JMLR: Workshop and Conference Proceedings vol 30 (2013) 1–18

Passive Learning with Target Risk

Mehrdad Mahdavi MAHDAVIM@CSE.MSU.EDU
Rong Jin RONGJIN@CSE.MSU.EDU
Department of Computer Science and Engineering
Michigan State University
East Lansing, MI 48824, USA

Abstract

In this paper we consider learning in passive setting but with a slight modification. We assume that the target expected loss, also referred to as *target risk*, is provided in advance for learner as prior knowledge. Unlike most studies in the learning theory that only incorporate the prior knowledge into the generalization bounds, we are able to explicitly utilize the target risk in the learning process. Our analysis reveals a surprising result on the sample complexity of learning: by exploiting the target risk in the learning algorithm, we show that when the loss function is both strongly convex and smooth, the sample complexity reduces to $\mathcal{O}(\log\left(\frac{1}{\epsilon}\right))$, an exponential improvement compared to the sample complexity $\mathcal{O}(\frac{1}{\epsilon})$ for learning with strongly convex loss functions. Furthermore, our proof is constructive and is based on a computationally efficient stochastic optimization algorithm for such settings which demonstrate that the proposed algorithm is practically useful.

Keywords: learning theory, risk minimization, stochastic optimization, sample complexity

1. Introduction

In the standard passive supervised learning setting, the learning algorithm is given a set of labeled examples $\mathcal{S} = ((\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n))$ drawn i.i.d. from a fixed but unknown distribution $\mathcal{D}$. The goal, with the help of labeled examples, is to output a classifier h from a predefined hypothesis class $\mathcal{H}$ that does well on unseen examples coming from the same distribution. The sample complexity of an algorithm is the number of examples which is sufficient to ensure that, with probability at least $1 - \delta$ (w.r.t. the random choice of $\mathcal{S}$), the algorithm picks a hypothesis with an error that is at most ϵ from the optimal one. Sample complexity of passive learning is well established and goes back to early works in the learning theory where the lower bounds $\Omega\left(\frac{1}{\epsilon}(\log\frac{1}{\epsilon} + \log\frac{1}{\delta})\right)$ and $\Omega\left(\frac{1}{\epsilon^2}(\log\frac{1}{\epsilon} + \log\frac{1}{\delta})\right)$ were obtained in classic PAC and general agnostic PAC settings, respectively (Ehrenfeucht et al., 1989; Blumer et al., 1989; Anthony and Bartlett, 1999).

In light of no free lunch theorem, learning is impossible unless we make assumptions regarding the nature of the problem at hand. Therefore, when approaching a particular learning problem, it is desirable to take into account some prior knowledge we might have about our problem and use a specialized algorithm that exploits this knowledge into a learning process or theoretical analysis. A key issue in this regard is the formalization of prior knowledge. Such prior knowledge can be expressed by restricting our hypothesis class, making assumptions on the nature of unknown distribution $\mathcal{D}$ or formalization of the data

space, analytical properties of the loss function being used to evaluate the performance, sparsity, and margin– to name a few.

There has been an upsurge of interest over the last decade in finding tight upper bounds on the sample complexity by utilizing prior knowledge on the analytical properties of the loss function, that led to stronger generalization bounds in agnostic PAC setting. In (Lee et al., 1998) *fast* rates obtained for squared loss, exploiting the strong convexity of this loss function, which only holds under pseudo-dimensionality assumption. With the recent development in online strongly convex optimization (Hazan et al., 2006), fast rates approaching $\mathcal{O}(\frac{1}{\epsilon}\log\frac{1}{\delta})$ for convex Lipschitz strongly convex loss functions has been obtained in (Sridharan et al., 2008; Kakade et al., 2008). For smooth non-negative loss functions, (Srebro et al., 2010) improved the sample complexity to *optimistic* rates

$$\mathcal{O}\left(\frac{1}{\epsilon}\left(\frac{\epsilon_{\text{opt}}+\epsilon}{\epsilon}\right)\left(\log^3\frac{1}{\epsilon}+\log\frac{1}{\delta}\right)\right)$$

for non-parametric learning using the notion of local Rademacher complexity (Bartlett et al., 2005), where ϵ_{opt} is the optimal risk.

In this work, we consider a slightly different setup for passive learning. We assume that before the start of the learning process, the learner has in mind a *target expected loss*, also referred to as *target risk*, denoted by ϵ_{prior}[1], and tries to learn a classifier with the expected risk of $O(\epsilon_{\text{prior}})$ by labeling a small number of training examples. We further assume the target risk ϵ_{prior} is feasible, i.e., $\epsilon_{\text{prior}} \geq \epsilon_{\text{opt}}$. To address this problem, we develop an efficient algorithm, based on stochastic optimization, for passive learning with target risk. The most surprising property of the proposed algorithm is that when the loss function is both smooth and strongly convex, it only needs $\mathcal{O}(d\log(1/\epsilon_{\text{prior}}))$ labeled examples to find a classifier with the expected risk of $O(\epsilon_{\text{prior}})$, where d is the dimension of data. This is a significant improvement compared to the sample complexity for empirical risk minimization.

The key intuition behind our algorithm is that by knowing target risk as prior knowledge, the learner has better control over the variance in stochastic gradients, which contributes mostly to the slow convergence in stochastic optimization and consequentially large sample complexity in passive learning. The trick is to run the stochastic optimization in multi-stages with a *fixed* size and decrease the variance of stochastically perturbed gradients at each iteration by a properly designed mechanism. Another crucial feature of the proposed algorithm is to utilize the target risk ϵ_{prior} to gradually refine the hypothesis space as the algorithm proceeds. Our algorithm differs significantly from standard stochastic optimization algorithms and is able to achieve a geometric convergence rate with the knowledge of target risk ϵ_{prior}.

We note that our work does not contradict the lower bound in (Srebro et al., 2010) because a *feasible* target risk ϵ_{prior} is given in our learning setup and is fully exploited by the proposed algorithm. Knowing that the target risk ϵ_{prior} is feasible makes it possible to improve the sample complexity from $\mathcal{O}(1/\epsilon_{\text{prior}})$ to $\mathcal{O}(\log(1/\epsilon_{\text{prior}}))$. We also note that although the logarithmic sample complexity is known for active learning (Hanneke, 2009; Balcan et al., 2010), we are unaware of any existing passive learning algorithm that is able to achieve a logarithmic sample complexity by incorporating any kind of prior knowledge.

1. We use ϵ_{prior} instead of ϵ to emphasize the fact that this parameter is known to the learner in advance.

1.1. More Related Work

Stochastic Optimization and Learnability Our work is related to the recent studies that examined the learnability from the viewpoint of stochastic convex optimization. In (Sridharan, 2012; Shalev-Shwartz et al., 2010), the authors presented learning problems that are learnable by stochastic convex optimization but not by empirical risk minimization (ERM). Our work follows this line of research. The proposed algorithm achieves the sample complexity of $O(d \log(1/\epsilon_{\text{prior}}))$ by explicitly incorporating the target expected risk ϵ_{prior} into the stochastic convex optimization algorithm. It is however difficult to incorporate such knowledge into the framework of ERM. Furthermore, it is worth noting that in (Ramdas and Singh, 2013; Sridharan, 2012; Rakhlin et al., 2010; Ben-David et al., 2009), the authors explored the connection between online optimization and statistical learning in the opposite direction. This was done by exploring the complexity measures developed in statistical learning for the learnability of online learning.

Online and Stochastic Optimization The proposed algorithm is closely related to the recent works that stated $O(1/n)$ is the optimal convergence rate for stochastic optimization when the objective function is strongly convex (Iouditski and Nesterov, 2010; Hazan and Kale, 2011; Rakhlin et al., 2012). In contrast, the proposed algorithm is able to achieve a geometric convergence rate for a target optimization error. Similar to the previous argument, our result does not contradict the lower bound given in (Hazan and Kale, 2011) because of the knowledge of a feasible optimization error. Moreover, in contrast to the multistage algorithm in (Hazan and Kale, 2011) where the size of stages increases exponentially, in our algorithm, the size of each stage is fixed to be a constant.

