}$$

Finally, setting $\delta = \min\{1, \sqrt{d}T^{-1/4}\}$ completes the proof:

$$\sum_{t=1}^{T} \mathbb{E}\left[\ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{u})\right] \leq 2 + \left(3\|\boldsymbol{u}\|Ld\sqrt{T} + 4\|\boldsymbol{u}\|L\sqrt{d}T^{\frac{3}{4}}\right)\left(1 + \sqrt{B(\|\boldsymbol{u}\|)}\right) \\ + \|\boldsymbol{u}\|_2 dLT^{3/4}.$$

□

CHAPTER 5

MetaGrad: Universal Adaptation using Multiple Learning Rates in Online Learning

This chapter is based on: Van Erven, T., Koolen, W. M., and Van der Hoeven, D. (2020a). Metagrad: Universal adaptation using multiple learning rates in online learning. *Manuscript in preparation.*[1]

Abstract

In online convex optimization it is well known that certain subclasses of objective functions are much easier than arbitrary convex functions. We are interested in designing universally adaptive methods that can automatically get fast rates in as many such subclasses as possible, without any manual tuning. We provide a new universally adaptive method, MetaGrad, that is robust to general convex losses but adapts to a broad class of functions, including exp-concave and strongly convex functions, but also various types of stochastic and non-stochastic functions without any curvature. For instance, MetaGrad can achieve logarithmic regret on the unregularized hinge loss over the unit ball, even though the hinge loss has no curvature, if the data come from a favorable probability distribution. We prove this by drawing a connection to the Bernstein condition, which is known to imply fast rates in offline statistical learning. MetaGrad further adapts automatically to the size of the gradients. Its main feature is that it simultaneously considers multiple learning rates. Unlike previous methods with provable regret guarantees, however, its learning rates are not monotonically decreasing over time and are not tuned based on a theoretically derived bound on the regret. Instead, they are weighted directly proportional to their empirical performance on the data using a tilted exponential

[1] The author of this dissertation performed the following tasks: performing the experiments, co-deriving part of the theoretical results, and co-writing the paper

weights meta-algorithm. We provide three versions of MetaGrad. The full matrix version maintains a full covariance matrix and is applicable to learning tasks for which we can afford update time quadratic in the dimension. The other two versions provide speed-ups for high-dimensional learning tasks with an update time that is linear in the dimension: one is based on sketching, the other on running a separate copy of the basic algorithm per coordinate. We compare all versions of MetaGrad on benchmark online classification and regression tasks, showing that they consistently outperform both online gradient descent and AdaGrad.

5.1 Introduction

Methods for *online convex optimization* (OCO) (Shalev-Shwartz, 2011; Hazan et al., 2016) make it possible to optimize parameters sequentially, by processing convex functions in a streaming fashion. This is important in time series prediction where the data are inherently online; but it may also be convenient to process offline data sets sequentially, for instance if the data do not all fit into memory at the same time or if parameters need to be updated quickly when extra data become available.

The difficulty of an OCO task depends on the convex functions $f_1, f_2, \ldots, f_T$ that need to be optimized. The argument of these functions is a d-dimensional parameter vector $\boldsymbol{w}$ from a convex domain $\mathcal{W}$. Although this is abstracted away in the general framework, each function f_t usually measures the loss of the parameters on an underlying example $(\boldsymbol{x}_t, y_t)$ in a machine learning task. For example, in classification f_t might be the *hinge loss* $f_t(\boldsymbol{w}) = \max\{0, 1 - y_t\langle \boldsymbol{w}, \boldsymbol{x}_t\rangle\}$ or the *logistic loss* $f_t(\boldsymbol{w}) = \log\left(1 + e^{-y_t\langle \boldsymbol{w}, \boldsymbol{x}_t\rangle}\right)$, with $y_t \in \{-1, +1\}$. Thus the difficulty depends both on the choice of loss and on the observed data.

There are different methods for OCO, depending on assumptions that can be made about the functions. The simplest and most commonly used strategy is *online gradient descent* (GD). GD updates parameters $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta_t \nabla f_t(\boldsymbol{w}_t)$ by taking a step in the direction of the negative gradient, where the step size is determined by a parameter η_t called the *learning rate*. The goal is to minimize the *regret* over T rounds, which measures the difference in cumulative loss between the online iterates $\boldsymbol{w}_t$ and the best offline parameters $\boldsymbol{u}$. For learning rates $\eta_t \propto 1/\sqrt{t}$, GD guarantees that the regret for general convex functions is bounded by $O(\sqrt{T})$ (Zinkevich, 2003). Alternatively, if it is known beforehand that the functions are of an easier type, then better regret rates are sometimes possible. For instance, if the functions are *strongly convex*, then logarithmic regret $O(\log T)$ can be achieved by GD with much smaller learning rates $\eta_t \propto 1/t$ (Hazan et al., 2007), and, if they are *exp-concave*, then logarithmic regret $O(d \log T)$ can be achieved by the *Online Newton Step* (ONS) algorithm (Hazan et al., 2007).

This partitions OCO tasks into categories, leaving it to the user to choose the appropriate algorithm for their setting. Such a strict partition, apart from being a burden on the user, depends on an extensive cataloguing of all types of easier functions that might occur in practice. (See Section 5.3 for several ways in which the existing list of easy functions can be extended.) It also immediately raises the question of whether there are cases in between logarithmic and square-root regret (there are, see Theorem 21 in Section 5.3), and which algorithm to use then. And, third, it presents the problem that the appropriate algorithm might depend on (the distribution of) the data (again see Section 5.3), which makes it entirely impossible to select the right algorithm beforehand.

These issues motivate the development of *adaptive* methods, which are no worse than $O(\sqrt{T})$ for general convex functions, but also automatically take advantage of easier functions whenever possible. An important step in this direction are the adaptive GD algorithm of Bartlett et al. (2007) and its proximal improvement by Do et al. (2009), which are able to interpolate between strongly convex and general convex functions if they are provided with a data-dependent strong convexity parameter in each round, and significantly outperform the main non-adaptive method (i.e. Pegasos, (Shalev-Shwartz et al., 2011)) in the experiments of Do et al.. Here we consider a significantly richer class of functions, which includes exp-concave functions, strongly convex functions, general convex functions that do not change between rounds (even if they have no curvature), and stochastic functions whose gradients satisfy the so-called Bernstein condition, which is well-known to enable fast rates in offline statistical learning (Bartlett and Mendelson, 2006; Van Erven et al., 2015; Koolen et al., 2016). The latter group can again include functions without curvature, like the unregularized hinge loss. All these cases are covered simultaneously by a new adaptive method we call *MetaGrad*, for multiple eta gradient algorithm. Theorem 23 below implies the following:

Theorem 19. *Suppose the diameter of the domain $\mathcal{W}$ and the ℓ_2-norms of the gradients $\boldsymbol{g}_t = \nabla f_t(\boldsymbol{w}_t)$ are both bounded by constants, and define $V_T^{\boldsymbol{u}} = \sum_{t=1}^T ((\boldsymbol{u} - \boldsymbol{w}_t)^\intercal \boldsymbol{g}_t)^2$. Then MetaGrad's regret is simultaneously bounded by $O(\sqrt{T \log \log T})$, and by*

$$\sum_{t=1}^T f(\boldsymbol{w}_t) - \sum_{t=1}^T f_t(\boldsymbol{u}) \leq \sum_{t=1}^T (\boldsymbol{w}_t - \boldsymbol{u})^\intercal \boldsymbol{g}_t \leq O\left(\sqrt{V_T^{\boldsymbol{u}}\, d \ln(T/d)} + d \ln(T/d)\right) \tag{5.1.1}$$

for any $\boldsymbol{u} \in \mathcal{W}$.

Theorem 19 bounds the regret in terms of a measure of variance $V_T^{\boldsymbol{u}}$ that depends on the distance of the algorithm's choices $\boldsymbol{w}_t$ to the optimum $\boldsymbol{u}$, and which, in favorable cases, may be significantly smaller than T. Intuitively, this happens, for instance, when there is a stable optimum $\boldsymbol{u}$ that the algorithm's choices $\boldsymbol{w}_t$ converge to. Formal consequences are given in Section 5.3, which shows that this bound implies faster than $O(\sqrt{T})$ regret rates, often logarithmic in T, for all functions in the rich class mentioned above. In all cases the dependence on T in the rates matches what we would expect based on related work in the literature, and in most cases the dependence on the dimension d is also what we would expect. Only for strongly convex functions is there an extra factor d. It seems that this is a real limitation of the method as presented here. In Section 5.9 we discuss a recent extension of MetaGrad by Zhang et al. (2019) that removes this limitation.

The main difficulty in achieving the regret guarantee from Theorem 19 is tuning a learning rate parameter η. In theory, η should be roughly proportional to $1/\sqrt{V_T^{\boldsymbol{u}}}$, but this is not possible using any existing techniques, because the optimum $\boldsymbol{u}$ is unknown in advance, and tuning in terms of a uniform upper bound $\max_{\boldsymbol{u}} V_T^{\boldsymbol{u}}$ ruins all desired benefits. MetaGrad therefore runs multiple supporting expert algorithms, each with a different learning rate η, and combines them with a novel controller algorithm that learns the empirically best learning rate for the OCO task in hand. Crucially, the overhead for learning the best expert is not of the usual order $O(\sqrt{T})$, which would ruin all desired benefits, but only costs a negligible $O(\log \log T)$.

The experts are instances of exponential weights on the continuous parameters $\boldsymbol{u}$ with a suitable surrogate loss function, which in particular causes the exponential weights distributions to be multivariate Gaussians. The resulting updates are closely related to the ONS algorithm on the original losses, where each expert receives the controller's gradients instead of its own. It is shown that $\lceil \log_2 T \rceil$ experts suffice, which is at most 30 as long as $T \leq 10^9$, and therefore seems computationally acceptable. If not, then the number of experts can be further reduced at the cost of slightly worse constants in the bound.

An important practical consideration for OCO algorithms is whether they can adapt to the Lipschitz-constant of the losses f_t, i.e. the maximum norm of the gradients. For instance, this is an important feature of AdaGrad (Duchi et al., 2011; McMahan and Streeter, 2010). The MetaGrad algorithm is also adaptive in this way. Our approach is a refinement of the techniques of Mhammedi et al. (2019): whereas their procedure may occasionally restart the whole MetaGrad algorithm, we only restart the controller but not the experts. Wherever possible, we further measure the size of the gradients by the (semi-)norm $\max_{\boldsymbol{w} \in \mathcal{W}} |\boldsymbol{w}^\intercal \boldsymbol{g}_t|$ instead of the larger $\max_{\boldsymbol{w}, \boldsymbol{v} \in \mathcal{W}} \|\boldsymbol{w} - \boldsymbol{v}\|_2 \|\boldsymbol{g}_t\|_2$. The difference is crucial in Section 5.5.1, where we consider a domain for which the diameter is infinite, but our norms are under control.

The version of MetaGrad described so far maintains a full covariance matrix of size $d \times d$, where d is the parameter dimension. This requires at least $O(d^2)$ computation steps per round to update, which is prohibitive for large d. We therefore also present two extensions: the first applies the matrix sketching approach of Luo et al. (2017) to approximate the matrix by a rank k sketch, and requires $O(kd)$ update time on average per round. Our second extension was inspired by the diagonal version of

AdaGrad (Duchi et al., 2011; McMahan and Streeter, 2010) and runs a separate copy of full MetaGrad per coordinate, which takes $O(d)$ computation per round, just like vanilla GD and AdaGrad. While the full matrix version of MetaGrad and its sketching approximation naturally favor parameters $\boldsymbol{u}$ with small ℓ_2-norm, the coordinatewise extension is appropriate for the ℓ_∞-norm.

Related Work If we disregard computational efficiency and omit Lipschitz-adaptivity, then the result of Theorem 19 can be achieved by finely discretizing the domain $\mathcal{W}$ and running the Squint algorithm for prediction with experts with each discretization point as an expert (Koolen and Van Erven, 2015). MetaGrad may therefore also be seen as a computationally efficient extension of Squint to the OCO setting.

As already mentioned, Zhang et al. (2019) extend MetaGrad to adapt to strongly convex functions. They further provide an extension for the case that the optimal parameters $\boldsymbol{u}$ vary over time, as measured in terms of the adaptive regret. See also the closely related extension of Squint for the adaptive regret by Neuteboom (2020).

Our focus in this work is on adapting to sequences of functions f_t that are easier than general convex functions, but we require an estimate $\hat{D}$ of the ℓ_2-norm of the optimum $\boldsymbol{u}$ as a hyperparameter. In contrast, a different line of work designs methods that can adapt to the norm of $\boldsymbol{u}$ over all of $\mathbb{R}^d$, but without providing adaptivity to the functions f_t (Mcmahan and Streeter, 2012; Orabona, 2014; Cutkosky and Orabona, 2018). It was thought for some time that these two directions could not be reconciled, because the impossibility result of Cutkosky and Boahen (2017) blocks simultaneous adaptivity to both the size of the gradients of the functions f_t and the norm of $\boldsymbol{u}$. The perspective has recently shifted, however, following discoveries of ways to partially circumvent this lower bound (Kempka et al., 2019; Cutkosky, 2019; Mhammedi and Koolen, 2020).

Another notion of adaptivity is explored in a series of work obtaining tighter bounds for linear functions f_t that vary little between rounds, as measured either by their deviation from the mean function or by successive differences (Hazan and Kale, 2010; Chiang et al., 2012; Steinhardt and Liang, 2014). Such bounds imply super fast rates for optimizing a fixed linear function, but reduce to slow $O(\sqrt{T})$ rates in the other cases of easy functions that we consider. Finally, the way MetaGrad's experts maintain a Gaussian distribution on parameters $\boldsymbol{u}$ is similar in spirit to AROW and related confidence weighted methods, as analyzed by (Crammer et al., 2009) in the mistake bound model.

Outline We start with the main definitions in the next section. Then Section 5.3 contains an extensive set of examples where Theorem 19 leads to fast rates, Section 5.4 presents the Full Matrix version of the MetaGrad algorithm, and Section 5.5 describes the faster sketching and coordinatewise extensions. Section 5.6 provides the analysis leading to Theorem 23 for the Full Matrix version of MetaGrad, which is a more detailed statement of Theorem 19 with several quantities replaced by data-dependent versions and with exact constants. Section 5.7 extends this analysis to the two other versions of MetaGrad. Then, in Section 5.8, we compare all versions of MetaGrad to GD and to AdaGrad in experiments with several benchmark classification and regression data sets. We conclude with possible further extensions of MetaGrad in Section 5.9.

5.2 Setup

We consider algorithms for OCO, which operate according to the protocol displayed in Protocol 9. In each round, the environment reveals a closed convex domain $\mathcal{W}_t$, which we assume contains the origin $\mathbf{0}$ (if not, it needs to be translated). In the introduction, we assumed that $\mathcal{W}_t = \mathcal{W}$ was fixed beforehand, but for the remainder of the paper we allow it to vary between rounds, which is needed in the context of the sketching version of MetaGrad (Section 5.5.1). Let $\boldsymbol{w}_t \in \mathcal{W}_t$ be the iterate produced by the algorithm in round t, let $f_t : \mathcal{W}_t \to \mathbb{R}$ be the convex loss function produced by the environment and let $\boldsymbol{g}_t = \nabla f_t(\boldsymbol{w}_t)$ be the (sub)gradient, which is the feedback given to the algorithm.[2] The *regret* over T rounds $R_T^{\boldsymbol{u}}$, its linearization $\tilde{R}_T^{\boldsymbol{u}}$ and our measure of variance $V_T^{\boldsymbol{u}}$ are defined as

$$\begin{aligned} R_T^{\boldsymbol{u}} &= \sum_{t=1}^T \left(f_t(\boldsymbol{w}_t) - f_t(\boldsymbol{u})\right), & \tilde{R}_T^{\boldsymbol{u}} &= \sum_{t=1}^T (\boldsymbol{w}_t - \boldsymbol{u})^\intercal \boldsymbol{g}_t, \\ V_T^{\boldsymbol{u}} &= \sum_{t=1}^T \left((\boldsymbol{u} - \boldsymbol{w}_t)^\intercal \boldsymbol{g}_t\right)^2 & &\text{with respect to any } \boldsymbol{u} \in \bigcap_{t=1}^T \mathcal{W}_t. \end{aligned}$$

By convexity of f_t, we always have $f_t(\boldsymbol{w}_t) - f_t(\boldsymbol{u}) \leq (\boldsymbol{w}_t - \boldsymbol{u})^\intercal \boldsymbol{g}_t$, which implies the first inequality in Theorem 19: $R_T^{\boldsymbol{u}} \leq \tilde{R}_T^{\boldsymbol{u}}$. Finally, wherever possible we measure the size of the gradient $\boldsymbol{g}_t$ in the *intrinsic (semi-)norm* for the domain $\mathcal{W}_t$:

$$\|\boldsymbol{g}\|_t = \max_{\boldsymbol{w} \in \mathcal{W}_t} |\boldsymbol{w}^\intercal \boldsymbol{g}|.$$

This is a norm in the typical case that $\mathcal{W}_t$ has full dimension d, and it is still a semi-norm in general. We note that the intrinsic norm is smaller than the usual

[2] If f_t is not differentiable at $\boldsymbol{w}_t$, any choice of subgradient $\boldsymbol{g}_t \in \partial f_t(\boldsymbol{w}_t)$ is allowed.

Algorithm 9 Online Convex Optimization from First-order Information

```
for $t = 1, 2, \dots$ do
    Environment reveals convex domain $\mathcal{W}_t \subseteq \mathbb{R}^d$ containing the origin $\mathbf{0}$
    Learner plays $\boldsymbol{w}_t \in \mathcal{W}_t$
    Environment chooses a convex loss function $f_t : \mathcal{W}_t \to \mathbb{R}$
    Learner incurs loss $f_t(\boldsymbol{w}_t)$ and observes (sub)gradient $\boldsymbol{g}_t = \nabla f_t(\boldsymbol{w}_t)$
end for
```

upper bounds based on Hölder's inequality: $\|\boldsymbol{g}\|_t \leq \|\boldsymbol{g}\| \max_{\boldsymbol{w} \in \mathcal{W}_t} \|\boldsymbol{w}\|^*$ for any dual norms $\|\cdot\|$ and $\|\cdot\|^*$. The difference becomes essential in Section 5.5.1, where we consider a domain $\mathcal{W}_t$ that has an infinite radius $\max_{\boldsymbol{w} \in \mathcal{W}_t} \|\boldsymbol{w}\|^*$ in any norm $\|\cdot\|^*$, but for which $\|\boldsymbol{g}_t\|_t$ is still bounded. MetaGrad depends on (upper bounds on) the sizes of the gradients per round b_t, as well as their running maximum B_t:

$$b_t \geq \|\boldsymbol{g}_t\|_t, \qquad\qquad B_t = \max_{s \leq t} b_s, \tag{5.2.1}$$

with the convention that $B_0 = 0$. We would normally take the best upper bound $b_t = \|\boldsymbol{g}_t\|_t$, except if this is difficult to compute. In such cases, we may for example let $b_t = \|\boldsymbol{g}_t\| \max_{\boldsymbol{u} \in \mathcal{W}_t} \|\boldsymbol{u}\|^*$.

Further Notation We denote by $\lceil z \rceil_+ = \max\{\lceil z \rceil, 1\}$ the smallest integer that is at least z and at least 1.

5.3 Fast Rates Examples

In this section, we motivate our interest in the adaptive bound (5.1.1) by giving a series of examples in which it provides fast rates. Although MetaGrad is designed to handle time varying domains, for simplicity we will assume that the domain is fixed in this section. In this section we will also assume that the following standard boundedness assumptions hold for all $\boldsymbol{u}, \boldsymbol{w} \in \mathcal{W}$ and all t: $\|\boldsymbol{u} - \boldsymbol{w}\|_2 \leq D'$ and $\|\boldsymbol{g}_t\|_2 \leq G'$. The fast rates are all derived from two general sufficient conditions: one based on the directional derivative of the functions f_t and one for stochastic gradients that satisfy the *Bernstein condition*, which is the standard condition for fast rates in off-line statistical learning. Simple simulations that illustrate the conditions are provided in Section 5.10.1 and proofs are also postponed to Section 5.10.

Directional Derivative Condition In order to control the regret with respect to some point $\boldsymbol{u}$, the first condition requires a quadratic lower bound on the curvature of the functions f_t in the direction of $\boldsymbol{u}$:

Theorem 20. *Suppose, for a given $\boldsymbol{u} \in \mathcal{W}$, there exist constants $a, b > 0$ such that the functions f_t all satisfy*

$$f_t(\boldsymbol{u}) \geq f_t(\boldsymbol{w}) + a(\boldsymbol{u}-\boldsymbol{w})^\intercal \nabla f_t(\boldsymbol{w}) + b\left((\boldsymbol{u}-\boldsymbol{w})^\intercal \nabla f_t(\boldsymbol{w})\right)^2 \qquad \textit{for all } \boldsymbol{w} \in \mathcal{W}. \tag{5.3.1}$$

Then any method with regret bound (5.1.1) *incurs logarithmic regret, $R_T^{\boldsymbol{u}} = O(d \ln T)$, with respect to $\boldsymbol{u}$.*

The case $a = 1$ of this condition was introduced by (Hazan et al., 2007), who show that it is satisfied for all $\boldsymbol{u} \in \mathcal{W}$ by exp-concave and strongly convex functions. These are both requirements on the curvature of f_t that are stronger than mere convexity: α-exp-concavity of f for $\alpha > 0$ means that $e^{-\alpha f}$ is concave or, equivalently, that $\nabla^2 f \succeq \alpha \nabla f \nabla f^\intercal$; α-strong convexity means that $\nabla^2 f \succeq \alpha \boldsymbol{I}$. We see that α-strong convexity implies $(\alpha/\|\nabla f\|_2^2)$-exp-concavity. The rate $O(d \log T)$ is also what we would expect by summing the asymptotic offline rate obtained by ridge regression on the squared loss (Srebro et al., 2010, Section 5.2), which is exp-concave. Our extension to $a > 1$ is technically a minor step, but it makes the condition much more liberal, because it may then also be satisfied by functions that do *not* have any curvature. For example, suppose that $f_t = f$ is a fixed convex function that does not change with t. Then, when $\boldsymbol{u}^* = \arg\min_{\boldsymbol{u}} f(\boldsymbol{u})$ is the offline minimizer, we have $(\boldsymbol{u}^* - \boldsymbol{w})^\intercal \nabla f(\boldsymbol{w}) \in [-G'D', 0]$, so that

$$\begin{aligned} f(\boldsymbol{u}^*) - f(\boldsymbol{w}) \geq& (\boldsymbol{u}^* - \boldsymbol{w})^\intercal \nabla f(\boldsymbol{w}) \\ \geq& 2(\boldsymbol{u}^* - \boldsymbol{w})^\intercal \nabla f(\boldsymbol{w}) + \frac{1}{D'G'}\left((\boldsymbol{u}^* - \boldsymbol{w})^\intercal \nabla f(\boldsymbol{w})\right)^2, \end{aligned}$$

where the first inequality uses only convexity of f. Thus condition (5.3.1) is satisfied by *any fixed convex function*, even if it does not have any curvature at all, with $a = 2$ and $b = 1/(G'D')$.

Bernstein Stochastic Gradients The possibility of getting fast rates even without any curvature is intriguing, because it goes beyond the usual strong convexity or exp-concavity conditions. In the online setting, the case of fixed functions $f_t = f$ seems rather restricted, however, and may in fact be handled by offline optimization methods. We therefore seek to loosen this requirement by replacing it by a stochastic condition on the distribution of the functions f_t. The relation between variance bounds like Theorem 19 and fast rates in the stochastic setting is studied in depth by (Koolen et al., 2016), who obtain fast rate results both in expectation and in probability. Here we provide a direct proof only for the expected regret, which allows a simplified analysis.

Suppose the functions f_t are independent and identically distributed (i.i.d.), with common distribution $\mathbb{P}$. Then we say that the gradients satisfy the (B, β)*-Bernstein condition* with respect to the stochastic optimum $\boldsymbol{u}^* = \arg\min_{\boldsymbol{u}\in\mathcal{W}} \mathbb{E}_{f\sim\mathbb{P}}[f(\boldsymbol{u})]$ if for all $\boldsymbol{w} \in \mathcal{W}$.

$$(\boldsymbol{w}-\boldsymbol{u}^*)^\intercal \mathop{\mathbb{E}}_{f}[\nabla f(\boldsymbol{w})\nabla f(\boldsymbol{w})^\intercal]\,(\boldsymbol{w}-\boldsymbol{u}^*) \;\le\; B\big((\boldsymbol{w}-\boldsymbol{u}^*)^\intercal \mathop{\mathbb{E}}_{f}[\nabla f(\boldsymbol{w})]\,\big)^{\beta}. \quad (5.3.2)$$

This is an instance of the well-known Bernstein condition from offline statistical learning (Bartlett and Mendelson, 2006; Van Erven et al., 2015), applied to the linearized excess loss $(\boldsymbol{w} - \boldsymbol{u}^*)^\intercal \nabla f(\boldsymbol{w})$. As shown in Section 5.14, imposing the condition for the linearized excess loss is a weaker requirement than imposing it for the original excess loss $f(\boldsymbol{w}) - f(\boldsymbol{u}^*)$.

Theorem 21. *If the gradients satisfy the* (B, β)*-Bernstein condition for* $B > 0$ *and* $\beta \in (0, 1]$ *with respect to* $\boldsymbol{u}^* = \arg\min_{\boldsymbol{u}\in\mathcal{W}} \mathbb{E}_{f\sim\mathbb{P}}[f(\boldsymbol{u})]$*, then any method with regret bound* (5.1.1) *incurs expected regret*

$$\mathbb{E}[R_T^{\boldsymbol{u}^*}] = O\left((Bd\ln T)^{1/(2-\beta)}\, T^{(1-\beta)/(2-\beta)} + d\ln T\right).$$

For $\beta = 1$, the rate becomes $O(d \ln T)$, just like for fixed functions, and for smaller β it is in between logarithmic and $O(\sqrt{dT})$. For instance, the hinge loss on the unit ball with i.i.d. data satisfies the Bernstein condition with $\beta = 1$, which implies an $O(d \log T)$ rate. (See Section 5.10.4.) It is common to add ℓ_2-regularization to the hinge loss to make it strongly convex, but this example shows that that is not necessary to get logarithmic regret.

5.4 Full Matrix Version of the MetaGrad Algorithm

In this section, we explain the full matrix version of the MetaGrad algorithm: MetaGrad$^{\text{FULL}}$. Computationally more efficient extensions follow in Section 5.5. MetaGrad$^{\text{FULL}}$ will be defined by means of the following *surrogate loss* $\ell_t^\eta(\boldsymbol{u})$:

$$\ell_t^\eta(\boldsymbol{u}) \;:=\; \eta(\boldsymbol{u} - \boldsymbol{w}_t)^\intercal \boldsymbol{g}_t + \big(\eta(\boldsymbol{u} - \boldsymbol{w}_t)^\intercal \boldsymbol{g}_t\big)^2. \quad (5.4.1)$$

This surrogate loss consists of a linear and a quadratic part, whose relative importance is controlled by a learning rate parameter $\eta > 0$. The sum of the quadratic parts is what appears in the regret bound of Theorem 19. They may be viewed as causing a "time-varying regularizer" (Orabona et al., 2015b) or "temporal adaptation of the proximal function" (Duchi et al., 2011).

MetaGrad$^{\text{FULL}}$ is a two-level hierarchical construction: at the top is a main controller, shown in Algorithm 10, which manages multiple η-experts, shown in Algorithm 11. Each η-expert produces predictions for the surrogate loss ℓ_t^η with its own value of η, and the controller is responsible for learning the best η by starting and stopping multiple η-experts on demand, and aggregating their predictions.

Algorithm 10 Full MetaGrad: Controller

```
 1: for $t = 1, 2, \dots$ do
 2:     Receive domain $\mathcal{W}_t$
 3:     Start and stop $\eta$-experts to manage active set $\mathcal{A}_t$ (see (5.4.2)).
        Give newly started $\eta$-experts weight $p_t(\eta) = 1$.
 4:     if Nobody active: $\mathcal{A}_t = \emptyset$ then
 5:         Predict $\boldsymbol{w}_t = \boldsymbol{0}$                    ▷ Make a default prediction
 6:     else
 7:         Have active $\eta$-experts project onto $\mathcal{W}_t$
 8:         Collect prediction $\boldsymbol{w}_t^\eta$ for every active $\eta$-expert
 9:         Predict
                $\boldsymbol{w}_t = \frac{\sum_{\eta \in \mathcal{A}_t} p_t(\eta)\eta\boldsymbol{w}_t^\eta}{\sum_{\eta \in \mathcal{A}_t} p_t(\eta)\eta}$
10:     end if
11:     Receive gradient $\boldsymbol{g}_t = \nabla f_t(\boldsymbol{w}_t)$ and range bound $b_t$ (see (5.2.1))
12:     Update every active $\eta$-expert with unclipped surrogate loss $\ell_t^\eta$
13:     if No reset needed after round $t$ (see (5.4.3)) then
14:         Update based on the clipped surrogate losses (see (5.4.4)):
            $p_{t+1}(\eta) = \frac{p_t(\eta)\exp\big(-\bar{\ell}_t^\eta(\boldsymbol{w}_t^\eta)\big)}{\sum_{\eta \in \mathcal{A}_t} p_t(\eta)\exp\big(-\bar{\ell}_t^\eta(\boldsymbol{w}_t^\eta)\big)}(\sum_{\eta \in \mathcal{A}_t} p_t(\eta))$    for all $\eta \in \mathcal{A}_t$.
15:     else
16:         Set $p_{t+1}(\eta) = 1$ for all $\eta \in \mathcal{A}_t$                ▷ Reset
17:     end if
18: end for
```

Controller Online learning of the best learning rate η is notoriously difficult because the regret is non-monotonic over rounds and may have multiple local minima as a function of η (see (Koolen et al., 2014) for a study in the expert setting). The standard technique is therefore to derive a monotonic upper bound on the regret and tune the learning rate optimally *for the bound*. In contrast, our approach, inspired by the approach for combinatorial games of Koolen and Van Erven (2015, Section 4), is to weigh the different η depending on their empirical performance using exponential weights with sleeping experts (line 14), except that in the predictions the weights of the η-experts are *tilted* by their learning rates (line 9), having the effect of giving a larger weight to larger η. Although we provide a formal analysis of the regret, the controller algorithm does not depend on the outcome of this analysis, so any slack in our bounds does not feed back into the algorithm.

To be able to adapt to the norms of the gradients, the controller maintains a finite grid $\mathcal{A}_t$ of active learning rates η, which is dynamically adjusted over time:

$$\mathcal{A}_t = \begin{cases} \emptyset & \text{while } B_{t-1} = 0, \\ \{2^i \mid i \in \mathbb{Z}\} \cap \left(\frac{1}{4\left(\sum_{s=1}^{t-1} b_s \frac{B_{s-1}}{B_s} + B_{t-1}\right)}, \frac{1}{4B_{t-1}} \right] & \text{afterwards.} \end{cases} \tag{5.4.2}$$

Using that $b_s \frac{B_{s-1}}{B_s} \leq B_{t-1}$, it can be seen that the number of active learning rates never exceeds $|\mathcal{A}_t| \leq \lceil \log_2 T \rceil$. In the first two rounds, or if there is a sudden enormous gradient such that B_{t-1} dwarfs $\sum_{s=1}^{t-1} b_s B_{s-1}/B_s$, it may also happen that $\mathcal{A}_t$ is empty, which signals that all previous rounds were negligible compared to the last round. In such cases the controller decides it has not yet learned anything, and makes a default prediction: $\boldsymbol{w}_t = \mathbf{0}$.

There are two further mechanisms to deal with extreme changes in the size of the gradients. The first mechanism is that extremely large gradients may trigger a *reset* of the controller's weights on η-experts. This splits the controller's learning process into epochs. When running in an epoch starting at time $\tau + 1$, a reset and new epoch will be triggered after the first round t such that

$$B_t > B_\tau \sum_{s=1}^{t} \frac{b_s}{B_s}. \tag{5.4.3}$$

As the sum on the right-hand side will typically grow linearly in t, we only expect a reset to occur when the effective size of the gradients grows by more than a factor t compared to the largest size seen before the start of the epoch. This should normally be very rare except perhaps for a few initial rounds when t is still small.

The second mechanism to protect against extreme gradients is that the controller measures performance of the experts by a *clipped* version of their corresponding surrogate losses:

$$\bar{\ell}_t^\eta(\boldsymbol{u}) \;:=\; \eta(\boldsymbol{u} - \boldsymbol{w}_t)^\intercal \bar{\boldsymbol{g}}_t + \left(\eta(\boldsymbol{u} - \boldsymbol{w}_t)^\intercal \bar{\boldsymbol{g}}_t\right)^2, \tag{5.4.4}$$

which are based on the clipped gradients

$$\bar{\boldsymbol{g}}_t \;:=\; \frac{B_{t-1}}{B_t} \boldsymbol{g}_t.$$

This is a trick first used by Cutkosky (2019), which makes the effective sizes of the gradients predictable one round in advance: $\max_{\boldsymbol{u} \in \mathcal{W}_t} |\boldsymbol{u}^\intercal \bar{\boldsymbol{g}}_t| \leq B_{t-1}$.

Algorithm 11 Full MetaGrad: η-Expert

```
Input: Learning rate $\eta > 0$, estimate $\hat{D} > 0$ of comparator norm $\|\boldsymbol{u}\|_2$,
       activation round $a \equiv a^\eta$
1: Initialize $\widetilde{\boldsymbol{w}}_a^\eta = \boldsymbol{0}, \Sigma_a^\eta = \hat{D}^2 \boldsymbol{I}$ and $\boldsymbol{\Lambda}_a^\eta = \frac{1}{\hat{D}^2}\boldsymbol{I}$
2: for $t = a, a+1, \dots$ do
3:     Project $\boldsymbol{w}_t^\eta = \arg\min_{\boldsymbol{u} \in \mathcal{W}_t} (\boldsymbol{u} - \widetilde{\boldsymbol{w}}_t^\eta)^\intercal \boldsymbol{\Lambda}_t^\eta (\boldsymbol{u} - \widetilde{\boldsymbol{w}}_t^\eta)$
4:     Predict $\boldsymbol{w}_t^\eta$
5:     Observe gradient $\boldsymbol{g}_t = \nabla f_t(\boldsymbol{w}_t)$ ▷ Gradient at controller prediction $\boldsymbol{w}_t$
6:     Update:
       $\Sigma_{t+1}^\eta = \Sigma_t^\eta - \frac{2\eta^2 (\Sigma_t^\eta \boldsymbol{g}_t)(\boldsymbol{g}_t^\intercal \Sigma_t^\eta)}{1 + 2\eta^2 \boldsymbol{g}_t^\intercal \Sigma_t^\eta \boldsymbol{g}_t}$
       $\boldsymbol{\Lambda}_{t+1}^\eta = \boldsymbol{\Lambda}_t^\eta + \boldsymbol{g}_s \boldsymbol{g}_s^\intercal$
       $\widetilde{\boldsymbol{w}}_{t+1}^\eta = \boldsymbol{w}_t^\eta - (1 + 2\eta(\boldsymbol{w}_t^\eta - \boldsymbol{w}_t)^\intercal \boldsymbol{g}_t)\, \eta \Sigma_{t+1}^\eta \boldsymbol{g}_t$
7: end for
```

η-Experts Each η-expert is active for a single contiguous sequence of rounds for which $\eta \in \mathcal{A}_t$. Upon activation, its job is to issue predictions $\boldsymbol{w}_t^\eta \in \mathcal{W}_t$ for the (unclipped) surrogate loss ℓ_t^η that achieve small regret compared to any $\boldsymbol{u} \in \bigcap_{t:\eta \in \mathcal{A}_t} \mathcal{W}_t$. This is a standard online convex optimization task with a quadratic loss function and time-varying domain, which we assume is non-empty. We use continuous exponential weights with a Gaussian prior, which is a standard approach for quadratic losses (Vovk, 2001), because the corresponding posterior exponential weights distribution is also Gaussian with mean $\boldsymbol{w}_t^\eta$ and covariance matrix $\Sigma_t^\eta = \left(\frac{1}{\hat{D}^2}\boldsymbol{I} + 2\eta^2 \sum_{s=a}^t \boldsymbol{g}_s \boldsymbol{g}_s^\intercal\right)^{-1}$. Algorithm 11 presents the update equations in a computationally efficient form. To avoid inverting Σ_t^η, it maintains its inverse $\boldsymbol{\Lambda}_t^\eta = (\Sigma_t^\eta)^{-1}$ separately. For a recent overview of continuous exponential weights see Van der Hoeven et al. (2018). It can be seen that our η-expert algorithm is nearly identical to Online Newton Step (ONS) (Hazan et al., 2007), which is not surprising because ONS is minimizing a quadratic loss that is nearly identical to our ℓ_t^η. The differences are that each η-expert receives the controller's gradient $\boldsymbol{g}_t = \nabla f_t(\boldsymbol{w}_t)$ instead of its own $\nabla f_t(\boldsymbol{w}_t^\eta)$, and that an additional term $(1 + 2\eta(\boldsymbol{w}_t^\eta - \boldsymbol{w}_t)^\intercal \boldsymbol{g}_t)$ in line 6 adjusts for the difference between the η-expert's parameters $\boldsymbol{w}_t^\eta$ and the controller's parameters $\boldsymbol{w}_t$. MetaGrad is therefore a bona fide first-order algorithm that only accesses f_t through $\boldsymbol{g}_t$. We also note that we have chosen the Greedy projections version that iteratively updates and projects (see line 6). One might alternatively consider the Lazy Projection version (as in Zinkevich (2004); Nesterov (2009); Xiao (2010)) that forgets past projections when updating on new data. Since projections are typically computationally expensive, we have opted for the Greedy projection version, which we expect to project less often, since a projected point seems less likely to update to a point outside of the domain than an unprojected point.

5.4.1 Practical Considerations

Although MetaGrad$^{\text{FULL}}$ is adaptive to the maximum effective size of the gradients B_T, its performance degrades when B_T becomes too large. In applications, it is therefore important that the domain $\mathcal{W}_t$ is small enough along the direction of $\boldsymbol{g}_t$ to keep the effective gradient size b_t under control.