Outline The remainder of the paper is organized as follows: In Section 2, we set up notation, describe the setting, and discuss the assumptions on which our algorithm relies. Section 3 motivates the problem and discusses the main intuition of our algorithm. The proposed algorithm and main result are discussed in Section 4. We prove the main result in Section 5. Section 6 concludes the paper and the appendix contains the omitted proofs.

2. Preliminaries

As usual in the framework of statistical learning theory, we consider a domain $\mathcal{Z} := \mathcal{X} \times \mathcal{Y}$ where $\mathcal{X} \subseteq \mathbb{R}^d$ is the space for instances and $\mathcal{Y}$ is the set of labels, and $\mathcal{H}$ is a hypothesis class. We assume that the domain space $\mathcal{Z}$ is endowed with an unknown Borel probability measure $\mathcal{D}$. We measure the performance of a specific hypothesis h by defining a nonnegative loss function $\ell : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}_+$. We denote the risk of a hypothesis h by $\mathcal{L}(h) = \mathbb{E}_{\mathbf{z}\sim\mathcal{D}}[\ell(h, \mathbf{z})]$. Given a sample $\mathcal{S} = (\mathbf{z}_1, \cdots, \mathbf{z}_n) = ((\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)) \sim \mathcal{D}^n$, the goal of a learning algorithm is to pick a hypothesis $h : \mathcal{X} \to \mathcal{Y}$ from $\mathcal{H}$ in such a way that its risk $\mathcal{L}(h)$ is close to the minimum possible risk of a hypothesis in $\mathcal{H}$.

Throughout this paper we pursue stochastic optimization viewpoint for risk minimization as detailed in Section 3. Precisely, we focus on the convex learning problems for which we assume that the hypothesis class $\mathcal{H}$ is a parametrized convex set $\mathcal{H} = \{h_{\mathbf{w}} : \mathbf{x} \mapsto \langle \mathbf{w}, \mathbf{x} \rangle : \mathbf{w} \in \mathbb{R}^d, \|\mathbf{w}\| \leq R\}$ and for all $\mathbf{z} = (\mathbf{x}, y) \in \mathcal{Z}$, the loss function $\ell(\cdot, \mathbf{z})$ is a non-negative convex function. Thus, in the remainder we simply use vector $\mathbf{w}$ to represent $h_{\mathbf{w}}$, rather than working with hypothesis $h_{\mathbf{w}}$. We will assume throughout that $\mathcal{X} \subseteq \mathbb{R}^d$ is the unit

ball so that $\|\mathbf{x}\| \leq 1$. Finally, the conditions under which we can get the desired result on sample complexity depend on analytic properties of the loss function. In particular, we assume that the loss function is strongly convex and smooth (Nesterov, 2004).

Definition 1 (Strong convexity) *A loss function $\ell(\mathbf{w})$ is said to be α-strongly convex w.r.t a norm $\|\cdot\|$[2], if there exists a constant $\alpha > 0$ (often called the modulus of strong convexity) such that, for any $\lambda \in [0,1]$ and for all $\mathbf{w}_1, \mathbf{w}_2 \in \mathcal{H}$, it holds that*

$$\ell(\lambda \mathbf{w}_1 + (1-\lambda)\mathbf{w}_2) \leq \alpha \ell(\mathbf{w}_1) + (1-\lambda)\ell(\mathbf{w}_2) - \frac{1}{2}\lambda(1-\lambda)\alpha\|\mathbf{w}_1 - \mathbf{w}_2\|^2.$$

When $\ell(\mathbf{w})$ is differentiable, the strong convexity is equivalent to

$$\ell(\mathbf{w}_1) \geq \ell(\mathbf{w}_2) + \langle \nabla \ell(\mathbf{w}_2), \mathbf{w}_1 - \mathbf{w}_2 \rangle + \frac{\alpha}{2}\|\mathbf{w}_1 - \mathbf{w}_2\|^2, \ \forall\, \mathbf{w}_1, \mathbf{w}_2 \in \mathcal{H}.$$

We would like to emphasize that in our setting, we only need that the expected loss function $\mathcal{L}(\mathbf{w})$ be strongly convex, without having to assume strong convexity for individual loss functions.
Another property of loss function that underline our analysis is its smoothness. Smooth functions arise, for instance, in logistic and least-squares regression, and in general for learning linear predictors where the loss function has a Lipschitz-continuous gradient.

Definition 2 (Smoothness) *A differentiable loss function $\ell(\mathbf{w})$ is said to be β-smooth with respect to a norm $\|\cdot\|$, if it holds that*

$$\ell(\mathbf{w}_1) \leq \ell(\mathbf{w}_2) + \langle \nabla \ell(\mathbf{w}_2), \mathbf{w}_1 - \mathbf{w}_2 \rangle + \frac{\beta}{2}\|\mathbf{w}_1 - \mathbf{w}_2\|^2, \ \forall\, \mathbf{w}_1, \mathbf{w}_2 \in \mathcal{H}. \tag{1}$$

3. The Curse of Stochastic Oracle

We begin by discussing stochastic optimization for risk minimization, convex learnability, and then the main intuition that motivates this work.

Most existing learning algorithms follow the framework of empirical risk minimizer (ERM) or regularized ERM, which was developed to great extent by Vapnik and Chervonenkis (Vapnik and Chervonenkis, 1971). Essentially, ERM methods use the empirical loss over $\mathcal{S}$, i.e., $\widehat{\mathcal{L}}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{w}, \mathbf{z}_i)$, as a criterion to pick a hypothesis. In regularized ERM methods, the learner picks a hypothesis that jointly minimizes $\widehat{\mathcal{L}}(\mathbf{w})$ and a regularization function over $\mathbf{w}$. We note that ERM resembles the widely used Sample Average Approximation (SAA) method in the optimization community when the hypothesis space and the loss function are convex. If uniform convergence holds, then the empirical risk minimizer is consistent, i.e., the population risk of the ERM converges to the optimal population risk, and the problem is learnable using ERM.

A rather different paradigm for risk minimization is stochastic optimization. Recall that the goal of learning is to approximately minimize the risk $\mathcal{L}(\mathbf{w}) = \mathbb{E}_{\mathbf{z}\sim\mathcal{D}}[\ell(\mathbf{w}, \mathbf{z})]$. However, since the distribution $\mathcal{D}$ is unknown to the learner, we can not utilize standard gradient methods to minimize the expected loss. Stochastic optimization methods circumvent this

2. Throughout this paper, we only consider the ℓ_2-norm.

problem by allowing the optimization method to take a step which is only in expectation along the negative of the gradient. To motivate stochastic optimization as an alternative to the ERM method, (Shalev-Shwartz et al., 2009b,a) challenged the ERM method and showed that there is a real gap between learnability and uniform convergence by investigating non-trivial problems where no uniform convergence holds, but they are still learnable using Stochastic Gradient Descent (SGD) algorithm (Nemirovski et al., 2009). These results uncovered an important relationship between learnability and stability, and showed that stability together with approximate empirical risk minimization, assures learnability (Shalev-Shwartz et al., 2010). We note that Lipschitzness or smoothness of loss function is necessary for an algorithm to be stable, and boundedness and convexity alone are not sufficient for ensuring that the convex learning problem is learnable.

To directly solve $\min_{\mathbf{w}\in\mathcal{H}} \mathcal{L}(\mathbf{w}) = \mathbb{E}_{\mathbf{z}\sim\mathcal{D}}[\ell(\mathbf{w},\mathbf{z})]$, a typical stochastic optimization algorithm initially picks some point in the feasible set $\mathcal{H}$ and iteratively updates these points based on first order perturbed gradient information about the function at those points. For instance, the widely used SGD algorithm starts with $\mathbf{w}_0 = \mathbf{0}$; at each iteration t, it queries the stochastic oracle ($\mathcal{SO}$) at $\mathbf{w}_t$ to obtain a perturbed but unbiased gradient $\hat{\mathbf{g}}_t$ and updates the current solution by

$$\mathbf{w}_{t+1} = \Pi_{\mathcal{H}}\left(\mathbf{w}_t - \eta_t \hat{\mathbf{g}}_t\right),$$

where $\Pi_{\mathcal{H}}(\mathbf{w})$ projects the solution $\mathbf{w}$ into the domain $\mathcal{H}$. To capture the efficiency of optimization procedures in a general sense, one can use oracle complexity of the algorithm which, roughly speaking, is the minimum number of calls to any oracle needed by any method to achieve desired accuracy (Nesterov, 2004). We note that the oracle complexity corresponds to the sample complexity of learning from the stochastic optimization viewpoint previously discussed. The following theorem states a lower bound on the sample complexity of stochastic optimization algorithms (Nemirovsky and Yudin, 1983).

Theorem 3 (Lower Bound on Oracle Complexity) *Suppose $\mathcal{L}(\mathbf{w}) = \mathbb{E}_{\mathbf{z}\sim\mathcal{D}}[\ell(\mathbf{w},\mathbf{z})]$ is α-strongly and β-smooth convex function defined over convex domain $\mathcal{H}$. Let $\mathcal{SO}$ be a stochastic oracle that for any point $\mathbf{w} \in \mathcal{H}$ returns an unbiased estimate $\hat{\mathbf{g}}$, i.e., $\mathbb{E}[\hat{\mathbf{g}}] = \nabla\mathcal{L}(\mathbf{w})$, such that $\mathbb{E}\left[\|\hat{\mathbf{g}} - \nabla\mathcal{L}(\mathbf{w})\|^2\right] \leq \sigma^2$ holds. Then for any stochastic optimization algorithm $\mathcal{A}$ to find a solution $\widehat{\mathbf{w}}$ with ϵ accuracy respect to the optimal solution $\mathbf{w}_*$, i.e., $\mathbb{E}\left[\mathcal{L}(\widehat{\mathbf{w}}) - \mathcal{L}(\mathbf{w}_*)\right] \leq \epsilon$, the number of calls to $\mathcal{SO}$ is lower bounded by*

$$\mathcal{O}(1)\left(\sqrt{\frac{\beta}{\alpha}}\log\left(\frac{\beta\|\mathbf{w}_0 - \mathbf{w}_*\|^2}{\epsilon}\right) + \frac{\sigma^2}{\alpha\epsilon}\right). \tag{2}$$

The first term in (2) comes from deterministic oracle complexity and the second term is due to noisy gradient information provided by $\mathcal{SO}$. As indicated in (2), the slow convergence rate for stochastic optimization is due to the variance in stochastic gradients, leading to at least $\mathcal{O}\left(\sigma^2/\epsilon\right)$ queries to be issued. We note that the idea of mini-batch (Cotter et al., 2011; Duchi et al., 2012), although it reduces the variance in stochastic gradients, does not reduce the oracle complexity.

We close this section by informally presenting why logarithmic sample complexity is, in principle, possible, under the assumption that target risk is known to the learner $\mathcal{A}$. To this end, consider the setting of Theorem 3 and assume that the learner $\mathcal{A}$ is given the prior accuracy ϵ_{prior} and is asked to find an ϵ_{prior}-accurate solution. If it happens that the variance of $\mathcal{SO}$ has the same magnitude as ϵ_{prior}, i.e., $\mathbb{E}\left[\|\hat{\mathbf{g}} - \nabla\mathcal{L}(\mathbf{w})\|^2\right] \leq \epsilon_{\text{prior}}$, then from (2) it follows that the second term vanishes and the learner $\mathcal{A}$ needs to issue only $\mathcal{O}\left(\log 1/\epsilon_{\text{prior}}\right)$ queries to find the solution. But, since there is no control on $\mathcal{SO}$, except that the variance of stochastic gradients are bounded, $\mathcal{A}$ needs a mechanism to manage the variance of perturbed gradients at each iteration in order to alleviate the influence of noisy gradients. One strategy is to replace the unbiased estimate of gradient with a biased one, which unfortunately may yield loose bounds. To overcome this problem, we introduce a strategy that shrinks the solution space with respect to the target risk ϵ_{prior} to control the damage caused by biased estimates.

4. Algorithm and Main Result

In this section we proceed to describe the proposed algorithm and state the main result on its sample complexity.

4.1. Description of Algorithm

We now turn to describing our algorithm. Interestingly, our algorithm is quite dissimilar to the classic stochastic optimization methods. It proceeds by running the algorithm online on fixed chunks of examples, and using the intermediate hypotheses and target risk ϵ_{prior} to gradually refine the hypothesis space. As mentioned above, we assume in our setting that the target expected risk ϵ_{prior} is provided to the learner a priori. We further assume the target risk ϵ_{prior} is feasible for the solution within the domain $\mathcal{H}$, i.e., $\epsilon_{\text{prior}} \geq \epsilon_{\text{opt}}$. The proposed algorithm explicitly takes advantage of the knowledge of expected risk ϵ_{prior} to attain an $O\left(\log(1/\epsilon_{\text{prior}})\right)$ sample complexity.

Throughout we shall consider linear predictors of form $\langle \mathbf{w}, \mathbf{x}\rangle$ and assume that the loss function of interest $\ell(\langle \mathbf{w}, \mathbf{x}\rangle, y)$ is β-smooth. It is straightforward to see that $\mathcal{L}(\mathbf{w}) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\ell(\langle \mathbf{w}, \mathbf{x}\rangle, y)\right]$ is also β-smooth. In addition to the smoothness of the loss function, we also assume that $\mathcal{L}(\mathbf{w})$ to be α-strongly convex. We denote by $\mathbf{w}_*$ the optimal solution that minimizes $\mathcal{L}(\mathbf{w})$, i.e., $\mathbf{w}_* = \arg\min_{\mathbf{w}\in\mathcal{H}} \mathcal{L}(\mathbf{w})$, and denote its optimal value by ϵ_{opt}.

Let $(\mathbf{x}_t, y_t), t = 1, \ldots, T$ be a sequence of i.i.d. training examples. The proposed algorithm divides the T iterations into the m stages, where each stage consists of T_1 training examples, i.e., $T = mT_1$. Let $(\mathbf{x}_k^t, y_k^t)$ be the t-th training example received at stage k, and let η be the step size used by all the stages. At the beginning of each stage k, we initialize the solution $\mathbf{w}$ by the average solution $\widehat{\mathbf{w}}_k$ obtained from the last stage, i.e.,

$$\widehat{\mathbf{w}}_k = \frac{1}{T_1}\sum_{t=1}^{T_1} \widehat{\mathbf{w}}_k^t, \tag{3}$$

where $\widehat{\mathbf{w}}_k^t$ denotes the tth solution at stage k. Another feature of the proposed algorithm is a domain shrinking strategy that adjusts the domain as the algorithm proceeds using