It is further required to choose the hyperparameter $\hat{D}$, which is an estimate of the ℓ_2-norm of the comparator $\boldsymbol{u}$. Theorem 23 quantifies the trade-off between underestimating and overestimating this parameter. Note that overestimating $\|\boldsymbol{u}\|_2$ only incurs a logarithmic penalty, so it is less expensive to use a too large value rather than a too small value.

Finally, we note that there is no gain in pre-processing the data by scaling all gradients by a fixed constant factor, since the regret bound in Theorem 23 is scale-free. In fact, the MetaGrad$^{\text{FULL}}$ algorithm is almost invariant under such rescaling, except for the term $\{2^i \mid i \in \mathbb{Z}\}$ in the definition of $\mathcal{A}_t$. If one wants to make the algorithm fully invariant under rescaling, this term may be replaced by $\{2^i/B_\tau \mid i \in \mathbb{Z}\}$, where τ is the first round that $B_\tau > 0$. Or, equivalently, one may replace all gradients by $\boldsymbol{g}_t/B_\tau$ for $t > \tau$. Since we do not expect any noticeable difference in performance from this modification, we have left it out.

Run Time The run time of MetaGrad$^{\text{FULL}}$ is dominated by computations for the η-experts. Ignoring the projection step, an η-expert takes $O(d^2)$ time to update. If there are at most k' active η-experts in any round, this makes the overall computational effort $O(k'd^2)$, both in time per round and in memory. Since $|\mathcal{A}_t| \leq \lceil \log_2 T \rceil$, it is guaranteed that $k' \leq 30$ as long as $T \leq 10^9$. We note that all η-experts share the same gradient $\boldsymbol{g}_t$, which is only computed once. We remark that a potential speed-up is possible by running the η-experts in parallel. If the factor k' is still considered too large, it is possible to reduce the size of $|\mathcal{A}_t|$ by spacing the learning rates by a factor larger than 2, at the cost of a worse constant in the regret bound.

In addition, each η-expert may incur the cost of a projection, which depends on the shape of the domain $\mathcal{W}_t$. To get a sense for the projection cost, we consider the Euclidean ball as a typical example. If the matrix Σ_t^η were diagonal, we could project to any desired precision using a few iterations of Newton's method. Since each such iteration takes $O(d)$ time, this would be affordable. But for the non-diagonal Σ_t^η that occur in the algorithm, we first need to reduce to the diagonal case by a basis transformation, which takes $O(d^3)$ to compute using a singular value decomposition. We therefore see that the projection dwarfs the other run time by an order of magnitude. This has motivated Luo et al. (2017) to define a different domain (see Section 5.5.1), for which projections can be computed in closed form with $O(d)$ computation steps. In this case, the computation for the projections is negligible and the total computational complexity is $O(d^2)$ per round. We refer to Duchi et al. (2011) for examples of how to compute projections for various other domains $\mathcal{W}_t$.

5.5 Faster Extension Algorithms

As discussed above, MetaGrad$^{\text{FULL}}$ requires at least $O(d^2)$ computation per round, which makes it slow in high dimensions. We therefore present two extensions to speed up the algorithm. The first is a straightforward adaption of the sketching approach of Luo et al. (2017), which we apply to approximate the matrix Σ_t^η in the η-experts. This reduces the computation per round to $O(kd)$, where k is a hyper-parameter that determines the sketch size. The second extension is to run a separate copy of the algorithm per dimension, which was inspired by the diagonal version of AdaGrad (Duchi et al., 2011). This requires $O(d)$ computation per round.

5.5.1 Sketched MetaGrad with Closed-form Projections

In this section, we are mixing matrices of different dimensions. The identity matrix $\boldsymbol{I}_d \in \mathbb{R}^d$ and the all-zeros matrix $\mathbf{0}_{a\times b} \in \mathbb{R}^{a\times b}$ are therefore annotated with subscripts to make their dimensions explicit. To simplify notation, we further assume without loss of generality that the η-experts are started in round $a^\eta = 1$.

Luo et al. (2017) develop several sketching approaches for Online Newton Step, which transfer directly to our η-experts. They combine these with a computationally efficient choice of the domain that applies to loss functions of the form $f_t(\boldsymbol{w}) = h_t(\boldsymbol{w}^\intercal \boldsymbol{x}_t)$, where the input vectors $\boldsymbol{x}_t \in \mathbb{R}^d$ are assumed to be known at the start of round t, but the convex functions $h_t : \mathbb{R} \to \mathbb{R}$ are not. They then choose the domain to be

$$\mathcal{W}_t = \{\boldsymbol{w} : |\boldsymbol{w}^\intercal \boldsymbol{x}_t| \leq C\} \qquad \text{for a fixed constant } C. \tag{5.5.1}$$

Let $\boldsymbol{G}_t = (\boldsymbol{g}_1, \ldots, \boldsymbol{g}_t)^\intercal \in \mathbb{R}^{t \times d}$, such that $\Sigma_{t+1}^\eta = (\frac{1}{\hat{D}^2}\boldsymbol{I}_d + 2\eta^2 \boldsymbol{G}_t^\intercal \boldsymbol{G}_t)^{-1}$. The idea of sketching is to replace Σ_{t+1}^η by an approximation

$$\widetilde{\Sigma}_{t+1}^\eta = \left(\tfrac{1}{\hat{D}^2}\boldsymbol{I}_d + 2\eta^2 \boldsymbol{S}_t^\intercal \boldsymbol{S}_t\right)^{-1},$$

where $\boldsymbol{S}_t \in \mathbb{R}^{k \times d}$ for a given *sketch size* $k \leq d$, so that $\boldsymbol{S}_t^\intercal \boldsymbol{S}_t$ has rank at most k. Abbreviating $\hat{\boldsymbol{g}}_t = (1 + 2\eta(\boldsymbol{w}_t^\eta - \boldsymbol{w}_t)^\intercal \boldsymbol{g}_t)\, \eta \boldsymbol{g}_t$, we then need to compute

$$\begin{aligned} \boldsymbol{w}_t^\eta &= \operatorname*{arg\,min}_{\boldsymbol{u} \in \mathcal{W}_t} \ (\boldsymbol{u} - \widetilde{\boldsymbol{w}}_t^\eta)^\intercal (\widetilde{\Sigma}_t^\eta)^{-1} (\boldsymbol{u} - \widetilde{\boldsymbol{w}}_t^\eta) && \text{(projection)} \\ \widetilde{\boldsymbol{w}}_{t+1}^\eta &= \boldsymbol{w}_t^\eta - \widetilde{\Sigma}_{t+1}^\eta \hat{\boldsymbol{g}}_t. && \text{(update)} \end{aligned}$$

The key to an efficient implementation of these steps is to rewrite $\widetilde{\Sigma}_{t+1}^\eta$ using the Woodbury identity (Golub and Van Loan, 2012):

$$\widetilde{\Sigma}_{t+1}^\eta = \hat{D}^2(\boldsymbol{I}_d - 2\eta^2 \boldsymbol{S}_t^\intercal (\tfrac{1}{\hat{D}^2}\boldsymbol{I}_k + 2\eta^2 \boldsymbol{S}_t \boldsymbol{S}_t^\intercal)^{-1} \boldsymbol{S}_t) = \hat{D}^2(\boldsymbol{I}_d - 2\eta^2 \boldsymbol{S}_t^\intercal \boldsymbol{H}_t^\eta \boldsymbol{S}_t),$$

where we have introduced the abbreviation $\boldsymbol{H}_t^\eta = (\frac{1}{\hat{D}^2}\boldsymbol{I}_k + 2\eta^2 \boldsymbol{S}_t \boldsymbol{S}_t^\intercal)^{-1}$. Let $\mathrm{s}_C(y) = \mathrm{sign}(y)\max\{|y| - C, 0\}$. By Lemma 1 of Luo et al. (2017), the projection step then becomes

$$\boldsymbol{w}_t^\eta = \widetilde{\boldsymbol{w}}_t^\eta - \frac{\mathrm{s}_C(\boldsymbol{x}_t^\intercal \widetilde{\boldsymbol{w}}_t^\eta)}{(\boldsymbol{x}_t^\intercal \boldsymbol{x}_t - 2\eta^2 \boldsymbol{x}_t^\intercal \boldsymbol{S}_t^\intercal \boldsymbol{H}_t^\eta \boldsymbol{S}_t \boldsymbol{x}_t)} (\boldsymbol{x}_t - 2\eta^2 \boldsymbol{S}_t^\intercal \boldsymbol{H}_t^\eta \boldsymbol{S}_t \boldsymbol{x}_t),$$

and the update step can be written as

$$\widetilde{\boldsymbol{w}}_{t+1}^\eta = \boldsymbol{w}_t^\eta - \hat{D}^2(\hat{\boldsymbol{g}}_t - 2\eta^2 \boldsymbol{S}_t^\intercal \boldsymbol{H}_t^\eta \boldsymbol{S}_t \hat{\boldsymbol{g}}_t).$$

Assuming that $\boldsymbol{S}_t$ and $\boldsymbol{H}_t^\eta$ can be efficiently maintained, the operations involving $\boldsymbol{S}_t \boldsymbol{x}_t$ or $\boldsymbol{S}_t \hat{\boldsymbol{g}}_t$ require $O(kd)$ computation time and matrix-vector products with $\boldsymbol{H}_t^\eta$ can be performed in $O(k^2)$ time. As noted by Luo et al. (2017), both of these are only a factor k more than the $O(d)$ time required by first-order methods. They describe two sketching techniques to maintain $\boldsymbol{S}_t$ and $\boldsymbol{H}_t^\eta$, each requiring $O(kd)$ storage and $O(kd)$ average computation time per round. The first technique is based on Frequent Directions (FD) sketching; the other one on Oja's algorithm. We adopt the FD approach, which comes with a guaranteed bound on the regret. Luo et al. (2017) further develop an extension of FD for sparse gradients, and yet another option would have been the Robust Frequent Directions sketching method of Luo et al. (2019).

Frequent Directions Sketching

Some sketching approaches are randomized, but Frequent Directions sketching (Ghashami et al., 2016) is a deterministic method. The simplest version (Luo et al., 2017, Algorithm 2) performs a singular value decomposition (SVD) of $\boldsymbol{S}_t$ every round at the cost of $O(k^2 d)$ computation time, but there also exists a refined epoch-based version which only performs an SVD once per epoch. Each epoch takes m rounds and $k = 2m$, leading to an average runtime of $O(kd)$ per round. We describe here the epoch version, adapted from Algorithm 6 of Luo et al. (2017) and summarized in Algorithm 12.

Algorithm 12 Frequent Directions Sketching

```
Initialize $\boldsymbol{S}_0 = \boldsymbol{0}_{2m \times d}$, and $\boldsymbol{H}_0^\eta = \hat{D}^2 \boldsymbol{I}_{2m}$.
for $t = 1, 2, \dots$ do
    Let $\tau = t \bmod m$ and insert $\boldsymbol{g}_t^\intercal$ in the $(m+\tau)$-th row of $\boldsymbol{S}_{t-1}$ to obtain $\tilde{\boldsymbol{S}}$.
    if $\tau \neq 0$ then
        Set $\boldsymbol{S}_t = \tilde{\boldsymbol{S}}$.
        Let $\boldsymbol{e} \in \mathbb{R}^{2m}$ be the basis vector in direction $m + \tau$ and $\boldsymbol{q} = 2\eta^2(\tilde{\boldsymbol{S}}\boldsymbol{g}_t - \frac{\boldsymbol{g}_t^\intercal \boldsymbol{g}_t}{2}\boldsymbol{e})$.
        Update $\boldsymbol{H}_t^\eta = \tilde{\boldsymbol{H}} - \frac{\tilde{\boldsymbol{H}}\boldsymbol{e}\boldsymbol{q}^\intercal \tilde{\boldsymbol{H}}}{1+\boldsymbol{q}^\intercal \tilde{\boldsymbol{H}}\boldsymbol{e}}$, where $\tilde{\boldsymbol{H}} = \boldsymbol{H}_{t-1}^\eta - \frac{\boldsymbol{H}_{t-1}^\eta \boldsymbol{q}\boldsymbol{e}^\intercal \boldsymbol{H}_{t-1}^\eta}{1+\boldsymbol{e}^\intercal \boldsymbol{H}_{t-1}^\eta \boldsymbol{q}}$.
    else
        From the SVD of $\tilde{\boldsymbol{S}}$, compute the top-$m$ singular values $\sigma_1 \geq \dots \geq \sigma_m$ and corresponding right-singular vectors as $\boldsymbol{V} \in \mathbb{R}^{d \times m}$.
        Set $\boldsymbol{S}_t = \mathrm{diag}(\sigma_1^2 - \sigma_m^2, \dots, \sigma_m^2 - \sigma_m^2)^{1/2} \boldsymbol{V}^\intercal$.
        Set $\boldsymbol{H}_t^\eta = \mathrm{diag}(\frac{1}{\hat{D}^{-2}+2\eta^2(\sigma_1^2-\sigma_m^2)}, \dots, \frac{1}{\hat{D}^{-2}+2\eta^2(\sigma_m^2-\sigma_m^2)}, \frac{1}{\hat{D}^{-2}}, \dots, \frac{1}{\hat{D}^{-2}})$.
    end if
end for
```

Recall that $\boldsymbol{S}_t^\intercal \boldsymbol{S}_t$ is an approximation of $\boldsymbol{G}_t^\intercal \boldsymbol{G}_t$. At the start of each epoch, we have the invariant that only the first $m-1$ rows of $\boldsymbol{S}_t$ contribute to this approximation and the remaining $m+1$ rows are filled with zeros. During the τ-th round in any epoch we first add the incoming gradient $\boldsymbol{g}_t^\intercal$ to row $m+\tau$ of $\boldsymbol{S}_{t-1}$ to obtain an intermediate result $\tilde{\boldsymbol{S}}$. If we are not yet in the last round of the epoch (i.e. $\tau < m$), then we simply set $\boldsymbol{S}_t = \tilde{\boldsymbol{S}}$, and we use that

$$(\boldsymbol{H}_t^\eta)^{-1} = (\boldsymbol{H}_{t-1}^\eta)^{-1} + \boldsymbol{q}\boldsymbol{e}^\intercal + \boldsymbol{e}\boldsymbol{q}^\intercal,$$

where $\boldsymbol{e} \in \mathbb{R}^{2m}$ is the basis vector in direction $m + \tau$ and $\boldsymbol{q} = 2\eta^2(\tilde{\boldsymbol{S}}\boldsymbol{g}_t - \frac{\boldsymbol{g}_t^\intercal \boldsymbol{g}_t}{2}\boldsymbol{e})$. It follows that we can compute $\boldsymbol{H}_t^\eta$ from $\boldsymbol{H}_{t-1}^\eta$ using two rank-one updates with the Sherman-Morrison formula:

$$\boldsymbol{H}_t^\eta = \tilde{\boldsymbol{H}} - \frac{\tilde{\boldsymbol{H}}\boldsymbol{e}\boldsymbol{q}^\intercal \tilde{\boldsymbol{H}}}{1 + \boldsymbol{q}^\intercal \tilde{\boldsymbol{H}}\boldsymbol{e}}, \text{ where } \tilde{\boldsymbol{H}} = \boldsymbol{H}_{t-1}^\eta - \frac{\boldsymbol{H}_{t-1}^\eta \boldsymbol{q}\boldsymbol{e}^\intercal \boldsymbol{H}_{t-1}^\eta}{1 + \boldsymbol{e}^\intercal \boldsymbol{H}_{t-1}^\eta \boldsymbol{q}}.$$

Otherwise, if we are in the last round of the epoch (i.e. $\tau = m$), the invariant is restored by eigen decomposing $\tilde{\boldsymbol{S}}^\intercal \tilde{\boldsymbol{S}}$ into $\boldsymbol{W}\boldsymbol{\Lambda}\boldsymbol{W}^\intercal$, where $\boldsymbol{\Lambda} = \operatorname{diag}(\lambda_1, \ldots, \lambda_{2m})$ contains the potentially non-zero eigenvalues in non-decreasing order $\lambda_1 \geq \cdots \geq \lambda_{2m}$ and the columns of $\boldsymbol{W} \in \mathbb{R}^{d \times 2m}$ contain the corresponding eigenvectors. Then we set $\boldsymbol{S}_t = \operatorname{diag}(\lambda_1 - \lambda_m, \ldots, \lambda_m - \lambda_m, 0, \ldots, 0)^{1/2}\boldsymbol{W}^\intercal$. Since the rows of $\boldsymbol{S}_t$ are now orthogonal,

$$\begin{aligned}\boldsymbol{H}_t^\eta &= (\tfrac{1}{\hat{D}^2}\boldsymbol{I}_{2m} + 2\eta^2 \boldsymbol{S}_t \boldsymbol{S}_t^\intercal)^{-1} \\ &= \operatorname{diag}\left(\frac{1}{\hat{D}^{-2} + 2\eta^2(\lambda_1 - \lambda_m)}, \ldots, \frac{1}{\hat{D}^{-2} + 2\eta^2(\lambda_m - \lambda_m)}, \frac{1}{\hat{D}^{-2}}, \ldots, \frac{1}{\hat{D}^{-2}}\right)\end{aligned}$$

is a diagonal matrix.

Implementation Details When implementing the FD procedure, we can calculate the eigen decomposition of $\tilde{\boldsymbol{S}}^\intercal \tilde{\boldsymbol{S}}$ via an SVD of $\tilde{\boldsymbol{S}}$, which can be performed in $O(m^2 d)$ computation steps. The eigenvalues λ_i then correspond to the squared singular values σ_i^2 of $\tilde{\boldsymbol{S}}$, and $\boldsymbol{W}$ contains the corresponding right-singular vectors. In fact, we only need the top-m singular values and the corresponding m right-singular vectors $\boldsymbol{V} \in \mathbb{R}^{d \times m}$ to compute $\boldsymbol{S}_t = \operatorname{diag}(\lambda_1 - \lambda_m, \ldots, \lambda_m - \lambda_m, 0, \ldots, 0)^{1/2}\boldsymbol{W}^\intercal = \operatorname{diag}(\sigma_1^2 - \sigma_m^2, \ldots, \sigma_m^2 - \sigma_m^2)^{1/2}\boldsymbol{V}^\intercal$.

We further observe that, since $\boldsymbol{S}_t$ does not depend on η, we only need to compute it once when sketching for multiple η-experts with different learning rates η. The matrix $\boldsymbol{H}_t^\eta$, however, does need to be computed for each η separately.

Practical Considerations

Sketching introduces an extra hyper-parameter $k = 2m$, which controls the sketch size. In theory, larger k provides a better approximation of the full version of MetaGrad, at the cost of more computation.

5.5.2 Coordinate MetaGrad

Duchi et al. (2011) introduce a full and a diagonal version of their AdaGrad algorithm. The diagonal version, which is the version that is widely used in applications, may be interpreted as running a copy of online gradient descent (Zinkevich, 2003) for each dimension separately, with a separate data-dependent tuning of the step size per dimension. This approach of running a separate copy per dimension can be applied to any online learning algorithm, and works out as follows.

We output a joint prediction $\boldsymbol{w}_t = (w_{t,1}, \dots, w_{t,d})^\intercal$, where each $w_{t,i}$ is the output of the copy of the algorithm for dimension i. Each of these copies gets as inputs the 1-dimensional losses $f_{t,i}(w) = wg_{t,i}$, where $g_{t,i}$ is the i-th component of the joint gradient $\boldsymbol{g}_t = \nabla f_t(\boldsymbol{w}_t)$. This works because the linearized regret decomposes per dimension:

$$\sum_{t=1}^{T} (\boldsymbol{w}_t - \boldsymbol{u})^\intercal \boldsymbol{g}_t = \sum_{i=1}^{d} \sum_{t=1}^{T} (f_{t,i}(w_{t,i}) - f_{t,i}(u_i)),$$

so our joint linearized regret is simply the sum of the linearized regrets per dimension.

One limitation of this approach, if we apply it as is, is that the domain cannot introduce dependencies between the dimensions, so we are limited to rectangular domains:

$$\mathcal{W}_t^{\text{rect}} = \{\boldsymbol{w} \in \mathbb{R}^d \mid a_{t,i} \leq w_i \leq z_{t,i} \text{ for } i = 1, \dots, d\},$$

with our only freedom consisting of choosing the boundaries $a_{t,i}$ and $z_{t,i}$.

Practical Considerations

Running a copy of MetaGrad per dimension potentially introduces a separate hyperparameter $\hat{D}_i$ per dimension i. Like Duchi et al. (2011), we reduce the complexity of hyperparameter tuning by letting $\hat{D}_i = \hat{D}$ be the same for all dimensions. If no specific domain is required and the components of the gradients are approximately standardized, it is also generally sufficient to set the dimensions of the rectangular domain to $a_{t,i} = -q$ and $z_{t,i} = q$ for a fixed parameter q.

5.6 Analysis of the Full Matrix Version of MetaGrad

The high-level goal of MetaGrad is to deliver a tight data-dependent regret bound. Such bounds could be achieved in principle by existing algorithms, were their learning rate tuned certain a-priori unknown data-dependent quantities. The practical approach implemented in MetaGrad is to run multiple instances of a baseline

"η-expert" algorithm, each with different candidate tuning. A controller then aggregates these η-expert predictions and manages their lifetimes to always have the required tuning present.

The MetaGrad$^{\text{FULL}}$ η-experts are Exponentially Weighted Average forecasters starting from a Gaussian prior and taking in our quadratic surrogate losses. Their efficient implementation is a variant of Online Newton Step, where the losses are centred at the prediction of the controller instead of that of the η-expert. In turn, the controller is a specialists (also known as sleeping experts) algorithm to deal with the starting and retiring of η-experts. It is further designed to give a non-uniform regret guarantee, obtaining especially small regret when the best learning rate turns out to be high. Finally, our approach for adapting to the Lipschitz constant is speculative. Starting at zero, we monitor the implied Lipschitz constant of the incoming gradients. If it is increasing slowly, the controller is able to accommodate the overshoots in a lower-order term. If it makes a large jump, then the controller may need to reset. We do so by resetting the controller weights without changing the state of the affected η-experts.

5.6.1 Controller

Denote by $\mathcal{G} = \{2^i : i \in \mathbb{Z}\}$ and by a^η the starting time of an η-expert (for the exact definition of a^η see definition 3 in Section 5.11). Let us introduce the concept of expiration.

Definition 2. *We say that $\eta \in \mathcal{G}$ is* expired *after T rounds (or, equivalently, after round T) if $\eta > \frac{1}{4B_{T-1}}$.*

Note that expiration can be checked *before* the round happens (it is "predictable"). All learning rates used by Algorithm 10 by means of the active set $\mathcal{A}_t$ (5.4.2) are not expired. Also note the "lifecycle" of any fixed learning rate η. It starts inactive unexpired. Then it becomes active unexpired. And finally it expires, after which it loses all relevance.

For the controller, we prove that it behaviour approximates that of any η-expert not expired, when measured in the η surrogate loss (5.4.1).

Lemma 15 (Controller Surrogate Regret Bound)**.** *For any learning rate $\eta \in \mathcal{G}$ not expired after T rounds and any comparator $\boldsymbol{u} \in \bigcap_{t=1}^{T} \mathcal{W}_t$,* MetaGrad$^{\text{FULL}}$

ensures

$$R_t^\eta(\boldsymbol{u}) \le \underbrace{\frac{1}{2} + 4\eta B_T}_{\text{small}} + \underbrace{2\ln\left\lceil 2\log_2\left(\sum_{t=1}^{T-1}\frac{b_t}{B_t}+1\right)\right\rceil_+}_{\text{specialist regret for epoch, } O(\ln\ln T)} + \underbrace{\sum_{t=a^\eta}^{T}\left(\ell_t^\eta(\boldsymbol{w}_t^\eta) - \ell_t^\eta(\boldsymbol{u})\right)}_{\ell^\eta\text{-regret of } \eta\text{-expert w.r.t. } \boldsymbol{u}}.$$

The proof is in Section 5.11. It follows the MetaGrad analysis of Mhammedi et al. (2019), including the range clipping technique due to Cutkosky (2019), and the reset technique of Mhammedi et al. (2019), which in particular ensures that whenever a reset occurs, the accumulated regret up until the *previous* reset is small. As such, we only have to pay for the controller regret for the last two epochs.

We further streamline the approach by using a standard specialists (sleeping experts) algorithm on a discrete grid of η-experts as our controller algorithm. Of note here is our use of the improper log-uniform prior. We also employ a slightly tightened measure b_t of the effective loss range.

To make further progress, we need to make use of the details of the η-experts.

5.6.2 Full η-Experts

Next we establish a $O(d\log T)$ regret bound in terms of the surrogate loss for each METAGRAD$^{\text{FULL}}$ η-expert. The η-experts implement Follow-the-Regularised-Leader with the quadratic losses ℓ_t^η and the squared Euclidean norm regulariser. Equivalently, we can see them as implementing the exponentially weighted average forecaster for the quadratic losses ℓ_t^η starting from a Gaussian prior. Algorithms for the specific quadratic loss arising in linear regression were designed and analysed by Vovk (2001). The general quadratic case goes back (at least) to Hazan et al. (2007), who unfortunately do not separate the analysis for general quadratic losses from the reduction of exp-concave losses to quadratics, even though these ideas are clearly present. The explicit analysis by van Erven and Koolen (2016) includes an unnecessary range restriction, which was subsequently removed by Van der Hoeven et al. (2018). As pointed out by Luo et al. (2017), the extension to time-varying domains is trivial.

Lemma 16 (Surrogate regret bound)**.** *Consider the* METAGRAD$^{\text{FULL}}$ *η-expert in Algorithm 11 with learning rate $\eta \le \frac{1}{4B_T}$ starting from time a^η. Its surrogate regret after round $T \ge a^\eta$ w.r.t. any comparator $\boldsymbol{u} \in \bigcap_{t=a^\eta}^T \mathcal{W}_t$ is bounded by*

$$\sum_{t=a^\eta}^{T}\left(\ell_t^\eta(\boldsymbol{w}_t^\eta) - \ell_t^\eta(\boldsymbol{u})\right) \le \frac{1}{2\hat{D}^2}\|\boldsymbol{u}\|_2^2 + \ln\det\left(\boldsymbol{I} + 2\eta^2\hat{D}^2\sum_{t=a^\eta}^{T}\boldsymbol{g}_t\boldsymbol{g}_t^\intercal\right).$$

The proof of Lemma 16 can be found in Section 5.12. We note that the condition on η in the lemma is slightly stricter than not being expired (Definition 2), which only requires $\eta \leq \frac{1}{4B_{T-1}}$. The reason is that the η-expert operates off the *unclipped* surrogate loss and gradients.

5.6.3 Composition (bounding the actual regret)

To complete the analysis of MetaGrad$^{\text{FULL}}$, we put the regret bounds for the controller and η-experts together. We then optimize η over the grid $\mathcal{G}$ to get our main result. For the purpose of this section, let us define the *essential horizon* and *gradient covariance* by

$$Q_T \;:=\; \sum_{t=1}^{T-1} \frac{b_t}{B_t} + 1 \qquad \text{and} \qquad \boldsymbol{F}_T \;:=\; \sum_{t=1}^{T} \boldsymbol{g}_t \boldsymbol{g}_t^\intercal .$$

Theorem 22 (Grid point regret)**.** *Let $\eta \in \mathcal{G}$ be such that $\eta \leq \frac{1}{4B_T}$. Then* MetaGrad$^{\text{FULL}}$ *guarantees that the linearized regret w.r.t. any comparator $\boldsymbol{u} \in \bigcap_{t=1}^{T} \mathcal{W}_t$ is at most*

$$\tilde{R}_T^{\boldsymbol{u}} \;\leq\; \eta V_T^{\boldsymbol{u}} + \frac{\ln\det\Big(\boldsymbol{I} + 2\eta^2 \hat{D}^2 \boldsymbol{F}_T\Big) + \frac{1}{2\hat{D}^2}\|\boldsymbol{u}\|_2^2 + 2\ln\lceil 2\log_2 Q_T\rceil_+ + \frac{1}{2}}{\eta} + 4B_T .$$

Proof. Combining the controller and η-expert surrogate regret bounds Lemma 15 and Lemma 16, we obtain

$$\begin{aligned}\sum_{t=1}^{T} \big(\ell_t^\eta(\boldsymbol{w}_t) - \ell_t^\eta(\boldsymbol{u})\big) \;\leq\; & \frac{1}{2} + 4\eta B_T + 2\ln\left\lceil 2\log_2\left(\sum_{t=1}^{T-1} \frac{b_t}{B_t} + 1\right)\right\rceil_+ \\ & + \frac{1}{2\hat{D}^2}\|\boldsymbol{u}\|_2^2 + \ln\det\left(\boldsymbol{I} + 2\eta^2\hat{D}^2 \sum_{t=1}^{T} \boldsymbol{g}_t\boldsymbol{g}_t^\intercal\right).\end{aligned}$$

The definition of the surrogate loss (5.4.1) gives $\ell_t^\eta(\boldsymbol{w}_t) - \ell_t^\eta(\boldsymbol{u}) = \eta(\boldsymbol{w}_t - \boldsymbol{u})^\intercal \boldsymbol{g}_t - \big(\eta(\boldsymbol{u} - \boldsymbol{w}_t)^\intercal \boldsymbol{g}_t\big)^2$ and the theorem follows by reorganising and dividing by η. □

The final step is to properly select the learning rate $\eta \in \mathcal{G}$ in the regret bound Theorem 22. This leads to our main result. The proof is in Section 5.13.

Theorem 23 (MetaGrad Full Regret Bound)**.** *For all $\boldsymbol{u} \in \bigcap_{t=1}^{T} \mathcal{W}_t$ the linearized regret of* MetaGrad$^{\text{FULL}}$ *is simultaneously bounded by*

$$\tilde{R}_T^{\boldsymbol{u}} \;\leq\; \frac{5}{2}\sqrt{V_T^{\boldsymbol{u}}\big(\tfrac{1}{2\hat{D}^2}\|\boldsymbol{u}\|_2^2 + Z_T\big)} + 10B_T\big(\tfrac{1}{2\hat{D}^2}\|\boldsymbol{u}\|_2^2 + Z_T\big) + 4B_T,$$

where $Z_T = \operatorname{rk}(\boldsymbol{F}_T)\ln\left(1+\frac{\hat{D}^2\sum_{t=1}^T\|\boldsymbol{g}_t\|_2^2}{8B_T^2\operatorname{rk}(\boldsymbol{F}_T)}\right) + 2\ln\lceil 2\log_2 T\rceil_+ + \frac{1}{2}$, *and by*

$$\tilde{R}_T^{\boldsymbol{u}} \le \frac{5}{2}\sqrt{\left(V_T^{\boldsymbol{u}} + 2\hat{D}^2\sum_{t=1}^T\|\boldsymbol{g}_t\|_2^2\right)\left(\tfrac{1}{2\hat{D}^2}\|\boldsymbol{u}\|_2^2 + Z'_T\right)} \\ + 10B_T\left(\tfrac{1}{2\hat{D}^2}\|\boldsymbol{u}\|_2^2 + Z'_T\right) + 4B_T,$$

where $Z'_T = 2\ln\lceil 2\log_2 T\rceil_+ + \frac{1}{2}$.

Since $\operatorname{rk}(\boldsymbol{F}_T) \le d$, Theorem 19 follows when we assume that the diameter of the domain $\mathcal{W}_t$ and the gradient norms are both uniformly bounded over all rounds, which implies $Z_T = O(d\log(T/d))$. If the eigenvalues of $\boldsymbol{F}_T$ satisfy a decay condition, then a more refined bound is possible instead of the first term in the definition of Z_T, as can be seen from the proof.

5.7 Extensions for Faster MetaGrad Analysis

5.7.1 Sketching: Analysis

The analysis for the frequent directions sketching version of MetaGrad with sketch size $k = 2m$ proceeds like the analysis of the full matrix version, except that we obtain a different bound for the η-expert regret. This bound depends on the spectral decay of $\boldsymbol{F}_T = \boldsymbol{G}_T^\intercal\boldsymbol{G}_T = \sum_{t=1}^T\boldsymbol{g}_t\boldsymbol{g}_t^\intercal$. Let λ_i be the i-th eigenvalue of $\boldsymbol{G}_T^\intercal\boldsymbol{G}_T$ and define $\Omega_q = \sum_{i=q+1}^d\lambda_i$. Then the surrogate regret of the η-expert algorithm with FD sketching is bounded as follows:

Lemma 17. *Consider the sketching version of the MetaGrad η-expert algorithm with learning rate $\eta \le \frac{1}{4B_T}$ starting from time a^η. Its surrogate regret after round $T \ge a^\eta$ w.r.t. any comparator $\boldsymbol{u} \in \bigcap_{t=a^\eta}^T\mathcal{W}_t$ is bounded by*

$$\sum_{t=a^\eta}^T\left(\ell_t^\eta(\boldsymbol{w}_t^\eta) - \ell_t^\eta(\boldsymbol{u})\right) \\ \le \frac{1}{2\hat{D}^2}\|\boldsymbol{u}\|_2^2 + \log(\det(\boldsymbol{I} + 2\eta^2\hat{D}^2\boldsymbol{S}_T^\intercal\boldsymbol{S}_T)) + \frac{2\eta^2\hat{D}^2m\Omega_q}{m-q}$$

for any $q = 0, \dots, m-1$.

Compared to Lemma 16, we see that $\boldsymbol{G}_T^\intercal\boldsymbol{G}_T = \sum_{t=1}^T\boldsymbol{g}_t\boldsymbol{g}_t^\intercal$ in the logarithmic term has been replaced by its sketching approximation $\boldsymbol{S}_T^\intercal\boldsymbol{S}_T$. We therefore pay logarithmically for the top m directions, which are captured by the sketch. What we lose is the rightmost term of order $O(\eta^2\Omega_q)$, which corresponds to the remaining $d-q$ directions that are not captured.

The proof of Lemma 17 is a straightforward adaptation of the proof of Theorem 3 by Luo et al. (2017). For the details, we refer to Chapter 4 of Deswarte (2018), with two minor remarks: the first is that Deswarte uses a slightly stricter bound on η, which allows him to bound $\frac{1}{2}\left(1+2\eta\left\langle \boldsymbol{w}_t - \boldsymbol{w}_t^\eta, \boldsymbol{g}_t\right\rangle\right)^2 \leq 1$, whereas we get an upper bound of 2 from (5.12.1) and therefore obtain a final result that is a factor of 2 larger. The other remark is that we have described the fast version of FD sketching, which corresponds to Algorithm 6 of Luo et al. (2017) instead of the simpler slow version in their Algorithm 2. They and Deswarte consider the slow version in their analysis, but this makes no difference for the proof because the fast algorithm satisfies the same guarantees (Ghashami et al., 2016).

Analogously with Theorem 22, we find:

Theorem 24 (Sketching Grid Point Regret)**.** *Let $\eta \in \mathcal{G}$ be such that $\eta \leq \frac{1}{4B_T}$. Then* METAGRAD$^{\text{SKETCH}}$ *guarantees that the linearized regret w.r.t. any comparator $\boldsymbol{u} \in \bigcap_{t=1}^T \mathcal{W}_t$ is at most*

$$\tilde{R}_T^{\boldsymbol{u}} \leq \frac{\ln\det\left(\boldsymbol{I} + 2\eta^2\hat{D}^2\boldsymbol{S}_T^\intercal\boldsymbol{S}_T\right) + \frac{1}{2\hat{D}^2}\|\boldsymbol{u}\|_2^2 + 2\ln\lceil 2\log_2 Q_T\rceil_+ + \frac{1}{2}}{\eta} + \eta V_T^{\boldsymbol{u}} + \frac{2\eta\hat{D}^2 m\Omega_q}{m-q} + 4B_T,$$

for any $q = 0,\dots,m-1$.

As shown in Section 5.13, optimizing η leads to the following final result:

Theorem 25 (MetaGrad Sketching Regret Bound)**.** *For all $\boldsymbol{u} \in \bigcap_{t=1}^T \mathcal{W}_t$ the linearized regret of* METAGRAD$^{\text{SKETCH}}$ *is simultaneously bounded by*

$$\tilde{R}_T^{\boldsymbol{u}} \leq \frac{5}{2}\sqrt{(V_T^{\boldsymbol{u}} + \frac{2\hat{D}^2 m\Omega_q}{m-q})(\tfrac{1}{2\hat{D}^2}\|\boldsymbol{u}\|_2^2 + Z_T) + 10B_T(\tfrac{1}{2\hat{D}^2}\|\boldsymbol{u}\|_2^2 + Z_T)} + 4B_T,$$

where $Z_T = \operatorname{rk}(\boldsymbol{S}_T^\intercal\boldsymbol{S}_T)\ln\left(1 + \frac{\hat{D}^2\operatorname{Tr}(\boldsymbol{S}_T^\intercal\boldsymbol{S}_T)}{8B_T^2\operatorname{rk}(\boldsymbol{S}_T^\intercal\boldsymbol{S}_T)}\right) + 2\ln\lceil 2\log_2 T\rceil_+ + \frac{1}{2}$, and by

$$\tilde{R}_T^{\boldsymbol{u}} \leq \frac{5}{2}\sqrt{\left(V_T^{\boldsymbol{u}} + 2\hat{D}^2\sum_{t=1}^T\|\boldsymbol{g}_t\|_2^2 + \frac{2\hat{D}^2 m\Omega_q}{m-q}\right)\left(\tfrac{1}{2\hat{D}^2}\|\boldsymbol{u}\|_2^2 + Z_T'\right) + 10B_T\left(\tfrac{1}{2\hat{D}^2}\|\boldsymbol{u}\|_2^2 + Z_T'\right)} + 4B_T,$$

for any $q = 0,\dots,m-1$, where $Z_T' = 2\ln\lceil 2\log_2 T\rceil_+ + \frac{1}{2}$.

Compared to Theorem 23, all occurrences of $\boldsymbol{G}_T^\intercal \boldsymbol{G}_T$ have been replaced by their sketched approximations: $\boldsymbol{F}_T = \boldsymbol{G}_T^\intercal \boldsymbol{G}_T$ has become $\boldsymbol{S}_T^\intercal \boldsymbol{S}_T$ everywhere, with $\mathrm{rk}(\boldsymbol{S}_T^\intercal \boldsymbol{S}_T) \leq 2m$, and $\sum_{t=1}^T \|\boldsymbol{g}_t\|_2^2 = \mathrm{tr}(\boldsymbol{G}_T^\intercal \boldsymbol{G}_T)$ is now $\mathrm{tr}(\boldsymbol{S}_T^\intercal \boldsymbol{S}_T)$. We further see the additional term involving Ω_k, which corresponds to the directions not captured by the sketch.