Algorithm 1 Convex Learning with Target Risk

Input: step size η, stage size T_1, number of stages m, target expected risk ϵ_{prior}, parameters $\varepsilon \in (0,1)$ and $\tau \in (0,1)$ used for updating domain size Δ_k, and parameter $\xi \geq 1$ used to clip the gradients
Initialization: $\widehat{\mathbf{w}}_1 = 0$, $\Delta_1 = R$, and $\mathcal{H}_1 = \mathcal{H}$
for $k = 1, \ldots, m$ **do**
  Set $\mathbf{w}_k^t = \widehat{\mathbf{w}}_k$ and $\gamma_k = 2\xi\beta\Delta_k$
  for $t = 1, \ldots, T_1$ **do**
    Receive training example $(\mathbf{x}_t, y_t)$
    Compute the gradient $\hat{\mathbf{g}}_k^t$ and the clipped version of the gradient $\mathbf{v}_k^t$ using Eq. (5)
    Update the solution $\mathbf{w}_k^t$ using Eq. (6).
  end for
  Update Δ_k using Eq. (7).
  Compute the average solution $\widehat{\mathbf{w}}_{k+1}$ according to Eq. (3), and update the domain $\mathcal{H}_{k+1}$ using the expression in (4).
end for

intermediate hypotheses and target risk. We define the domain $\mathcal{H}_k$ used at stage k as

$$\mathcal{H}_k = \{\mathbf{w} \in \mathcal{H} : \|\mathbf{w} - \widehat{\mathbf{w}}_k\| \leq \Delta_k\}, \tag{4}$$

where Δ_k is the domain size, whose value will be discussed later. Similar to the SGD method, at each iteration of stage k, we receive a training example $(\mathbf{x}_k^t, y_k^t)$, and compute the gradient $\hat{\mathbf{g}}_k^t = \ell'\left(\langle \mathbf{w}_k^t, \mathbf{x}_k^t \rangle, y_t\right) \mathbf{x}_k^t$. Instead of using the gradient directly, following (Hazan and Koren, 2011), a clipped version of the gradient, denoted by $\mathbf{v}_k^t = \text{clip}\left(\gamma_k, \hat{\mathbf{g}}_k^t\right)$, will be used for updating the solution. More specifically, the clipped vector $\mathbf{v}_k^t \in \mathbb{R}^d$ is defined as

$$[\mathbf{v}_k^t]_i = \text{clip}\left(\gamma_k, [\hat{\mathbf{g}}_k^t]_i\right) = \text{sign}\left([\hat{\mathbf{g}}_k^t]_i\right) \min\left(\gamma_k, \left|[\hat{\mathbf{g}}_k^t]_i\right|\right), i = 1, \ldots, d \tag{5}$$

where $\gamma_k = 2\xi\beta\Delta_k$ with $\xi \geq 1$. Given the clipped gradient $\mathbf{v}_k^t$, we follow the standard framework of stochastic gradient descent, and update the solution by

$$\mathbf{w}_k^{t+1} = \Pi_{\mathcal{H}_k}\left(\mathbf{w}_k^t - \eta \mathbf{v}_k^t\right). \tag{6}$$

The purpose of introducing the clipped version of the gradient is to effectively control the variance in stochastic gradients, an important step toward achieving the geometric convergence rate. At the end of each stage, we will update the domain size by explicitly exploiting the target expected risk ϵ_{prior} as

$$\Delta_{k+1} = \sqrt{\varepsilon\Delta_k^2 + \tau\epsilon_{\text{prior}}}, \tag{7}$$

where $\varepsilon \in (0,1)$ and $\tau \in (0,1)$ are two parameters, both of which will be discussed later.

Algorithm 1 gives the detailed steps for the proposed method. The three important aspects of Algorithm 1, all crucial to achieve a geometric convergence rate, are highlighted as follows:

- Each stage of the proposed algorithm is comprised of the same number of training examples. This is in contrast to the epoch gradient algorithm (Hazan and Kale, 2011) which divides m iterations into exponentially increasing epochs, and runs SGD with averaging on each epoch. Also, in our case the learning rate is fixed for all iterations.
- The proposed algorithm uses a clipped gradient for updating the solution in order to better control the variance in stochastic gradients; this stands in contrast to the SGD method, which uses original gradients to update the solution.
- The proposed algorithm takes into account the targeted expected risk and intermediate hypotheses when updating the domain size at each stage. The purpose of domain shrinking is to reduce the damage caused by biased gradients that resulted from clipping operation.

4.2. Main Result on Sample Complexity

The main theoretical result of Algorithm 1 is given in the following theorem.

Theorem 4 (Convergence Rate) *Assume that the hypothesis space $\mathcal{H}$ is compact and the loss function ℓ is α-strongly convex and β-smooth. Let $T = mT_1$ be the size of the sample and ϵ_{prior} be the target expected loss given to the learner in advance such that $\epsilon_{\text{opt}} \leq \epsilon_{\text{prior}}$ holds. Given $\varepsilon \in (0,1)$ and $\tau \in (0,1)$, set ξ, η, and T_1 as*

$$\xi = \frac{4\beta}{\alpha\tau},\ T_1 = 4\max\left\{\frac{\xi^3\beta d + 2\xi\beta\sqrt{d}}{\varepsilon\alpha}\ln\frac{ms}{\delta}, \frac{16\xi^2\beta^2}{\alpha^2\varepsilon^2}\right\},\ \eta = \frac{1}{2\xi\beta\sqrt{T_1}},$$

where

$$s = \left\lceil \log_2 \frac{\xi\beta R^2}{\epsilon_{\text{prior}}} \right\rceil. \tag{8}$$

After running Algorithm 1 over m stages, we have, with a probability $1-\delta$,

$$\mathcal{L}(\widehat{\mathbf{w}}_{m+1}) \leq \frac{\beta R^2}{2}\varepsilon^m + \left(1 + \frac{\tau}{1-\varepsilon}\right)\epsilon_{\text{prior}},$$

implying that only $O(d\log[1/\epsilon_{\text{prior}}])$ training examples are needed in order to achieve a risk of $O(\epsilon_{\text{prior}})$.

We note that comparing to the bound in Theorem 3, for Algorithm 1 the level of error to which the linear convergence holds is not determined by the noise level in stochastic gradients, but by the target risk. In other words, the algorithm is able to tolerate the noise by knowing the target risk as prior knowledge and achieves a linear convergence to the level of the target risk even when the variance of stochastic gradients is much larger than the target risk. In addition, although the result given in Theorem 4 assumes a bounded domain with $\|\mathbf{w}\| \leq R$, however, this assumption can be lifted by effectively exploring the strong convexity of the loss function and further assuming that the loss function is Lipschitz continuous with constant G, i.e., $|\mathcal{L}(\mathbf{w}_1) - \mathcal{L}(\mathbf{w}_2)| \leq G\|\mathbf{w}_1 - \mathbf{w}_2\|,\ \forall\ \mathbf{w}_1, \mathbf{w}_2 \in \mathcal{H}$. More

specifically, the fact that the $\mathcal{L}(\mathbf{w})$ is α-strongly convex with first order optimality condition, for the optimal solution $\mathbf{w}_* = \arg\min_{\mathbf{w}\in\mathcal{H}} \mathcal{L}(\mathbf{w})$, we have

$$\mathcal{L}(\mathbf{w}) - \mathcal{L}(\mathbf{w}_*) \geq \frac{\alpha}{2}\|\mathbf{w} - \mathbf{w}_*\|^2, \quad \forall \mathbf{w} \in \mathcal{H}.$$

This inequality combined with Lipschitz continuous assumption implies that for any $\mathbf{w} \in \mathcal{H}$ the inequality $\|\mathbf{w} - \mathbf{w}_*\| \leq R_* := 2G/\alpha$ holds, and therefore we can simply set $R = R_*$. We also note that this dependency can be resolved with a weaker assumption than Lipschitz continuity, which only depends on the gradient of loss function at origin. To this end, we define $|\ell'(0, y)| = G$. Using the fact that $\mathcal{L}(\mathbf{w})$ is α-strongly, it is easy to verify that $\frac{\alpha}{2}\|\mathbf{w}_*\|^2 - G\|\mathbf{w}_*\| \leq 0$, leading to $\|\mathbf{w}_*\| \leq R_* := \frac{2}{\alpha}G$ and, therefore, we can simply set $R = R_*$.

We now use our analysis of Algorithm 1 to obtain a sample complexity analysis for learning smooth strongly convex problems with a bounded hypothesis class. To make it easier to parse, we only keep the dependency on the main parameters d, α, β, T, and ϵ_{prior} and hide the dependency on other constants in $\mathcal{O}(\cdot)$ notation. Let $\widehat{\mathbf{w}}$ denote the output of Algorithm 1. By setting $\varepsilon = 0.5$ and letting $c = O(\tau)$ to be an arbitrary small number, Theorem 4 yields the following:

Corollary 5 (Sample Complexity) *Under the same conditions as Theorem 4, by running Algorithm 1 for minimizing $\mathcal{L}(\mathbf{w})$ with a number of iterations (i.e., number of training examples) T, if it holds that,*

$$T \geq \mathcal{O}\left(d\kappa^4\left(\log\frac{1}{\epsilon_{\text{prior}}}\log\log\frac{1}{\epsilon_{\text{prior}}} + \log\frac{1}{\delta}\right)\right)$$

where $\kappa = \beta/\alpha$ denotes the condition number of the loss function and d is the dimension of data, then with a probability $1 - \delta$, $\widehat{\mathbf{w}}$ attains a risk of $O(\epsilon_{\text{prior}})$, i.e., $\mathcal{L}(\widehat{\mathbf{w}}) \leq (1 + c)\epsilon_{\text{prior}}$.

As an example of a concrete problem that may be put into the setting of the present work is the regression problem with squared loss. It is easy to show that average square loss function is Lipschitz continuous with a Lipschitz constant $\beta = \lambda_{\max}(X^\top X)$ which denotes the largest eigenvalue of matrix $X^\top X$ where X is the data matrix. The strong convexity is guaranteed as long as the population data covariance matrix is not rank-deficient and its minimum eigenvalue is lower bounded by a constant $\alpha > 0$. For this problem, the optimal minimax sample complexity is known to be $O(\frac{1}{\epsilon})$, but as it implies from Corollary 5, by the knowledge of target risk ϵ_{prior}, it is possible to reduce the sample complexity to $O(\log(1/\epsilon_{\text{prior}}))$.

Remark 6 *It is indeed remarkable that the sample complexity of Theorem 4 has $\kappa^4 = (\beta/\alpha)^4$ dependency on the condition number of the loss function, which is worse than the $\sqrt{\beta/\alpha}$ dependency in the lower bound in (2). Also, the explicit dependency of sample complexity on dimension d makes the proposed algorithm inappropriate for non-parametric settings.*

5. Analysis

Now we turn to proving the main theorem. The proof will be given in a series of lemmas and theorems where the proof of few are given in the appendix. The proof makes use of the Bernstein inequality for martingales, idea of peeling process, self-bounding property of smooth loss functions, standard analysis of stochastic optimization, and novel ideas to derive the claimed sample complexity for the proposed algorithm.

The proof of Theorem 4 is by induction and we start with the key step given in the following theorem.

Theorem 7 *Assume $\epsilon_{\text{prior}} \geq \epsilon_{\text{opt}}$. For a fixed stage k, if $\|\widehat{\mathbf{w}}_k - \mathbf{w}_*\| \leq \Delta_k$, then, with a probability $1-\delta$, we have*

$$\|\widehat{\mathbf{w}}_{k+1} - \mathbf{w}_*\|^2 \leq a\Delta_k^2 + b\epsilon_{\text{prior}}$$

where

$$a = \frac{2}{\alpha T_1}\left(2\xi\beta\sqrt{T_1} + \left[\xi^3\beta d + 2\xi\beta\sqrt{d}\right]\ln\frac{s}{\delta}\right), \quad b = \frac{8}{\alpha\xi} \tag{9}$$

and s is given in (8), provided that $\xi \geq 16\beta/\alpha$ and $\eta = 1/(2\xi\beta\sqrt{T_1})$ hold.

Taking this statement as given for the moment, we proceed with the proof of Theorem 4, returning later to establish the claim stated in Theorem 7.

Proof [of Theorem 4] By setting a and b in (9) in Theorem 7 as $a \leq \varepsilon$ and $b \leq 2\tau/\beta$, we have $\xi \geq 4\beta/(\alpha\tau)$ and

$$T_1 \leq \frac{2}{\alpha\varepsilon}\left(2\xi\beta\sqrt{T_1} + \left[\xi^3\beta d + 2\xi\beta\sqrt{d}\right]\ln\frac{s}{\delta}\right)$$

implying that

$$T_1 \geq 4\max\left\{\frac{\xi^3\beta d + 2\xi\beta\sqrt{d}}{\varepsilon\alpha}\ln\frac{s}{\delta}, \frac{16\xi^2\beta^2}{\alpha^2\varepsilon^2}\right\}.$$

Thus, using Theorem 7 and the definition of ξ and T_1, we have, with a probability $1-\delta$,

$$\Delta_{k+1}^2 \leq \varepsilon\Delta_k^2 + \frac{2\tau}{\beta}\epsilon_{\text{prior}}.$$

After m stages, with a probability $1 - m\delta$, we have

$$\Delta_{m+1}^2 \leq \varepsilon^m\Delta_1^2 + \frac{2\tau}{\beta}\epsilon_{\text{prior}}\sum_{i=0}^{m-1}\varepsilon^i \leq \varepsilon^m\Delta_1^2 + \frac{2\tau}{\beta(1-\varepsilon)}\epsilon_{\text{prior}}.$$

By the β-smoothness of $\mathcal{L}(\mathbf{w})$, it implies that

$$\begin{aligned}\mathcal{L}(\widehat{\mathbf{w}}_{m+1}) - \mathcal{L}(\mathbf{w}_*) \leq \frac{\beta}{2}\|\widehat{\mathbf{w}}_{m+1} - \mathbf{w}_*\|^2 &\leq \frac{\beta}{2}\varepsilon^m\Delta_1^2 + \frac{\tau}{1-\varepsilon}\epsilon_{\text{prior}}, \\ &\leq \frac{\beta R^2}{2}\varepsilon^m + \frac{\tau}{1-\varepsilon}\epsilon_{\text{prior}},\end{aligned}$$

where the last inequality follows from $\Delta_1 \leq R$. The bound stated in the theorem follows the assumption that $\mathcal{L}(\mathbf{w}_*) = \epsilon_{\text{opt}} \leq \epsilon_{\text{prior}}$. ■

5.1. Proof of Theorem 7

To bound $\|\widehat{\mathbf{w}}_{k+1}-\mathbf{w}_*\|$ in terms of Δ_k, we start with the standard analysis of online learning. In particular, from the strong convexity assumption of $\mathcal{L}(\mathbf{w})$ and updating rule (6) we have,

$$\begin{aligned}
\mathcal{L}(\mathbf{w}_k^t)-\mathcal{L}(\mathbf{w}_*) &\leq \langle \nabla\mathcal{L}(\mathbf{w}_k^t), \mathbf{w}_k^t-\mathbf{w}_*\rangle - \frac{\alpha}{2}\|\mathbf{w}_k^t-\mathbf{w}_*\|^2 \\
&= \langle \mathbf{v}_k^t, \mathbf{w}_k^t-\mathbf{w}_*\rangle + \langle \nabla\mathcal{L}(\mathbf{w}_k^t)-\mathbf{v}_k^t, \mathbf{w}_k^t-\mathbf{w}_*\rangle - \frac{\alpha}{2}\|\mathbf{w}_t-\mathbf{w}_*\|^2 \\
&\leq \frac{\|\mathbf{w}_k^{t+1}-\mathbf{w}_*\|^2-\|\mathbf{w}_k^{t+1}-\mathbf{w}_*\|^2}{2\eta} + \frac{\eta d}{2}\gamma_k^2 \\
&\quad + \underbrace{\langle \nabla\mathcal{L}(\mathbf{w}_k^t)-\mathbf{v}_k^t, \mathbf{w}_k^t-\mathbf{w}_*\rangle}_{\triangleq v_k^t} - \frac{\alpha}{2}\|\mathbf{w}_t-\mathbf{w}_*\|^2, \qquad (10)
\end{aligned}$$