5.7.2 MetaGrad Coordinate: Analysis

First, we define $b_{t,i} = |g_{t,i}| \max\{|a_{t,i}|, |z_{t,i}|\}$ and $B_{t,i} = \max_{s \leq t} b_{s,i}$. The analysis of the coordinate version of MetaGrad, which we denote by MetaGrad$^{\text{COOR}}$, is straightforward as we can simply apply the regret bound of MetaGrad$^{\text{FULL}}$ to each dimension. The formal statement can be found below.

Theorem 26. *Let $V_{T,i}^{u_i} = (u_i - w_{t,i})^2 g_{t,i}^2$. For any $\boldsymbol{u} \in \bigcap_{t=1}^T \mathcal{W}_t^{\text{rect}}$, the linearized regret of* MetaGrad$^{\text{COOR}}$ *is simultaneously bounded by*

$$\tilde{R}_T^{\boldsymbol{u}} \leq \sum_{i=1}^d \frac{5}{2}\sqrt{V_{T,i}^{u_i}(\tfrac{1}{2\hat{D}^2}u_i^2 + Z_{T,i})} + 10B_{T,i}(\tfrac{1}{2\hat{D}^2}u_i^2 + Z_{T,i}) + 4B_{T,i},$$

where $Z_{T,i} \;=\; \ln\left(1 + \frac{\hat{D}^2 \sum_{t=1}^T g_{t,i}^2}{8B_{T,i}^2}\right) + 2\ln\lceil 2\log_2 T\rceil + \frac{1}{2}$, *and by*

$$\begin{aligned}\tilde{R}_T^{\boldsymbol{u}} \;\leq\; & \sum_{i=1}^d \frac{5}{2}\sqrt{\left(V_{T,i}^{u_i} + 2\hat{D}^2\sum_{t=1}^T g_{t,i}^2\right)\left(\tfrac{1}{2\hat{D}^2}u_i^2 + Z'_T\right)} \\ & + 10B_{T,i}\left(\tfrac{1}{2\hat{D}^2}u_i^2 + Z'_T\right) + 4B_{T,i},\end{aligned}$$

where $Z'_T = 2\ln\lceil 2\log_2 T\rceil + \frac{1}{2}$.

5.8 Experiments

For an experimental evaluation we implemented six versions of MetaGrad, AdaGrad, Online Gradient Descent with learning rate $\eta_T = \frac{\hat{D}}{\sqrt{\sum_{s=1}^t \|\boldsymbol{g}_s\|_2^2 + 1e^{-8}}}$ (abbreviated as GDn), Online Gradient Descent with learning rate $\eta_t = \frac{\hat{D}}{G\sqrt{t}}$ (abbreviated as GDt) in python. The six versions of MetaGrad we used are the full version(abbreviated as MGFull), the coordinate version(abbreviated as MGCo), and two versions of MetaGrad that employ either Frequent Directions sketching with $m = 1$, $m = \min\{10, d\}$, $m = \min\{25, d\}$, and $m = \min\{50, d\}$ (abbreviated as MGF1, MGF10, MGF25, and MGF50 respectively). We compared the algorithms on seventeen datasets from the LIBSVM library (Chang and Lin, 2011), with T

ranging from 252 to 581012 and d ranging from 6 to 300. Of the seventeen datasets six had real outcomes and eleven had binary outcomes. A summary of the datasets can be found in Table 5.2 in Section 5.15. If available we used scaled versions of the datasets. For all datasets we used the features $\boldsymbol{x}_t$ without any adjustments in the following manner: we estimate $\hat{y} = \boldsymbol{w}_t^\intercal \boldsymbol{x}_t$ and feed this prediction to loss function $f(\hat{y}_t, y_t)$. For binary datasets we used logistic loss and hinge loss. For datasets with real outcomes we made use of squared loss and absolute loss. The settings of the algorithms can be found in Table 5.1 in Section 5.15. For AdaGrad and the coordinate version of MetaGrad we used $U = \{\boldsymbol{w} : \|\boldsymbol{w}\|_\infty \leq D\}$, where D was set to $3\|\boldsymbol{u}\|_\infty$, where $\boldsymbol{u} = \arg\min_{\boldsymbol{u}} \sum_{t=1}^T f(\boldsymbol{u}^\intercal \boldsymbol{x}_t, y_t)$. For the other algorithms we used $\mathcal{W} = \cap_{t=1}^T \mathcal{W}_t = \{\boldsymbol{w} : |\boldsymbol{x}_t^T \boldsymbol{w}| \leq C\}$ as domain, where $C = 3\max_t |\boldsymbol{u}^\intercal \boldsymbol{x}_t|$. Hyperparamaters for algorithms involving a norm of the minimizer were set to three times the norm of the comparator $\boldsymbol{u}$. In other words $\hat{D} = D$. Hyperparameters involving an upper bound on a norm of $\boldsymbol{g}_t$ were set as follows. For $t = 1$, $G_1 = \|\boldsymbol{g}_1\|_2$ and then we update. For any subsequent round, if $G_{t-1} \leq \|\boldsymbol{g}_t\|_2$ set $G_t = \|\boldsymbol{g}_t\|_2$, otherwise $G_t = G_{t-1}$.

5.8.1 Experimental Results

In Figure 5.1 we plotted the regret of three versions of MetaGrad and AdaGrad versus the regret of GDt on a logarithmic scale. We decided to use GDt as a baseline algorithm since it is the algorithm with the lowest regret that is not MetaGrad (in nine experiments either AdaGrad or GDn had lower regret than GDt). Table 5.3 in Section 5.15 contains the regrets of all algorithms on all datasets.

Out of 34 experiments in nine experiments a version of MetaGrad did not have the lowest regret. Among the six version of MetaGrad MGFull appears to be the best version as it had the lowest regret for the most datasets (thirteen). As predicted by theory, increasing the sketching size mostly improved the performance of Frequent Directions. With $m = \min\{50, d\}$, the Frequent Directions version of MetaGrad is very close to the performance of the full version of MetaGrad. Overall, the coordinate version of MetaGrad is close to the performance of the Full version of MetaGrad, which suggests that on the datasets that we used the correlations between the features are of little importance.

At first sight it may seem surprising that Online Gradient Descent outperformed MetaGrad on a9a, bodyfat, housing, ijcnn, and mg when the loss had curvature. However, upon closer inspection of the regret bounds we see that even in theory the regret bound of GDt is no worse than the regret of MetaGrad. For example, on the dataset bodyfat ($d = 14$, $T = 252$) with the squared loss the full version

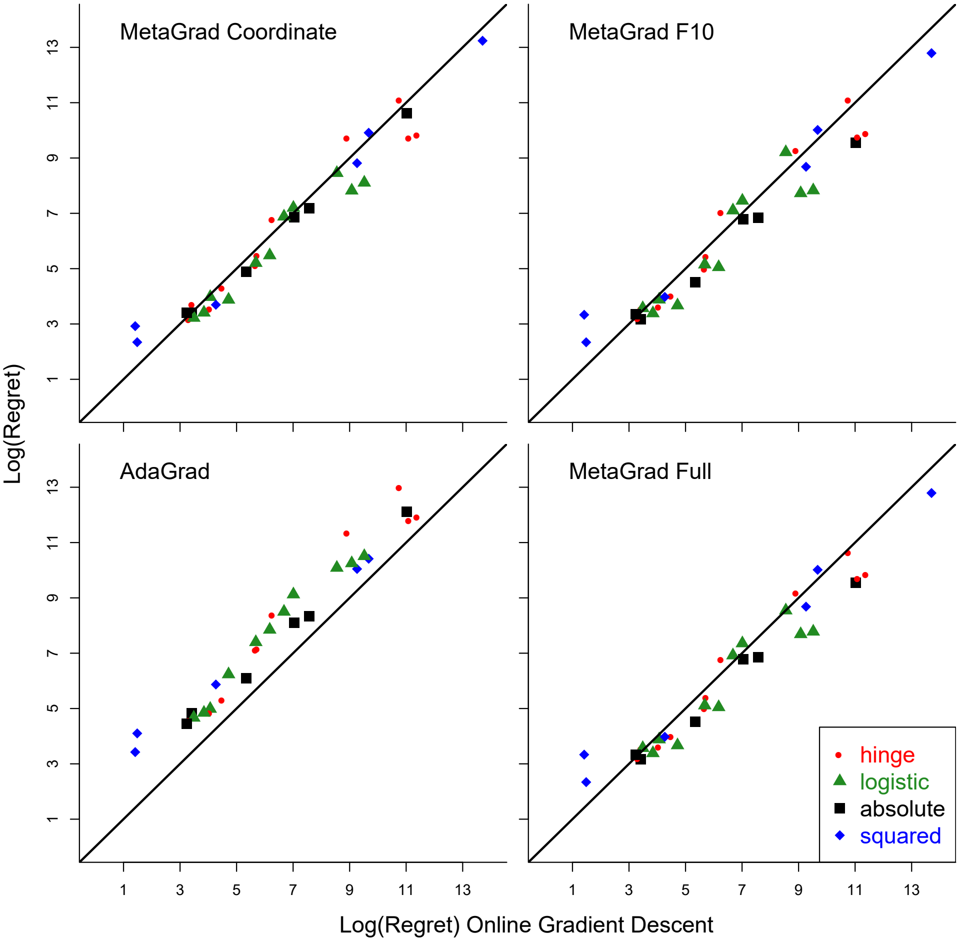


Figure 5.1: Comparison of the logarithm of the regret of three versions of MetaGrad and AdaGrad and the logarithm of the regret of GDt.

of MetaGrad has $O(\min\{d \log T, \sqrt{T}\}) = O(\sqrt{T})$ regret, whereas GDt also has $O(\sqrt{T})$ regret.

To our surprise, AdaGrad had the worst performance of all algorithms. However, upon closer review of the literature we see that in the experiments of Luo et al. (2017) and Chen et al. (2018) AdaGrad also had the worst performance, albeit in a different measure of performance (progressive misclassification rate and log objective gap, respectively).

Overall, we see that MetaGrad often outperforms AdaGrad and Online Gradient Descent with various learning rates.

5.9 Conclusion and Possible Extensions

We provide a new universally adaptive method, MetaGrad, which is robust to general convex losses but simultaneously can take advantage of special structure in the losses, like curvature or if the data come from a fixed distribution. The main new technique is to consider multiple learning rates in parallel: each learning rate η has its own surrogate loss (5.4.1) and there is a single controller method that aggregates the predictions of η-experts corresponding to the different surrogate losses.

An important feature of the controller is that its contribution to the final regret is the log of the number of experts, which is typically dominated by other terms in the bound. It is therefore cheap to add more experts for possibly different surrogate losses. To make the proof go through, a sufficient requirement on any such surrogates is that they replace the term $\big(\eta(\boldsymbol{u}-\boldsymbol{w}_t)^\intercal \boldsymbol{g}_t\big)^2$ in (5.4.1) by an upper bound. This possibility is exploited by Zhang et al. (2019), who add extra experts with surrogates that contain $\big(\eta\|\boldsymbol{g}_t\|_2\|\boldsymbol{u}-\boldsymbol{w}_t\|_2\big)^2$ instead. Since these surrogates are quadratic in all directions, and not just in the direction of $\boldsymbol{g}_t$, they are better suited for strongly convex losses, which then leads to an even more universally applicable extension of MetaGrad that also gets the optimal rate $O(\log T)$ for strongly convex losses.

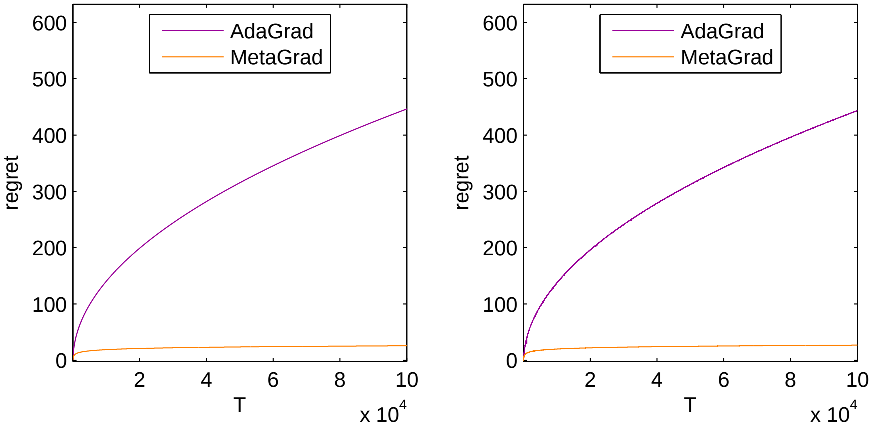
Another way to extend MetaGrad is to replace the exponential weights update in the controller by a different experts algorithm. Zhang et al. (2019) use this to extend MetaGrad for the case that the optimal parameters $\boldsymbol{u}$ vary over time, as measured in terms of the adaptive regret. See also Neuteboom (2020), who provides a similar extension of the closely related Squint algorithm for adaptive regret.

As a final possible extension, we mention the sliding window variant of Full Matrix AdaGrad (Agarwal et al., 2018). The same sliding window idea could be used to base the covariance matrix Σ_t^η in our Algorithm 11 only on the k most recent gradients. This has both computational advantages, because Σ_t^η then becomes a matrix of fixed rank $d+k$, and it could be beneficial for non-convex optimization when older covariance information needs to be discarded.

5.10 Extra Material Related to Section 5.3

In this section we gather extra material related to the fast rate examples from Section 5.3. We first provide simulations. Then we present the proofs of Theorems 20 and 21. And finally we give an example in which the unregularized hinge loss satisfies the Bernstein condition.

5.10.1 Simulations: Logarithmic Regret without Curvature



(a) Offline: $f_t(u) = |u - 1/4|$

(b) Stochastic Online: $f_t(u) = |u - x_t|$ where $x_t = \pm\frac{1}{2}$ i.i.d. with probabilities 0.4 and 0.6.

Figure 5.2: Examples of fast rates on functions without curvature. MetaGrad incurs logarithmic regret $O(\log T)$, while AdaGrad incurs $O(\sqrt{T})$ regret, matching its bound.

We provide two simple simulation examples to illustrate the sufficient conditions from Theorems 20 and 21, and to show that such fast rates are not automatically obtained by previous methods for general functions. Both our examples are one-dimensional, and have a stable optimum (that good algorithms will converge to); yet the functions are based on absolute values, which are neither strongly convex nor smooth, so the gradient norms do not vanish near the optimum. As our baseline we include AdaGrad (Duchi et al., 2011), because it is commonly used in practice (Mikolov et al., 2013; Schmidhuber, 2015) and because, in the one-dimensional case, it implements GD with an adaptive tuning of the learning rate that is applicable to general convex functions.

In the first example, we consider offline convex optimization of the fixed function $f_t(u) \equiv f(u) = |u - \frac{1}{4}|$, which satisfies (5.3.1), because it is convex. In the second example, we look at stochastic optimization with convex functions $f_t(u) = |u - x_t|$, where the outcomes $x_t = \pm\frac{1}{2}$ are chosen i.i.d. with probabilities 0.4 and 0.6. These probabilities satisfy (5.3.2) with $\beta = 1$. Their values are by no means essential, as long we avoid the worst case where the probabilities are equal.

Figure 5.2 graphs the results. We see that in both cases the regret of AdaGrad follows its $O(\sqrt{T})$ bound, while MetaGrad achieves an $O(\ln T)$ rate, as predicted by Theorems 20 and 21. This shows that MetaGrad achieves a type of adaptivity that is not achieved by AdaGrad. We should be careful in extending this conclusion to higher dimensions, though: whereas (the diagonal version of) AdaGrad uses a separate learning rate per dimension, MetaGrad$^{\text{FULL}}$ shares learning rates between dimensions (unless we run a MetaGrad$^{\text{COOR}}$ rather than MetaGrad$^{\text{FULL}}$).

5.10.2 Proof of Theorem 20

Proof. By (5.3.1), applied with $\boldsymbol{w} = \boldsymbol{w}_t$, and Theorem 19, there exists a $C > 0$ (depending on a) such that, for all sufficiently large T,

$$\begin{aligned} R_T^{\boldsymbol{u}} &\leq a\tilde{R}_T^{\boldsymbol{u}} - bV_T^{\boldsymbol{u}} \leq C\sqrt{V_T^{\boldsymbol{u}}\, d \ln T} + Cd\ln T - bV_T^{\boldsymbol{u}} \\ &\leq \frac{\gamma}{2} C V_T^{\boldsymbol{u}} + \left(\frac{1}{2\gamma} + 1\right) Cd\ln T - bV_T^{\boldsymbol{u}} \qquad \text{for all } \gamma > 0, \end{aligned}$$

where the last inequality is based on $\sqrt{xy} = \min_{\gamma>0} \frac{\gamma}{2}x + \frac{y}{2\gamma}$ for all $x, y > 0$. The result follows upon taking $\gamma = \frac{2b}{C}$. □

5.10.3 Proof of Theorem 21

Proof. Abbreviate $\tilde{r}_t^{\boldsymbol{u}} = (\boldsymbol{w}_t - \boldsymbol{u})^\intercal \boldsymbol{g}_t$. Then, by (5.1.1), Jensen's inequality and the Bernstein condition, there exists a constant $C > 0$ such that, for all sufficiently large T, the expected linearized regret is at most

$$\begin{aligned} \mathbb{E}\left[\tilde{R}_T^{\boldsymbol{u}^*}\right] &\leq C\,\mathbb{E}\left[\sqrt{V_T^{\boldsymbol{u}^*}\, d\ln T}\right] + Cd\ln T \leq C\sqrt{\mathbb{E}\left[V_T^{\boldsymbol{u}^*}\right]\, d\ln T} + Cd\ln T \\ &\leq C\sqrt{B\sum_{t=1}^T \left(\mathbb{E}\left[\tilde{r}_t^{\boldsymbol{u}^*}\right]\right)^\beta d\ln T} + Cd\ln T. \end{aligned}$$

We will repeatedly use the fact that

$$x^\alpha y^{1-\alpha} = c_\alpha \inf_{\gamma>0}\left(\frac{x}{\gamma} + \gamma^{\frac{\alpha}{1-\alpha}} y\right) \qquad \text{for any } x, y \geq 0 \text{ and } \alpha \in (0,1), \quad (5.10.1)$$

where $c_\alpha = (1-\alpha)^{1-\alpha}\alpha^\alpha$. Applying this first with $\alpha = 1/2$, $x = Bd\ln T$ and $y = \sum_{t=1}^T \left(\mathbb{E}[\tilde{r}_t^{\boldsymbol{u}^*}]\right)^\beta$, we obtain

$$\sqrt{B\sum_{t=1}^T \left(\mathbb{E}[\tilde{r}_t^{\boldsymbol{u}^*}]\right)^\beta d\ln T} \leq c_{1/2}\gamma_1 \sum_{t=1}^T \left(\mathbb{E}[\tilde{r}_t^{\boldsymbol{u}^*}]\right)^\beta + \frac{c_{1/2}}{\gamma_1} Bd\ln T \;\text{ for any } \gamma_1 > 0.$$

If $\beta = 1$, then $\sum_{t=1}^T \left(\mathbb{E}[\tilde{r}_t^{\boldsymbol{u}^*}]\right)^\beta = \mathbb{E}[\tilde{R}_T^{\boldsymbol{u}^*}]$ and the result follows by taking $\gamma_1 = \frac{1}{2Cc_{1/2}}$. Alternatively, if $\beta < 1$, then we apply (5.10.1) a second time, with $\alpha = \beta$,

$x = \mathbb{E}[\tilde{r}_t^{\boldsymbol{u}^*}]$ and $y = 1$, to find that, for any $\gamma_2 > 0$,

$$\begin{aligned}
&\sqrt{B \sum_{t=1}^{T} \left(\mathbb{E}[\tilde{r}_t^{\boldsymbol{u}^*}]\right)^{\beta} \, d \ln T} \\
&\leq c_\beta c_{1/2} \gamma_1 \sum_{t=1}^{T} \left(\frac{\mathbb{E}[\tilde{r}_t^{\boldsymbol{u}^*}]}{\gamma_2} + \gamma_2^{\beta/(1-\beta)} \right) + \frac{c_{1/2}}{\gamma_1} B d \ln T \\
&= \frac{c_\beta c_{1/2} \gamma_1}{\gamma_2} \mathbb{E}[\tilde{R}_T^{\boldsymbol{u}^*}] + c_\beta c_{1/2} \gamma_1 \gamma_2^{\beta/(1-\beta)} T + \frac{c_{1/2}}{\gamma_1} B d \ln T.
\end{aligned}$$

Taking $\gamma_1 = \frac{\gamma_2}{2 c_\beta c_{1/2} C}$, this yields

$$\mathbb{E}[\tilde{R}_T^{\boldsymbol{u}^*}] \leq \gamma_2^{1/(1-\beta)} T + \frac{4 C^2 c_{1/2}^2 c_\beta B d \ln T}{\gamma_2} + 2 C d \ln T.$$

We may optimize over γ_2 by a third application of (5.10.1), now with $x = 4C^2 c_{1/2}^2 c_\beta B d \ln T$, $y = T$ and $\alpha = 1/(2-\beta)$, such that $\alpha/(1-\alpha) = 1/(1-\beta)$:

$$\begin{aligned}
\mathbb{E}[\tilde{R}_T^{\boldsymbol{u}^*}] &\leq \frac{1}{c_{1/(2-\beta)}} \left(4 C^2 c_{1/2}^2 c_\beta B d \ln T \right)^{1/(2-\beta)} T^{(1-\beta)/(2-\beta)} + 2 C d \ln T \\
&= O\left((B d \ln T)^{1/(2-\beta)} T^{(1-\beta)/(2-\beta)} + d \ln T \right),
\end{aligned}$$

which completes the proof. □

5.10.4 Unregularized Hinge Loss Example

As shown by Koolen et al. (2016), the Bernstein condition is satisfied in the following classification task:

Lemma 18 (Unregularized Hinge Loss Example)**.** *Suppose that* $(\boldsymbol{X}_1, Y_1), (\boldsymbol{X}_2, Y_2), \ldots$ *are i.i.d. with* Y_t *taking values in* $\{-1, +1\}$*, and let* $f_t(\boldsymbol{u}) = \max\{0, 1 - Y_t \langle \boldsymbol{u}, \boldsymbol{X}_t \rangle\}$ *be the* hinge loss. *Assume that both* $\mathcal{W}$ *and the domain for* $\boldsymbol{X}_t$ *are the* d*-dimensional unit ball. Then the* (B, β)*-Bernstein condition is satisfied with* $\beta = 1$ *and* $B = \frac{2\lambda_{max}}{\|\boldsymbol{\mu}\|_2}$*, where* λ_{max} *is the maximum eigenvalue of* $\mathbb{E}[\boldsymbol{X}\boldsymbol{X}^\intercal]$ *and* $\boldsymbol{\mu} = \mathbb{E}[Y\boldsymbol{X}]$*, provided that* $\|\boldsymbol{\mu}\|_2 > 0$.

In particular, if $\boldsymbol{X}_t$ *is uniformly distributed on the sphere and* $Y_t = \operatorname{sign}(\langle \bar{\boldsymbol{u}}, \boldsymbol{X}_t \rangle)$ *is the noiseless classification of* $\boldsymbol{X}_t$ *according to the hyperplane with normal vector* $\bar{\boldsymbol{u}}$*, then* $B \leq \frac{c}{\sqrt{d}}$ *for some absolute constant* $c > 0$.

Thus the version of the Bernstein condition that implies an $O(d \log T)$ rate is always satisfied for the hinge loss on the unit ball, except when $\|\boldsymbol{\mu}\|_2 = 0$, which is very natural to exclude, because it implies that the expected hinge loss is 1 (its maximal value) for all $\boldsymbol{u}$, so there is nothing to learn. It is common to add ℓ_2-regularization to the hinge loss to make it strongly convex, but this example shows that that is not necessary to get logarithmic regret.

5.11 Controller Regret Bound (Proof of Lemma 15)

We prove Lemma 15 in two parts.

5.11.1 Decomposing the Surrogate Regret

Fix a comparator point $\boldsymbol{u} \in \bigcap_{t=1}^T \mathcal{W}_t$. We will first bound the surrogate regret

$$R_T^\eta(\boldsymbol{u}) \;:=\; \sum_{t=1}^T \left(\ell_t^\eta(\boldsymbol{w}_t) - \ell_t^\eta(\boldsymbol{u})\right)$$

for any $\eta \in \mathcal{G}$ not expired after T rounds (see Definition 2). Note that by definition (5.4.1), the surrogate loss $\ell_t^\eta(\boldsymbol{w}_t)$ of the controller is always zero, but we believe writing it helps interpretation. We will then use this surrogate regret bound to control the (non-surrogate) regret.

For the first half of this section, we fix a final time T, and a grid-point $\eta \in \mathcal{G}$ that is still not expired after time T, (see Definition 2)

Definition 3. *We define the wakeup time of learning rate $\eta \in \mathcal{G}$ by*

$$a^\eta \;:=\; \inf\left\{ t \le T \,\middle|\, \eta > \frac{1}{4\left(\sum_{s=1}^{t-1} b_s \frac{B_{s-1}}{B_s} + B_{t-1}\right)} \right\} \wedge (T+1).$$

Note that we manually set to $T+1$ the wakeup time of an η that does not wake up during the first T rounds. We do this so that $[1, a^\eta - 1]$ and $[a^\eta, T]$ always partition rounds $[1, T]$.

Our strategy will be to split the regret in three parts, which we will analyse separately.

Proposition 1. *We have*

$$\begin{aligned} R_T^\eta(\boldsymbol{u}) \;=\; & \underbrace{\sum_{t=1}^{a^\eta-1} \left(\ell_t^\eta(\boldsymbol{w}_t) - \ell_t^\eta(\boldsymbol{u})\right)}_{\ell^\eta\text{-regret of controller w.r.t. } \boldsymbol{u}} + \underbrace{\sum_{t=a^\eta}^{T} \left(\ell_t^\eta(\boldsymbol{w}_t) - \ell_t^\eta(\boldsymbol{w}_t^\eta)\right)}_{\ell^\eta\text{-regret of controller w.r.t. } \eta\text{-expert}} \\ & + \underbrace{\sum_{t=a^\eta}^{T} \left(\ell_t^\eta(\boldsymbol{w}_t^\eta) - \ell_t^\eta(\boldsymbol{u})\right)}_{\ell^\eta\text{-regret of } \eta\text{-expert w.r.t. } \boldsymbol{u}} \end{aligned}$$

Proof. The choice of a^η makes all $\boldsymbol{w}_t^\eta$ defined. We can hence merge the sums. □

We think of the three sums as follows. The first sum is "startup nuisance", and it will turn out to be small. The second sum is controlled by the controller, and it only depends on its construction. The third sum is controlled by the η-experts, and it only depends on their construction.

We will now proceed to bound the three parts above. First, we reduce to the clipped surrogate losses (5.4.4) at almost negligible cumulative cost using the clipping technique of Cutkosky (2019).

Lemma 19 (Clipping in the controller is cheap)**.**

$$\begin{aligned} & \underbrace{\sum_{t=1}^{a^\eta-1} \left(\ell_t^\eta(\boldsymbol{w}_t) - \ell_t^\eta(\boldsymbol{u})\right)}_{\ell^\eta\text{-regret of controller w.r.t. } \boldsymbol{u}} + \underbrace{\sum_{t=a^\eta}^{T} \left(\ell_t^\eta(\boldsymbol{w}_t) - \ell_t^\eta(\boldsymbol{w}_t^\eta)\right)}_{\ell^\eta\text{-regret of controller w.r.t. } \eta\text{-expert}} \\ & \leq \underbrace{\sum_{t=1}^{a^\eta-1} \left(\bar\ell_t^\eta(\boldsymbol{w}_t) - \bar\ell_t^\eta(\boldsymbol{u})\right)}_{\bar\ell^\eta\text{-regret of controller w.r.t. } \boldsymbol{u}} + \underbrace{\sum_{t=a^\eta}^{T} \left(\bar\ell_t^\eta(\boldsymbol{w}_t) - \bar\ell_t^\eta(\boldsymbol{w}_t^\eta)\right)}_{\bar\ell^\eta\text{-regret of controller w.r.t. } \eta\text{-expert}} + 2\eta B_T \end{aligned}$$

Proof. For any $\boldsymbol{u} \in \mathcal{W}_t$ (which includes the case $\boldsymbol{u} = \boldsymbol{w}_t^\eta$), we may use the definition of the range bound (5.2.1), the surrogate loss (5.4.1) and its clipped version (5.4.4) to find

$$\begin{aligned} & \left(\ell_t^\eta(\boldsymbol{w}_t) - \ell_t^\eta(\boldsymbol{u})\right) - \left(\bar\ell_t^\eta(\boldsymbol{w}_t) - \bar\ell_t^\eta(\boldsymbol{u})\right) \\ & = \eta \frac{B_t - B_{t-1}}{B_t} (\boldsymbol{w}_t - \boldsymbol{u})^\intercal \boldsymbol{g}_t - \underbrace{\eta^2 \frac{B_t^2 - B_{t-1}^2}{B_t^2} \left((\boldsymbol{u} - \boldsymbol{w}_t)^\intercal \boldsymbol{g}_t\right)^2}_{\geq 0} \\ & \leq 2\eta \frac{B_t - B_{t-1}}{B_t} b_t \;\leq\; 2\eta \left(B_t - B_{t-1}\right). \end{aligned}$$

Summing over rounds completes the proof. □

Next we deal with the clipped surrogate regret. We first handle the case of the early rounds before a^η. The key idea is that when η has not yet woken up, it is very small. Since the surrogate loss scales with η, it is small as well, even in sum.

Lemma 20. *For any η and any $\boldsymbol{u} \in \bigcap_{s=1}^{a^\eta-1} \mathcal{W}_s$*

$$\underbrace{\sum_{t=1}^{a^\eta-1} \left(\bar{\ell}_t^\eta(\boldsymbol{w}_t) - \bar{\ell}_t^\eta(\boldsymbol{u})\right)}_{\bar{\ell}^\eta\text{-regret of controller w.r.t. } \boldsymbol{u}} \leq \frac{1}{2}.$$

Proof. By definition of the clipped surrogate loss $\bar{\ell}_t^\eta$ in (5.4.4), the range bound b_t in (5.2.1) and the wakeup time a_t in Definition 3,

$$\begin{aligned}\sum_{t=1}^{a^\eta-1} \bar{\ell}_t^\eta(\boldsymbol{w}_t) - \bar{\ell}_t^\eta(\boldsymbol{u}) &\leq \sum_{t=1}^{a^\eta-1} \eta(\boldsymbol{w}_t - \boldsymbol{u})^\intercal \bar{\boldsymbol{g}}_t \\ &\leq \sum_{t:\eta \leq \frac{1}{4\left(\sum_{s=1}^{t-1} b_s \frac{B_{s-1}}{B_s} + B_{t-1}\right)}} \eta 2 b_t \frac{B_{t-1}}{B_t} \leq \frac{1}{2}.\end{aligned}$$

□

In the next subsection we deal with the middle sum in Proposition 1. This part only depends on the construction of the controller. We deal with the final sum in the section after that.

5.11.2 Controller surrogate regret bound

The controller is a specialists algorithm, which sometimes resets. We call the time segments between resets epochs. In every epoch, the controller guarantees a certain specialists regret bound w.r.t. any η-expert in its grid.

The η-expert that we need can be active during several epochs. Our strategy, following Mhammedi et al. (2019), will be the following. We incur the controller regret in the last and one-before-last epochs. We further separately prove, using the reset condition, that the total regret in all earlier epochs is small.

Theorem 27. *Consider an epoch starting at time $\tau + 1$ and fix any later time t in that same epoch. Fix any grid point $\eta \in \mathcal{G}$ not expired after t rounds (meaning*

$\eta \leq \frac{1}{2B_{t-1}}$*). Then the MetaGrad controller guarantees*

$$\underbrace{\sum_{s \in (\tau,t]: s \geq a^\eta} \left(\bar{\ell}_t^\eta(\boldsymbol{w}_t) - \bar{\ell}_t^\eta(\boldsymbol{w}_t^\eta)\right)}_{\text{specialist } \bar{\ell}^\eta\text{-regret of controller w.r.t. } \eta\text{-expert on } (\tau,t]} \leq \ln \left\lceil 2 \log_2 \left(\sum_{s=1}^{t-1} \frac{b_s}{B_s} + 1 \right) \right\rceil_+ .$$

Note that it is not important what the η-experts do at this point, the only feature that we use in the proof is that $\boldsymbol{w}_t^\eta \in \mathcal{W}_t$ for each active η. Also, note that the right-hand side is $O(\ln \ln T)$. We choose to stay with the current more detailed expression as it can be much smaller. This occurs whenever the actually observed loss ranges b_s are smaller than their respective upper bounds B_s.

Proof. We first observe that Algorithm 10, as far as it maintains the weights $p_t(\eta)$ between resets, implements Specialists Exponential Weights (called SBayes by Freund et al., 1997). In our particular case it is applied to specialists $\eta \in \mathcal{G}$, loss function $\eta \mapsto \ell_t^\eta(\boldsymbol{w}_t^\eta)$, active set $\mathcal{A}_t \subseteq \mathcal{G}$ and uniform (improper) prior on $\mathcal{G}$. The specialists regret bound (Freund et al., 1997, Theorem 1) directly yields[3]

$$\sum_{s \in (\tau,t]: s \geq a^\eta} - \ln \mathop{\mathbb{E}}_{p_t(\eta)} \left[e^{-\bar{\ell}_t^\eta(\boldsymbol{w}_t^\eta)} \right] \leq \ln \left| \bigcup_{s \in (\tau,t]} \mathcal{A}_s \right| + \sum_{s \in (\tau,t]: s \geq a^\eta} \bar{\ell}_t^\eta(\boldsymbol{w}_t^\eta).$$

Algorithm 10 further chooses the controller iterate

$$\boldsymbol{w}_t = \frac{\mathbb{E}_{p_t(\eta)} \left[\eta \boldsymbol{w}_t^\eta\right]}{\mathbb{E}_{p_t(\eta)} \left[\eta\right]}$$

which we claim ensures that

$$0 \leq - \ln \mathop{\mathbb{E}}_{p_t(\eta)} \left[e^{-\bar{\ell}_t^\eta(\boldsymbol{w}_t^\eta)} \right].$$

To see why, we use the definition (5.4.4) of clipped loss and gradient to obtain $(\boldsymbol{w}_t - \boldsymbol{w}_t^\eta)^\intercal \bar{\boldsymbol{g}}_t \geq -2B_{t-1}$, and we further use that p_t is supported on $\mathcal{A}_t$, which implies that $\eta \leq \frac{1}{4B_{t-1}}$. Together these license[4] the "prod bound" ($e^{x-x^2} \leq 1 + x$

[3] Our improper prior does not cause any trouble here, because renormalizing the prior, in hindsight, to the finite set of η-experts that were ever active preserves the algorithm's output and hence its regret bound.

[4] Here we motivate our controller algorithm using the loss function $\eta \mapsto \bar{\ell}_t^\eta(\boldsymbol{w}_t^\eta)$. One can alternatively base it on the loss function $\eta \mapsto -\ln(1 + \eta(\boldsymbol{w}_t - \boldsymbol{w}_t^\eta)^\intercal \bar{\boldsymbol{g}}_t)$ (These two versions are called Squint and iProd respectively by Koolen and Van Erven, 2015). As the second is always smaller (by the prod bound), using it would give a strictly tighter theorem here. We do not see a way to ultimately harvest this gain, as we would still need to invoke the prod bound at a later point in the analysis to express our regret bound in second-order form. We chose to present the "Squint-style" version here as we believe it is the more intuitive of the two.

for $x \geq -\frac{1}{2}$) yielding

$$-\ln \mathop{\mathbb{E}}_{p_t(\eta)}\left[e^{-\bar{\ell}_t^\eta(\boldsymbol{w}_t^\eta)}\right] \;\geq\; -\ln \mathop{\mathbb{E}}_{p_t(\eta)}\left[1+\eta(\boldsymbol{w}_t-\boldsymbol{w}_t^\eta)^\intercal \bar{\boldsymbol{g}}_t\right] \;=\; 0.$$

Inserting $\ell_t^\eta(\boldsymbol{w}_t)=0$, this implies

$$\sum_{s\in(\tau,t]:s\geq a^\eta}\left(\bar{\ell}_t^\eta(\boldsymbol{w}_t)-\bar{\ell}_t^\eta(\boldsymbol{w}_t^\eta)\right) \;\leq\; \ln\left|\bigcup_{s\in(\tau,t]}\mathcal{A}_s\right|.$$

It remains to bound the maximum number of awake grid-points during any epoch. Recall that the active set at any time t is

$$\mathcal{A}_t \;=\; \left(\frac{1}{4\left(\sum_{s=1}^{t-1} b_s\frac{B_{s-1}}{B_s}+B_{t-1}\right)}, \frac{1}{4B_{t-1}}\right]$$

Both endpoints are decreasing with t. Since our epoch starts at time $\tau+1$, the maximal η awake in the epoch is

$$\max\left\{\eta\in\mathcal{G}\;\middle|\;\eta\leq\frac{1}{4B_\tau}\right\}.$$

As the epoch lasts until at least time $t\geq\tau+1$, the smallest η active in the epoch is

$$\min\left\{\eta\in\mathcal{G}\;\middle|\;\eta\geq\frac{1}{4\left(\sum_{s=1}^{t-1} b_s\frac{B_{s-1}}{B_s}+B_{t-1}\right)}\right\}.$$

And since $\mathcal{G}$ is exponentially spaced with base 2, the maximum number of η that could possibly have been awake is

$$\begin{aligned}\left\lceil\log_2\frac{\left(\sum_{s=1}^{t-1} b_s\frac{B_{s-1}}{B_s}+B_{t-1}\right)}{B_\tau}\right\rceil &\leq \left\lceil\log_2\frac{B_{t-1}\left(\sum_{s=1}^{t-1}\frac{b_s}{B_s}+1\right)}{B_\tau}\right\rceil\\ &\overset{(5.4.3)}{\leq} \left\lceil\log_2\left(\left(\sum_{s=1}^{t-1}\frac{b_s}{B_s}\right)\left(\sum_{s=1}^{t-1}\frac{b_s}{B_s}+1\right)\right)\right\rceil\\ &\leq \left\lceil 2\log_2\left(\sum_{s=1}^{t-1}\frac{b_s}{B_s}+1\right)\right\rceil_+,\end{aligned}$$

so our prior costs for the improper (uniform on $\mathcal{G}$) prior are upper bounded by

$$\ln\left|\bigcup_{s\in(\tau,t]}\mathcal{A}_s\right| \;\leq\; \ln\left\lceil 2\log_2\left(\sum_{s=1}^{t-1}\frac{b_s}{B_s}+1\right)\right\rceil_+. \tag{5.11.1}$$

□

We now have a specialists regret bound that we can apply to each epoch.

Lemma 21 (Total regret in far past is small)**.** *Consider two consecutive epochs, starting after $\tau_1 < \tau_2$, and let η be not expired after τ_1 rounds. Then*

$$\sum_{s \in [1,\tau_1], s \geq a^\eta} \left(\bar{\ell}_s^\eta(\boldsymbol{w}_s) - \bar{\ell}_s^\eta(\boldsymbol{w}_s^\eta) \right) \;\leq\; 2\eta B_{\tau_2}$$

Proof.

$$\begin{aligned} - \sum_{s \in [1,\tau_1], s \geq a^\eta} \bar{\ell}_s^\eta(\boldsymbol{w}_s^\eta) \;\leq\; 2\eta \sum_{s=1}^{\tau_1} b_s \frac{B_{s-1}}{B_s} &\leq 2\eta B_{\tau_1} \sum_{s=1}^{\tau_1} \frac{b_s}{B_s} \\ &\leq 2\eta B_{\tau_1} \sum_{s=1}^{\tau_2} \frac{b_s}{B_s} \\ &\leq 2\eta B_{\tau_2}, \end{aligned}$$

where the last inequality is the reset condition (5.4.3) at time τ_2. □

We are now ready to compose the main theorem.

Theorem 28 (Overall controller specialists regret bound)**.** *Let η be not expired after T rounds. Then*

$$\sum_{t=a^\eta}^{T} \left(\bar{\ell}_t^\eta(\boldsymbol{w}_t) - \bar{\ell}_t^\eta(\boldsymbol{w}_t^\eta) \right) \;\leq\; 2\eta B_T + 2 \ln \left[2 \log_2 \left(\sum_{t=1}^{T-1} \frac{b_t}{B_t} + 1 \right) \right]. \quad (5.11.2)$$

Proof. We make a case distinction based on the number of epochs started by the algorithm. First, let us check the general case of ≥ 3 epochs (at least two normal epochs after the startup epoch). We apply the controller regret bound, Theorem 27, to the last two epochs each. Suppose these start after τ_1 and τ_2. For any $\eta \in \mathcal{G}$ not expired, we find

$$\begin{aligned} &- \sum_{t \in (\tau_1,\tau_2], t \geq a^\eta} \bar{\ell}_t^\eta(\boldsymbol{w}_t^\eta) - \sum_{t \in (\tau_2,T], t \geq a^\eta} \bar{\ell}_t^\eta(\boldsymbol{w}_t^\eta) \\ &\leq\; \ln \left[2 \log_2 \left(\sum_{s=1}^{\tau_2 - 1} \frac{b_s}{B_s} + 1 \right) \right] + \ln \left[2 \log_2 \left(\sum_{t=1}^{T-1} \frac{b_t}{B_t} + 1 \right) \right]. \end{aligned}$$

The regret on all epochs except the last two is bounded by Lemma 21. So together we obtain the theorem. Alternatively, suppose there are 2 epochs. Then, since we get no clipped regret in the 1st epoch, we apply the controller regret bound only in the second epoch to get

$$-\sum_{t\in[1,T],t\geq a^\eta} \bar{\ell}^\eta_t(\boldsymbol{w}^\eta_t) \leq \ln\left\lceil 2\log_2\left(\sum_{t=1}^{T-1}\frac{b_t}{B_t}+1\right)\right\rceil,$$

and (5.11.2) also holds. Finally, if there is only 1 epoch, then our clipped regret is 0 because nobody is awake, so (5.11.2) also holds. □

5.12 Proof of Lemma 16

Proof. The η-expert algorithm implements the exponentially weighted average forecaster with ℓ^η_t as the quadratic loss, unit learning rate, and with greedy projections (of the mean) onto $\mathcal{W}_t$. By (Hazan et al., 2007, Proof of Theorem 2), we obtain that

$$\sum_{t=a^\eta}^{T}\left(\ell^\eta_t(\boldsymbol{w}^\eta_t)-\ell^\eta_t(\boldsymbol{u})\right) \;\leq\; \frac{\|\boldsymbol{u}\|_2^2}{2\hat{D}^2}+\frac{1}{2}\sum_{t=a^\eta}^{T}\boldsymbol{g}'^{\intercal}_t\Sigma^\eta_{t+1}\boldsymbol{g}'_t$$

where $\boldsymbol{g}'_t = \eta\left(1+2\eta\left\langle\boldsymbol{w}_t-\boldsymbol{w}^\eta_t,\boldsymbol{g}_t\right\rangle\right)\boldsymbol{g}_t$ and where we recall that $(\Sigma^\eta_{t+1})^{-1} = \frac{1}{\hat{D}^2}\boldsymbol{I}+2\eta^2\sum_{s=a^\eta}^{t}\boldsymbol{g}_s\boldsymbol{g}_s^\intercal$. Expanding, we obtain

$$\boldsymbol{g}'^{\intercal}_t\Sigma^\eta_{t+1}\boldsymbol{g}'_t \;=\; \frac{1}{2}\left(1+2\eta\left\langle\boldsymbol{w}_t-\boldsymbol{w}^\eta_t,\boldsymbol{g}_t\right\rangle\right)^2\cdot 2\eta^2\boldsymbol{g}_t^\intercal\left(\frac{1}{\hat{D}^2}\boldsymbol{I}+2\eta^2\sum_{s=a^\eta}^{t}\boldsymbol{g}_s\boldsymbol{g}_s^\intercal\right)^{-1}\boldsymbol{g}_t$$

Now we may use that

$$\frac{1}{2}\left(1+2\eta\left\langle\boldsymbol{w}_t-\boldsymbol{w}^\eta_t,\boldsymbol{g}_t\right\rangle\right)^2 \leq \frac{1}{2}(1+4\eta b_t)^2 \leq \frac{1}{2}(1+1)^2 = 2 \qquad (5.12.1)$$

by the assumed upper bound on η. Moreover, by concavity of the log determinant, we have

$$\begin{aligned}&2\eta^2\boldsymbol{g}_t^\intercal\left(\frac{1}{\hat{D}^2}\boldsymbol{I}+2\eta^2\sum_{s=a^\eta}^{t}\boldsymbol{g}_s\boldsymbol{g}_s^\intercal\right)^{-1}\boldsymbol{g}_t\\ &\leq\; \ln\det\left(\frac{1}{\hat{D}^2}\boldsymbol{I}+2\eta^2\sum_{s=a^\eta}^{t}\boldsymbol{g}_s\boldsymbol{g}_s^\intercal\right)-\ln\det\left(\frac{1}{\hat{D}^2}\boldsymbol{I}+2\eta^2\sum_{s=a^\eta}^{t-1}\boldsymbol{g}_s\boldsymbol{g}_s^\intercal\right).\end{aligned}$$

Summing over rounds and telescoping, we find

$$\frac{1}{2}\sum_{t=a^\eta}^{T}\boldsymbol{g}'^{\intercal}_t\Sigma^\eta_{t+1}\boldsymbol{g}'_t \;\leq\; \ln\det\Big(\boldsymbol{I}+2\eta^2\hat{D}^2\sum_{t=a^\eta}^{T}\boldsymbol{g}_t\boldsymbol{g}_t^\intercal\Big)$$

and obtain the result. □

5.13 Composition Proofs of Theorems 23 and 25

We combine the proofs of Theorems 23 and 25, which are both special cases of the following more abstract result:

Theorem 29. *Suppose there exist a number $\omega \geq 0$ and a positive semi-definite matrix $\boldsymbol{F}'$ such that the linearized regret is at most*

$$\tilde{R}_T^{\boldsymbol{u}} \;\leq\; \eta\omega + \frac{\ln\det\left(\boldsymbol{I} + 2\eta^2\hat{D}^2\boldsymbol{F}'\right) + \frac{1}{2\hat{D}^2}\|\boldsymbol{u}\|_2^2 + 2\ln\lceil 2\log_2 Q_T\rceil_+ + \frac{1}{2}}{\eta} + 4B_T.$$

Then the linearized regret is both bounded by

$$\tilde{R}_T^{\boldsymbol{u}} \;\leq\; \frac{5}{2}\sqrt{\omega(\tfrac{1}{2\hat{D}^2}\|\boldsymbol{u}\|_2^2 + Z_T)} + 10B_T(\tfrac{1}{2\hat{D}^2}\|\boldsymbol{u}\|_2^2 + Z_T) + 4B_T,$$

where $Z_T = \operatorname{rk}(\boldsymbol{F}')\ln\left(1 + \frac{\hat{D}^2\sum_{t=1}^T\|\boldsymbol{g}_t\|_2^2}{8B_T^2\operatorname{rk}(\boldsymbol{F}')}\right) + 2\ln\lceil 2\log_2 T\rceil_+ + \frac{1}{2}$*, and by*

$$\tilde{R}_T^{\boldsymbol{u}} \;\leq\; \frac{5}{2}\sqrt{\left(\omega + 2\hat{D}^2\operatorname{tr}(\boldsymbol{F}')\right)\left(\tfrac{1}{2\hat{D}^2}\|\boldsymbol{u}\|_2^2 + Z'_T\right)} + 10B_T\left(\tfrac{1}{2\hat{D}^2}\|\boldsymbol{u}\|_2^2 + Z'_T\right) + 4B_T,$$

where $Z'_T = 2\ln\lceil 2\log_2 T\rceil_+ + \frac{1}{2}$.

Theorem 23 corresponds to the case $\omega = V_T^{\boldsymbol{u}}$ and $\boldsymbol{F}' = \boldsymbol{F}_T$, such that $\operatorname{Tr}(\boldsymbol{F}') = \sum_{t=1}^T\|\boldsymbol{g}_t\|_2^2$; Theorem 25 is obtained with $\omega = V_T^{\boldsymbol{u}} + \frac{2\hat{D}^2 m\Omega_q}{m-q}$ and $\boldsymbol{F}' = \boldsymbol{S}_T^\intercal\boldsymbol{S}_T$. The precondition of Theorem 29 is established by Theorems 22 and 24, respectively.

To prove Theorem 29 we start with a general lemma about optimizing in η:

Lemma 22. *For any $X, Y > 0$,*

$$\min_{\eta\in\mathcal{G}\,:\,\eta\leq\frac{1}{4B_T}} \eta X + \frac{Y}{\eta} \;\leq\; \frac{5}{2}\sqrt{XY} + 10B_T Y.$$

Proof. Let us denote the unconstrained optimizer of the left-hand side by $\hat{\eta} = \sqrt{Y/X}$. We distinguish two cases: first, when $\hat{\eta} \leq \frac{1}{4B_T}$, we upper bound the left-hand side by choosing the closest grid point $\eta \in \mathcal{G}$ below $\hat{\eta}$ (which, in the worst case, is at $\hat{\eta}/2$) to obtain

$$\min_{\eta\in\mathcal{G}\,:\,\eta\leq\frac{1}{4B_T}} \eta X + \frac{Y}{\eta} \;\leq\; \max_{\eta\in[\hat{\eta}/2,\hat{\eta}]} \eta X + \frac{Y}{\eta} = \frac{5}{2}\sqrt{XY}.$$

CHAPTER 5

In the second case, if $\hat{\eta} > \frac{1}{4B_T}$, we plug in the highest available grid point (for which the worst case is $\frac{1}{8B_T}$) to find

$$\min_{\eta \in \mathcal{G}\,:\,\eta \leq \frac{1}{4B_T}} \eta X + \frac{Y}{\eta} \;\leq\; \frac{1}{8B_T} X + 8B_T Y \;<\; 10 B_T Y,$$

where the second inequality follows by the assumption that $\hat{\eta} > \frac{1}{4B_T}$. In both cases the conclusion of the lemma follows. □

Proof. **(Theorem 29)** We start with the first claim of the theorem. By assumption, for any $\eta \leq \frac{1}{4B_T}$ in the grid $\mathcal{G}$, we have

$$\begin{aligned} \tilde{R}_T^{\boldsymbol{u}} \;&\leq\; \eta\omega + \frac{A}{\eta} + 4B_T \\ \text{where} \qquad A \;&=\; \ln\det\left(\boldsymbol{I} + \frac{1}{8B_T^2}\hat{D}^2 \boldsymbol{F}'\right) + \frac{1}{2\hat{D}^2}\|\boldsymbol{u}\|_2^2 + 2\ln\lceil 2\log_2 T\rceil + \frac{1}{2}. \end{aligned}$$

Lemma 22 therefore implies that

$$\tilde{R}_T^{\boldsymbol{u}} \;\leq\; \frac{5}{2}\sqrt{\omega A} + 10 B_T A + 4B_T.$$

The proof of the first claim follows by applying the inequality $\log\det(\boldsymbol{I} + \boldsymbol{M}) \leq \operatorname{rk}(\boldsymbol{M}) \log\left(1 + \frac{\operatorname{Tr}(\boldsymbol{M})}{\operatorname{rk}(\boldsymbol{M})}\right)$ (see Lemma 23 below) to the matrix $\boldsymbol{M} = \frac{1}{8B_T^2}\hat{D}^2\boldsymbol{F}'$.

For the second claim of the theorem, we again start from Theorem 22 and now bound $\ln\det(\boldsymbol{I} + \boldsymbol{M}) \leq \operatorname{Tr}(\boldsymbol{M})$ for $\boldsymbol{M} = 2\eta^2\hat{D}^2\boldsymbol{F}'$ (again see Lemma 23) to obtain

$$\tilde{R}_T^{\boldsymbol{u}} \;\leq\; \eta\omega + 2\eta\hat{D}^2 \operatorname{tr}(\boldsymbol{F}') + \frac{A'}{\eta} + 4B_T$$

where

$$A' \;=\; \frac{1}{2\hat{D}^2}\|\boldsymbol{u}\|_2^2 + 2\ln\lceil 2\log_2 T\rceil + \frac{1}{2}.$$

Using Lemma 22, the second claim follows, which completes the proof of the theorem. □

Lemma 23. *For any positive semi-definite matrix $\boldsymbol{M} \in \mathbb{R}^d$*

$$\log\det(\boldsymbol{I} + \boldsymbol{M}) \leq \operatorname{rk}(\boldsymbol{M}) \log\left(1 + \frac{\operatorname{Tr}(\boldsymbol{M})}{\operatorname{rk}(\boldsymbol{M})}\right)$$

and

$$\log\det(\boldsymbol{I} + \boldsymbol{M}) \leq \operatorname{Tr}(\boldsymbol{M}).$$

Proof. Let $\lambda_1, \dots, \lambda_d$ be the eigenvalues of $\boldsymbol{M}$. Then $(1+\lambda_1), \dots, (1+\lambda_d)$ are the eigenvalues of $\boldsymbol{I}+\boldsymbol{M}$, and Jensen's inequality implies

$$\begin{aligned}\log\det(\boldsymbol{I}+\boldsymbol{M}) &= \sum_{i=1}^{d} \log(1+\lambda_i) = \operatorname{rk}(\boldsymbol{M}) \sum_{i:\lambda_i\neq 0} \frac{1}{\operatorname{rk}(\boldsymbol{M})} \log(1+\lambda_i)\\ &\leq \operatorname{rk}(\boldsymbol{M}) \log\left(1 + \sum_{i:\lambda_i \neq 0} \frac{\lambda_i}{\operatorname{rk}(\boldsymbol{M})}\right) = \operatorname{rk}(\boldsymbol{M}) \log\left(1 + \frac{\operatorname{Tr}(\boldsymbol{M})}{\operatorname{rk}(\boldsymbol{M})}\right),\end{aligned}$$

which proves the first inequality. The second inequality follows because

$$\log\det(\boldsymbol{I}+\boldsymbol{M}) = \sum_{i=1}^{d} \log(1+\lambda_i) \leq \sum_{i=1}^{d} \lambda_i = \operatorname{Tr}(\boldsymbol{M}).$$

□

5.14 Bernstein for Linearized Excess Loss

Let $f : \mathcal{W} \to \mathbb{R}$ be a convex function drawn from distribution $\mathbb{P}$ with stochastic optimum $\boldsymbol{u}^* = \arg\min_{\boldsymbol{u}\in\mathcal{W}} \mathbb{E}_{f\sim\mathbb{P}}[f(\boldsymbol{u})]$. For any $\boldsymbol{w} \in \mathcal{W}$, we now show that the Bernstein condition for the excess loss $X := f(\boldsymbol{w}) - f(\boldsymbol{u}^*)$ implies the Bernstein condition with the same exponent β for the linearized excess loss $Y := (\boldsymbol{w} - \boldsymbol{u}^*)^\intercal \nabla f(\boldsymbol{w})$. These variables satisfy $Y \geq X$ by convexity of f and $Y \leq C := D'G'$.

Lemma 24. *For $\beta \in (0,1]$, let X be a (B,β)-Bernstein random variable:*

$$\mathbb{E}[X^2] \leq B\,\mathbb{E}[X]^\beta.$$

Then any bounded random variable $Y \leq C$ with $Y \geq X$ pointwise satisfies the (B',β)-Bernstein condition

$$\mathbb{E}[Y^2] \leq B'\,\mathbb{E}[Y]^\beta$$

for $B' = \max\left\{B, \frac{2}{\beta}C^{2-\beta}\right\}$.

Proof. For $\beta \in (0,1)$ we will use the fact that

$$z^\beta \;=\; c_\beta \inf_{\gamma>0}\left(\frac{z}{\gamma} + \gamma^{\frac{\beta}{1-\beta}}\right) \qquad \text{for any } z \geq 0,$$

with $c_\beta = (1-\beta)^{1-\beta}\beta^\beta$. For $\gamma = \left(\frac{1-\beta}{\beta}\,\mathbb{E}[Y]\right)^{1-\beta}$ we therefore have

$$\begin{aligned}
\mathbb{E}[X^2] - B'\,\mathbb{E}[X]^\beta &\geq \mathbb{E}[X^2] - B'c_\beta\left(\frac{\mathbb{E}[X]}{\gamma} + \gamma^{\frac{\beta}{1-\beta}}\right) \\
&\geq \mathbb{E}[Y^2] - B'c_\beta\left(\frac{\mathbb{E}[Y]}{\gamma} + \gamma^{\frac{\beta}{1-\beta}}\right) \qquad (5.14.1) \\
&= \mathbb{E}[Y^2] - B'\,\mathbb{E}[Y]^\beta, \qquad (5.14.2)
\end{aligned}$$

where the second inequality holds because $x^2 - c_\beta B' x/\gamma$ is a decreasing function of $x \leq C$ for $\gamma \leq \frac{c_\beta B'}{2C}$, which is satisfied by the choice of B'. This proves the lemma for $\beta \in (0,1)$. The claim for $\beta = 1$ follows by taking the limit $\beta \to 1$ in (5.14.2). □

5.15 Details of Experiments

Algorithm	$\hat{D}$	Domain	Domain Parameter	G
AdaGrad	$3\|\boldsymbol{u}\|_\infty$	$\mathcal{W}_t = \{\boldsymbol{w} : \|\boldsymbol{w}\|_\infty \leq D\}$	$D = 3\|\boldsymbol{u}\|_\infty$	·
GDn	$3\|\boldsymbol{u}\|_2$	$\mathcal{W}_t = \{\boldsymbol{w} : \|\langle \boldsymbol{w}, \boldsymbol{x}_t\rangle\| \leq C\}$	$C = 3\max_t \|\langle \boldsymbol{u}, \boldsymbol{x}_t\rangle\|$	·
GDt	$3\|\boldsymbol{u}\|_2$	$\mathcal{W}_t = \{\boldsymbol{w} : \|\langle \boldsymbol{w}, \boldsymbol{x}_t\rangle\| \leq C\}$	$C = 3\max_t \|\langle \boldsymbol{u}, \boldsymbol{x}_t\rangle\|$	$\max_{s\leq t}\|\boldsymbol{g}_s\|_2$
MGFull	$3\|\boldsymbol{u}\|_2$	$\mathcal{W}_t = \{\boldsymbol{w} : \|\langle \boldsymbol{w}, \boldsymbol{x}_t\rangle\| \leq C\}$	$C = 3\max_t \|\langle \boldsymbol{u}, \boldsymbol{x}_t\rangle\|$	·
MGDiag	$3\|\boldsymbol{u}\|_\infty$	$\mathcal{W}_t = \{\boldsymbol{w} : \|\boldsymbol{w}\|_\infty \leq D\}$	$D = 3\|\boldsymbol{u}\|_\infty$	·
MGF1	$3\|\boldsymbol{u}\|_2$	$\mathcal{W}_t = \{\boldsymbol{w} : \|\langle \boldsymbol{w}, \boldsymbol{x}_t\rangle\| \leq C\}$	$C = 3\max_t \|\langle \boldsymbol{u}, \boldsymbol{x}_t\rangle\|$	·
MGF10	$3\|\boldsymbol{u}\|_2$	$\mathcal{W}_t = \{\boldsymbol{w} : \|\langle \boldsymbol{w}, \boldsymbol{x}_t\rangle\| \leq C\}$	$C = 3\max_t \|\langle \boldsymbol{u}, \boldsymbol{x}_t\rangle\|$	·

Table 5.1: The settings of each algorithm.

Dataset	T	d	Outcome	P(y = 1)
a9a	32561	123	binary	0.24
abalone	4177	8	real	
australian	690	14	binary	0.44
bodyfat	252	14	real	
breast-cancer	683	9	binary	0.35
covtype	581012	54	binary	0.49
cpusmall	8192	12	real	
diabetes	768	8	binary	0.65
heart	270	13	binary	0.44
housing	506	13	real	
ijcnn1	91701	22	binary	0.10
ionosphere	351	34	binary	0.64
mg	1385	6	real	
space_ga	3107	6	real	
splice	1000	60	binary	0.52
w1atest	47272	300	binary	0.03
w8a	49479	300	binary	0.03

Table 5.2: Summary of the datasets used in the experiments.

Dataset	Loss	AdaGrad	GDnorm	GDt	MGCo	MGF1	MGF10	MGF25	MGF50	MGFull
a9a	hinge	85411	12712	**7619**	17012	14064	10821	10354	9858	9743
a9a	logistic	9185	2158	**1109**	1340	2306	1732	1668	1590	1568
abalone	absolute	4144	2529	1959	1317	2738	**938**	**938**	**938**	939
abalone	squared	23051	12484	10545	6725	9607	**5900**	**5900**	**5900**	5901
australian	hinge	124	42	**32**	41	46	35	35	35	35
australian	logistic	511	156	112	48	42	39	39	39	**39**
bodyfat	absolute	125	38	30	30	34	24	24	24	**24**
bodyfat	squared	60	5	**4**	10	10	10	10	10	10
breast-cancer	hinge	98	36	28	**24**	25	25	25	25	25
breast-cancer	logistic	107	41	33	**25**	36	36	36	36	36
covtype	hinge	445023	83124	48461	66797	121064	67126	59301	42043	**41985**
covtype	logistic	24043	11065	5157	**4713**	17698	10009	8662	5155	5147
cpusmall	absolute	183731	67098	61563	40537	89234	13974	**13818**	**13818**	13946
cpusmall	squared	2671536	806408	894232	561505	728070	**358831**	358832	358832	358833
diabetes	hinge	203	107	91	75	95	56	56	56	**55**
diabetes	logistic	147	80	58	53	54	**49**	**49**	**49**	49
heart	hinge	127	77	59	**35**	46	38	38	38	38
heart	logistic	127	71	47	30	30	30	30	30	**30**
housing	absolute	3301	1282	1147	946	1044	888	886	886	**886**
housing	squared	33324	**15560**	15909	20191	22244	22333	22336	22336	22336
ijcnn1	hinge	4413	1216	**537**	885	1522	1156	883	883	883
ijcnn1	logistic	4912	1404	**795**	976	1559	1219	1013	1013	1013
ionosphere	hinge	1245	431	299	169	166	**150**	150	150	150
ionosphere	logistic	2564	730	480	240	172	157	156	**156**	156
mg	absolute	85	33	**26**	30	45	29	29	29	28
mg	squared	31	10	**4**	19	28	28	28	28	28
space_ga	absolute	441	314	208	133	370	92	92	92	**91**
space_ga	squared	354	115	71	**40**	69	53	53	53	53
splice	hinge	1296	369	315	243	242	235	234	234	**225**
splice	logistic	1636	323	293	183	175	173	171	168	**166**
w1atest	hinge	134146	52780	67627	16910	17951	17436	16815	17143	**16636**
w1atest	logistic	28340	7500	8735	2498	2436	2271	2229	2207	**2182**
w8a	hinge	152227	60303	90302	**18789**	20764	19872	19783	19239	19229
w8a	logistic	36683	8370	13620	3324	2725	2519	2449	2421	**2392**

Table 5.3: The regret of each algorithm for the various datasets and loss functions. Boldface indicates smallest regret.

CHAPTER 6

Exploiting the Surrogate Gap in Online Multiclass Classification

This chapter is based on Van der Hoeven, D. (2020). Exploiting the surrogate gap in online multiclass classification. *To Appear in Advances in Neural Information Processing Systems 33*.

Abstract

We present GAPTRON, a randomized first-order algorithm for online multiclass classification. In the full information setting we show expected mistake bounds with respect to the logistic loss, hinge loss, and the smooth hinge loss with $O(K)$ expected surrogate regret, where the expectation is with respect to the learner's randomness and K is the number of classes. In the bandit classification setting we show that GAPTRON is the first linear time algorithm with $O(K\sqrt{T})$ expected surrogate regret. Additionally, the expected mistake bound of GAPTRON does not depend on the dimension of the feature vector, contrary to previous algorithms with $O(K\sqrt{T})$ surrogate regret in the bandit classification setting. We present a new proof technique that exploits the gap between the zero-one loss and surrogate losses rather than exploiting properties such as exp-concavity or mixability, which are traditionally used to prove logarithmic or constant regret bounds.

6.1 Introduction

In online multiclass classification a learner has to repeatedly predict the label that corresponds to a feature vector. Algorithms in this setting have a wide range of applications ranging from predicting the outcomes of sport matches to recommender systems. In some applications such as sport forecasting the learner obtains the true label regardless of what outcome the learner predicts, but in other applications such as recommender systems the learner only learns whether or not the label he predicted was the true label. The setting in which the learner receives the true label is called the full information multiclass classification setting and the setting in which the learner only receives information about the predicted label is called the bandit multiclass classification setting.

In this chapter we consider both the full information and bandit multiclass classification settings. In both settings the environment chooses the true outcome $y_t \in \{1, \ldots, K\}$ and feature vector $\boldsymbol{x}_t \in \mathbb{R}^d$. The environment then reveals the feature vector to the learner, after which the learner issues a (randomized) prediction $y'_t \in \{1, \ldots, K\}$. The goal of both settings is to minimize the number of expected mistakes the learner makes with respect to the best offline linear predictor $\boldsymbol{U} \in \mathbb{R}^{K \times d}$, where each row of $\boldsymbol{U}$ essentially keeps track of a linear predictor for each class. Standard practice in both settings is to upper bound the non-convex zero-one loss with a convex surrogate loss ℓ_t (see for example Bartlett et al. (2006)). This leads to guarantees of the form

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}[y'_t \neq y_t]\right] = \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(\boldsymbol{U})\right] + \tilde{\mathcal{R}}_T,$$

where $\mathbb{1}$ is the indicator function, y_t is the true label, the expectation is taken with respect to the learner's randomness, and $\tilde{\mathcal{R}}_T$ is the surrogate regret after T rounds.

We introduce GAPTRON, which is a randomized first-order algorithm that exploits the gap between the zero-one loss and the surrogate loss. In the full information multiclass classification setting GAPTRON has $O(K)$ surrogate regret with respect various surrogate losses. In the bandit multiclass classification setting we show that GAPTRON has $O(K\sqrt{T})$ surrogate regret with respect to the same surrogate losses as in the full information setting. Importantly, our surrogate regret bounds do not depend on the dimension of the feature vector in either the full or bandit information setting, contrary to previous results with comparable surrogate regret bounds. Furthermore, in the bandit multiclass classification setting GAPTRON is the first $O(dK)$ running time algorithm with $O(K\sqrt{T})$ surrogate regret.

To achieve these results we develop a new proof technique. Standard approaches that lead to small surrogate regret bounds exploit properties of the surrogate loss function such as strong convexity, exp-concavity (Hazan et al., 2007), or mixability (Vovk, 2001). Instead, inspired by the recent success of Neu and Zhivotovskiy (2020) in online classification with abstention[1], we exploit the *gap* between the zero-one loss, which is used to measure the performance of the learner, and the surrogate loss, which is used to measure the performance of the comparator $\boldsymbol{U}$, hence the name GAPTRON.

For an overview of our results and a comparison to previous work see Table 6.1. Here we briefly discuss the most relevant literature to place our results into perspective. A more detailed comparison can be found in the relevant sections. The full information multiclass classification setting is well understood and has been studied by many authors. Perhaps the most well known algorithm in this setting is the PERCEPTRON (Rosenblatt, 1958) and its multiclass versions (Crammer and Singer, 2003; Fink et al., 2006). The PERCEPTRON is a deterministic first-order algorithm which has $O(\sqrt{T})$ surrogate regret with respect to the hinge loss in the worst-case. Variants of the PERCEPTRON such as AROW (Crammer et al., 2009) and the second-order PERCEPTRON (Cesa-Bianchi et al., 2005) are second-order methods which result in a possibly smaller surrogate regret at the cost of longer running time. Online logistic regression (Berkson, 1944) is an alternative to the PERCEPTRON which has been thoroughly studied. For an overview of results for online logistic regression we refer the reader to Shamir (2020). We mention a recent result by Foster et al. (2018a), who use Exponential Weights (Vovk, 1990; Littlestone and Warmuth, 1994) to optimize the logistic loss and obtain a surrogate regret bound of order $O(dK\ln(DT+1))$, where D is an upper bound on the Frobenius norm of $\boldsymbol{U}$, with a polynomial time algorithm.

The bandit multiclass classification setting was first studied by Kakade et al. (2008) and is a special case of the contextual bandit setting (Langford and Zhang, 2008). Kakade et al. (2008) present a first-order algorithm called BANDITRON with a $O((DK)^{1/3}T^{2/3})$ surrogate regret bound with respect to the hinge loss. The impractical EXP4 algorithm (Auer et al., 2002) has a $O(\sqrt{TdK\ln(T+1)})$ surrogate regret bound and Abernethy and Rakhlin (2009) posed the problem of obtaining a practical algorithm which attains an $O(K\sqrt{T})$ surrogate regret bound. Several authors have proposed polynomial running time algorithms that have a surrogate regret bound of order $O(K\sqrt{dT\log(T+1)})$ such as NEWTRON (Hazan and Kale, 2011), SOBA (Beygelzimer et al., 2017), and OBAMA (Foster et al., 2018a).

[1] In fact, in Section 6.10 we slightly generalize the results of Neu and Zhivotovskiy (2020).

[2] These results hold for a family of loss functions parametrized by $\kappa \in [0, 1]$, which includes the

Table 6.1: Main results and comparisons with previous work (see Section 6.2 for notation). The references are for the surrogate regret bounds, not necessarily for the first analysis of the algorithm. For this table we assume that $\|\boldsymbol{x}_t\| \leq 1 \; \forall t$ and denote by $L_T = \sum_{t=1}^T \ell_t(\boldsymbol{U})$ the sum of the surrogate losses of the comparator.

Algorithm	Loss	surrogate regret full information setting	surrogate regret bandit setting	Time (per round)
PERCEPTRON(Fink et al., 2006; Kakade et al., 2008)	hinge	$O(\|\boldsymbol{U}\|^2 + \|\boldsymbol{U}\|\sqrt{L_T})$	$O((DK)^{1/3}T^{2/3})$	$O(dK)$
Second-Order PERCEPTRON (Orabona et al., 2012; Beygelzimer et al., 2017)	hinge^2	$O(\frac{\kappa}{2-\kappa}\|\boldsymbol{U}\|^2 + \frac{dK}{\kappa(2-\kappa)}\ln(L_T))$	$O(\|\boldsymbol{U}\|^2 + \frac{K}{\kappa}\sqrt{dT\ln(T)})$	$O((dK)^2)$
ONS (Hazan et al., 2014; Hazan and Kale, 2011)	logistic	$O(\exp(D)dK\ln(T))$	$O(dK^3DT^{2/3})$	$O((dK)^2)$
Vovk's Aggregating Algorithm (Foster et al., 2018a)	logistic	$O(dK\ln(DT))$	$O(K\sqrt{dT\ln(DT)})$	$O(\max\{dK, T\}^{12})$
GAPTRON (This work)	logistic, hinge, smooth hinge	$O(K\|\boldsymbol{U}\|^2)$	$O(KD\sqrt{T})$	$O(dK)$

6.2 Preliminaries

Notation. Let $\mathbf{1}$ and $\mathbf{0}$ denote vectors with only ones and zeros respectively and let $\boldsymbol{e}_k$ denote the basis vector in direction k. The inner product between vectors $\boldsymbol{g} \in \mathbb{R}^d$ and $\boldsymbol{w} \in \mathbb{R}^d$ is denoted by $\langle \boldsymbol{w}, \boldsymbol{g} \rangle$. The rows of matrix $\boldsymbol{W} \in \mathbb{R}^{K \times d}$ are denoted by $\boldsymbol{W}^1, \ldots, \boldsymbol{W}^K$. We will interchangeably use $\boldsymbol{W}$ to denote a matrix and a column vector in $\mathbb{R}^{Kd}$ to avoid unnecessary notation. The vector form of matrix $\boldsymbol{W}$ is $(\boldsymbol{W}^1, \ldots, \boldsymbol{W}^K)^\intercal$. The Frobenius norm of matrix $\boldsymbol{W}$ is denoted by $\|\boldsymbol{W}\| = \sqrt{\sum_{k=1}^K \sum_{i=1}^d W_{k,i}^2}$. Likewise the l_2 norm of vector $\boldsymbol{x}$ is denoted by $\|\boldsymbol{x}\| = \sqrt{\sum_{i=1}^d x_i^2}$. We denote the Kronecker product between matrices $\boldsymbol{W}$ and $\boldsymbol{U}$ by $\boldsymbol{W} \otimes \boldsymbol{U}$. For a given round t we use $\mathbb{E}_t[\cdot]$ to denote the conditional expectation given the predictions $y_1', y_2', \ldots, y_{t-1}'$.

6.2.1 Multiclass Classification

The multiclass classification setting proceeds in rounds $t = 1, \ldots, T$. In each round t the environment first picks an outcome $y_t \in \{1, \ldots K\}$ and feature vector $\boldsymbol{x}_t$ such that $\|\boldsymbol{x}_t\| \leq X$ for all t. Before the learner makes his prediction y_t' the environment reveals the feature vector $\boldsymbol{x}_t$ which the learner may use to form y_t'. In the full information multiclass classification setting, after the learner has issued y_t', the environment reveals the outcome y_t to the learner. In the bandit multiclass

hinge loss.

Algorithm 13 GAPTRON

```
Input: Learning rate $\eta > 0$, exploration rate $\gamma \in [0, 1]$, and gap map $a : \mathbb{R}^{K \times d} \times \mathbb{R}^d \to [0, 1]$
Initialize $\boldsymbol{W}_1 = \boldsymbol{0}$
for $t = 1 \dots T$ do
    Obtain $\boldsymbol{x}_t$
    Let $y_t^\star = \arg\max_k \langle \boldsymbol{W}_t^k, \boldsymbol{x}_t \rangle$
    Set $\boldsymbol{p}_t' = (1 - \max\{a(\boldsymbol{W}_t, \boldsymbol{x}_t), \gamma\})\boldsymbol{e}_{y_t^\star} + \max\{a(\boldsymbol{W}_t, \boldsymbol{x}_t), \gamma\}\frac{1}{K}\boldsymbol{1}$
    Predict with label $y_t' \sim \boldsymbol{p}_t'$
    Obtain $\mathbb{1}[y_t' \neq y_t]$ and set $\boldsymbol{g}_t = \nabla \ell_t(\boldsymbol{W}_t)$
    Update $\boldsymbol{W}_{t+1} = \arg\min_{\boldsymbol{W} \in \mathcal{W}} \eta \langle \boldsymbol{g}_t, \boldsymbol{W} \rangle + \frac{1}{2}\|\boldsymbol{W} - \boldsymbol{W}_t\|^2$
end for
```

classification setting (Kakade et al., 2008) the environment only reveals whether the prediction of the learner was correct or not, i.e. $\mathbb{1}[y_t' \neq y_t]$. We only consider the adversarial setting, which means that we make no assumptions on how y_t or $\boldsymbol{x}_t$ is generated. In both settings we allow the learner to use randomized predictions. The goal of the multiclass classification setting is to control the number of expected mistakes the learner makes in T rounds: $M_T = \mathbb{E}\left[\sum_{t=1}^T \mathbb{1}[y_t' \neq y_t]\right]$, where the expectation is taken with respect to the learner's randomness.

Since the zero-one loss is non-convex a standard approach is to use a surrogate loss ℓ_t as a function of a weight matrix $\boldsymbol{W}_t \in \mathcal{W}$, where $\mathcal{W} = \{\boldsymbol{W} : \|\boldsymbol{W}\| \leq D\}$. The surrogate loss function is a convex upper bound on the zero-one loss, which is then optimized using an Online Convex Optimization algorithm such as Online Gradient Descent (OGD) (Zinkevich, 2003), Online Newton Step (ONS) (Hazan et al., 2007), or Exponential Weights (EW) (Vovk, 1990; Littlestone and Warmuth, 1994). In this chapter we treat three surrogate loss functions: logistic loss, the hinge loss, and the smooth hinge loss, all of which result in different guarantees on the number of expected mistakes a learner makes.