where the last step follows from $\|\mathbf{v}_k^t\| \leq \gamma_k\sqrt{d}$. By adding all the inequalities of (10) at stage k, we have

$$\begin{aligned}
\sum_{t=1}^{T_1}\mathcal{L}(\mathbf{w}_k^t)-\mathcal{L}(\mathbf{w}_*) &\leq \frac{\|\widehat{\mathbf{w}}_k-\mathbf{w}_*\|^2}{2\eta} + \frac{d\eta}{2}\gamma_k^2 T_1 + \sum_{t=1}^{T_1} v_k^t - \frac{\alpha}{2}\sum_{t=1}^{T_1}\|\mathbf{w}_t-\mathbf{w}_*\|^2 \\
&\leq \frac{\Delta_k^2}{2\eta} + \frac{d\eta}{2}\gamma_k^2 T_1 + V_k - \frac{\alpha}{2}W_k, \qquad (11)
\end{aligned}$$

where V_k and W_k are defined as $V_k = \sum_{t=1}^{T_1} v_k^t$ and $W_k = \sum_{t=1}^{T_1}\|\mathbf{w}_k^t-\mathbf{w}_*\|^2$, respectively. In order to bound V_k, using the fact that $\nabla\mathcal{L}(\mathbf{w}_k^t) = \mathbb{E}_t[\hat{\mathbf{g}}_k^t]$, we rewrite V_k as

$$\begin{aligned}
V_k &= \sum_{t=1}^{T_1}\underbrace{\langle -\mathbf{v}_k^t + \mathbb{E}_t[\mathbf{v}_k^t], \mathbf{w}_k^t-\mathbf{w}_*\rangle}_{\triangleq d_k^t} + \sum_{t=1}^{T_1}\underbrace{\langle \mathbb{E}_t\left[\hat{\mathbf{g}}_k^t\right] - \mathbb{E}_t[\mathbf{v}_k^t], \mathbf{w}_k^t-\mathbf{w}_*\rangle}_{\triangleq e_k^t} \\
&= D_k + E_k,
\end{aligned}$$

where $D_k = \sum_{t=1}^{T_1} d_k^t$ and $E_k = \sum_{t=1}^{T_1} e_k^t$ which represent the variance and bias of the clipped gradient $\mathbf{v}_k^t$, respectively. We now turn to separately upper bound each term.

The following lemma bounds the variance term D_k using the Bernstein inequality for martingale. Its proof can be found in Appendix A.

Lemma 8 *For any $L>0$ and $\mu>0$, we have*

$$\Pr\left(W_k \leq \frac{\epsilon_{\text{prior}} T_1}{2\mu\beta}\right) + \Pr\left(D_k \leq \frac{1}{L}W_k + \left(L\gamma_k^2 d + \gamma_k\Delta_k\sqrt{d}\right)\ln\frac{s}{\delta}\right) \geq 1-\delta$$

where s is given by

$$s = \left\lceil \log_2 \frac{8\beta\mu R^2}{\epsilon_{\text{prior}}} \right\rceil.$$

The following lemma bounds E_k using the self-bounding property of smooth functions and the proof is deferred to Appendix B.

Lemma 9

$$E_k \leq \frac{4T_1}{\xi}\epsilon_{\text{opt}} + \frac{4\beta}{\xi}W_k \leq \frac{4T_1}{\xi}\epsilon_{\text{prior}} + \frac{4\beta}{\xi}W_k.$$

Note that without the knowledge of ϵ_{prior}, we have to bound ϵ_{opt} by $\Omega(1)$, resulting in a very loose bound for the bias term E_k. It is knowledge of the target expected risk ϵ_{prior} that allows us to come up with a significantly more accurate bound for the bias term E_k, which consequentially leads to a geometric convergence rate.

We now proceed to bound $\sum_{t=1}^{T_1}\mathcal{L}(\mathbf{w}_k^t) - \mathcal{L}(\mathbf{w}_*)$ using the two bounds in Lemma 8 and 9. To this end, based on the result obtained in Lemma 8, we consider two scenarios. In the first scenario, we assume

$$W_k \leq \frac{\epsilon_{\text{prior}}T_1}{2\mu\beta} \tag{12}$$

In this case, we have

$$\sum_{t=1}^{T_1}\mathcal{L}(\mathbf{w}_k^t) - \mathcal{L}(\mathbf{w}_*) \leq \frac{\beta}{2}W_k \leq \frac{\epsilon_{\text{prior}}}{2\mu}T_1. \tag{13}$$

In the second scenario, we assume

$$D_k \leq \frac{1}{L}W_T + \left(L\gamma_k^2 d + \gamma_k\Delta_k\sqrt{d}\right)\ln\frac{s}{\delta}. \tag{14}$$

In this case, by combining the bounds for D_k and E_k and setting $L = \frac{\xi}{4\beta}$, we have

$$\begin{aligned} V_k &\leq \frac{8\beta}{\xi}W_k + \left(\frac{\xi d}{4\beta}\gamma_k^2 + \gamma_k\Delta_k\sqrt{d}\right)\ln\frac{s}{\delta} + \frac{4T_1}{\xi}\epsilon_{\text{prior}} \\ &= \frac{8\beta}{\xi}W_k + \left(\xi^3\beta d + 2\xi\beta\sqrt{d}\right)\Delta_k^2\ln\frac{s}{\delta} + \frac{4T_1}{\xi}\epsilon_{\text{prior}}, \end{aligned}$$

where the last equality follows from the fact $\gamma_k = 2\xi\beta\Delta_k$. If we choose ξ such that $\frac{8\beta}{\xi} \leq \frac{\alpha}{2}$ or $\xi \geq \frac{16\beta}{\alpha} > 1$ holds, we get

$$V_k \leq \frac{\alpha}{2}W_k + \left(\xi^3\beta d + 2\xi\beta\sqrt{d}\right)\Delta_k^2\ln\frac{s}{\delta} + \frac{4T_1}{\xi}\epsilon_{\text{prior}}$$

Substituting the above bound for V_k into the inequality of (11), we have

$$\sum_{t=1}^{T_1}\mathcal{L}(\mathbf{w}_k^t) - \mathcal{L}(\mathbf{w}_*) \leq \frac{\Delta_k^2}{2\eta} + \frac{\eta}{2}\gamma_k^2 T_1 + \left(\xi^3\beta d + 2\xi\beta\sqrt{d}\right)\Delta_k^2\ln\frac{s}{\delta} + \frac{4T_1}{\xi}\epsilon_{\text{prior}}$$

By choosing η as $\eta = \frac{\Delta_k}{\gamma_k\sqrt{T_1}} = \frac{1}{2\xi\beta\sqrt{T_1}}$, we have

$$\mathcal{L}(\widehat{\mathbf{w}}_{k+1}) - \mathcal{L}(\mathbf{w}_*) \leq \frac{1}{T_1}\left(2\xi\beta\sqrt{T_1} + \left[\xi^3\beta d + 2\xi\beta\sqrt{d}\right]\ln\frac{s}{\delta}\right)\Delta_k^2 + \frac{4}{\xi}\epsilon_{\text{prior}}. \tag{15}$$

By combining the bounds in (13) and (15), under the assumption that at least one of the two conditions in (12) and (14) is true, by setting $\mu = B/8$, we have

$$\mathcal{L}(\widehat{\mathbf{w}}_{k+1}) - \mathcal{L}(\mathbf{w}_*) \leq \frac{1}{T_1}\left(2\xi\beta\sqrt{T_1} + \left[\xi^3\beta d + 2\xi\beta\sqrt{d}\right]\ln\frac{s}{\delta}\right)\Delta_k^2 + \frac{4}{\xi}\epsilon_{\text{prior}},$$

implying

$$\|\widehat{\mathbf{w}}_{k+1} - \mathbf{w}_*\| \leq \frac{2}{\alpha T_1}\left(2\xi\beta\sqrt{T_1} + \left[\xi^3\beta d + 2\xi\beta\sqrt{d}\right]\ln\frac{s}{\delta}\right)\Delta_k^2 + \frac{8}{\alpha\xi}\epsilon_{\text{prior}}.$$

We complete the proof by using Lemma 8, which states that the probability for either of the two conditions hold is no less than $1 - \delta$.

6. Conclusions

In this paper, we have studied the sample complexity of passive learning when the target expected risk is given to the learner as prior knowledge. The crucial fact about target risk assumption is that, it can be fully exploited by the learning algorithm and stands in contrast to most common types of prior knowledges that usually enter into the generalization bounds and are often perceived as a rather crude way to incorporate such assumptions. We showed that by explicitly employing the target risk ϵ_{prior} in a properly designed stochastic optimization algorithm, it is possible to attain the given target risk ϵ_{prior} with a logarithmic sample complexity $\log\left(\frac{1}{\epsilon_{\text{prior}}}\right)$, under the assumption that the loss function is both strongly convex and smooth.

There are various directions for future research. The current study is restricted to the parametric setting where the hypothesis space is of finite dimension. It would be interesting to see how to achieve a logarithmic sample complexity in a non-parametric setting where hypotheses lie in a functional space of infinite dimension. Evidently, it is impossible to extend the current algorithm for the non-parametric setting; therefore additional analysis tools are needed to address the challenge of infinite dimension arising from the non-parametric setting. It is also an interesting problem to relate target risk assumption we made here to the low noise margin condition which is often made in active learning for binary classification since both settings appear to share the same sample complexity. However it is currently unclear how to derive a connection between these two settings. We believe this issue is worthy of further exploration and leave it as an open problem.

Acknowledgments

The authors would like to thank the PC for insightful discussions and three anonymous reviewers for their constructive comments and helpful suggestions on the original version of this paper. This work is partially supported by Office of Navy Research (ONR Award N00014-09-1-0663 and N000141210431).

References

M. Anthony and P.L. Bartlett. *Neural network learning: Theoretical foundations*. Cambridge University Press, 1999.

Maria-Florina Balcan, Steve Hanneke, and Jennifer Wortman Vaughan. The true sample complexity of active learning. *Machine Learning*, 80(2-3):111–139, 2010.

P.L. Bartlett, O. Bousquet, and S. Mendelson. Local rademacher complexities. *The Annals of Statistics*, 33(4):1497–1537, 2005.

Shai Ben-David, David Pal, and Shai Shalev-Shwartz. Agnostic online learning. In *COLT*, 2009.

Anselm Blumer, A. Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the vapnik-chervonenkis dimension. *J. ACM*, 36(4):929–965, 1989.

Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.

Andrew Cotter, Ohad Shamir, Nati Srebro, and Karthik Sridharan. Better mini-batch algorithms via accelerated gradient methods. In *NIPS*, pages 1647–1655, 2011.

John C. Duchi, Peter L. Bartlett, and Martin J. Wainwright. Randomized smoothing for stochastic optimization. *SIAM Journal on Optimization*, 22(2):674–701, 2012.

A. Ehrenfeucht, D. Haussler, M. Kearns, and L. Valiant. A general lower bound on the number of examples needed for learning. *Information and Computation*, 82(3):247–261, 1989.

Steve Hanneke. *Theoretical Foundations of Active Learning*. PhD thesis, 2009.

Elad Hazan and Satyen Kale. Beyond the regret minimization barrier: an optimal algorithm for stochastic strongly-convex optimization. *COLT*, 2011.

Elad Hazan and Tomer Koren. Optimal algorithms for ridge and lasso regression with partially observed attributes. *CoRR*, 2011.

Elad Hazan, Adam Kalai, Satyen Kale, and Amit Agarwal. Logarithmic regret algorithms for online convex optimization. In *COLT*, pages 499–513, 2006.

Anatoli Iouditski and Yuri Nesterov. Primal-dual subgradient methods for minimizing uniformly convex functions. available at http://hal.archives-ouvertes.fr/docs/00/50/89/33/PDF/Strong-hal.pdf, 2010.

Sham M. Kakade, Karthik Sridharan, and Ambuj Tewari. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In *NIPS*, pages 793–800, 2008.

Vladimir Koltchinskii. *Oracle Inequalities in Empirical Risk Minimization and Sparse Recovery Problems*. Lecture Notes in mathematics. Springer, 2011.

Wee Sun Lee, Peter L. Bartlett, and Robert C. Williamson. The importance of convexity in learning with squared loss. *IEEE Transactions on Information Theory*, 44(5):1974–1980, 1998.

A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

A.S. Nemirovsky and D.B. Yudin. *Problem complexity and method efficiency in optimization.* Wiley Interscience Series in Discrete Mathematics, 1983.

Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publishers, 2004.

Alexander Rakhlin, Karthik Sridharan, and Ambuj Tewari. Online learning: Random averages, combinatorial parameters, and learnability. *CoRR*, abs/1006.1138, 2010.

Alexander Rakhlin, Ohad Shamir, and Karthik Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. In *ICML*, 2012.

Aaditya Ramdas and Aarti Singh. Optimal stochastic convex optimization through the lens of active learning. In *ICML*, 2013.

S. Shalev-Shwartz, O. Shamir, K. Sridharan, and N. Srebro. Learnability and stability in the general learning setting. *COLT*, 2009a.

S. Shalev-Shwartz, O. Shamir, K. Sridharan, and N. Srebro. Stochastic convex optimization. *COLT*, 2009b.

Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability, stability and uniform convergence. *Journal of Machine Learning Research*, 11:2635–2670, 2010.

Nathan Srebro, Karthik Sridharan, and Ambuj Tewari. Smoothness, low noise and fast rates. In *NIPS*, pages 2199–2207, 2010.

Karthik Sridharan. Learning from an optimization viewpoint. PhD Thesis, 2012.

Karthik Sridharan, Shai Shalev-Shwartz, and Nathan Srebro. Fast rates for regularized objectives. In *NIPS*, pages 1545–1552, 2008.

V.N. Vapnik and A.Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and Its Applications*, 16(2):264–280, 1971.

Appendix A. Proof of Lemma 8

The proof is based on the Bernstein inequality for martingales (see, e.g., Cesa-Bianchi and Lugosi (2006)).

Lemma 10 *(Bernstein inequality for martingales). Let $X_1, \ldots, X_n$ be a bounded martingale difference sequence with respect to the filtration $\mathcal{F} = (\mathcal{F}_i)_{1 \le i \le n}$ and with $\|X_i\| \le M$. Let $S_i = \sum_{j=1}^{i} X_j$ be the associated martingale. Denote the sum of the conditional variances by*

$$\Sigma_n^2 = \sum_{t=1}^{n} \mathbb{E}\left[X_t^2 | \mathcal{F}_{t-1}\right]$$

Then for all constants κ, $\nu > 0$,

$$\Pr\left[\max_{i=1,\ldots,n} S_i > \rho \text{ and } \Sigma_n^2 \le \nu\right] \le \exp\left(-\frac{\rho^2}{2(\nu + M\rho/3)}\right)$$

and therefore,

$$\Pr\left[\max_{i=1,\ldots,n} S_i > \sqrt{2\nu\rho} + \frac{\sqrt{2}}{3} M\rho \text{ and } \Sigma_n^2 \le \nu\right] \le e^{-\rho}.$$