6.3 GAPTRON

In this section we discuss GAPTRON (Algorithm 13). The prediction y_t' is sampled from

$$\boldsymbol{p}_t' = (1 - \max\{a(\boldsymbol{W}_t, \boldsymbol{x}_t), \gamma\})\boldsymbol{e}_{y_t^\star} + \max\{a(\boldsymbol{W}_t, \boldsymbol{x}_t), \gamma\}\frac{1}{K}\boldsymbol{1},$$

where $\gamma \in [0, 1]$, $a : \mathbb{R}^{K \times d} \times \mathbb{R}^d \to [0, 1]$, $y_t^\star = \arg\max_k \langle \boldsymbol{W}_t^k, \boldsymbol{x}_t \rangle$, and $\boldsymbol{e}_{y_t^\star}$ is the basis vector in direction $y_t^\star$. In the full information setting γ is set to 0 but in the bandit setting γ is used to guarantee that each label is sampled with at least probability $\frac{\gamma}{K}$, which is a common strategy in bandit algorithms (see for example Auer et al. (2002)). The fact that each label is sampled with at least probability $\frac{\gamma}{K}$ is important because in the bandit setting we use importance weighting to form estimated surrogate losses ℓ_t and their gradients $\boldsymbol{g}_t = \nabla \ell_t(\boldsymbol{W}_t)$ and we need to control the variance of these estimates. The main difference between GAPTRON and standard algorithms for multiclass classification is the a function, which governs the mixture that forms $\boldsymbol{p}_t'$. In fact, if we set $a(\boldsymbol{W}, \boldsymbol{x}) = 0$, $\gamma = 0$, and choose ℓ_t to be the hinge loss we recover an algorithm that closely resembles the PERCEPTRON (Rosenblatt, 1958), which can be interpreted as OGD on the hinge loss[3]. GAPTRON also uses OGD, which is used to update weight matrix $\boldsymbol{W}_t$, which in turn is used to form distribution $\boldsymbol{p}_t'$. For convenience we will define $a_t = a(\boldsymbol{W}_t, \boldsymbol{x}_t)$.

The role of a, which we will refer to as the gap map, is to exploit the gap between the surrogate loss and the zero-one loss. Before we explain how we exploit said gap we first present the expected mistake bound of GAPTRON in Lemma 25. The proof of Lemma 25 follows from applying the regret bound of OGD and working out the expected number of mistakes. The formal proof can be found in Section 6.7.

Lemma 25. *For any $\boldsymbol{U} \in \mathcal{W}$ Algorithm 13 satisfies*

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}[y_t' \neq y_t]\right] \\
&\leq \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(\boldsymbol{U})\right] + \frac{\|\boldsymbol{U}\|^2}{2\eta} + \gamma\frac{K-1}{K}T \\
&+ \sum_{t=1}^{T} \underbrace{\mathbb{E}\left[(1-a_t)\mathbb{1}[y_t^\star \neq y_t] + a_t\frac{K-1}{K} - \ell_t(\boldsymbol{W}_t) + \frac{\eta}{2}\|\boldsymbol{g}_t\|^2\right]}_{\text{surrogate gap}}.
\end{aligned}$$

As we mentioned before, standard classifiers such as the PERCEPTRON simply set $a(\boldsymbol{W}, \boldsymbol{x}) = 0$ and upper bound $\mathbb{1}[y_t^\star \neq y_t] - \ell_t(\boldsymbol{W}_t)$ by 0. In the full information setting we can set $\gamma = 0$ and $\eta = \sqrt{\frac{\|\boldsymbol{U}\|^2}{\sum_{t=1}^T \|\boldsymbol{g}_t\|^2}}$ to obtain[4] $M_T \leq \sum_{t=1}^T \ell_t(\boldsymbol{U}) + \|\boldsymbol{U}\|\sqrt{\sum_{t=1}^T \|\boldsymbol{g}_t\|^2}$. However, the gap between the surrogate loss and the zero-one

[3] Other interpretations exist which lead to possibly better guarantees, see for example Beygelzimer et al. (2017).

[4] Although such tuning is impossible due to not knowing $\|\boldsymbol{U}\|$ or $\sum_{t=1}^T \|\boldsymbol{g}_t\|^2$ there exist algorithms that are able to achieve the same guarantee up to logarithmic factors, see for example Cutkosky and Orabona (2018).

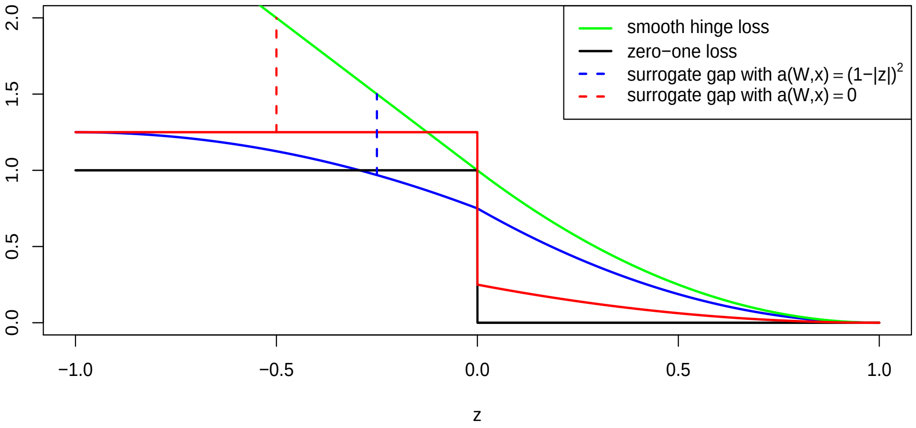


Figure 6.1: The surrogate gap for the smooth hinge loss as a function of margin z with $\eta = \frac{1}{8}$, $\gamma = 0$, and $\|\boldsymbol{x}\| = 1$. The solid red line is given by $\mathbb{1}[z \leq 0] + \frac{\eta}{2}\|\boldsymbol{g}\|^2$, where $\|\boldsymbol{g}\|^2 = 4(1-z)^2$ if $z > 0$ and $\|\boldsymbol{g}\|^2 = 4$ otherwise. The solid blue line is given by $(1-(1-|z|)^2)\mathbb{1}[z \leq 0] + \frac{1}{2}(1-|z|)^2 + \frac{\eta}{2}\|\boldsymbol{g}\|^2$. The surrogate gap is positive whenever the red or blue line is above the green line.

loss can be large. In fact, even with $a(\boldsymbol{W}, \boldsymbol{x}) = 0$, the gap between the zero-one loss and the surrogate loss is large enough to bound $\mathbb{1}[y_t^\star \neq y_t] - \ell_t(\boldsymbol{W}_t) + \frac{\eta}{2}\|\boldsymbol{g}_t\|^2$ by 0 for some loss functions and values of $\boldsymbol{W}_t$ and $\boldsymbol{x}_t$.

In Figure 6.1 we can see a depiction of the surrogate gap for the smooth hinge loss for $K = 2$ (Rennie and Srebro, 2005) in the full information setting (see Section 6.4.3 for the definition of the smooth multiclass hinge loss). In the case where $K = 2$, $\boldsymbol{W}$ is a vector rather than a matrix and outcomes y_t are coded as $\{-1, 1\}$. We see that with $a(\boldsymbol{W}, \boldsymbol{x}) = 0$, only when margin $z = y\langle \boldsymbol{W}, \boldsymbol{x}\rangle \in [-0.125, 0]$ the surrogate gap is not upper bounded by 0. Decreasing η would increase the range for which the surrogate gap is bounded by zero, but only for $\eta = 0$ the surrogate gap is bounded by 0 everywhere. However, with $a(\boldsymbol{W}, \boldsymbol{x}) = (1 - |z|)^2$ the surrogate gap is upper bounded by 0 for all z, which leads to constant surrogate regret. The remainder of the chapter is concerned with deriving different a for different loss functions for which the surrogate gap is bounded by 0. In the following section we start in the full information setting.

6.4 Full Information Multiclass Classification

In this section we derive gap maps that allow us to upper bound the surrogate gap by 0 for the logistic loss, hinge loss, and smooth hinge loss in the full information setting. Throughout this section we will set $\gamma = 0$. We start with the result for logistic loss.

6.4.1 Logistic Loss

The logistic loss is defined as

$$\ell_t(\boldsymbol{W}) = -\log_2(\sigma(\boldsymbol{W}, \boldsymbol{x}_t, y_t)), \tag{6.4.1}$$

where $\sigma(\boldsymbol{W}, \boldsymbol{x}, k) = \frac{\exp(\langle \boldsymbol{W}^k, \boldsymbol{x}\rangle)}{\sum_{\tilde{k}=1}^K \exp(\langle \boldsymbol{W}^{\tilde{k}}, \boldsymbol{x}\rangle)}$ is the softmax function. For the logistic loss we will use the following gap map:

$$a(\boldsymbol{W}_t, \boldsymbol{x}_t) = 1 - \mathbb{1}[p_t^\star \geq 0.5]p_t^\star,$$

where $p_t^\star = \max_k \sigma(\boldsymbol{W}_t, \boldsymbol{x}_t, k)$. This means that GAPTRON samples a label uniformly at random as long as $p_t^\star \leq 0.5$. While this may appear counter-intuitive at first sight note that when $p_t^\star < 0.5$ the zero-one loss is upper bounded by the logistic loss regardless of what we play since $-\log_2(p) \geq 1$ for $p \in [0, 0.5]$, which we use to show that the surrogate gap is bounded by 0 whenever $p_t^\star < 0.5$. The mistake bound of GAPTRON can be found in Theorem 30. To prove Theorem 30 we show that the surrogate gap is bounded by 0 and then use Lemma 25. The formal proof can be found in Section 6.8.1.

Theorem 30. *Let* $a(\boldsymbol{W}_t, \boldsymbol{x}_t) = 1 - \mathbb{1}[p_t^\star \geq 0.5]p_t^\star$, $\eta = \frac{\ln(2)}{2KX^2}$, $\gamma = 0$, *and let* ℓ_t *be the logistic loss defined in* (6.4.1). *Then for any* $\boldsymbol{U} \in \mathcal{W}$ *Algorithm 13 satisfies*

$$\mathbb{E}\left[\sum_{t=1}^T \mathbb{1}[y_t' \neq y_t]\right] \leq \sum_{t=1}^T \ell_t(\boldsymbol{U}) + \frac{KX^2\|\boldsymbol{U}\|^2}{\ln(2)}.$$

Let us compare the mistake bound of GAPTRON with other results for logistic loss. Foster et al. (2018a) circumvent a lower bound for online logistic regression by Hazan et al. (2014) by using an improper learning algorithm and achieve $O(dK\ln(DT+1))$ surrogate regret. Unfortunately this algorithm is impractical since the running time can be of order $O(D^6 \max\{dK, T\}^{12}T)$. In the case where $K = 2$ Jézéquel et al. (2020) provide a faster improper learning algorithm called AIOLI based on the Vovk-Azoury-Warmuth forecaster (Vovk, 2001; Azoury and Warmuth, 2001) that has running time $O(d^2T)$ and a surrogate regret of order $O(dD\ln(T))$. Unfortunately it is not known if AIOLI can be extended to $K > 2$. An alternative algorithm is ONS, which has running time $O((dK)^2T)$ but a surrogate regret bound of order $O(\exp(D)dK\ln(T+1))$. With standard OGD we could degrade the dependence on T to improve the dependence on D to find a surrogate regret of order $O(D\sqrt{T})$ with an algorithm that has running time $O(dKT)$. Depending on $\|\boldsymbol{U}\|^2$ the surrogate regret of GAPTRON can be significantly smaller than the surrogate regret of the aforementioned algorithms as the surrogate regret of GAPTRON is independent of T and d. Furthermore, since GAPTRON uses OGD to update $\boldsymbol{W}_t$ the running time is $O(dKT)$, significantly improving upon the running time of previous algorithms with comparable mistake bounds.

6.4.2 Multiclass Hinge Loss

We use a variant of the multiclass hinge loss of Crammer and Singer (2001), which is defined as:

$$\ell_t(\boldsymbol{W}) = \begin{cases} \max\{1 - m_t(\boldsymbol{W}, y_t), 0\} & \text{if } m_t^\star \le \beta \\ \max\{1 - m_t(\boldsymbol{W}, y_t), 0\} & \text{if } y_t^\star \neq y_t \text{ and } m_t^\star > \beta \\ 0 & \text{if } y_t^\star = y_t \text{ and } m_t^\star > \beta, \end{cases} \tag{6.4.2}$$

where $m_t(\boldsymbol{W}_t, y) = \langle \boldsymbol{W}_t^y, \boldsymbol{x}_t \rangle - \max_{k \neq y} \langle \boldsymbol{W}_t^k, \boldsymbol{x}_t \rangle$ and $m_t^\star = \max_k m_t(\boldsymbol{W}_t, k)$. Note that we set $\ell_t(\boldsymbol{W}) = 0$ when $y_t^\star = y_t$ and $m_t^\star > \beta$. In common implementations of the PERCEPTRON $\ell_t(\boldsymbol{W}) = 0$ whenever $y_t^\star = y_t$ (see for example Kakade et al. (2008)). However, for the surrogate gap to be bounded by zero we need ℓ_t to be positive whenever $a_t > 0$ otherwise there is nothing to cancel out the $a_t \frac{K-1}{K}$ term. The gap map for the hinge loss is $a(\boldsymbol{W}_t, \boldsymbol{x}_t) = 1 - \max\{\mathbb{1}[m_t^\star > \beta], m_t^\star\}$. This means that whenever $m_t^\star > \beta$ the predictions of GAPTRON are identical to the predictions of the PERCEPTRON. The mistake bound of GAPTRON for the hinge loss can be found in Theorem 31 (its proof is deferred to Section 6.8.2).

Theorem 31. *Set* $a(\boldsymbol{W}_t, \boldsymbol{x}_t) = 1 - \max\{\mathbb{1}[m_t^\star > \beta], m_t^\star\}$, $\eta = \frac{1-\beta}{KX^2}$, $\gamma = 0$, *and let* ℓ_t *be the multiclass hinge loss defined in* (6.4.2) *with* $\beta = \frac{1}{K}$. *Then for any* $\boldsymbol{U} \in \mathcal{W}$ *Algorithm 13 satisfies*

$$\mathbb{E}\left[\sum_{t=1}^T \mathbb{1}[y_t' \neq y_t]\right] \le \sum_{t=1}^T \ell_t(\boldsymbol{U}) + \frac{K^2 X^2 \|\boldsymbol{U}\|^2}{2(K-1)}.$$

Let us compare the mistake bound of GAPTRON with the mistake bound of the PERCEPTRON. The PERCEPTRON guarantees $M_T \le \sum_{t=1}^T \ell_t(\boldsymbol{U}) + X^2\|\boldsymbol{U}\|^2 + 2X\|\boldsymbol{U}\|\sqrt{2\sum_{t=1}^T \ell_t(\boldsymbol{U})}$ (see Beygelzimer et al. (2017) for a proof). The factor K in the surrogate regret of GAPTRON is due to the cost of exploring uniformly at random. For small K the mistake bound of GAPTRON can be significantly smaller in the adversarial case, but for large K the cost of sampling uniformly at random can be too high and the mistake bound of GAPTRON can be larger than that of the PERCEPTRON. In the separable case the PERCEPTRON has a strictly better guarantee for any K since then only the $X^2\|\boldsymbol{U}\|^2$ term remains.

Orabona et al. (2012) show that for all loss functions of the form $\ell_t(\boldsymbol{W}) = \max\{1-\frac{2}{2-\kappa}m_t(\boldsymbol{W}, y_t) + \frac{\kappa}{2-\kappa}m_t(\boldsymbol{W}, y_t)^2, 0\}$ the second-order PERCEPTRON guarantees $M_T \leq \sum_{t=1}^T \ell_t(\boldsymbol{U}) + O(\frac{\kappa}{2-\kappa}X^2\|\boldsymbol{U}\|^2 + \frac{dK}{\kappa(2-\kappa)}\ln(\sum_{t=1}^T \ell_t(\boldsymbol{U}) + 1)$. Thus, for small K GAPTRON always has a smaller surrogate regret term but for larger K the guarantee of GAPTRON can be worse, although this also depends on the performance and norm of the comparator $\boldsymbol{U}$.

6.4.3 Smooth Multiclass Hinge Loss

The smooth multiclass hinge loss (Rennie and Srebro, 2005) is defined as

$$\ell_t(\boldsymbol{W}) = \begin{cases} \max\{1 - 2m_t(\boldsymbol{W}, y_t), 0\} & \text{if } m_t(\boldsymbol{W}, y_t) \leq 0 \\ \max\{(1 - m_t(\boldsymbol{W}, y_t))^2, 0\} & \text{if } m_t(\boldsymbol{W}, y_t) > 0, \end{cases} \tag{6.4.3}$$

where $m_t(\boldsymbol{W}_t, y) = \langle \boldsymbol{W}_t^y, \boldsymbol{x}_t\rangle - \max_{k\neq y}\langle \boldsymbol{W}_t^k, \boldsymbol{x}_t\rangle$ as in Section 6.4.3. This loss function is not exp-concave nor is it strongly-convex. This means that with standard methods from Online Convex Optimization we cannot hope to achieve a better surrogate regret bound than $O(D\sqrt{T})$ in the worst-case. Theorem 32 shows that with gap map $a(\boldsymbol{W}_t, \boldsymbol{x}_t) = (1 - \min\{1, m_t^\star\})^2$, where $m_t^\star = \max_k m_t(\boldsymbol{W}_t, k)$, GAPTRON has a $O(K)$ surrogate regret bound. The proof of Theorem 32 follows from bounding the surrogate gap by zero and can be found in Section 6.8.3.

Theorem 32. *Set* $a(\boldsymbol{W}_t, \boldsymbol{x}_t) = (1 - \min\{1, m_t^\star\})^2$, $\eta = \frac{1}{4KX^2}$, $\gamma = 0$, *and let* ℓ_t *be the smooth multiclass hinge loss defined in* (6.4.3). *Then for any* $\boldsymbol{U} \in \mathcal{W}$ *Algorithm 13 satisfies*

$$\mathbb{E}\left[\sum_{t=1}^T \mathbb{1}[y_t' \neq y_t]\right] \leq \sum_{t=1}^T \ell_t(\boldsymbol{U}) + 2KX^2\|\boldsymbol{U}\|^2.$$

6.5 Bandit Multiclass Classification

In this section we will analyse GAPTRON in the bandit multiclass classification setting. While in the full information setting the fact that GAPTRON is a randomized algorithm can be seen as a drawback, in the adversarial bandit setting it is actually a requirement (see for example chapter 11 by Lattimore and Szepesvári (2018)). We will use the same gap maps as in the full information setting. The only difference is how we feed the surrogate loss to GAPTRON. We will use the same loss functions as in the full information setting but now multiplied by $\mathbb{1}[y'_t = y_t]p'_t(y'_t)^{-1}$, which is simply importance weighting. This also means that, compared to the full information setting, the gradients that OGD uses to update weight matrix $\boldsymbol{W}_t$ are multiplied by $\mathbb{1}[y'_t = y_t]p'_t(y'_t)^{-1}$. To control the surrogate gap we set $\gamma > 0$, which allows us to bound the variance of the norm of the gradients. The proofs in this section follow the same structure as in the full information setting, with the notable change that we suffer increased surrogate regret due to the $\gamma\frac{K-1}{K}T$ bias term and the increased $\mathbb{E}[\|\boldsymbol{g}_t\|^2] = O(\frac{K}{\gamma})$ term.

The results in this section provide three new answers to the open problem by Abernethy and Rakhlin (2009), who posed the problem of obtaining an efficient algorithm with $O(K\sqrt{T})$ surrogate regret. Several solutions with various loss function have been proposed. Beygelzimer et al. (2017) solved the open problem using an algorithm called SOBA. SOBA is a second-order algorithm which is analysed using a family of surrogate loss functions introduced by Orabona et al. (2012) ranging from the standard multiclass hinge loss to the squared multiclass hinge loss. The loss functions are parameterized by κ, where $\kappa = 0$ corresponds to the multiclass hinge loss and $\kappa = 1$ corresponds to the squared hinge loss. Simultaneously for all surrogate loss functions in the family of loss functions SOBA suffers a surrogate regret of order $O(\|\boldsymbol{U}\|^2X^2 + \frac{K}{\kappa}\sqrt{dT\ln(T+1)})$ and has a running time of order $O((dK)^2T)$. Hazan and Kale (2011) consider the logistic loss and obtain surrogate regret of order $O(dK^3 \min\{\exp(DX)\ln(T+1), DXT^{\frac{2}{3}}\})$. Hazan and Kale (2011) also obtain $DX\sqrt{T}$ surrogate regret for a variant of the logistic loss function we consider in this chapter. Both results of Hazan and Kale (2011) are obtained by running ONS on (a variant of) the logistic loss, which has running time $O((dK)^2T)$. Foster et al. (2018a) introduce OBAMA, which improves the results of Hazan and Kale (2011) and suffers $O(\min\{dK^2\ln(TDX+1), K\sqrt{dT\ln(TDX+1)}\})$ surrogate regret for the logistic loss. Unfortunately, OBAMA has running time $O(D^6\max\{dK, T\}^{12}T)$.

GAPTRON is the first $O(dKT)$ running time algorithm which has $O(DK\sqrt{T})$ surrogate regret in bandit multiclass classification with respect to the logistic, hinge, or smooth hinge loss. GAPTRON also improves the surrogate regret bounds of previous algorithms with $O(DK\sqrt{T})$ surrogate regret by a factor $O(\sqrt{d\log(T+1)})$. The remainder of this section provides the settings for GAPTRON to achieve these results, starting with the logistic loss.

6.5.1 Bandit Logistic Loss

The bandit version of the logistic loss is defined as:

$$\ell_t(\boldsymbol{W}) = -\mathbb{1}[y'_t = y_t]p'_t(y'_t)^{-1}\log_2(\sigma(\boldsymbol{W}, \boldsymbol{x}_t, y_t)). \tag{6.5.1}$$

A similar definition of the bandit logistic loss is used by Hazan and Kale (2011); Foster et al. (2018a). It is straightforward to verify that $\mathbb{E}_t[\ell_t(\boldsymbol{w})]$ is equivalent to its full information counterpart (6.4.1). This loss is a factor $\frac{1}{\ln(2)}$ larger than the loss used by Hazan and Kale (2011); Foster et al. (2018a), who use the natural logarithm instead of the logarithm with base 2. To stay consistent with the full information setting we opt to use base 2 in the bandit setting. Using GAPTRON with the natural logarithm will give similar results.

The mistake bound of GAPTRON for this loss can be found in Theorem 33 (its proof can be found in Section 6.9.1). Compared to OBAMA, which achieves a surrogate regret bound of order $O(\min\{dK^2\ln(TDX+1), K\sqrt{dT\ln(TDX+1)}\})$, GAPTRON has a larger dependency on D and X. However, the mistake bound of GAPTRON does not depend on d, which can be a significant improvement over the surrogate regret bound of OBAMA. Theorem 33 answers the two questions by Hazan and Kale (2011) affirmatively; GAPTRON is a linear time algorithm with exponentialy improved constants in the surrogate regret bound compared to NEWTRON.

Theorem 33. *Let* $a(\boldsymbol{W}_t, \boldsymbol{x}_t) = 1 - \mathbb{1}[p^\star_t \geq 0.5]p^\star_t$, $\eta = \frac{\ln(2)((1-\gamma)\exp(-2DX)\frac{1}{K}+\gamma)}{2K^2X^2}$, *and let* ℓ_t *be the bandit logistic loss* (6.5.1). *Then there exists a setting of* γ *such that Algorithm 13 satisfies*

$$\mathbb{E}\left[\sum_{t=1}^T \mathbb{1}[y'_t \neq y_t]\right]$$
$$\leq \mathbb{E}\left[\sum_{t=1}^T \ell_t(\boldsymbol{U})\right] + KXD\min\left\{\max\left\{\frac{2KXD}{\ln(2)}, 2\sqrt{\frac{T}{\ln(2)}}\right\}, \frac{KXD}{e^{-2DX}\ln(2)}\right\}.$$

6.5.2 Bandit Multiclass Hinge Loss

We use the following definition of the bandit multiclass hinge loss:

$$\ell_t(\boldsymbol{W}_t) =$$
$$\begin{cases} \mathbb{1}[y'_t = y_t]p'_t(y'_t)^{-1}\max\{1 - m_t(\boldsymbol{W}_t, y_t), 0\} & \text{if } m^\star_t \leq \beta \\ \mathbb{1}[y'_t = y_t]p'_t(y'_t)^{-1}\max\{1 - m_t(\boldsymbol{W}_t, y_t), 0\} & \text{if } y^\star_t \neq y_t \text{ and } m^\star_t > \beta \\ 0 & \text{if } y'_t = y^\star_t = y_t \text{ and } m^\star_t > \beta. \end{cases} \tag{6.5.2}$$

It is straightforward to see that the conditional expectation of the bandit multiclass hinge loss is the full information multiclass hinge loss. Both the BANDITRON algorithm (Kakade et al., 2008) and SOBA (Beygelzimer et al., 2017) use a similar loss function.

As we mentioned before, Beygelzimer et al. (2017) present SOBA, which is a second-order algorithm with surrogate regret $O(\|\boldsymbol{U}\|^2X^2 + \frac{K}{\kappa}\sqrt{dT\ln(T+1)})$. BANDITRON is a first-order algorithm based on the PERCEPTRON algorithm and suffers $O((KDX)^{1/3}T^{2/3})$ surrogate regret. For the more general setting of contextual bandits (Foster and Krishnamurthy, 2018) use continuous Exponential Weights with the hinge loss to also obtain an $O(KDX\sqrt{dT\ln(T+1)})$ surrogate regret bound with a polynomial time algorithm. The expected mistake bound of GAPTRON can be found in Theorem 34 and its proof can be found in Section 6.9.2. Compared to the BANDITRON GAPTRON has larger surrogate regret in terms of D, K, and X, but smaller surrogate regret in terms of T. Compared to the surrogate regret of SOBA the surrogate regret of GAPTRON does not contain a factor $\sqrt{d\ln(T+1)}$.

Theorem 34. *Set* $a(\boldsymbol{W}_t, \boldsymbol{x}_t) = 1 - \max\{\mathbb{1}[m_t^\star > \beta], m_t^\star\}$, $\eta = \frac{\gamma(1-\beta)}{K^2X^2}$, $\gamma = \min\left\{1, \sqrt{\frac{K^3X^2D^2}{2(1-\beta)(K-1)T}}\right\}$, *and let* ℓ_t *be the bandit multiclass hinge loss defined in* (6.5.2) *with* $\beta = \frac{1}{K}$. *Then for any* $\boldsymbol{U} \in \mathcal{W}$ *Algorithm 13 satisfies*

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}[y_t' \neq y_t]\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(\boldsymbol{U})\right] + \max\left\{\frac{K^3X^2D^2}{K-1},\ 2KXD\sqrt{\frac{T}{2}}\right\}.$$

6.5.3 Bandit Smooth Multiclass Hinge Loss

In this section we use the following loss function:

$$\ell_t(\boldsymbol{W}) = \begin{cases} \mathbb{1}[y_t' = y_t]p_t'(y_t')^{-1}\max\{1 - 2m_t(\boldsymbol{W}, y_t), 0\} & \text{if } m_t(\boldsymbol{W}, y_t) \leq 0 \\ \mathbb{1}[y_t' = y_t]p_t'(y_t')^{-1}\max\{(1 - m_t(\boldsymbol{W}, y_t))^2, 0\} & \text{if } m_t(\boldsymbol{W}, y_t) > 0. \end{cases} \tag{6.5.3}$$

This loss function is the bandit version of the smooth multiclass hinge loss that we we used in Section 6.4.3 and its expectation is equivalent to its full information counterpart in equation (6.4.3). The surrogate regret of GAPTRON with this loss function can be found in Theorem 35. The proof of Theorem 35 can be found in Section 6.9.3.

Theorem 35. *Set* $a(\boldsymbol{W}_t, \boldsymbol{x}_t) = (1 - \min\{1, m_t^\star\})^2$, $\eta = \frac{\gamma}{4K^2X^2}$, $\gamma = \min\left\{1, \sqrt{\frac{4K^2X^2D^2}{T}}\right\}$, *and let* ℓ_t *be the bandit smooth multiclass hinge loss* *defined in* (6.5.3)*. Then for any $\boldsymbol{U} \in \mathcal{W}$ Algorithm 13 satisfies*

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}[y_t' \neq y_t]\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(\boldsymbol{U})\right] + \max\left\{4K^2X^2D^2,\ 2KXD\sqrt{2T}\right\}.$$

6.6 Conclusion

In this chapter we introduced GAPTRON, a randomized first-order algorithm for the full and bandit information multiclass classification settings. Using a new technique we showed that GAPTRON has a $O(K)$ surrogate regret bound in the full information setting and a surrogate regret bound of order $O(K\sqrt{T})$ in the bandit setting. One of the main drawbacks of GAPTRON is that it is a randomized algorithm. Our bounds only hold in expectation and it would be interesting to show similar bounds also hold with high probability. Another interesting venue to explore is how to extend the ideas in this chapter to the stochastic setting or the more general contextual bandit setting. In future work we would like to conduct experiments to compare GAPTRON with other algorithms, particularly in the bandit setting.

6.7 Details of Section 6.3

Proof of Lemma 25. As we said before, the updates of $\boldsymbol{W}_t$ are Online Gradient Descent (Zinkevich, 2003), which guarantees

$$\sum_{t=1}^{T} (\ell_t(\boldsymbol{W}_t) - \ell_t(\boldsymbol{U})) \leq \frac{\|\boldsymbol{U}\|^2}{2\eta} + \sum_{t=1}^{T} \frac{\eta}{2}\|\boldsymbol{g}_t\|^2. \tag{6.7.1}$$

Now, by using (6.7.1) and $\mathbb{E}_t[\mathbb{1}[y'_t \neq y_t]] = (1 - \max\{a_t, \gamma\})\mathbb{1}[y^\star_t \neq y_t] + \max\{a_t, \gamma\}\frac{K-1}{K}$ we find

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T}\left(\mathbb{1}[y'_t \neq y_t] - \ell_t(\boldsymbol{U})\right)\right] \\
&= \mathbb{E}\left[\sum_{t=1}^{T}\left(\mathbb{1}[y'_t \neq y_t] - \ell_t(\boldsymbol{W}_t)\right) + \sum_{t=1}^{T}\left(\ell_t(\boldsymbol{W}_t) - \ell_t(\boldsymbol{U})\right)\right] \\
&\leq \frac{\|\boldsymbol{U}\|^2}{2\eta} + \mathbb{E}\left[\sum_{t=1}^{T}\left(\mathbb{1}[y'_t \neq y_t] - \ell_t(\boldsymbol{W}_t) + \frac{\eta}{2}\|\boldsymbol{g}_t\|^2\right)\right] \\
&= \frac{\|\boldsymbol{U}\|^2}{2\eta} + \mathbb{E}\left[\sum_{t=1}^{T}\Big((1 - \max\{a_t, \gamma\})\mathbb{1}[y^\star_t \neq y_t]\right. \qquad (6.7.2) \\
&\qquad \left. + \max\{a_t, \gamma\}\frac{K-1}{K} - \ell_t(\boldsymbol{W}_t) + \frac{\eta}{2}\|\boldsymbol{g}_t\|^2\Big)\right] \\
&\leq \frac{\|\boldsymbol{U}\|^2}{2\eta} + \gamma\frac{K-1}{K}T + \mathbb{E}\left[\sum_{t=1}^{T}\Big((1 - a_t)\mathbb{1}[y^\star_t \neq y_t]\right. \\
&\qquad \left. + a_t\frac{K-1}{K} - \ell_t(\boldsymbol{W}_t) + \frac{\eta}{2}\|\boldsymbol{g}_t\|^2\Big)\right],
\end{aligned}$$

where in the last inequality we used $(1-\max\{a_t, \gamma\}) \leq (1-a_t)$ and $\max\{a_t, \gamma\} \leq a_t + \gamma$. Adding $\mathbb{E}\left[\sum_{t=1}^{T} \ell_t(\boldsymbol{U})\right]$ to both sides of equation (6.7.2) completes the proof. □

6.8 Details of Full Information Multiclass Classification

6.8.1 Details of Section 6.4.1

Proof of Theorem 30. We will prove the Theorem by showing that the surrogate gap is bounded by 0 and then using Lemma 25. The gradient of the logistic loss evaluated at $\boldsymbol{W}_t$ is given by:

$$\nabla\ell_t(\boldsymbol{W}_t) = \frac{1}{\ln(2)}(\tilde{\boldsymbol{p}}_t - \boldsymbol{e}_{y_t}) \otimes \boldsymbol{x}_t,$$

where $\tilde{\boldsymbol{p}}_t = (\tilde{p}_t(1), \ldots, \tilde{p}_t(k))^\intercal$ and $\tilde{p}_t(k) = \sigma(\boldsymbol{W}_t, \boldsymbol{x}_t, k)$.

We continue by writing out the surrogate gap:

$$
\begin{aligned}
&(1-a_t)\mathbb{1}[y_t^\star \neq y_t] + a_t\frac{K-1}{K} - \ell_t(\boldsymbol{W}_t) + \frac{\eta}{2}\|\boldsymbol{g}_t\|^2 \\
&\leq (1-a_t)\mathbb{1}[y_t^\star \neq y_t] + a_t\frac{K-1}{K} - \ell_t(\boldsymbol{W}_t) - \frac{\eta}{\ln(2)}\|\boldsymbol{x}_t\|^2\log_2(\tilde{p}_t(y_t)) \\
&\leq (1-a_t)\mathbb{1}[y_t^\star \neq y_t] + a_t\frac{K-1}{K} - \ell_t(\boldsymbol{W}_t) - \frac{\eta}{\ln(2)}X^2\log_2(\tilde{p}_t(y_t)) \\
&= \begin{cases}
0 + \frac{K-1}{K} + \log_2(\tilde{p}_t(y_t)) - \frac{\eta}{\ln(2)}X^2\log_2(\tilde{p}_t(y_t)) & \text{if } p_t^\star < 0.5 \\
p_t^\star + (1-p_t^\star)\frac{K-1}{K} + \log_2(\tilde{p}_t(y_t)) & \\
\quad -\frac{\eta}{\ln(2)}X^2\log_2(\tilde{p}_t(y_t)) & \text{if } y_t^\star \neq y_t \text{ and } p_t^\star \geq 0.5 \\
(1-p_t^\star)\frac{K-1}{K} + \log_2(p_t^\star) - \frac{\eta}{\ln(2)}X^2\log_2(p_t^\star) & \text{if } y_t^\star = y_t \text{ and } p_t^\star \geq 0.5,
\end{cases}
\end{aligned}
\tag{6.8.1}
$$

where the first inequality is due to Lemma 26 below.

We now split the analysis into the cases in (6.8.1). We start with $p_t^\star < 0.5$. In this case we use $1 \leq -\log_2(x)$ for $x \in [0, \frac{1}{2}]$ and obtain

$$
\begin{aligned}
&\frac{K-1}{K} + \log_2(\tilde{p}_t(y_t)) - \frac{\eta}{\ln(2)}X^2\log_2(\tilde{p}_t(y_t)) \\
&\leq -\frac{K-1}{K}\log_2(\tilde{p}_t(y_t)) + \log_2(\tilde{p}_t(y_t)) - \frac{\eta}{\ln(2)}X^2\log_2(\tilde{p}_t(y_t)) \\
&= \frac{1}{K}\log_2(\tilde{p}_t(y_t)) - \frac{\eta}{\ln(2)}X^2\log_2(\tilde{p}_t(y_t)),
\end{aligned}
$$

which is bounded by 0 since $\eta < \frac{\ln(2)}{KX^2}$.

The second case we consider is when $y_t^\star \neq y_t$ and $p_t^\star \geq 0.5$. In this case we use $x \leq -\frac{1}{2}\log_2(1-x)$ for $x \in [0.5, 1]$ and $1 - x \leq -\frac{1}{2}\log_2(1-x)$ for $x \in [0.5, 1]$

and obtain

$$\begin{aligned}
&p_t^\star + (1-p_t^\star)\frac{K-1}{K} + \log_2(\tilde{p}_t(y_t)) - \frac{\eta}{\ln(2)}X^2\log_2(\tilde{p}_t(y_t))\\
&\leq -\tfrac{1}{2}\log_2(1-p_t^\star) - \frac{K-1}{K}\tfrac{1}{2}\log_2(1-p_t^\star) + \log_2(\tilde{p}_t(y_t))\\
&\quad - \frac{\eta}{\ln(2)}X^2\log_2(\tilde{p}_t(y_t))\\
&= -\tfrac{1}{2}\log_2\left(\sum_{k\neq y_t}^{K}\tilde{p}_t(k)\right) - \frac{K-1}{K}\tfrac{1}{2}\log_2\left(\sum_{k\neq y_t}^{K}\tilde{p}_t(k)\right) + \log_2(\tilde{p}_t(y_t))\\
&\quad - \frac{\eta}{\ln(2)}X^2\log_2(\tilde{p}_t(y_t))\\
&\leq -\tfrac{1}{2}\log_2\left(\tilde{p}_t(y_t)\right) - \frac{K-1}{K}\tfrac{1}{2}\log_2\left(\tilde{p}_t(y_t)\right) + \log_2(\tilde{p}_t(y_t))))\\
&\quad - \frac{\eta}{\ln(2)}X^2\log_2(\tilde{p}_t(y_t))\\
&= \frac{1}{2K}\log_2\left(\tilde{p}_t(y_t)\right) - \frac{\eta}{\ln(2)}X^2\log_2(\tilde{p}_t(y_t)),
\end{aligned}$$

which is 0 since $\eta = \frac{\ln(2)}{2KX^2}$.

The last case we need to consider is $y_t^\star = y_t$ and $p_t^\star \geq 0.5$. In this case we use $1 - x \leq -\log_2(x)$ and obtain

$$\begin{aligned}
&(1-p_t^\star)\frac{K-1}{K} + \log_2(p_t^\star) - \frac{\eta}{\ln(2)}X^2\log_2(p_t^\star)\\
&\leq -\frac{K-1}{K}\log_2(p_t^\star) + \log_2(p_t^\star) - \frac{\eta}{\ln(2)}X^2\log_2(p_t^\star),
\end{aligned}$$

which is bounded by 0 since $\eta = \frac{\ln(2)}{2KX^2}$.

We now apply Lemma 25, plug in $\gamma = 0$, and use the above to find:

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}[y_t' \neq y_t]\right] \\
&\leq \frac{\|\boldsymbol{U}\|^2}{2\eta} + \sum_{t=1}^{T} \ell_t(\boldsymbol{U}) + \gamma \frac{K-1}{K} T \\
&\quad + \sum_{t=1}^{T} \Big((1-a_t)\mathbb{1}[y_t^\star \neq y_t] + a_t \frac{K-1}{K} - \ell_t(\boldsymbol{W}_t) + \frac{\eta}{2}\|\boldsymbol{g}_t\|^2 \Big) \\
&\leq \frac{\|\boldsymbol{U}\|^2}{2\eta} + \sum_{t=1}^{T} \ell_t(\boldsymbol{U}).
\end{aligned}$$

Using $\eta = \frac{\ln(2)}{2KX^2}$ completes the proof.

□

Lemma 26. *Let ℓ_t be the logistic loss* (6.4.1)*, then*

$$\|\nabla \ell_t(\boldsymbol{W}_t)\|^2 \leq \frac{2}{\ln(2)} \|\boldsymbol{x}_t\|^2 \ell_t(\boldsymbol{W}_t).$$

Proof. We have

$$\begin{aligned}
\|\nabla \ell_t(\boldsymbol{W}_t)\|^2 =& \frac{1}{\ln(2)^2} \|\boldsymbol{x}_t\|^2 \left(\sum_{k=1}^{K} (\mathbb{1}[y_t = k] - \tilde{p}_t(k))^2 \right) \\
\leq& \frac{1}{\ln(2)^2} \|\boldsymbol{x}_t\|^2 \left(\sum_{k=1}^{K} |\mathbb{1}[y_t = k] - \tilde{p}_t(k)| \right)^2 \\
\leq& -2 \frac{1}{\ln(2)} \|\boldsymbol{x}_t\|^2 \log_2(\tilde{p}_t(y_t)) \\
=& 2\frac{1}{\ln(2)} \|\boldsymbol{x}_t\|^2 \ell_t(\boldsymbol{W}_t),
\end{aligned}$$

where the last inquality follows from Pinsker's inequality (Cover and Thomas, 1991, Lemma 12.6.1). □

6.8.2 Details of Section 6.4.2

Proof of Theorem 31. We will prove the Theorem by showing that the surrogate gap is bounded by 0 and then using Lemma 25. Let $\tilde{k} = \arg\max_{k \neq y_t} \langle \boldsymbol{W}_t^k, \boldsymbol{x}_t \rangle$.

The gradient of the smooth multiclass hinge loss is given by

$$\nabla \ell_t(\boldsymbol{W}_t) = \begin{cases} (\boldsymbol{e}_{\tilde{k}} - \boldsymbol{e}_{y_t}) \otimes \boldsymbol{x}_t & \text{if } y_t^\star \neq y_t \\ (\boldsymbol{e}_{\tilde{k}} - \boldsymbol{e}_{y_t}) \otimes \boldsymbol{x}_t & \text{if } y_t^\star = y_t \text{ and } m_t^\star \leq \beta \\ 0 & \text{if } y_t^\star = y_t \text{ and } m_t^\star > \beta. \end{cases}$$

We continue by writing out the surrogate gap:

$$(1 - a_t)\mathbb{1}[y_t^\star \neq y_t] + a_t\frac{K-1}{K} - \ell_t(\boldsymbol{W}_t) + \frac{\eta}{2}\|\boldsymbol{g}_t\|^2 =$$
$$\begin{cases} m_t^\star + (1 - m_t^\star)\frac{K-1}{K} - (1 - m_t(\boldsymbol{W}_t, y_t)) + \eta\|\boldsymbol{x}_t\|^2 & \text{if } y_t^\star \neq y_t \text{ and } m_t^\star \leq \beta \\ (1 - m_t^\star)\frac{K-1}{K} - (1 - m_t^\star) + \eta\|\boldsymbol{x}_t\|^2 & \text{if } y_t^\star = y_t \text{ and } m_t^\star \leq \beta \\ 1 - (1 - m_t(\boldsymbol{W}_t, y_t)) + \eta\|\boldsymbol{x}_t\|^2 & \text{if } y_t^\star \neq y_t \text{ and } m_t^\star > \beta \\ 0 & \text{if } y_t^\star = y_t \text{ and } m_t^\star > \beta. \end{cases} \tag{6.8.2}$$

In the remainder of the proof we will repeatedly use the following useful inequality for whenever $y_t \neq y_t^\star$:

$$\begin{aligned} m_t^\star + m_t(\boldsymbol{W}_t, y_t) =& \langle \boldsymbol{W}_t^{y_t^\star}, \boldsymbol{x}_t \rangle - \max_{k \neq y_t^\star} \langle \boldsymbol{W}_t^k, \boldsymbol{x}_t \rangle + \langle \boldsymbol{W}_t^{y_t}, \boldsymbol{x}_t \rangle - \max_{k \neq y_t} \langle \boldsymbol{W}_t^k, \boldsymbol{x}_t \rangle \\ =& \langle \boldsymbol{W}_t^{y_t}, \boldsymbol{x}_t \rangle - \max_{k \neq y_t^\star} \langle \boldsymbol{W}_t^k, \boldsymbol{x}_t \rangle \\ \leq& \langle \boldsymbol{W}_t^{y_t}, \boldsymbol{x}_t \rangle - \langle \boldsymbol{W}_t^{y_t}, \boldsymbol{x}_t \rangle = 0. \end{aligned} \tag{6.8.3}$$

We now split the analysis into the cases in (6.8.2). We start with $y_t^\star \neq y_t$ and $m_t^\star \leq \beta$, in which case the surrogate gap can be bounded by 0 when $\eta \leq \frac{1}{KX^2}$:

$$\begin{aligned} & m_t^\star + (1 - m_t^\star)\frac{K-1}{K} - (1 - m_t(\boldsymbol{W}_t, y_t)) + \eta\|\boldsymbol{x}_t\|^2 \\ &= m_t^\star + m_t(\boldsymbol{W}_t, y_t) + (1 - m_t^\star)\frac{K-1}{K} - 1 + \eta\|\boldsymbol{x}_t\|^2 \\ &\leq -\frac{1}{K} + \eta X^2 && \text{(by equation (6.8.3))} \\ &\leq 0. \end{aligned}$$

We continue with the case where $y_t^\star = y_t$ and $m_t^\star \leq \beta$. In this case we have:

$$\begin{aligned} (1 - m_t^\star)\frac{K-1}{K} - (1 - m_t^\star) + \eta\|\boldsymbol{x}_t\|^2 &= -(1 - m_t^\star)\frac{1}{K} + \eta\|\boldsymbol{x}_t\|^2 \\ &\leq -\frac{1-\beta}{K} + \eta X^2, \end{aligned}$$

which is zero since $\eta = \frac{1-\beta}{KX^2}$.

Finally, in the case where $y_t^\star \neq y_t$ and $m_t^\star > \beta$ we have:

$$\begin{aligned} 1 - (1 - m_t(\boldsymbol{W}_t, y_t)) + \eta\|\boldsymbol{x}_t\|^2 =& m_t(\boldsymbol{W}_t, y_t) + \eta\|\boldsymbol{x}_t\|^2 \\ \leq& - m_t^\star + \eta\|\boldsymbol{x}_t\|^2 \qquad \text{(by equation (6.8.3))} \\ \leq& -\beta + \eta X^2, \end{aligned}$$

which is bounded by zero since $\beta = \frac{1}{K}$ and $\eta \leq \frac{1}{KX^2}$.

We now apply Lemma 25, plug in $\gamma = 0$, and use the above to find:

$$\begin{aligned} \mathbb{E}\left[\sum_{t=1}^T \mathbb{1}[y_t' \neq y_t]\right] \leq& \frac{\|\boldsymbol{U}\|^2}{2\eta} + \sum_{t=1}^T \ell_t(\boldsymbol{U}) + \gamma T + \\ & \sum_{t=1}^T \Big((1-a_t)\mathbb{1}[y_t^\star \neq y_t] + a_t\frac{K-1}{K} - \ell_t(\boldsymbol{W}_t) + \frac{\eta}{2}\|\boldsymbol{g}_t\|^2\Big) \\ \leq& \frac{\|\boldsymbol{U}\|^2}{2\eta} + \sum_{t=1}^T \ell_t(\boldsymbol{U}). \end{aligned}$$

Using $\eta = \frac{1-\beta}{KX^2} = \frac{K-1}{K^2X^2}$ completes the proof. □

6.8.3 Details of Section 6.4.3

Proof of Theorem 32. We will prove the Theorem by showing that the surrogate gap is bounded by 0 and then using Lemma 25. Let $\tilde{k} = \arg\max_{k \neq y_t}\langle \boldsymbol{W}_t^k, \boldsymbol{x}_t\rangle$. The gradient of the smooth multiclass hinge loss is given by

$$\nabla \ell_t(\boldsymbol{W}_t) = \begin{cases} 2(\boldsymbol{e}_{\tilde{k}} - \boldsymbol{e}_{y_t}) \otimes \boldsymbol{x}_t & \text{if } y_t^\star \neq y_t \\ 2(\boldsymbol{e}_{\tilde{k}} - \boldsymbol{e}_{y_t})(1 - m_t^\star) \otimes \boldsymbol{x}_t & \text{if } y_t^\star = y_t \text{ and } m_t^\star < 1 \\ 0 & \text{if } y_t^\star = y_t \text{ and } m_t^\star \geq 1. \end{cases}$$

We continue by writing out the surrogate gap:

$$\begin{aligned} &(1-a_t)\mathbb{1}[y_t^\star \neq y_t] + a_t\frac{K-1}{K} - \ell_t(\boldsymbol{W}_t) + \frac{\eta}{2}\|\boldsymbol{g}_t\|^2 = \\ &\begin{cases} 2m_t^\star - m_t^{\star 2} + (1-m_t^\star)^2\frac{K-1}{K} & \\ -(1 - 2m_t(\boldsymbol{W}_t, y_t)) + \eta 4\|\boldsymbol{x}_t\|^2 & \text{if } y_t^\star \neq y_t \text{ and } m_t^\star < 1 \\ (1-m_t^\star)^2\frac{K-1}{K} - (1-m_t^\star)^2 + \eta 4\|\boldsymbol{x}_t\|^2(1-m_t^\star)^2 & \text{if } y_t^\star = y_t \text{ and } m_t^\star < 1 \\ 1 - (1 - 2m_t(\boldsymbol{W}_t, y_t)) + \eta 4\|\boldsymbol{x}_t\|^2 & \text{if } y_t^\star \neq y_t \text{ and } m_t^\star \geq 1 \\ 0 & \text{if } y_t^\star = y_t \text{ and } m_t^\star \geq 1. \end{cases} \end{aligned} \tag{6.8.4}$$

We now split the analysis into the cases in (6.8.4). We start with the case where $y_t^\star \neq y_t$ and $m_t^\star < 1$. By using (6.8.3) we can see that with $\eta = \frac{1}{4KX^2}$ the surrogate gap is bounded by 0:

$$\begin{aligned}
&2m_t^\star - m_t^{\star 2} + (1 - m_t^\star)^2\frac{K-1}{K} - (1 - 2m_t(\boldsymbol{W}_t, y_t)) + \eta 4\|\boldsymbol{x}_t\|^2 \\
&= 2(m_t^\star + m_t(\boldsymbol{W}_t, y_t)) - m_t^{\star 2} + (1 - m_t^\star)^2\frac{K-1}{K} - 1 + \eta 4\|\boldsymbol{x}_t\|^2 \\
&\leq -m_t^{\star 2} + (1 - m_t^\star)^2\frac{K-1}{K} - 1 + \eta 4X^2 \qquad \text{(by equation (6.8.3))} \\
&\leq -\frac{1}{K} + \eta 4X^2 \leq 0.
\end{aligned}$$

The next case we consider is when $y_t^\star = y_t$ and $m_t^\star < 1$. In this case we have

$$\begin{aligned}
&(1 - m_t^\star)^2\frac{K-1}{K} - (1 - m_t^\star)^2 + \eta 4\|\boldsymbol{x}_t\|^2(1 - m_t^\star)^2 \\
&= -(1 - m_t^\star)^2\frac{1}{K} + \eta 4\|\boldsymbol{x}_t\|^2(1 - m_t^\star)^2,
\end{aligned}$$

which is bounded by 0 since $\eta = \frac{1}{4KX^2}$.

Finally, if $y_t^\star \neq y_t$ and $m_t^\star \geq 1$ then

$$\begin{aligned}
1 - (1 - 2m_t(\boldsymbol{W}_t, y_t)) + \eta 4\|\boldsymbol{x}_t\|^2 =& 2m_t(\boldsymbol{W}_t, y_t) + \eta 4\|\boldsymbol{x}_t\|^2 \\
\leq& - 2m_t^\star + \eta 4\|\boldsymbol{x}_t\|^2 \quad \text{(by equation (6.8.3))} \\
\leq& - 2 + \eta 4X^2,
\end{aligned}$$

which is bounded by 0 since $\eta < \frac{1}{2X^2}$. We apply Lemma 25 with $\gamma = 0$ and use the above to find:

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^T \mathbb{1}[y_t' \neq y_t]\right] \leq& \frac{\|\boldsymbol{U}\|^2}{2\eta} + \sum_{t=1}^T \ell_t(\boldsymbol{U}) + \gamma\frac{K-1}{K}T + \\
& \sum_{t=1}^T \Big((1 - a_t)\mathbb{1}[y_t^\star \neq y_t] + a_t\frac{K-1}{K} - \ell_t(\boldsymbol{W}_t) + \frac{\eta}{2}\|\boldsymbol{g}_t\|^2\Big) \\
\leq& \frac{\|\boldsymbol{U}\|^2}{2\eta} + \sum_{t=1}^T \ell_t(\boldsymbol{U}).
\end{aligned}$$

Using $\eta = \frac{1}{4KX^2}$ completes the proof.

□

6.9 Details of Bandit Multiclass Classification

6.9.1 Details of Section 6.5.1

Proof of Theorem 33. First, by straightforward calculations we can see that $p'_t(y_t) \geq \frac{(1-\gamma)\exp(-2DX)+\gamma}{K} = \delta$. As in the full information case we will prove the Theorem by showing that the surrogate gap is bounded by 0 and then using Lemma 25. By using $\mathbb{E}_t[\ell_t(\boldsymbol{W}_t)] = -\log_2(\tilde{p}_t(y_t))$ and $\mathbb{E}_t\left[\|\boldsymbol{g}_t\|^2\right] = \frac{1}{\ln(2)p'_t(y_t)}\|(\tilde{\boldsymbol{p}}_t - \boldsymbol{e}_{y_t}) \otimes \boldsymbol{x}_t\|^2$ we write out the surrogate gap:

$$
\begin{aligned}
&\mathbb{E}\left[(1-a_t)\mathbb{1}[y_t^\star \neq y_t] + a_t\frac{K-1}{K} - \ell_t(\boldsymbol{W}_t) + \frac{\eta}{2}\|\boldsymbol{g}_t\|^2\right] \\
&= \mathbb{E}\Bigg[(1-a_t)\mathbb{1}[y_t^\star \neq y_t] + a_t\frac{K-1}{K} + \log_2(\tilde{p}_t(y_t)) \\
&\quad + \frac{\eta}{2\ln(2)^2 p'_t(y_t)}\|(\tilde{\boldsymbol{p}}_t - \boldsymbol{e}_{y_t}) \otimes \boldsymbol{x}_t\|^2\Bigg] \\
&\leq \mathbb{E}\Bigg[(1-a_t)\mathbb{1}[y_t^\star \neq y_t] + a_t\frac{K-1}{K} + \log_2(\tilde{p}_t(y_t)) \\
&\quad - \frac{\eta}{\ln(2)p'_t(y_t)}X^2\log_2(\tilde{p}_t(y_t))\Bigg] \\
&= \begin{cases}
\frac{K-1}{K} + \mathbb{E}\Big[\log_2(\tilde{p}_t(y_t)) \\
\quad -\frac{\eta}{\ln(2)p'_t(y_t)}X^2\log_2(\tilde{p}_t(y_t))\Big] & \text{if } p_t^\star < 0.5 \\
\mathbb{E}\Big[p_t^\star + (1-p_t^\star)\frac{K-1}{K} + \log_2(\tilde{p}_t(y_t)) \\
\quad -\frac{\eta}{\ln(2)p'_t(y_t)}X^2\log_2(\tilde{p}_t(y_t))\Big] & \text{if } y_t^\star \neq y_t \text{ and } p_t^\star \geq 0.5 \\
\mathbb{E}\Big[(1-p_t^\star)\frac{K-1}{K} + \log_2(p_t^\star) \\
\quad -\frac{\eta}{\ln(2)p'_t(y_t^\star)}X^2\log_2(p_t^\star)\Big] & \text{if } y_t^\star = y_t \text{ and } p_t^\star \geq 0.5,
\end{cases}
\end{aligned}
\tag{6.9.1}
$$

where the first inequality is due to Lemma 26.

We now split the analysis into the cases in (6.9.1). We start with $p_t^\star < 0.5$. In this case we use $1 \leq -\log_2(x)$ for $x \in [0, \frac{1}{2}]$ and obtain

$$\begin{aligned}
&\frac{K-1}{K} + \mathbb{E}[\log_2(\tilde{p}_t(y_t)) - \frac{\eta}{\ln(2)p'_t(y_t)} X^2 \log_2(\tilde{p}_t(y_t))] \\
&\leq \mathbb{E}\left[-\frac{K-1}{K}\log_2(\tilde{p}_t(y_t)) + \log_2(\tilde{p}_t(y_t)) - \frac{\eta}{\ln(2)p'_t(y_t)} X^2 \log_2(\tilde{p}_t(y_t))\right] \\
&\leq \mathbb{E}\left[-\frac{K-1}{K}\log_2(\tilde{p}_t(y_t)) + \log_2(\tilde{p}_t(y_t)) - \frac{\eta}{\ln(2)\delta} X^2 \log_2(\tilde{p}_t(y_t))\right]
\end{aligned}$$

which is bounded by 0 when $\eta \leq \frac{\ln(2)\delta}{KX^2}$.

The second case we consider is when $y_t^\star \neq y_t$ and $p_t^\star \geq 0.5$. In this case we use $x \leq -\frac{1}{2}\log_2(1-x)$ for $x \in [0.5, 1]$ and $1 - x \leq -\frac{1}{2}\log_2(1-x)$ for $x \in [0.5, 1]$ and obtain

$$\begin{aligned}
&\mathbb{E}\left[p_t^\star + (1-p_t^\star)\frac{K-1}{K} + \log_2(\tilde{p}_t(y_t)) - \frac{\eta}{\ln(2)p'_t(y_t)} X^2 \log_2(\tilde{p}_t(y_t))\right] \\
&\leq \mathbb{E}\Bigg[-\tfrac{1}{2}\log_2(1-p_t^\star) - \frac{K-1}{K}\tfrac{1}{2}\log_2(1-p_t^\star) + \log_2(\tilde{p}_t(y_t)) \\
&\quad - \frac{\eta}{\ln(2)\delta} X^2 \log_2(\tilde{p}_t(y_t))\Bigg] \\
&= \mathbb{E}\Bigg[-\tfrac{1}{2}\log_2\left(\sum_{k\neq y_t}^{K} \tilde{p}_t(k)\right) - \frac{K-1}{K}\tfrac{1}{2}\log_2\left(\sum_{k\neq y_t}^{K} \tilde{p}_t(k)\right) + \log_2(\tilde{p}_t(y_t)) \\
&\quad - \frac{\eta}{\ln(2)\delta} X^2 \log_2(\tilde{p}_t(y_t))\Bigg] \\
&\leq \mathbb{E}\Bigg[-\tfrac{1}{2}\log_2\left(\tilde{p}_t(y_t)\right) - \frac{K-1}{K}\tfrac{1}{2}\log_2\left(\tilde{p}_t(y_t)\right) + \log_2(\tilde{p}_t(y_t)) \\
&\quad - \frac{\eta}{\ln(2)\delta} X^2 \log_2(\tilde{p}_t(y_t))\Bigg] \\
&= \mathbb{E}\left[\frac{1}{2K}\log_2(\tilde{p}_t(y_t)) - \frac{\eta}{\ln(2)\delta} X^2 \log_2(\tilde{p}_t(y_t))\right],
\end{aligned}$$

which is bounded by 0 since $\eta = \frac{\ln(2)\delta}{2KX^2}$.

The last case we need to consider is when $y_t^\star = y_t$ and $p_t^\star \geq 0.5$. In this case we use $1 - x \leq -\log_2(x)$ and obtain

$$\begin{aligned}
&\mathbb{E}\left[(1-p_t^\star)\frac{K-1}{K} + \log_2(p_t^\star) - \frac{\eta}{\ln(2)p_t'(y_t^\star)}X^2\log_2(p_t^\star)\right] \\
&\leq \mathbb{E}\left[-\frac{K-1}{K}\log_2(p_t^\star) + \log_2(p_t^\star) - \frac{\eta}{\ln(2)\delta}X^2\log_2(p_t^\star)\right],
\end{aligned}$$

which is bounded by 0 when $\eta \leq \frac{\ln(2)\delta}{KX^2}$.

We now apply Lemma 25 and use the above to find:

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^T \mathbb{1}[y_t' \neq y_t]\right] \\
&\leq \frac{\|\boldsymbol{U}\|^2}{2\eta} + \mathbb{E}\left[\sum_{t=1}^T \ell_t(\boldsymbol{U})\right] + \gamma\frac{K-1}{K}T \\
&\quad + \sum_{t=1}^T \mathbb{E}\left[(1-a_t)\mathbb{1}[y_t^\star \neq y_t] + a_t\frac{K-1}{K} - \ell_t(\boldsymbol{W}_t) + \frac{\eta}{2}\|\boldsymbol{g}_t\|^2\right] \\
&\leq \frac{\|\boldsymbol{U}\|^2}{2\eta} + \gamma T + \mathbb{E}\left[\sum_{t=1}^T \ell_t(\boldsymbol{U})\right].
\end{aligned}$$

Using $\eta = \frac{\ln(2)\delta}{2KX^2}$ gives us:

$$\mathbb{E}\left[\sum_{t=1}^T \mathbb{1}[y_t' \neq y_t]\right] \leq \frac{K^2X^2\|\boldsymbol{U}\|^2}{\ln(2)((1-\gamma)\exp(-2DX)+\gamma)} + \gamma T + \mathbb{E}\left[\sum_{t=1}^T \ell_t(\boldsymbol{U})\right],$$

Setting $\gamma = 0$ gives us

$$\mathbb{E}\left[\sum_{t=1}^T \mathbb{1}[y_t' \neq y_t]\right] \leq \frac{K^2X^2D^2}{\ln(2)\exp(-2DX)} + \mathbb{E}\left[\sum_{t=1}^T \ell_t(\boldsymbol{U})\right].$$

If instead we set $\gamma = \min\left\{1, \sqrt{\frac{K^2X^2D^2}{\ln(2)T}}\right\}$ we consider two cases. In the case where $1 \leq \sqrt{\frac{K^2X^2D^2}{T}}$ we have that $T \leq K^2X^2D^2$ and therefore

$$\mathbb{E}\left[\sum_{t=1}^T \mathbb{1}[y_t' \neq y_t]\right] \leq 2\frac{K^2X^2D^2}{\ln(2)} + \mathbb{E}\left[\sum_{t=1}^T \ell_t(\boldsymbol{U})\right].$$

In the case where $1 > \sqrt{\frac{K^2X^2D^2}{T}}$ we have that

$$\mathbb{E}\left[\sum_{t=1}^T \mathbb{1}[y_t' \neq y_t]\right] \leq 2KXD\sqrt{\frac{T}{\ln(2)}} + \mathbb{E}\left[\sum_{t=1}^T \ell_t(\boldsymbol{U})\right],$$

which after combining the above completes the proof. □

6.9.2 Details of Section 6.5.2

Proof of Theorem 34. First, note that $p'_t(y_t) \geq \frac{\gamma}{K}$. The proof proceeds in a similar way as in the full information setting (Theorem 31), except now we use that $p'_t(y_t) \geq \frac{\gamma}{K}$ to bound $\mathbb{E}_t[\|\boldsymbol{g}_t\|^2]$. We will prove the Theorem by showing that the surrogate gap is bounded by 0 and then using Lemma 25. By using $\mathbb{E}_t\left[\mathbb{1}[y'_t = y_t]p'_t(y'_t)^{-1}\right] = 1$ and $\mathbb{E}_t\left[\left(\mathbb{1}[y'_t = y_t]p'_t(y'_t)^{-1}\right)^2\right] = \mathbb{1}[y'_t = y_t]p'_t(y'_t)^{-1}$ we start by splitting the surrogate gap in cases:

$$\mathbb{E}\left[(1-a_t)\mathbb{1}[y_t^\star \neq y_t] + a_t\frac{K-1}{K} - \ell_t(\boldsymbol{W}_t) + \frac{\eta}{2}\|\boldsymbol{g}_t\|^2\right]$$
$$= \begin{cases} \mathbb{E}\left[m_t^\star + (1-m_t^\star)\frac{K-1}{K} - (1-m_t(\boldsymbol{W}_t, y_t)) \right. \\ \left. + \frac{\eta}{p'_t(y_t)}\|\boldsymbol{x}_t\|^2\right] & \text{if } y_t^\star \neq y_t \text{ and } m_t^\star \leq \beta \\ \mathbb{E}\left[(1-m_t^\star)\frac{K-1}{K} - (1-m_t^\star) + \frac{\eta}{p'_t(y_t)}\|\boldsymbol{x}_t\|^2\right] & \text{if } y_t^\star = y_t \text{ and } m_t^\star \leq \beta \\ \mathbb{E}\left[1 - (1-m_t(\boldsymbol{W}_t, y_t)) + \frac{\eta}{p'_t(y_t)}\|\boldsymbol{x}_t\|^2\right] & \text{if } y_t^\star \neq y_t \text{ and } m_t^\star > \beta \\ 0 & \text{if } y_t^\star = y_t \text{ and } m_t^\star > \beta. \end{cases} \tag{6.9.2}$$

We now split the analysis into the cases in (6.9.2). We start with $y_t^\star \neq y_t$ and $m_t^\star \leq \beta$. The surrogate gap can now be bounded by 0 when $\eta \leq \frac{\gamma}{K^2X^2}$:

$$\begin{aligned}
&\mathbb{E}\left[m_t^\star + (1-m_t^\star)\frac{K-1}{K} - (1-m_t(\boldsymbol{W}_t, y_t)) + \frac{\eta}{p'_t(y_t)}\|\boldsymbol{x}_t\|^2\right] \\
&= \mathbb{E}\left[m_t^\star + m_t(\boldsymbol{W}_t, y_t) + (1-m_t^\star)\frac{K-1}{K} - 1 + \frac{\eta}{p'_t(y_t)}\|\boldsymbol{x}_t\|^2\right] \\
&\leq -\frac{1}{K} + \frac{K\eta}{\gamma}X^2 && \text{(equation (6.8.3))} \\
&\leq 0.
\end{aligned}$$

We continue with the case where $y_t^\star = y_t$ and $m_t^\star \leq \beta$. In this case we have:

$$\begin{aligned}
\mathbb{E}\left[(1-m_t^\star)\frac{K-1}{K} - (1-m_t^\star) + \eta\|\boldsymbol{x}_t\|^2\right] &= \mathbb{E}\left[-(1-m_t^\star)\frac{1}{K} + \frac{\eta}{p'_t(y_t)}\|\boldsymbol{x}_t\|^2\right] \\
&\leq -\frac{1-\beta}{K} + \frac{K\eta}{\gamma}X^2,
\end{aligned}$$

which is bounded by zero since $\eta = \frac{\gamma(1-\beta)}{K^2X^2}$.

Finally, in the case where $y_t^\star \neq y_t$ and $m_t^\star > \beta$ we have:

$$\begin{aligned}
&\mathbb{E}\left[1 - (1 - m_t(\boldsymbol{W}_t, y_t)) + \frac{\eta}{p_t'(y_t)}\|\boldsymbol{x}_t\|^2\right] \\
&= \mathbb{E}\left[m_t(\boldsymbol{W}_t, y_t) + \frac{\eta}{p_t'(y_t)}\|\boldsymbol{x}_t\|^2\right] \\
&\leq \mathbb{E}\left[-m_t^\star + \frac{\eta}{p_t'(y_t)}\|\boldsymbol{x}_t\|^2\right] && \text{(by equation (6.8.3))} \\
&\leq -\beta + \frac{K\eta}{\gamma}X^2,
\end{aligned}$$

which is bounded by zero since $\eta = \frac{\gamma(1-\beta)}{K^2X^2}$ and $\beta \leq 0.5$.

We now apply Lemma 25 and use the above to find:

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}[y_t' \neq y_t]\right] \\
&\leq \frac{\|\boldsymbol{U}\|^2}{2\eta} + \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(\boldsymbol{U})\right] + \gamma\frac{K-1}{K}T \\
&\quad + \sum_{t=1}^{T} \mathbb{E}\left[(1 - a_t)\mathbb{1}[y_t^\star \neq y_t] + a_t\frac{K-1}{K} - \ell_t(\boldsymbol{W}_t) + \frac{\eta}{2}\|\boldsymbol{g}_t\|^2\right] \\
&\leq \frac{D^2}{2\eta} + \gamma\frac{K-1}{K}T + \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(\boldsymbol{U})\right].
\end{aligned}$$

Plugging in $\eta = \frac{\gamma(1-\beta)}{K^2X^2}$ and $\beta = \frac{1}{K}$ gives us:

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}[y_t' \neq y_t]\right] \leq \frac{K^3X^2D^2}{2\gamma(K-1)} + \gamma\frac{K-1}{K}T + \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(\boldsymbol{U})\right].$$

We now set $\gamma = \min\left\{1, \sqrt{\frac{K^3X^2D^2}{2(1-\beta)(K-1)T}}\right\}$. In the case where $1 \leq \sqrt{\frac{K^3X^2D^2}{2(1-\beta)(K-1)T}}$ we have

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}[y_t' \neq y_t]\right] \leq \frac{K^3X^2D^2}{K-1} + \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(\boldsymbol{U})\right].$$

In the case where $1 > \sqrt{\frac{K^3X^2D^2}{2(1-\beta)(K-1)T}}$ we have

$$\mathbb{E}\left[\sum_{t=1}^{T}\mathbb{1}[y_t' \neq y_t]\right] \leq 2KXD\sqrt{\frac{T}{2}} + \mathbb{E}\left[\sum_{t=1}^{T}\ell_t(\boldsymbol{U})\right],$$

which completes the proof. □

6.9.3 Details of Section 6.5.3

Proof of Theorem 35. First, note that $p_t'(y_t) \geq \frac{\gamma}{K}$. The proof proceeds in a similar way as in the full information case. We will prove the Theorem by showing that the surrogate gap is bounded by 0 and then using Lemma 25. By using $\mathbb{E}_t\left[\mathbb{1}[y_t' = y_t]p_t'(y_t')^{-1}\right] = 1$ and $\mathbb{E}_t\left[\left(\mathbb{1}[y_t' = y_t]p_t'(y_t')^{-1}\right)^2\right] = \mathbb{1}[y_t' = y_t]p_t'(y_t')^{-1}$ we can expand the surrogate gap:

$$\begin{aligned}
&\mathbb{E}\left[(1-a_t)\mathbb{1}[y_t^\star \neq y_t] + a_t\frac{K-1}{K} - \ell_t(\boldsymbol{W}_t) + \frac{\eta}{2}\|\boldsymbol{g}_t\|^2\right] \\
&= \begin{cases}
\mathbb{E}\Big[2m_t^\star - m_t^{\star 2} + (1-m_t^\star)^2\frac{K-1}{K} & \\
\quad -(1-2m_t(\boldsymbol{W}_t, y_t)) + \frac{\eta}{p_t'(y_t)}4\|\boldsymbol{x}_t\|^2\Big] & \text{if } y_t^\star \neq y_t \text{ and } m_t^\star < 1 \\
\mathbb{E}\Big[(1-m_t^\star)^2\frac{K-1}{K} - (1-m_t^\star)^2 & \\
\quad +\frac{\eta}{p_t'(y_t)}4\|\boldsymbol{x}_t\|^2(1-m_t^\star)^2\Big] & \text{if } y_t^\star = y_t \text{ and } m_t^\star < 1 \\
\mathbb{E}\left[1-(1-2m_t(\boldsymbol{W}_t, y_t)) + \frac{\eta}{p_t'(y_t)}4\|\boldsymbol{x}_t\|^2\right] & \text{if } y_t^\star \neq y_t \text{ and } m_t^\star \geq 1 \\
0 & \text{if } y_t^\star = y_t \text{ and } m_t^\star \geq 1.
\end{cases}
\end{aligned} \tag{6.9.3}$$

We now split the analysis into the cases in (6.9.3). We start with the case where $y_t^\star \neq y_t$ and $m_t^\star < 1$. By using (6.8.3) we can see that for $\eta = \frac{\gamma}{4K^2X^2}$

$$\begin{aligned}
&\mathbb{E}\left[2m_t^\star - m_t^{\star 2} + (1-m_t^\star)^2\frac{K-1}{K} - (1-2m_t(\boldsymbol{W}_t, y_t)) + \frac{\eta}{p_t'(y_t)}4\|\boldsymbol{x}_t\|^2\right] \\
&= \mathbb{E}\left[2(m_t^\star + m_t(\boldsymbol{W}_t, y_t)) - m_t^{\star 2} + (1-m_t^\star)^2\frac{K-1}{K} - 1 + \frac{\eta}{p_t'(y_t)}4\|\boldsymbol{x}_t\|^2\right] \\
&\leq \mathbb{E}\left[-m_t^{\star 2} + (1-m_t^\star)^2\frac{K-1}{K} - 1 + \frac{\eta}{p_t'(y_t)}4X^2\right] \qquad \text{(by equation (6.8.3))} \\
&\leq -\frac{1}{K} + \frac{K\eta}{\gamma}4X^2 \leq 0.
\end{aligned}$$

The next case we consider is when $y_t^\star = y_t$ and $m_t^\star < 1$. In this case we have

$$\begin{aligned}
&\mathbb{E}\left[(1 - m_t^\star)^2 \frac{K-1}{K} - (1 - m_t^\star)^2 + \frac{\eta}{p_t'(y_t)} 4\|\boldsymbol{x}_t\|^2 (1 - m_t^\star)^2\right] \\
&= \mathbb{E}\left[-(1 - m_t^\star)^2 \frac{1}{K} + \frac{\eta}{p_t'(y_t)} 4\|\boldsymbol{x}_t\|^2 (1 - m_t^\star)^2\right] \\
&= \mathbb{E}\left[-(1 - m_t^\star)^2 \frac{1}{K} + \frac{K\eta}{\gamma} 4X^2 (1 - m_t^\star)^2\right],
\end{aligned}$$

which is bounded by 0 since $\eta = \frac{\gamma}{4K^2X^2}$.

Finally, if $y_t^\star \neq y_t$ and $m_t^\star \geq 1$ then

$$\begin{aligned}
&\mathbb{E}\left[1 - (1 - 2m_t(\boldsymbol{W}_t, y_t)) + \frac{\eta}{p_t'(y_t)} 4\|\boldsymbol{x}_t\|^2\right] \\
&= \mathbb{E}\left[2m_t(\boldsymbol{W}_t, y_t) + \frac{\eta}{p_t'(y_t)} 4\|\boldsymbol{x}_t\|^2\right] \\
&\leq \mathbb{E}\left[-2m_t^\star + \frac{\eta}{p_t'(y_t)} 4\|\boldsymbol{x}_t\|^2\right] && \text{(by equation (6.8.3))} \\
&\leq -2 + \frac{K\eta}{\gamma} 4X^2,
\end{aligned}$$

which is bounded by 0 since $\eta < \frac{\gamma}{2K^2X^2}$. We apply Lemma 25 and use the above to find:

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^T \mathbb{1}[y_t' \neq y_t]\right] \\
&\leq \frac{\|\boldsymbol{U}\|^2}{2\eta} + \mathbb{E}\left[\sum_{t=1}^T \ell_t(\boldsymbol{U})\right] + \gamma T \\
&\quad + \sum_{t=1}^T \mathbb{E}\left[(1 - a_t)\mathbb{1}[y_t^\star \neq y_t] + a_t \frac{K-1}{K} - \ell_t(\boldsymbol{W}_t) + \frac{\eta}{2}\|\boldsymbol{g}_t\|^2\right] \\
&\leq \frac{D^2}{2\eta} + \gamma T + \mathbb{E}\left[\sum_{t=1}^T \ell_t(\boldsymbol{U})\right].
\end{aligned}$$

Plugging in $\eta = \frac{\gamma}{4K^2X^2}$ gives us:

$$\mathbb{E}\left[\sum_{t=1}^T \mathbb{1}[y_t' \neq y_t]\right] \leq \frac{2K^2X^2D^2}{\gamma} + \gamma T + \mathbb{E}\left[\sum_{t=1}^T \ell_t(\boldsymbol{U})\right].$$

Algorithm 14 ADAHEDGE with abstention

```
Input: ADAHEDGE
for $t = 1 \dots T$ do
    Obtain expert predictions $\boldsymbol{y}_t = (y_t^1, \dots, y_t^d)^\intercal$
    Obtain expert distribution $\hat{\boldsymbol{p}}_t$ from ADAHEDGE
    Set $\hat{y}_t = \langle \hat{\boldsymbol{p}}_t, \boldsymbol{y}_t \rangle$
    Let $y_t^\star = \mathrm{sign}(\hat{y}_t)$
    Set $b_t = 1 - |\hat{y}_t|$
    Predict $y_t' = y_t^\star$ with probability $1 - b_t$ and predict $y_t' = *$ with probability $b_t$
    Obtain $\ell_t$ and send $\ell_t$ to ADAHEDGE
end for
```

Now we set $\gamma = \min\left\{1, \sqrt{\frac{2K^2X^2D^2}{T}}\right\}$. In the case where $1 \leq \sqrt{\frac{2K^2X^2D^2}{T}}$ we have

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}[y_t' \neq y_t]\right] \leq 4K^2X^2D^2 + \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(\boldsymbol{U})\right].$$

In the case where $1 > \sqrt{\frac{2K^2X^2D^2}{T}}$ we have

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}[y_t' \neq y_t]\right] \leq 2DKX\sqrt{2T} + \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(\boldsymbol{U})\right],$$

which completes the proof.

□

6.10 Online Classification with Abstention

The online classification with abstention setting was introduced by Neu and Zhivotovskiy (2020) and is a special case of the prediction with expert advice setting Vovk (1990); Littlestone and Warmuth (1994). For brevity we only consider the case where there are only 2 labels, -1 and 1. The online classification with abstention setting is different from the standard classification setting in that the learner has access to a third option, abstaining. Neu and Zhivotovskiy (2020) show that when the cost for abstaining is smaller than $\frac{1}{2}$ in all rounds it is possible to tune Exponential Weights such that it suffers constant regret with respect to the best expert in hindsight. Neu and Zhivotovskiy (2020) only consider the zero-one loss, but we show that a similar bound also holds for the hinge loss (and also for the zero one loss as a special case of the hinge loss). We use a different proof technique from Neu and Zhivotovskiy (2020), which was the inspiration for the proofs of the mistake bounds of GAPTRON. Instead of vanilla Exponential Weights we use a slight adaptation of ADAHEDGE (De Rooij et al., 2014) to prove constant regret bounds when all abstention costs c_t are smaller than $\frac{1}{2}$. In online classification with abstention, in each round t

1 the learner observes the predictions $y_t^i \in [-1, 1]$ of experts $i = 1, \dots, d$

2 based on the experts' predictions the learner predicts $y_t' \in [-1, 1] \cup *$, where $*$ stands for abstaining

3 the environment reveals $y_t \in \{-1, 1\}$

4 the learner suffers loss $\ell_t(y_t') = \frac{1}{2}(1 - y_t y_t')$ if $y_t' \in [-1, 1]$ and c_t otherwise.

The algorithm we use can be found in Algorithm 14. A parallel result to Lemma 25 can be found in Lemma 27, which we will use to derive the regret of Algorithm 14.

Lemma 27. *For any expert i, the expected loss of Algorithm 14 satisfies:*

$$\begin{aligned}
&\sum_{t=1}^{T} ((1 - b_t)\ell_t(y_t^\star) + b_t c_t) \\
&\leq \sum_{t=1}^{T} \ell_t(y_t^i) + \inf_{\eta > 0} \left\{ \frac{\ln(d)}{\eta} + \sum_{t=1}^{T} \underbrace{((1 - b_t)\ell_t(y_t^\star) + c_t b_t + \eta v_t - \ell_t(\hat{y}_t))}_{\text{Abstention gap}} \right\} \\
&\quad + \frac{4}{3}\ln(d) + 2,
\end{aligned}$$

where $v_t = \mathbb{E}_{i\sim\hat{\boldsymbol{p}}_t}[(\ell_t(\hat{y}_t) - \ell_t(y_t^i))^2]$.

Before we prove Lemma 27 let us compare Algorithm 14 with GAPTRON. The updates of weight matrix $\boldsymbol{W}_t$ in GAPTRON are performed with OGD. In Algorithm 14 the updates or $\hat{p}_t$ are performed using ADAHEDGE. The roles of a_t in GAPTRON and b_t in Algorithm 14 are similar. The role of a_t is to ensure that the surrogate gap is bounded by 0, the role of b_t is to ensure that the abstention gap is bounded by 0.

Proof of Lemma 27. First, ADAHEDGE guarantees that

$$\sum_{t=1}^{T} \left(\ell_t(\hat{y}_t) - \ell_t(y_t^i)\right) \leq 2\sqrt{\ln(d) \sum_{t=1}^{T} v_t} + 4/3\ln(d) + 2.$$

Using the regret bound of AdaHedge we can upper bound the expectation of the loss of the learner as

$$\begin{aligned}
&\sum_{t=1}^{T} ((1-b_t)\ell_t(y_t^\star) + b_t c_t) \\
&= \sum_{t=1}^{T} \big((1-b_t)\ell_t(y_t^\star) + b_t c_t + \ell_t(y_t^i) - \ell_t(\hat{y}_t)\big) + \sum_{t=1}^{T} \big(\ell_t(\hat{y}_t) - \ell_t(y_t^i)\big) \\
&\leq \sum_{t=1}^{T} \big((1-b_t)\ell_t(y_t^\star) + b_t c_t + \ell_t(y_t^i) - \ell_t(\hat{y}_t)\big) + 2\sqrt{\ln(d)\sum_{t=1}^{T} v_t} \\
&\quad + 4/3\ln(d) + 2 \\
&= \sum_{t=1}^{T} \ell_t(y_t^i) + \inf_{\eta>0}\left\{\frac{\ln(d)}{\eta} + \sum_{t=1}^{T}((1-b_t)\ell_t(y_t^\star) + b_t c_t + \eta v_t - \ell_t(\hat{y}_t))\right\} \\
&\quad + 4/3\ln(d) + 2.
\end{aligned}$$

□

To upper bound the abstention gap by 0 is more difficult than to upper bound the surrogate gap as the negative term is no longer an upper bound on the zero-one loss. Hence, the abstention cost has to be strictly better than randomly guessing as otherwise there is no η or b_t such that the abstention gap is smaller than 0. The result for abstention can be found in Theorem 36 below.

Theorem 36. *Suppose* $\max_t c_t < \frac{1}{2}$ *for all* T. *Then Algorithm 14 guarantees*

$$\begin{aligned}
&\sum_{t=1}^{T} ((1-b_t)\ell_t(y_t^\star) + b_t c_t) \\
&\leq \sum_{t=1}^{T} \ell_t(y_t^i) + \min\left\{\frac{\ln(d)}{1-2\max_t c_t}, 2\sqrt{\ln(d)\sum_{t=1}^{T} v_t}\right\} + 4/3\ln(d) + 2.
\end{aligned}$$

Proof. We start by upper bounding the v_t term. We have

$$v_t = \frac{1}{4}\mathbb{E}_{\hat{\boldsymbol{p}}_t}\left[(y_t^i - \hat{y}_t)^2\right] \leq \frac{1}{4}(1-\hat{y}_t)(\hat{y}_t+1) \leq \tfrac{1}{2}(1-|\hat{y}_t|),$$

where the first inequality is the Bhatia-Davis inequality (Bhatia and Davis, 2000).

As with the proofs of GAPTRON we split the abstention gap in cases:

$$
\begin{aligned}
&(1-b_t)\ell_t(y_t^\star) + c_t b_t + \eta v_t - \ell_t(\hat{y}_t) \\
&\leq (1-b_t)\ell_t(y_t^\star) + c_t b_t + \eta\tfrac{1}{2}(1-|\hat{y}_t|) - \ell_t(\hat{y}_t) \\
&= \begin{cases} c_t(1-|\hat{y}_t|) + \eta\frac{1}{2}(1-|\hat{y}_t|) - \frac{1}{2}(1-|\hat{y}_t|) & \text{if } y_t^\star = y_t \\ |\hat{y}_t| + c_t(1-|\hat{y}_t|) + \eta\frac{1}{2}(1-|\hat{y}_t|) - \frac{1}{2}(1+|\hat{y}_t|) & \text{if } y_t^\star \neq y_t. \end{cases}
\end{aligned} \tag{6.10.1}
$$

Note that regardless of the the true label $(1-b_t)\ell_t(y_t^\star) + c_t b_t - \ell_t(\hat{y}_t) \leq 0$ since $c_t < \frac{1}{2}$. Hence, by using Lemma 27, we can see that as long as $c_t < \frac{1}{2}$

$$
\sum_{t=1}^T ((1-b_t)\ell_t(y_t^\star) + b_t c_t) \leq \sum_{t=1}^T \ell_t(y_t^i) + 2\sqrt{\ln(d)\sum_{t=1}^T v_t} + 4/3\ln(d) + 2.
$$

Now consider the case where $y_t^\star = y_t$. In this case, as long as $\eta \leq 1 - 2c_t$ the abstention gap is bounded by 0. If $y_t^\star \neq y_t$ then

$$
\begin{aligned}
&|\hat{y}_t| + c_t(1-|\hat{y}_t|) + \eta\tfrac{1}{2}(1-|\hat{y}_t|) - \tfrac{1}{2}(1+|\hat{y}_t|) \\
&= c_t(1-|\hat{y}_t|) + \eta\tfrac{1}{2}(1-|\hat{y}_t|) - \tfrac{1}{2}(1-|\hat{y}_t|).
\end{aligned}
$$

So as long as $\eta \leq 1 - 2c_t$ the abstention gap is bounded by 0. Applying Lemma 27 now gives us

$$
\begin{aligned}
&\sum_{t=1}^T \big((1-b_t)\ell_t(y_t^\star) + b_t c_t - \ell_t(y_t^i)\big) \\
&\leq \inf_{\eta>0}\left\{\frac{\ln(d)}{\eta} + \sum_{t=1}^T ((1-b_t)\ell_t(y_t^\star) + c_t b_t + \eta v_t - \ell_t(\hat{y}_t))\right\} + 4/3\ln(d) + 2 \\
&\leq \frac{\ln(d)}{1 - 2\max_t c_t} + 4/3\ln(d) + 2,
\end{aligned}
$$

which completes the proof. □

With a slight modification of the proof of Theorem 36 one can also show a similar result as Theorem 8 by Neu and Zhivotovskiy (2020), albeit with slightly worse constants. We leave this as an exercise for the reader.

CHAPTER 7

Open Problem: Fast and Optimal Online Portfolio Selection

This chapter is based on Van Erven, T., Van der Hoeven, D., Kotłowski, W., and Koolen, W. M. (2020b). Open problem: Fast and optimal online portfolio selection. In *Proceedings of the 33rd Annual Conference on Learning Theory (COLT)*, pages 3864–3869.[1]

Abstract

Online portfolio selection has received much attention since its introduction by Cover, but all state-of-the-art methods fall short in at least one of the following ways: they are either i) computationally infeasible; or ii) they do not guarantee optimal regret; or iii) they assume the gradients are bounded, which is unnecessary and cannot be guaranteed. We are interested in a natural follow-the-regularized-leader (FTRL) approach based on the log barrier regularizer, which is computationally feasible. The open problem we put before the community is to formally prove whether this approach achieves the optimal regret. Resolving this question will likely lead to new techniques to analyse FTRL algorithms. There are also interesting technical connections to self-concordance, which has previously been used in the context of bandit convex optimization.

[1] The author of this dissertation performed the following tasks: co-deriving the theoretical results and co-writing the paper.

7.1 Introduction

Online portfolio selection (Cover, 1991) may be viewed as an instance of online convex optimization (OCO) (Hazan et al., 2016): in each of $t = 1, \dots, T$ rounds, a learner has to make a prediction $\boldsymbol{w}_t$ in a convex domain $\mathcal{W}$ before observing a convex loss function $f_t : \mathcal{W} \to \mathbb{R}$. The goal is to obtain a guaranteed bound on the regret $\mathcal{R}_T = \sum_{t=1}^T f_t(\boldsymbol{w}_t) - \min_{\boldsymbol{w}\in\mathcal{W}} \sum_{t=1}^T f_t(\boldsymbol{w})$ that holds for any possible sequence of loss functions f_t. Online portfolio selection corresponds to the special case that the domain $\mathcal{W} = \{\boldsymbol{w} \in \mathbb{R}_+^d \mid \sum_{i=1}^d w_i = 1\}$ is the probability simplex and the loss functions are restricted to be of the form $f_t(\boldsymbol{w}) = -\ln(\boldsymbol{w}^\intercal \boldsymbol{x}_t)$ for vectors $\boldsymbol{x}_t \in \mathbb{R}_+^d$. It was introduced by Cover (1991) with the interpretation that $x_{t,i}$ represents the factor by which the value of an asset $i \in \{1, \dots, d\}$ grows in round t and $w_{t,i}$ represents the fraction of our capital we re-invest in asset i in round t. The factor by which our initial capital grows over T rounds then becomes $\prod_{t=1}^T \boldsymbol{w}_t^\intercal \boldsymbol{x}_t = e^{-\sum_{t=1} f_t(\boldsymbol{w}_t)}$. An alternative interpretation in terms of mixture learning is given by Orseau et al. (2017).

For an extensive survey of online portfolio selection we refer to Li and Hoi (2014). Here we review only the results that are most relevant to our open problem. Cover (1991); Cover and Ordentlich (1996) show that the best possible guarantee on the regret is of order $\mathcal{R}_T = O(d \ln T)$ and that this is achieved by choosing $\boldsymbol{w}_{t+1}$ as the mean of a continuous exponential weights distribution $\mathrm{d}P_{t+1}(\boldsymbol{w}) \propto e^{-\sum_{s=1}^t f_s(\boldsymbol{w})} \mathrm{d}\pi(\boldsymbol{w})$ with Dirichlet-prior π (and learning rate $\eta = 1$). Unfortunately, this approach has a run-time of order $O(T^d)$, which scales exponentially in the number of assets d, and is therefore computationally infeasible when d exceeds, say, 3. A sampling-based implementation by Kalai and Vempala (2002) greatly improves the run-time to $\tilde{O}(T^4(T+d)d^2)$, but even this is still infeasible already for modest d and T.

As shown in Table 7.1, much faster algorithms are available, but they either do not achieve the optimal regret or they assume that the gradients are uniformly bounded by a *known* bound G: $\|\nabla f_t(\boldsymbol{w}_t)\|_2 \leq G$, and the bounds deteriorate rapidly when G is large. Bounding the gradients is very restrictive: we either need to (i) assume that the asset prices do not fluctuate too rapidly, which defeats the purpose of using adversarial online learning; or (ii) we need to allocate a minimum amount of capital $w_{t,i} \geq \alpha$ to each asset, which means we cannot drop any poorly performing assets from our portfolio.

We are interested in a natural follow-the-regularized-leader algorithm, previously

Table 7.1: Overview of achievable trade-offs between regret and run-time

Method	Regret	Run-time	Assumes Bounded Gradients	References
Universal Portfolio	$O(d\ln(T))$	$\tilde{O}(T^4(T+d)d^2)$	No	(Cover and Ordentlich, 1996; Kalai and Vempala, 2002)

Online Newton Step	$O(Gd\ln(T))$	$O(d^3T)$	Yes	(Agarwal et al., 2006;
				Hazan et al., 2007;
				Hazan and Kale, 2015)

Exponentiated Gradient	$O(G\sqrt{T\ln(d)})$	$O(dT)$	Yes	Helmbold et al. (1998)
Gradient Descent	$O(G\sqrt{dT})$	$O(dT)$	Yes	Zinkevich (2003)
Soft-Bayes	$O(\sqrt{dT\ln(d)})$	$O(dT)$	No	Orseau et al. (2017)
Ada-BARRONS	$O(d^2\ln^4(T))$	$O(d^{2.5}T^2)$	No	Luo et al. (2018)
FTRL	?	$O(d^2T^2)$	No	Agarwal and Hazan (2005)

proposed by Agarwal and Hazan (2005):

$$\boldsymbol{w}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{w}\in\mathcal{W}} \left\{ \sum_{s=1}^{t} f_s(\boldsymbol{w}) + \lambda \sum_{i=1}^{d} -\ln w_i \right\} \tag{7.1.1}$$

for some $\lambda > 0$. The regularizer $R(\boldsymbol{w}) = \sum_{i=1}^{d} -\ln w_i$ is a self-concordant barrier function (Nesterov and Nemirovskii, 1994) that is the log barrier for the positive orthant and has a natural interpretation as adding d extra rounds in which $\boldsymbol{x}$ equals $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d$.

The optimization problem (7.1.1) can be solved to machine precision in $O(d^2t)$ steps using Newton's method, so a naive implementation in which we solve the optimization problem independently for each round would already lead to a total run-time of $O(d^2T^2)$, which is computationally feasible for practical values of d and T. One might further hope that sharing calculations between rounds or solving (7.1.1) approximately may lead to additional speed-ups, similar to those obtained for FTRL with linear losses by Abernethy et al. (2008). Thus the method is computationally feasible, at least for an interesting range of d and T. The open problem we now pose is whether it is also worst-case optimal in terms of regret:

Open Problem: *Does the FTRL algorithm* (7.1.1) *guarantee the optimal regret* $O(d\ln T)$ *without further assumptions like bounded gradients?*

Our motivation is twofold: efficient algorithms for portfolio selection (and beyond) are desirable, and FTRL is the simplest natural candidate. In addition, our current inability to analyse it highlights frustrating blind spots in our FTRL toolbox, which solving this problem will need to address.

Agarwal and Hazan (2005) already prove $O(G^2 d \ln(dT))$ regret when the gradients are bounded, but we believe that the bound should not depend on G at all. It seems that the key difficulty in analyzing the regret is to control the sum of so-called local norms of the gradients. As we will discuss below, this is possible at least in several encouraging special cases.

7.2 Technical Discussion

It is convenient to reparametrize by $\boldsymbol{v} \in \mathbb{R}_+^{d-1}$ such that $\sum_{i=1}^{d-1} v_i \leq 1$, obtaining $\boldsymbol{w}_t = A\boldsymbol{v}_t + \boldsymbol{b}$ for $A = \begin{pmatrix} \boldsymbol{I} \\ -\mathbf{1}^\intercal \end{pmatrix}$, and $\boldsymbol{b} = \boldsymbol{e}_d$. With some abuse of notation, we will also write $f_t(\boldsymbol{v})$ for $f_t(A\boldsymbol{v} + \boldsymbol{b})$ and $R(\boldsymbol{v})$ for $R(A\boldsymbol{v} + \boldsymbol{b})$. Then the criterion being minimized is

$$\phi_T(\boldsymbol{v}) = \sum_{t=1}^{T} f_t(\boldsymbol{v}) + \lambda R(\boldsymbol{v}).$$

As the loss is 1-exp-concave, we have $\nabla^2 f_t(\boldsymbol{v}) \succeq \nabla f_t(\boldsymbol{v}) \nabla f_t(\boldsymbol{v})^\intercal$ (Bubeck, 2015, pp. 324–325). In fact, this holds with equality in the present case:

$$\nabla f_t(\boldsymbol{v}) = \frac{-A^\intercal \boldsymbol{x}_t}{(A\boldsymbol{v} + \boldsymbol{b})^\intercal \boldsymbol{x}_t}, \quad \nabla^2 f_t(\boldsymbol{v}) = \frac{A^\intercal \boldsymbol{x}_t \boldsymbol{x}_t^\intercal A}{\big((A\boldsymbol{v} + \boldsymbol{b})^\intercal \boldsymbol{x}_t\big)^2} = \nabla f_t(\boldsymbol{v}) \nabla f_t(\boldsymbol{v})^\intercal.$$

7.2.1 Regret Bounded by Local Norms via Self-concordance

We observe that both the losses f_t and the regularizer R are self-concordant functions (Abernethy et al., 2008). Assume for simplicity that $\lambda \geq 1$, in which case ϕ_T is a sum of self-concordant functions and hence also self-concordant. Like Abernethy et al. (2008), define the local norms $\|\boldsymbol{g}\|_t = \sqrt{\boldsymbol{g}^\intercal \nabla^{-2} \phi_t(\boldsymbol{v}_t) \boldsymbol{g}}$. By Lemma 28 below we know that the gradients are always bounded in these local norms.

Lemma 28. $\|\nabla f_t(\boldsymbol{v}_t)\|_t^2 \leq \frac{1}{\lambda+1}$

Proof. We start by observing that

$$\begin{aligned}
\|\nabla f_t(\boldsymbol{v}_t)\|_t^2 \leq& \nabla f_t(\boldsymbol{v}_t)^\intercal (\nabla f_t(\boldsymbol{v}_t) \nabla f_t(\boldsymbol{v}_t)^\intercal + \lambda \nabla^2 R(\boldsymbol{v}_t))^{-1} \nabla f_t(\boldsymbol{v}_t) \\
=& \lambda^{-1} \|\nabla f_t(\boldsymbol{v}_t)\|_{R(\boldsymbol{v}_t)}^2 - \frac{\lambda^{-2} \|\nabla f_t(\boldsymbol{v}_t)\|_{R(\boldsymbol{v}_t)}^4}{1 + \lambda^{-1} \|\nabla f_t(\boldsymbol{v}_t)\|_{R(\boldsymbol{v}_t)}^2} \\
=& \frac{\|\nabla f_t(\boldsymbol{v}_t)\|_{R(\boldsymbol{v}_t)}^2}{\lambda + \|\nabla f_t(\boldsymbol{v}_t)\|_{R(\boldsymbol{v}_t)}^2},
\end{aligned}$$

where $\|\boldsymbol{g}\|_{R(\boldsymbol{v}_t)} = \sqrt{\boldsymbol{g}^\intercal \nabla^{-2} R(\boldsymbol{v}_t)\boldsymbol{g}}$ and the first equality follows from the Sherman-Morrison formula. Note that $\nabla^2 R(\boldsymbol{v}_t)$ is positive definite, so $\nabla f_t(\boldsymbol{v}_t)^\intercal \nabla^2 R(\boldsymbol{v}_t)\nabla f_t(\boldsymbol{v}_t) = 0$ only when $\nabla f_t(\boldsymbol{v}_t) = \mathbf{0}$, for which the result holds. If $\nabla f_t(\boldsymbol{v}_t)^\intercal \nabla^2 R(\boldsymbol{v}_t)\nabla f_t(\boldsymbol{v}_t) > 0$, because $\nabla^2 R(\boldsymbol{v}) \succeq \nabla f_t(\boldsymbol{v})\nabla f_t(\boldsymbol{v})^\intercal$ for all $\boldsymbol{v}$ by Lemma 29 below we have

$$\begin{aligned}&\nabla f_t(\boldsymbol{v}_t)^\intercal \nabla^{-2} R(\boldsymbol{v}_t)\nabla f_t(\boldsymbol{v}_t)\nabla f_t(\boldsymbol{v}_t)^\intercal \nabla^2 R(\boldsymbol{v}_t)\nabla f_t(\boldsymbol{v}_t)\\ &\le \nabla f_t(\boldsymbol{v}_t)^\intercal \nabla^2 R(\boldsymbol{v}_t)\nabla f_t(\boldsymbol{v}_t)\end{aligned}$$

and thus $\nabla f_t(\boldsymbol{v}_t)^\intercal \nabla^{-2} R(\boldsymbol{v}_t)\nabla f_t(\boldsymbol{v}_t) \le 1$. Using that $s(x) = x/(\lambda + x)$ is increasing for $x > -\lambda$ we conclude that the gradients are indeed bounded in the local norms:

$$\|\nabla f_t(\boldsymbol{v}_t)\|_t^2 \le \frac{\|\nabla f_t(\boldsymbol{v}_t)\|_{R(\boldsymbol{v}_t)}^2}{\lambda + \|\nabla f_t(\boldsymbol{v}_t)\|_{R(\boldsymbol{v}_t)}^2} \le \frac{1}{\lambda+1}. \tag{7.2.1}$$

□

Lemma 29. $\nabla^2 R(\boldsymbol{v}) \succeq \nabla f_t(\boldsymbol{v})\nabla f_t(\boldsymbol{v})^\intercal$ *for all* $\boldsymbol{v}$.

Proof. We need to show that, for all $\boldsymbol{x}$ and $\boldsymbol{w}$:

$$A^\intercal \Big(\sum_{i=1}^d \frac{\boldsymbol{e}_i\boldsymbol{e}_i^\intercal}{\big(\boldsymbol{w}^\intercal \boldsymbol{e}_i\big)^2}\Big)A \succeq \frac{A^\intercal \boldsymbol{x}\boldsymbol{x}^\intercal A}{\big(\boldsymbol{w}^\intercal\boldsymbol{x}\big)^2}.$$

It is sufficient to show that:

$$\sum_{i=1}^d \frac{\boldsymbol{e}_i\boldsymbol{e}_i^\intercal}{\big(\boldsymbol{w}^\intercal \boldsymbol{e}_i\big)^2} \succeq \frac{\boldsymbol{x}\boldsymbol{x}^\intercal}{\big(\boldsymbol{w}^\intercal\boldsymbol{x}\big)^2}.$$

Both sides are positive semi-definite. The right-hand side is rank 1, with eigenvector $\boldsymbol{x}$. Hence it is sufficient to show that

$$\begin{aligned}\boldsymbol{x}^\intercal\Big(\sum_{i=1}^d \frac{\boldsymbol{e}_i\boldsymbol{e}_i^\intercal}{\big(\boldsymbol{w}^\intercal \boldsymbol{e}_i\big)^2}\Big)\boldsymbol{x} &\ge \boldsymbol{x}^\intercal\Big(\frac{\boldsymbol{x}\boldsymbol{x}^\intercal}{\big(\boldsymbol{w}^\intercal\boldsymbol{x}\big)^2}\Big)\boldsymbol{x}\\ \sum_{i=1}^d \frac{x_i^2}{w_i^2} &\ge \frac{\|\boldsymbol{x}\|_2^4}{\big(\boldsymbol{w}^\intercal\boldsymbol{x}\big)^2}\\ \big(\boldsymbol{w}^\intercal\boldsymbol{x}\big)^2\sum_{i=1}^d \frac{x_i^2}{w_i^2} &\ge \|\boldsymbol{x}\|_2^4\\ \|\boldsymbol{y}\|_1^2\|\boldsymbol{z}\|_2^2 &\ge (\boldsymbol{y}^\intercal\boldsymbol{z})^2\\ \|\boldsymbol{y}\|_1\|\boldsymbol{z}\|_2 &\ge \boldsymbol{y}^\intercal\boldsymbol{z}\end{aligned}$$

where $\boldsymbol{y} = (y_1, \ldots, y_d)$ for $y_i = w_i x_i$, $\boldsymbol{z} = (z_1, \ldots, z_d)$ for $z_i = x_i/w_i$, and we are using that $w_i, x_i \geq 0$. The result then follows upon observing that $\|\boldsymbol{z}\|_1 \geq \|\boldsymbol{z}\|_2$, and applying the Cauchy-Schwarz inequality. □

To bound the regret the following Lemma is useful.

Lemma 30. *For $\lambda \geq \frac{5}{4}$, the regret is bounded in terms of the local norms:*

$$\mathcal{R}_T \leq \lambda d \ln(2T) + 1 + \sum_{t=1}^{T} \|\nabla f_t(\boldsymbol{v}_t)\|_t^2.$$

Proof. Let $\boldsymbol{v}^* \in \arg\min_{\boldsymbol{v}} \sum_{t=1}^{T} f_t(\boldsymbol{v})$. Then

$$\mathcal{R}_T = \phi_T(\boldsymbol{v}_{T+1}) - \lambda R(\boldsymbol{v}_1) - \sum_{t=1}^{T} f_t(\boldsymbol{v}^*) + \sum_{t=1}^{T} \big(\phi_t(\boldsymbol{v}_t) - \phi_t(\boldsymbol{v}_{t+1})\big).$$

We start by bounding

$$\begin{aligned}
\phi_T(\boldsymbol{v}_{T+1}) - \lambda R(\boldsymbol{v}_1) \leq & \phi_T\big((1 - \tfrac{1}{2T})\boldsymbol{v}^* + \tfrac{1}{2T}\boldsymbol{v}_1\big) - \lambda R(\boldsymbol{v}_1) \\
\leq & \sum_{t=1}^{T} - \ln((1 - \tfrac{1}{2T})(A\boldsymbol{v}^* + \boldsymbol{b})^\intercal \boldsymbol{x}_t) \\
& + \sum_{i=1}^{d} -\lambda \ln(\tfrac{1}{2T}(A\boldsymbol{v}_1 + \boldsymbol{b})^\intercal \boldsymbol{e}_i) - \lambda R(\boldsymbol{v}_1) \\
= & \sum_{t=1}^{T} f_t(\boldsymbol{v}^*) - T\ln(1 - \tfrac{1}{2T}) + d\lambda \ln(2T) \\
\leq & \sum_{t=1}^{T} f_t(\boldsymbol{v}^*) + \lambda d \ln(2T) + \frac{1}{2(1 - \frac{1}{2T})} \\
\leq & \sum_{t=1}^{T} f_t(\boldsymbol{v}^*) + \lambda d \ln(2T) + 1
\end{aligned}$$

Next, by using (2.16) of Nemirovski (2004) for the self-concordant function ϕ_t we find

$$\begin{aligned}
\phi_t(\boldsymbol{v}_t) - \phi_t(\boldsymbol{v}_{t+1}) &\leq - \ln(1 - \|\nabla f_t(\boldsymbol{v}_t)\|_t) - \|\nabla f_t(\boldsymbol{v}_t)\|_t \\
&\leq \|\nabla f_t(\boldsymbol{v}_t)\|_t^2,
\end{aligned}$$

where we used that $\nabla\phi_t(\boldsymbol{v}_t) = \nabla f_t(\boldsymbol{v}_t)$, $-\ln(1-s) - s \leq s^2$ for $s \in [0, \frac{2}{3}]$, and $\|\nabla f_t(\boldsymbol{v}_t)\|_t^2 \leq \frac{4}{9}$ by equation (7.2.1). To complete the proof we combine the above and find

$$\begin{aligned}\mathcal{R}_T =& \phi_T(\boldsymbol{v}_{T+1}) - \lambda R(\boldsymbol{v}_1) - \sum_{t=1}^T f_t(\boldsymbol{v}^*) + \sum_{t=1}^T \big(\phi_t(\boldsymbol{v}_t) - \phi_t(\boldsymbol{v}_{t+1})\big) \\ \leq& \lambda d \ln(2T) + 1 + \sum_{t=1}^T \big(\phi_t(\boldsymbol{v}_t) - \phi_t(\boldsymbol{v}_{t+1})\big) \\ \leq& \lambda d \ln(2T) + 1 + \sum_{t=1}^T \|\nabla f_t(\boldsymbol{v}_t)\|_t^2.\end{aligned}$$

□

Combining Lemma 30 with (7.2.1), we immediately see that the regret is bounded by

$$\mathcal{R}_T = O(\sqrt{dT\ln T}) \qquad \text{for } \lambda \approx \sqrt{\tfrac{T}{d\ln T}},$$

but if we hope to get the optimal rate, we need to use constant λ, so this is what we will assume from now on. Below we list several promising corollaries of Lemma 30.

7.2.2 Assuming Bounded Gradients

Suppose that, for some reason, the gradients with respect to $\boldsymbol{w}$ (not $\boldsymbol{v}$!) are bounded: $\|\nabla f_t(\boldsymbol{w}_t)\|_2 = \|\frac{-\boldsymbol{x}_t}{\boldsymbol{w}_t^\intercal \boldsymbol{x}_t}\|_2 \leq G$. Then, abbreviating $\boldsymbol{y}_t = A^\intercal \boldsymbol{x}_t / \|\boldsymbol{x}_t\|_2$, we can use that $\nabla^2 f_t(\boldsymbol{v}_t) \succeq \frac{A^\intercal \boldsymbol{x}_t \boldsymbol{x}_t^\intercal A}{\|\boldsymbol{x}_t\|_\infty^2} \succeq \frac{A^\intercal \boldsymbol{x}_t \boldsymbol{x}_t^\intercal A}{\|\boldsymbol{x}_t\|_2^2} = \boldsymbol{y}_t \boldsymbol{y}_t^\intercal$ to get

$$\begin{aligned}\sum_{t=1}^T \|\nabla f_t(\boldsymbol{v}_t)\|_t^2 \leq& G^2 \sum_{t=1}^T \|\boldsymbol{y}_t\|_t^2 \\ \leq& G^2 \sum_{t=1}^T \boldsymbol{y}_t^\intercal \Big(\sum_{s=1}^t \boldsymbol{y}_s \boldsymbol{y}_s^\intercal + \lambda A^\intercal A\Big)^{-1} \boldsymbol{y}_t \\ =& O\Big(G^2 d \ln T\Big),\end{aligned}$$

where the last step follows analogously to Hazan et al. (2007, Lemma 11) and using that $\det(A^\intercal A) = \det(\boldsymbol{I} + \boldsymbol{1}\boldsymbol{1}^\intercal) = (1 + \boldsymbol{1}^\intercal\boldsymbol{1})\det(\boldsymbol{I}) = d$ by Sylvester's determinant theorem. This gives the optimal rate if G is small.

7.2.3 Source Coding and $\boldsymbol{x}_t$ in Finite Set

We call the case that $\boldsymbol{x}_t \in \{\boldsymbol{e}_1, \dots, \boldsymbol{e}_d\}$ the source coding setting. This case is easy to analyse, because $\boldsymbol{w}_t$ has a simple closed-form solution that coincides with Cover's universal portfolio algorithm. More generally, let us assume that $\boldsymbol{x}_t$ takes values in some finite set $\mathcal{X}$ of size k, so $k = d$ in the source coding setting, and let $n_t(\boldsymbol{x})$ denote the number of times that $\boldsymbol{x}_s = \boldsymbol{x}$ for $s \leq t$. Then

$$\begin{aligned}
&\sum_{t=1}^{T} \|\nabla f_t(\boldsymbol{v}_t)\|_t^2 \\
&\leq \sum_{t=1}^{T} \nabla f_t(\boldsymbol{v}_t)^\intercal \big(n_t(\boldsymbol{x}_t)\nabla f_t(\boldsymbol{v}_t)\nabla f_t(\boldsymbol{v}_t)^\intercal + \lambda \nabla^2 R(\boldsymbol{v}_t)\big)^{-1} \nabla f_t(\boldsymbol{v}_t) \\
&\leq \sum_{t=1}^{T} \frac{1}{n_t(\boldsymbol{x}_t) + \lambda} = \sum_{\boldsymbol{x} \in \mathcal{X}} \sum_{j=1}^{n_T(\boldsymbol{x})} \frac{1}{j + \lambda} = O\big(\sum_{\boldsymbol{x} \in \mathcal{X}} \ln n_T(\boldsymbol{x})\big) = O\big(k \ln T\big).
\end{aligned}$$

In particular, algorithm (7.1.1) achieves the optimal rate in the source coding setting.

7.2.4 A (Suboptimal) General Bound without Bounded Gradients

Since $R(\boldsymbol{v})$ is a barrier, it should be the case that $w_{t,i} \geq C/t$ for some constant $C > 0$. We may therefore cover the effective domain of $\boldsymbol{w}_t$ by $m = O((\ln T)^d)$ sets $B_1, \dots, B_m$ such that $\boldsymbol{w}^\intercal \boldsymbol{x} \leq 2\boldsymbol{u}^\intercal \boldsymbol{x}$ for all $\boldsymbol{w}, \boldsymbol{u} \in B_i$. It follows that

$$\begin{aligned}
&\sum_{t=1}^{T} \|\nabla f_t(\boldsymbol{v}_t)\|_t^2 \\
&\leq \sum_{i=1}^{m} \sum_{t: \boldsymbol{v}_t \in B_i} \nabla f_t(\boldsymbol{v}_t)^\intercal \big(\sum_{s \leq t: \boldsymbol{v}_s \in B_i} \nabla f_s(\boldsymbol{v}_t) \nabla f_s(\boldsymbol{v}_t)^\intercal + \lambda \nabla^2 R(\boldsymbol{v}_t)\big)^{-1} \nabla f_t(\boldsymbol{v}_t) \\
&\leq 4 \sum_{i=1}^{m} \sum_{t: \boldsymbol{w}_t \in B_i} \nabla f_t(\boldsymbol{v}_t)^\intercal \big(\sum_{s \leq t: \boldsymbol{w}_s \in B_i} \nabla f_s(\boldsymbol{v}_s) \nabla f_s(\boldsymbol{v}_s)^\intercal + \lambda A^\intercal A\big)^{-1} \nabla f_t(\boldsymbol{v}_t) \\
&= O(md \ln T) = O\big(d(\ln T)^{d+1}\big),
\end{aligned}$$

where the first equality follows like Lemma 11 of Hazan et al. (2007) with $\|\nabla f_t(\boldsymbol{v}_t)\| \leq t/C$. This of course has wildly suboptimal dependence in d, but shows near-optimal regret for very small d.

7.3 Discussion

The partial analysis presented above relies on (2.16) of Nemirovski (2004). An alternative approach could be to use (2.4) of Nemirovski (2004) instead, as is done by Bilodeau et al. (2020) to analyse the logarithmic loss. Another attempt could be made to improve the approach of Section 7.2.4 by employing other techniques from literature on self-concordant barriers. Inside the Dikin ellipsoid the hessians of self-concordant barriers are roughly proportional (see for example Proposition 2.3.2 by Nesterov and Nemirovskii (1994) or (2.2) by Nemirovski (2004)). Instead of covering the domain as described in Section 7.2.4 perhaps it is possible to cover the domain in Dikin ellipsoids more efficiently, although several unfruitful attempts have already been made.

Bibliography

Abernethy, J., Hazan, E., and Rakhlin, A. (2008). Competing in the dark: An efficient algorithm for bandit linear optimization. In *Proceedings of the 21st Annual Conference on Learning Theory (COLT)*, pages 263–274.

Abernethy, J., Hazan, E., and Rakhlin, A. (2012). Interior-point methods for full-information and bandit online learning. *IEEE Trans. Information Theory*, 58(7):4164–4175.

Abernethy, J. and Rakhlin, A. (2009). An efficient bandit algorithm for $\sqrt{T}$-regret in online multiclass prediction? In *Proceedings of the 22nd Annual Conference on Learning Theory (COLT)*.

Abernethy, J. D., Jung, Y. H., Lee, C., McMillan, A., and Tewari, A. (2019). Online learning via the differential privacy lens. In *Advances in Neural Information Processing Systems 32*, pages 8894–8904.

Agarwal, A., Dekel, O., and Xiao, L. (2010). Optimal algorithms for online convex optimization with multi-point bandit feedback. In *Proceedings of the 23rd Annual Conference on Learning Theory (COLT)*, pages 28–40. Citeseer.

Agarwal, A., Foster, D. P., Hsu, D. J., Kakade, S. M., and Rakhlin, A. (2011). Stochastic convex optimization with bandit feedback. In *Advances in Neural Information Processing Systems 24*, pages 1035–1043.

Agarwal, A. and Hazan, E. (2005). Efficient algorithms for online game playing and universal portfolio management. *ECCC, TR06-033*.

Agarwal, A., Hazan, E., Kale, S., and Schapire, R. E. (2006). Algorithms for portfolio management based on the newton method. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, pages 9–16.

Agarwal, N., Bullins, B., Chen, X., Hazan, E., Singh, K., Zhang, C., and Zhang, Y. (2018). The case for full-matrix adaptive regularization. *arXiv preprint arXiv:1806.02958*.

Agarwal, N. and Singh, K. (2017). The price of differential privacy for online learning. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 32–40.

Arora, S., Hazan, E., and Kale, S. (2012). The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164.

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. (2002). The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77.

Azoury, K. S. and Warmuth, M. K. (2001). Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine Learning*, 43(3):211–246.

Banerjee, A., Merugu, S., Dhillon, I. S., and Ghosh, J. (2005). Clustering with bregman divergences. *The Journal of Machine Learning Research*, 6:1705–1749.

Bartlett, P. L., Hazan, E., and Rakhlin, A. (2007). Adaptive online gradient descent. In *Advances in Neural Information Processing Systems 20*, pages 65–72.

Bartlett, P. L., Jordan, M. I., and McAuliffe, J. D. (2006). Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156.

Bartlett, P. L. and Mendelson, S. (2006). Empirical minimization. *Probability Theory and Related Fields*, 135(3):311–334.

Beck, A. and Teboulle, M. (2003). Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175.

Berkson, J. (1944). Application of the logistic function to bio-assay. *Journal of the American Statistical Association*, 39(227):357–365.

Beygelzimer, A., Orabona, F., and Zhang, C. (2017). Efficient online bandit multiclass learning with o($\sqrt{T}$) regret. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 488–497.

Bhatia, R. and Davis, C. (2000). A better bound on the variance. *The American Mathematical Monthly*, 107(4):353–357.

Bilodeau, B., Foster, D. J., and Roy, D. M. (2020). Improved bounds on minimax regret under logarithmic loss via self-concordance. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*.

Bubeck, S. (2015). Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8(3–4):231–357.

Bubeck, S., Cesa-Bianchi, N., and Kakade, S. M. (2012). Towards minimax policies for online linear optimization with bandit feedback. In *Proceedings of the 25th Annual Conference on Learning Theory (COLT)*, pages 41.1–41.14.

Bubeck, S., Dekel, O., Koren, T., and Peres, Y. (2015). Bandit convex optimization: $\sqrt{T}$ regret in one dimension. In *Proceedings of the 28th Annual Conference on Learning Theory (COLT)*, pages 266–278.

Bubeck, S. and Eldan, R. (2015). The entropic barrier: a simple and optimal universal self-concordant barrier. In *Proceedings of the 28th Annual Conference on Learning Theory (COLT)*, pages 279–279.

Bubeck, S. and Eldan, R. (2016). Multi-scale exploration of convex functions and bandit convex optimization. In *Proceedings of the 29th Annual Conference on Learning Theory (COLT)*, pages 583–589.

Bubeck, S., Lee, Y. T., and Eldan, R. (2017). Kernel-based methods for bandit convex optimization. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 72–85. ACM.

Cesa-Bianchi, N., Conconi, A., and Gentile, C. (2005). A second-order perceptron algorithm. *SIAM Journal on Computing*, 34(3):640–668.

Cesa-Bianchi, N. and Lugosi, G. (2006). *Prediction, learning, and games*. Cambridge university press.

Cesa-Bianchi, N., Mansour, Y., and Stoltz, G. (2007). Improved second-order bounds for prediction with expert advice. *Machine Learning*, 66(2-3):321–352.

Chang, C.-C. and Lin, C.-J. (2011). Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):27.

Chen, Y., Lee, C.-W., Luo, H., and Wei, C.-Y. (2019). A new algorithm for non-stationary contextual bandits: Efficient, optimal, and parameter-free. In *Proceedings of the 32nd Annual Conference On Learning Theory (COLT)*, pages 696–726.

Chen, Z., Xu, Y., Chen, E., and Yang, T. (2018). Sadagrad: Strongly adaptive stochastic gradient methods. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 912–920.

Chernov, A. and Vovk, V. (2010). Prediction with advice of unknown number of experts. *Uncertainty in Aritificial Intelligence*, pages 117–125.

Chiang, C.-K., Yang, T., Le, C.-J., Mahdavi, M., Lu, C.-J., Jin, R., and Zhu, S. (2012). Online optimization with gradual variations. In *Proc. of the 25th Annual Conference on Learning Theory (COLT)*, pages 6.1–6.20.

Cover, T. M. (1991). Universal portfolios. *Mathematical Finance*, 1(1):1–29.

Cover, T. M. and Ordentlich, E. (1996). Universal portfolios with side information. *IEEE Transactions on Information Theory*, 42(2):348–363.

Cover, T. M. and Thomas, J. A. (1991). *Elements of information theory*. John Wiley & Sons.

Crammer, K., Kulesza, A., and Dredze, M. (2009). Adaptive regularization of weight vectors. In *Advances in Neural Information Processing Systems 22*, pages 414–422.

Crammer, K. and Singer, Y. (2001). On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2(Dec):265–292.

Crammer, K. and Singer, Y. (2003). Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3(Jan):951–991.

Csiszár, I. (1975). I-divergence geometry of probability distributions and minimization problems. *The Annals of Probability*, 3(1):146–158.

Cutkosky, A. (2019). Artificial constraints and hints for unbounded online learning. In *Proceedings of the 32nd Annual Conference on Learning Theory (COLT)*, pages 874–894.

Cutkosky, A. and Boahen, K. (2017). Online learning without prior information. In *Proceedings of the 30th Annual Conference on Learning Theory (COLT)*, pages 643–677.

Cutkosky, A. and Orabona, F. (2018). Black-box reductions for parameter-free online learning in banach spaces. In *Proceedings of the 31th Annual Conference on Learning Theory (COLT)*, pages 1493–1529.

Dani, V., Kakade, S. M., and Hayes, T. P. (2008). The price of bandit information for online optimization. In *Advances in Neural Information Processing Systems 21*, pages 345–352.

Daniely, A., Gonen, A., and Shalev-Shwartz, S. (2015). Strongly adaptive online learning. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 1405–1411.

Deswarte, R. (2018). *Linear regression and learning: contributions to regularization and aggregation methods*. PhD thesis, Université Paris-Saclay.

Dick, T., György, A., and Szepesvári, C. (2014). Online learning in Markov decision processes with changing cost sequences. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, pages 512–520.

Do, C. B., Le, Q. V., and Foo, C.-S. (2009). Proximal regularization for online and batch learning. In *Proceedings of the 26th Annual International Conf. on Machine Learning (ICML)*, pages 257–264.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.

Duchi, J. C., Jordan, M. I., and Wainwright, M. J. (2014). Privacy aware learning. *Journal of the ACM (JACM)*, 61(6):38.

Dwork, C. and Roth, A. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407.

Van Erven, T., Grünwald, P. D., Mehta, N. A., Reid, M. D., and Williamson, R. C. (2015). Fast rates in statistical and online learning. *Journal of Machine Learning Research*, 16:1793–1861.

Fink, M., Shalev-Shwartz, S., Singer, Y., and Ullman, S. (2006). Online multiclass learning by interclass hypothesis sharing. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, pages 313–320.

Flaxman, A. D., Kalai, A. T., Kalai, A. T., and McMahan, H. B. (2005). Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the 16th annual ACM-SIAM symposium on Discrete algorithms*, pages 385–394.

Foster, D. J., Kale, S., Luo, H., Mohri, M., and Sridharan, K. (2018a). Logistic regression: The importance of being improper. In *Proceedings of the 31st Annual Conference On Learning Theory (COLT)*, pages 167–208.

Foster, D. J., Kale, S., Mohri, M., and Sridharan, K. (2017). Parameter-free online learning via model selection. In *Advances in Neural Information Processing Systems 30*, pages 6020–6030.

Foster, D. J. and Krishnamurthy, A. (2018). Contextual bandits with surrogate losses: Margin bounds and efficient algorithms. In *Advances in Neural Information Processing Systems 31*, pages 2621–2632.

Foster, D. J., Krishnamurthy, A., and Luo, H. (2019). Model selection for contextual bandits. In *Advances in Neural Information Processing Systems 32*, pages 14714–14725.

Foster, D. J., Rakhlin, A., and Sridharan, K. (2015). Adaptive online learning. In *Advances in Neural Information Processing Systems 28*, pages 3375–3383.

Foster, D. J., Rakhlin, A., and Sridharan, K. (2018b). Online learning: Sufficient statistics and the burkholder method. In *Proceedings of the 31th Annual Conference on Learning Theory (COLT)*, pages 3028–3064.

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139.

Freund, Y., Schapire, R. E., Singer, Y., and Warmuth, M. K. (1997). Using and combining predictors that specialize. In *Proc. 29th Annual ACM Symposium on Theory of Computing*, pages 334–343. ACM.

Ghashami, M., Liberty, E., Phillips, J. M., and Woodruff, D. P. (2016). Frequent directions: Simple and deterministic matrix sketching. *SIAM Journal on Computing*, 45(5):1762–1792.

Golub, G. H. and Van Loan, C. F. (2012). *Matrix computations*, volume 3. JHU Press.

Grünwald, P. D. (2007). *The minimum description length principle*. MIT press.

Hazan, E., Agarwal, A., and Kale, S. (2007). Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192.

Hazan, E. et al. (2016). Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325.

Hazan, E. and Kale, S. (2010). Extracting certainty from uncertainty: Regret bounded by variation in costs. *Machine learning*, 80(2-3):165–188.

Hazan, E. and Kale, S. (2011). Newtron: an efficient bandit algorithm for online multiclass prediction. In *Advances in Neural Information Processing Systems 24*, pages 891–899.

Hazan, E. and Kale, S. (2015). An online portfolio selection algorithm with regret logarithmic in price variation. *Mathematical Finance*, 25(2):288–310.

Hazan, E., Koren, T., and Levy, K. Y. (2014). Logistic regression: Tight bounds for stochastic and online optimization. In *Proceedings of the 27th Annual Conference on Learning Theory (COLT)*, pages 197–209.

Hazan, E. and Levy, K. (2014). Bandit convex optimization: Towards tight bounds. In *Advances in Neural Information Processing Systems 27*, pages 784–792.

Helmbold, D. P., Schapire, R. E., Singer, Y., and Warmuth, M. K. (1998). Online portfolio selection using multiplicative updates. *Mathematical Finance*, 8(4):325–347.

Helmbold, D. P. and Warmuth, M. K. (2009). Learning permutations with Exponential Weights. *Journal of Machine Learning Research*, 10:1705–1736.

Hiriart-Urruty, J.-B. (2006). A note on the legendre-fenchel transform of convex composite functions. In *Nonsmooth Mechanics and Analysis*, pages 35–46. Springer.

Ihara, S. (1993). *Information Theory for Continuous Systems*, volume 2. World Scientific.

Jain, P., Kothari, P., and Thakurta, A. (2012). Differentially private online learning. In *Proceedings of the 25th Annual Conference on Learning Theory (COLT)*, pages 24–1.

Jain, P. and Thakurta, A. G. (2014). (near) dimension independent risk bounds for differentially private learning. In *Proceedings of the 31th International Conference on Machine Learning (ICML)*, pages 476–484.

Jun, K.-S. and Orabona, F. (2019). Parameter-free online convex optimization with sub-exponential noise. *Proceedings of the 32nd Annual Conference on Learning Theory (COLT)*, pages 1802–1823.

Jézéquel, R., Gaillard, P., and Rudi, A. (2020). Efficient improper learning for online logistic regression. In Abernethy, J. and Agarwal, S., editors, *Proceedings of 33rd Annual Conference on Learning Theory (COLT)*, volume 125 of *Proceedings of Machine Learning Research*, pages 2085–2108. PMLR.

Kakade, S. M., Shalev-Shwartz, S., and Tewari, A. (2008). Efficient bandit algorithms for online multiclass prediction. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*, pages 440–447.

Kalai, A. and Vempala, S. (2002). Efficient algorithms for universal portfolios. *Journal of Machine Learning Research*, 3(Nov):423–440.

Kasiviswanathan, S. P., Lee, H. K., Nissim, K., Raskhodnikova, S., and Smith, A. (2011). What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826.

Kempka, M., Kotlowski, W., and Warmuth, M. K. (2019). Adaptive scale-invariant online algorithms for learning linear models. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 3321–3330.

Kivinen, J. and Warmuth, M. K. (1997). Exponentiated Gradient versus Gradient Descent for linear predictors. *Information and Computation*, 132(1):1–63.

Kleinberg, R. and Leighton, T. (2003). The value of knowing a demand curve: Bounds on regret for online posted-price auctions. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pages 594–605. IEEE.

Kleinberg, R. D. (2005). Nearly tight bounds for the continuum-armed bandit problem. In *Advances in Neural Information Processing Systems 18*, pages 697–704.

Koolen, W. M. (2015). The relative entropy bound for Squint. *Blog August 13:* `http://blog.wouterkoolen.info/Squint_PAC/post.html`.

Koolen, W. M. (2016). Exploiting curvature using Exponential Weights. *Blog September 6:* `http://blog.wouterkoolen.info/EW4Quadratic/post.html`.

Koolen, W. M., Van Erven, T., and Grünwald, P. D. (2014). Learning the learning rate for prediction with expert advice. In *Advances in Neural Information Processing Systems 27*, pages 2294–2302.

Koolen, W. M., Grünwald, P., and Van Erven, T. (2016). Combining adversarial guarantees and stochastic fast rates in online learning. In *Advances in Neural Information Processing Systems 29*, pages 4457–4465.

Koolen, W. M. and Van Erven, T. (2015). Second-order quantile methods for experts and combinatorial games. In *Proceedings of the 28th Annual Conference on Learning Theory (COLT)*, pages 1155–1175.

Kotłowski, W. (2017). Scale-invariant unconstrained online learning. In *Proceedings of the 28th International Conference on Algorithmic Learning Theory (ALT)*, pages 412–433.

Krichevsky, R. and Trofimov, V. (1981). The performance of universal encoding. *IEEE Transactions on Information Theory*, 27(2):199–207.

Langford, J. and Zhang, T. (2008). The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in Neural Information Processing Systems 21*, pages 817–824.

Lattimore, T. (2015). The pareto regret frontier for bandits. In *Advances in Neural Information Processing Systems 28*, pages 208–216.

Lattimore, T. and Szepesvári, C. (2018). *Bandit algorithms*. Cambridge University Press.

Li, B. and Hoi, S. C. (2014). Online portfolio selection: A survey. *ACM Computing Surveys (CSUR)*, 46(3):1–36.

Littlestone, N. and Warmuth, M. K. (1994). The Weighted Majority algorithm. *Information and Computation*, 108(2):212–261.

Luo, H., Agarwal, A., Cesa-Bianchi, N., and Langford, J. (2017). Efficient second order online learning by sketching. *ArXiv preprint: arXiv:1602.02202*.

Luo, H., Wei, C.-Y., and Zheng, K. (2018). Efficient online portfolio with logarithmic regret. In *Advances in Neural Information Processing Systems 31*, pages 8235–8245.

Luo, L., Chen, C., Zhang, Z., Li, W.-J., and Zhang, T. (2019). Robust frequent directions with application in online learning. *Journal of Machine Learning Research*, 20(45):1–41.

Mcmahan, B. and Streeter, M. (2012). No-regret algorithms for unconstrained online convex optimization. In *Advances in Neural Information Processing Systems 25*, pages 2402–2410.

McMahan, H. B. and Orabona, F. (2014). Unconstrained online linear learning in hilbert spaces: Minimax algorithms and normal approximations. In *Proceedings of the 27th Annual Conference on Learning Theory (COLT)*, pages 1020–1039.

McMahan, H. B. and Streeter, M. (2010). Adaptive bound optimization for online convex optimization. In *Proceedings of the 23rd Annual Conference on Learning Theory (COLT)*, pages 244–256.

Mhammedi, Z. and Koolen, W. M. (2020). Lipschitz and comparator-norm adaptivity in online learning. In *Proceedings of the 33rd Annual Conference on Learning Theory (COLT)*, pages 2858–2887.

Mhammedi, Z., Koolen, W. M., and Van Erven, T. (2019). Lipschitz adaptivity with multiple learning rates in online learning. In Beygelzimer, A. and Hsu, D., editors, *Proceedings of the 32nd Annual Conference on Learning Theory (COLT)*, pages 2490–2511.

Mikolov, T., Chen, K., Corrado, G. S., and Dean, J. (2013). Efficient estimation of word representations in vector space. International Conference on Learning Representations. Arxiv.org/abs/1301.3781.

Narayanan, H. and Rakhlin, A. (2017). Efficient sampling from time-varying log-concave distributions. *The Journal of Machine Learning Research*, 18(1):4017–4045.

Nemirovski, A. (2004). Lecture notes: Interior point polynomial time methods in convex programming. *Spring Semester*.

Nesterov, Y. (2009). Primal-dual subgradient methods for convex problems. *Mathematical programming*, 120(1):221–259.

Nesterov, Y. and Nemirovskii, A. (1994). *Interior-point polynomial algorithms in convex programming*. SIAM.

Neu, G. and Zhivotovskiy, N. (2020). Fast rates for online prediction with abstention. In *Proceedings of the 33rd Annual Conference on Learning Theory (COLT)*, pages 3030–3048.

Neuteboom, T. W. H. (2020). Modifying Squint for prediction with expert advice in a changing environment. *Bachelor Thesis*. To appear at `https://www.universiteitleiden.nl/en/science/mathematics/education/theses#bachelor-theses-mathematics`.

Nielsen, F. and Nock, R. (2010). Entropies and cross-entropies of exponential families. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 3621–3624. IEEE.

Orabona, F. (2014). Simultaneous model selection and optimization through parameter-free stochastic learning. In *Advances in Neural Information Processing Systems 27*, pages 1116–1124.

Orabona, F., Cesa-Bianchi, N., and Gentile, C. (2012). Beyond logarithmic bounds in online learning. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 823–831.

Orabona, F., Crammer, K., and Cesa-Bianchi, N. (2015a). A generalized online mirror descent with applications to classification and regression. *Machine Learning*, 99(3):411–435.

Orabona, F., Crammer, K., and Cesa-Bianchi, N. (2015b). A generalized online mirror descent with applications to classification and regression. *Machine Learning*, 99(3):411–435.

Orabona, F. and Pál, D. (2016). Coin betting and parameter-free online learning. In *Advances in Neural Information Processing Systems 29*, pages 577–585.

Orabona, F. and Pál, D. (2018). Scale-free online learning. *Theoretical Computer Science*, 716:50–69.

Orabona, F. and Tommasi, T. (2017). Training deep networks without learning rates through coin betting. In *Advances in Neural Information Processing Systems 30*, pages 2160–2170.

Orseau, L., Lattimore, T., and Legg, S. (2017). Soft-bayes: Prod for mixtures of experts with log-loss. In *Proceedings of the 28th International Conference on Algorithmic Learning Theory (ALT)*, pages 372–399.

Rennie, J. D. and Srebro, N. (2005). Loss functions for preference levels: Regression with discrete ordered labels. In *Proceedings of the IJCAI multidisciplinary workshop on advances in preference handling*, volume 1, pages 180–186.

De Rooij, S., Van Erven, T., Grünwald, P. D., and Koolen, W. M. (2014). Follow the Leader if you can, Hedge if you must. *Journal of Machine Learning Research*, 15:1281–1316.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

Saha, A. and Tewari, A. (2011). Improved regret guarantees for online smooth convex optimization with bandit feedback. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 636–642.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117.

Shalev-Shwartz, S. (2011). Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194.

Shalev-Shwartz, S., Singer, Y., Srebro, N., and Cotter, A. (2011). Pegasos: Primal estimated sub-gradient solver for SVM. *Mathematical Programming*, 127(1):3–30.

Shamir, G. I. (2020). Logistic regression regret: What's the catch? In *Proceedings of the 33rd Annual Conference on Learning Theory (COLT)*, pages 3296–3319.

Song, S., Chaudhuri, K., and Sarwate, A. (2015). Learning from data with heterogeneous noise using sgd. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 894–902.

Srebro, N., Sridharan, K., and Tewari, A. (2010). Smoothness, low noise and fast rates. In *Advances in Neural Information Processing Systems 23*, pages 2199–2207.

Steinhardt, J. and Liang, P. (2014). Adaptivity and optimism: An improved exponentiated gradient algorithm. In *Proceedings of the 31th Annual International Conf. on Machine Learning (ICML)*, pages 1593–1601.

Thakurta, A. G. and Smith, A. (2013). (nearly) optimal algorithms for private online learning in full-information and bandit settings. In *Advances in Neural Information Processing Systems 26*, pages 2733–2741.

Van der Hoeven, D. (2019). User-specified local differential privacy in unconstrained adaptive online learning. In *Advances in Neural Information Processing Systems 32*, pages 14103–14112.

Van der Hoeven, D. (2020). Exploiting the surrogate gap in online multiclass classification. *To Appear in Advances in Neural Information Processing Systems 33*.

Van der Hoeven, D., Cutkosky, A., and Luo, H. (2020). Comparator-adaptive convex bandits. *To Appear in Advances in Neural Information Processing Systems 33*.

Van der Hoeven, D., Van Erven, T., and Kotłowski, W. (2018). The many faces of exponential weights in online learning. In *Proceedings of the 31st Annual Conference on Learning Theory (COLT)*, pages 2067–2092.

van Erven, T. and Koolen, W. M. (2016). Metagrad: Multiple learning rates in online learning. In *Advances in Neural Information Processing Systems 29*, pages 3666–3674.

Van Erven, T., Koolen, W. M., and Van der Hoeven, D. (2020a). Metagrad: Universal adaptation using multiple learning rates in online learning. *Manuscript in preparation.*

Van Erven, T., Van der Hoeven, D., Kotłowski, W., and Koolen, W. M. (2020b). Open problem: Fast and optimal online portfolio selection. In *Proceedings of the 33rd Annual Conference on Learning Theory (COLT)*, pages 3864–3869.

Vovk, V. (1990). Aggregating strategies. *Proceedings of 3rd Annual Workshop on Computational Learning Theory (COLT)*, page pages 371–383.

Vovk, V. and Zhdanov, F. (2009). Prediction with expert advice for the brier game. *Journal of Machine Learning Research*, 10(Nov):2445–2471.

Vovk, V. G. (2001). Competitive on-line statistics. *International Statistical Review*, 69(2):213–248.

Wasserman, L. and Zhou, S. (2010). A statistical framework for differential privacy. *Journal of the American Statistical Association*, 105(489):375–389.

Xiao, L. (2010). Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11:2543–2596.

Xie, Q. and Barron, A. R. (2000). Asymptotic minimax regret for data compression, gambling, and prediction. *IEEE Transactions on Information Theory*, 46(2):431–445.

Zhang, L., Wang, G., Tu, W., and Zhou, Z. (2019). Dual adaptivity: A universal algorithm for minimizing the adaptive regret of convex functions. *CoRR*, abs/1906.10851.

Zinkevich, M. (2003). Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, pages 928–936.

Zinkevich, M. (2004). *Theoretical Guarantees for Algorithms in Multi-Agent Settings*. PhD thesis, Carnegie Mellon University.

Samenvatting

Online Learning is een fundamentele machine learning setting waarin een leerder sequentieel voorspellingen moet doen gegeven (partiële) informatie over voorgaande correcte voorspellingen en mogelijk extra informatie. Over de omgeving van de leerder wordt vaak aangenomen dat het een vijandige omgeving is die de taak van de leerder, zo min mogelijk verlies lijden, zo moeilijk mogelijk maakt. Desalniettemin zijn er in de afgelopen drie decennia veel Online Learning algoritmes ontwikkeld die bevredigende garanties bieden in verschillende settings. De garanties van Online Learning algoritmes gaan over spijt: het verschil tussen het cumulatieve verlies van de leerder en het cumulatieve verlies van een offline optimizer van het cumulatieve verlies, waar de offline optimizer ook wel bekend staat als de vergelijker. In dit proefschrift presenteren wij verschillende nieuwe inzichten in verschillende settings van Online Learning. Vandaar ook de titel van het proefschrift: de vele gezichten van Online Learning.

In Hoofdstuk 2 bestuderen we een van de fundamenteelste algoritmes in Online Learning: Exponential Weights. We laten zien hoe Exponential Weights moet worden afgestemd zodat het in verschillende settings kan worden toegepast en we laten tevens zien hoe met specifieke keuzes voor de parameters verschillende belangrijke algoritmes worden teruggevonden als een speciaal geval van Exponential Weights. Dit inzicht leidt tot een gecentraliseerd begrip van verscheidene algoritmes in Online Learning en verenigt de analyse van deze algoritmes.

Een belangrijk onderscheid in Online Learning is het verschil tussen de volledige-informatie en bandit settings. In de volledige-informatie setting onthult de omgeving de gehele verliesfunctie, maar in de meer uitdagende bandit setting onthult de omgeving enkel partiële informatie. Een belangrijke eigenschap van veel algoritmes in zowel de volledige-informatie setting als de bandit setting is dat ze gepaste grenzen op de spijt garanderen, zelfs in een vijandige omgeving. Echter worden deze algoritmes dusdanig afgesteld dat ze enkel met een vijandige omgeving om kunnen gaan en niet meer goedaardige omgevingen kunnen uitbuiten. Een van de hoofdonderwerpen in dit proefschrift is hoe men algoritmes kan ontwerpen die zowel in een vijandige omgeving als in een goedaardige omgeving bevredigende garanties bieden, zonder dat van tevoren bekend is in wat voor omgeving de leerder zich bevindt. Omdat deze algoritmes zich aanpassen aan de omgeving staan dit soort algoritmes bekend als adaptieve algoritmes. In Hoofdstuk 2 laten we zien hoe we verschillende adaptieve algoritmes terugvinden als een speciaal geval van Exponential Weights. In Hoofdstukken 3 en 4 bestuderen we een speciaal soort adaptief algoritme, namelijk vergelijker-adaptieve algoritmes. De spijt grens van een vergelijker-adaptief algoritme hangt af van de norm van de vergelijker, wat in sommige gevallen tot minder spijt kan leiden vergeleken met standaard algoritmes. In Hoofdstuk 3 laten we zien hoe we vergelijker-adaptieve algoritmes kunnen aanpassen zodat ze adaptief zijn aan onbekende ruis. Dit is nuttig wanneer mensen willen kiezen hoeveel privacy zij hebben maar dit niet willen laten weten. In Hoofdstuk 4 ontwikkelen wij de eerste vergelijker-adaptieve algoritmes in de Bandit Convex Optimisation setting. In Hoofdstuk 5 presenteren wij MetaGrad, dat zich kan aanpassen aan een grote klasse van functies.

Omdat de leerder in elke ronde zijn voorspellingen update is een belangrijke eigenschap van een Online Learning algoritme de looptijd. De per ronde looptijd wordt vaak als te hoog gezien wanneer de updates meer tijd in beslag nemen dan kwadratisch in de dimensie van het probleem. Hierdoor is er een aanzienlijke hoeveelheid werk verzet door verschillende auteurs om de looptijd van Online Learning algoritmes te verminderen. Ook in dit proefschrift is er aandacht besteed aan het versnellen van Online Learning algoritmes. In Hoofdstuk 6 presenteren we een nieuw algoritme voor de Online Multiclass Classification setting die vaak vergelijkbare of betere garanties bied op de spijt dan tragere algoritmes. In deze setting moet de leerder in elke ronde een label voorspellen gegeven een d-dimensionale feature vector die mogelijk extra informatie bevat. In de Online Multiclass Classification setting lijdt de leerder de één-nul verliesfunctie. De één-nul verliesfunctie is één wanneer de voorspelling van de leerder incorrect is en nul wanneer de voorspelling van de leerder correct is. De benchmark in de Online Multiclass Classification setting is een convexe surrogaat verliesfunctie. Deze surrogaat verliesfunctie is een bovengrens op de één-nul verliesfunctie. Het doel van de leerder in de Online Multiclass Classification setting is de surrogaat spijt te minimaliseren: het verschil tussen de som van de één-nul verliesfuncties en het offline minimum van de som van de surrogaat verliesfuncties. Voorgaande algoritmes in de Online Multiclass Classification setting bouwden vaak voort op tweede-orde algoritmes om lage spijt te garanderen. Tweede-orde algoritmes houden een d bij d matrix bij die elke ronde wordt geupdate, wat de per ronde looptijd minstens d^2 maakt. Wij introduceren een nieuw algoritme genaamd GAPTRON dat een per ronde looptijd heeft van $O(d)$. GAPTRON heeft vaak een vergelijkbare of zelfs betere garantie op de spijt dan tragere algoritmes. Bijvoorbeeld, in de Bandit Online Multiclass Classification setting is de surrogaat spijt bovengrens van GAPTRON een factor $\sqrt{d}$ kleiner dan de surrogaat spijt bovengrens van langzamere algoritmes. We behalen deze resultaten door het gat tussen de één-nul verliesfunctie en de surrogaat verliesfunctie uit te buiten. Door deze nieuwe aanpak kan de leerder gebruik maken van eerste-orde algoritmes om zijn voorspellingen te updaten en tegelijkertijd kleine surrogaat spijt te garanderen.

Verdere verbeteringen van looptijd worden gemaakt in Hoofdstuk 5. In Hoofdstuk 5 laten we zien hoe de looptijd van MetaGrad kan worden verlaagd met behulp van sketching methoden. In Hoofdstuk 7 bestuderen we de online portfolio selectie setting. Het optimale algoritme voor de online portfolio selectie setting is een versie van Exponential Weights. Helaas is de looptijd van deze specifieke versie van Exponential Weights te groot om als praktisch te worden beschouwd. Voor de online portfolio selectie setting zijn veel andere algoritmes overwogen, maar allemaal hebben ze tekortkomingen. We stellen een open probleem waarin we vragen om een snel en optimaal algoritme. Wij geven een gedeeltelijke analyse van wat wij denken dat een snel en optimaal algoritme is. Hierbij laten we zien dat het algoritme wat wij voorstellen inderdaad de optimale bovengrens op de spijt behaalt in specifieke gevallen.

Summary

Online Learning is a fundamental machine learning setting in which a learner is to sequentially issue predictions given some (partial) knowledge about previous correct predictions and possibly additional information. The environment of the learner is often assumed to be adversarial, making the learner's task of suffering as little loss as possible difficult. Nevertheless, in the last three decades many different Online Learning algorithms have been successfully shown to provide satisfying guarantees in various settings. The guarantees in Online Learning are about regret, which is the difference between the cumulative loss of the learner and the cumulative loss the offline optimizer of the loss, which is also known as the comparator. In this dissertation we provide several new insights in many different settings of Online Learning, hence the title of the dissertation.

In Chapter 2 we study one of the most fundamental algorithms in Online Learning: Exponential Weights. We show how to tune Exponential Weights such that it can be applied to several different settings and show that with specific parameter choices we recover several other important algorithms in Online Learning as special cases of Exponential Weights. This provides a centralized understanding of many algorithms in the Online Learning setting and unifies the analysis of these algorithms.

An important distinction in Online Learning is between the full-information and bandit settings. In the full-information setting the environment reveals all information to the learner but in the more challenging bandit setting the environment only reveals partial information. An important property of many algorithms in both the full-information and bandit settings is that they are able to provide suitable regret bounds even in adversarial environments. However, these algorithms are often tuned to only deal with adversarial environments and are not able to exploit more benign environments. A recurring subject in this dissertation is how to design algorithms that are able to exploit benign environments but also provide suitable guarantees in adversarial environments, without knowing what type of environment the learner faces beforehand. These algorithms are known as adaptive algorithms as they adapt to the environment. In Chapter 2 we show how we can recover several adaptive algorithms as special cases of Exponential Weights. In Chapters 3 and 4 we study a particular type of adaptive algorithms, namely comparator-adaptive algorithms. The regret bounds of comparator-adaptive algorithms depend on properties of the offline

minimizer of the loss, which in some cases can lead to smaller regret compared to the regret of standard algorithms. In Chapter 3 we show how we can modify comparator-adaptive algorithms to adapt to unknown noise, which is useful when people want to choose how much privacy they have without disclosing how much they value their privacy. Additionally, when the losses are nice in a particular sense we show that our modified comparator-adaptive algorithm has low regret. In Chapter 4 we provide the first comparator-adaptive algorithms for the Bandit Convex Optimization setting. This can be especially advantageous when the comparator is small when measured in a particular norm, as this leads to smaller regret bounds compared to non-adaptive algorithms. In Chapter 5 we present MetaGrad, which adapts to a broad class of functions. The class of functions to which MetaGrad is adaptive includes exp-concave losses, losses with unknown lipschitz constants, and various other types of stochastic or non-stochastic functions.

Since the learner updates his prediction in each round an important property of Online Learning algorithms is the running time. The per round running time is often considered too high to be practical whenever the updates take more than quadratic time in the dimension of the problem. Because of this reason considerable effort has been made to improve the running time of many Online Learning algorithms, including in this dissertation. In Chapter 6 provide a new algorithm with often similar or better guarantees than slower algorithms for the Online Multiclass Classification setting. In this setting in each round the learner has to predict a label given a d-dimensional feature vector which contains additional information. In the Online Multiclass Classification setting the learner suffers the zero-one loss, which is one whenever the learner makes a mistake and zero whenever he correctly predicts the label. The benchmark in the Online Multiclass Classification setting is a convex surrogate loss which upper bounds the zero-one loss and the goal of the learner is to minimize the surrogate regret: the difference between the sum of the zero-one losses and the offline minimum of the sum of the surrogate losses. Previous algorithms in the Online Multiclass Classification setting often relied on second-order algorithms to guarantee small regret. Second-order algorithms keep track of a d by d matrix of parameters, which the learner updates in each round, making the per round running time at least d^2. We introduce a novel algorithm called GAPTRON which has a per round running time of order $O(d)$. Surprisingly, GAPTRON often matches or improves upon the guarantees of slower algorithms. For example, in the Bandit Multiclass Classification setting the surrogate regret bound of GAPTRON is a factor $\sqrt{d}$ smaller than slower algorithms. We achieve our results by using a new approach which exploits the gap between the zero-one loss and a surrogate loss. This new approach allows the learner to use a linear time algorithm to update the predictions while still obtaining small surrogate regret bounds. Other improvements in running time of algorithms are made in Chapter 5, in which we show how to improve the running time of the aforementioned adaptive algorithm MetaGrad by using sketching methods. In Chapter 7 we consider online portfolio selection. The optimal algorithm for online portfolio selection is a version of Exponential Weights. Unfortunately the running time of this particular version of Exponential Weights is too high to be considered practical. Many different algorithms for online portfolio selection have been considered, all of them with different shortcomings. We pose an open problem which asks for a fast and optimal algorithm and provide the first steps of the analysis of an algorithm we think is the answer to the open problem. We then show that in particular cases the proposed algorithm indeed yields the optimal regret bound.

Acknowledgements

Allereerst wil ik mijn supervisor Tim van Erven bedanken. Van het begin van mijn masterscriptie tot het einde van mijn PhD heeft Tim mij geadviseerd en onderwezen over alle mogelijke facetten van het leven als wetenschapper. Zonder Tim's geduld en onderwijstalent had ik nooit mijn vaardigheden als onderzoeker kunnen ontwikkelen zoals ik dat de afgelopen jaren heb gedaan. Hoewel Tim als onderwijzer heel geduldig is, is hij ook een zeer strenge reviewer. De afgelopen jaren had ik, in mijn ogen, verschillende fantastische onderzoeksideeën die na enkele bedenkelijke blikken en kritische vragen van Tim altijd een stuk minder fantastisch waren. Een groot pluspunt aan dit proces is ook dat, zodra ik Tim had overtuigd, ik vol vertrouwen een paper kon insturen: van de strengste reviewer had ik namelijk al goedkeuring gekregen. Naderhand ben ik uitzonderlijk dankbaar voor de blikken en vragen. Hoewel ik er op die momenten niet blij mee was, ben ik er van overtuigd dat deze interacties mij tot een betere onderzoeker hebben gemaakt. Waren de blikken en vragen mij onthouden gebleven had ik niet zelfstandig kritisch leren nadenken en ik denk dat dit een heel belangrijke stap is om een volwaardig wetenschapper te worden.

Ook ben ik zeer dankbaar voor Tim's aandacht voor de menselijke kant van het werk. Tim heeft nooit een kans gemist om mij te steunen als persoon. Wanneer ik teleurgesteld was, bijvoorbeeld na een kritische vergadering, gaf hij bemoedigende woorden. Wanneer er iets te vieren viel spoorde hij me aan om het ervan te nemen. En wanneer ik vastzat was hij er altijd om te helpen. Ik hoop dat ik op de goede weg zit om mezelf te ontwikkelen tot een kritische, ondersteunende, en vriendelijke onderwijzer van een kaliber als Tim.

I would also like to thank my coauthors for their patience and guidance during the writing of our papers. Insight into the way my coauthors work and think was incredibly educational and will shape my future career. In particular I would like to thank Haipeng Luo, whom I visited at the University of Southern California during the summer of 2019. Haipeng educated me in the world of bandits and showed me a different strategy than I was used to in Leiden to attack research problems. These lessons and the other experiences I had while visiting Haipeng will be with me during my career and I am grateful for Haipeng's willingness to welcome me in Los Angeles.

Verder wil ik graag mijn collega's op het Mathematisch Instituut bedanken. Mijn kantoorgenoten Anja, Carlo (never forget), Lasse, en Magnus ben ik erg dankbaar voor de gezelligheid en hulp. Ook het Mathematisch Instituut zelf en haar vertegenwoordigers ben ik erg dankbaar, vooral voor de steun gedurende de laatste maanden van mijn PhD.

Ik wil ook graag mijn vrienden en familie bedanken voor hun eindeloze steun gedurende mijn PhD. Mijn ouders en broertje voor hun geloof in mijn kunnen. Ik denk dat dit geloof essentieel is geweest om mijn onderzoek uit te kunnen voeren. Ook wil ik graag Remco en Wouter bedanken voor het luisteren naar verhalen over zwabberende stochasten en gaten tussen (surrogaat) verliesfuncties. Naar alle andere vrienden en familie gaat eveneens veel dank uit voor het geduldig aanhoren van onderzoeksverhalen.

Als laatste, maar zeer zeker niet de minste, wil ik graag Stefanie bedanken voor haar liefde en steun gedurende de afgelopen jaren. Mede dankzij Stefanie ben ik de laatste zware maanden van mijn PhD goed doorgekomen en heb ik me de eerste paar jaar van mijn wetenschappelijke carrière kunnen ontwikkelen.

Curriculum Vitae

Dirk van der Hoeven was born on April 8th 1992 in Leidschendam. After he completed his pre-university education at the Veurs Lyceum in 2010 he started his education in Leiden, where he would study and work for the coming decade. The first three years in Leiden were spent studying psychology in which he obtain his bachelor of science. In the subsequent year he obtained his first master of science degree, which is in psychology: methodology and statistics. During this first master education he spent half a year working as an intern at Unilever Research and Development in Vlaardingen, where he developed a new statistical test for the temporal dominance of sensation methodology. After completing his first master degree he continued his education by relocating to the Mathematical Institute in Leiden to study the master of science program statistics for the life and behavioral sciences. To complete his second master degree he wrote his thesis with dr. Tim van Erven, which sparked his interest in Online Learning and this master thesis was the basis for Chapter 2. Both master of science degrees were obtained *cum laude*. While working to obtain his master degrees he worked as a team leader at Jumbo supermarkets in Leidschendam and Scheveningen, caregiver at elderly center Prinsenhof in Leidschendam, and teaching assistant for various master and bachelor courses at Leiden University. While working on his second master thesis he also worked on establishing the education committee for the master program statistics for the life and behavioral sciences.

The last four years in Leiden Dirk worked as a PhD candidate under the supervision of dr. Tim van Erven, which resulted in the present dissertation. During these four years he was a teaching assistant for the course Statistical Learning Theory and was a lecturer of Statistics at Leiden University College (Den Haag). In the third year of his PhD he spent two months in Los Angeles, United States of America, to visit Haipeng Luo at the University of Southern California. The work done there resulted in Chapter 4.