Proof [of Lemma 8] Define martingale difference $d_k^t = \left\langle \mathbf{w}_k^t - \mathbf{w}_*, \mathbb{E}_t[\mathbf{v}_k^t] - \mathbf{v}_k^t \right\rangle$ and martingale $D_k = \sum_{t=1}^{T_1} d_k^t$. Let Σ_T^2 denote the conditional variance as

$$\begin{aligned}
\Sigma_T^2 = \sum_{t=1}^{T_1} \mathbb{E}_t\left[(d_k^t)^2\right] &\le \sum_{t=1}^{T_1} \mathbb{E}_t\left[\left\|\mathbb{E}_t[\mathbf{v}_k^t] - \mathbf{v}_k^t\right\|^2\right] \|\mathbf{w}_k^t - \mathbf{w}_*\|^2 \\
&\le \sum_{t=1}^{T} d\gamma_k^2 \|\mathbf{w}_k^t - \mathbf{w}\|^2 = d\gamma_k^2 W_k,
\end{aligned}$$

which follows from the Cauchy's Inequality and the definition of clipping. Define $M = \max_t |d_k^t| \le 2\sqrt{d}\gamma_k \Delta_k$. To prove the inequality in Lemma 8, we follow the idea of peeling process (Koltchinskii, 2011). Since $W_k \le 4R^2 T_1$, we have

$$\begin{aligned}
&\Pr\left(D_k \ge 2\gamma_k\sqrt{W_k d\rho} + \sqrt{2}M\rho/3\right) \\
&= \Pr\left(D_k \ge 2\gamma_k\sqrt{W_k d\rho} + \sqrt{2}M\rho/3, W_k \le 4R^2T_1\right) \\
&= \Pr\left(D_k \ge 2\gamma_k\sqrt{W_k d\rho} + \sqrt{2}M\rho/3, \Sigma_T^2 \le \gamma_k^2 d W_k, W_k \le 4R^2T_1\right) \\
&\le \Pr\left(D_k \ge 2\gamma_k\sqrt{W_k d\rho} + \sqrt{2}M\rho/3, \Sigma_T^2 \le \gamma_k^2 d W_k, W_k \le \epsilon_{\text{prior}} T_1/(2\beta\mu)\right) \\
&\quad + \sum_{i=1}^{s} \Pr\left(D_k \ge 2\gamma_k\sqrt{W_k d\rho} + \sqrt{2}M\rho/3, \Sigma_T^2 \le \gamma_k^2 d W_k, \frac{\epsilon_{\text{prior}} 2^{i-1} T_1}{2\beta\mu} < W_k \le \frac{\epsilon_{\text{prior}} 2^i T_1}{2\beta\mu}\right) \\
&\le \Pr\left(W_k \le \frac{\epsilon_{\text{prior}} T_1}{2\beta\mu}\right) + \sum_{i=1}^{s} \Pr\left(D_k \ge \sqrt{\frac{\epsilon_{\text{prior}} 2^{i+1} T_1 \gamma_k^2 d}{2\beta\mu}}\rho + \frac{\sqrt{2}}{3}M\rho, \Sigma_T^2 \le \frac{\epsilon_{\text{prior}} 2^i T_1 \gamma_k^2 d}{2\beta\mu}\right) \\
&\le \Pr\left(W_k \le \frac{\epsilon_{\text{prior}} T_1}{2\beta\mu}\right) + se^{-\rho},
\end{aligned}$$

where s is given by

$$s = \left\lceil \log_2 \frac{8\beta\mu R^2}{\epsilon_{\text{prior}}} \right\rceil .$$

The last step follows the Bernstein inequality for martingales. We complete the proof by setting $\rho = \ln(s/\delta)$ and using the fact that

$$2\gamma_k \sqrt{W_k \rho d} \leq \frac{1}{L} W_k + \gamma_k^2 \rho d L.$$

■

Appendix B. Proof of Lemma 9

To bound E_k, we need the following two lemmas. The first lemma bounds the deviation of the expected value of a clipped random variable from the original variable, in terms of its variance (Lemma A.2 from (Hazan and Koren, 2011)).

Lemma 11 *Let X be a random variable, let $\widetilde{X} = \text{clip}(\text{X}, \text{C})$ and assume that $|\mathbb{E}[X]| \leq C/2$ for some $C > 0$. Then*

$$|\mathbb{E}[\widetilde{X}] - \mathbb{E}[X]| \leq \frac{2}{C} |\text{Var[X]}|$$

Another key observation used for bounding E_k is the fact that for any non-negative β-smooth convex function, we have the following self-bounding property. We note that this self-bounding property has been used in (Srebro et al., 2010) to get better (optimistic) rates of convergence for non-negative smooth losses.

Lemma 12 *For any β-smooth non-negative function $f : \mathbb{R} \to \mathbb{R}$, we have $|f'(w)| \leq \sqrt{4\beta f(w)}$*

As a simple proof, first from the smoothness assumption, by setting $w_1 = w_2 - \frac{1}{\beta} f'(w_2)$ in (1) and rearranging the terms we obtain $f(w_2) - f(w_1) \geq \frac{1}{2\beta} |f'(w_2)|^2$. On the other hand, from the convexity of loss function we have $f(w_1) \geq f'(w_2) + \langle f'(w_1), w_1 - w_2 \rangle$. Combining these inequalities and considering the fact that the function is non-negative gives the desired inequality.

Proof [of Lemma 9] To apply the above lemmas, we write e_k^t as

$$e_k^t = \sum_{i=1}^{d} \mathbb{E}_t \left[\ell'(\langle \mathbf{w}_k^t, \mathbf{x}_k^t \rangle, y_t)[\mathbf{x}_k^t]_i - \text{clip}\left(\gamma_k, \ell'(\langle \mathbf{w}_k^t, \mathbf{x}_k^t \rangle, y_t)[\mathbf{x}_k^t]_i\right) \right] [\mathbf{w}_k^t - \mathbf{w}_*]_i$$

In order to apply Lemma 11, we check if the following condition holds

$$\gamma_k \geq 2 \left| \mathbb{E}_t \left[\ell' \left(\langle \mathbf{w}_k^t, \mathbf{x}_k^t \rangle, y_t \right) [\mathbf{x}_k^t]_i \right] \right| \tag{16}$$

Since

$$\begin{aligned} & \left|\mathbb{E}_t\left[\ell'\left(\langle \mathbf{w}_k^t, \mathbf{x}_k^t\rangle, y_t\right)[\mathbf{x}_k^t]_i\right]\right| \\ \leq & \left|\mathbb{E}_t\left[\left\{\ell'\left(\langle \mathbf{w}_k^t, \mathbf{x}_k^t\rangle, y_t\right) - \ell'\left(\langle \mathbf{w}_*, \mathbf{x}_k^t\rangle, y_t\right)\right\}[\mathbf{x}_k^t]_i\right]\right| + \left|\mathbb{E}_t\left[\ell'\left(\langle \mathbf{w}_*, \mathbf{x}_k^t\rangle, y_t\right)[\mathbf{x}_k^t]_i\right]\right| \\ \leq & \beta\|\mathbf{w}_k^t - \mathbf{w}_*\| \leq \beta\Delta_k \end{aligned}$$

where the last inequality follows from $\mathbb{E}_t\left[\ell'\left(\langle \mathbf{w}_*, \mathbf{x}_k^t\rangle, y_t\right)[\mathbf{x}_k^t]_i\right] = 0$ since $\mathbf{w}_*$ is the minimizer of $\mathcal{L}(\mathbf{w})$, we thus have

$$\gamma_k = 2\xi\beta\Delta_k \geq 2\beta\Delta_k \geq 2\left|\mathbb{E}_t\left[\ell'\left(\langle \mathbf{w}_k^t, \mathbf{x}_k^t\rangle, y_t\right)[\mathbf{x}_k^t]_i\right]\right|$$

where $\xi \geq 1$, implying that the condition in (16) holds. Thus, using Lemma 11, we have

$$\begin{aligned} e_k^t & \leq \sum_{i=1}^d \left|[\mathbf{w}_k^t - \mathbf{w}_*]_i\right| \frac{1}{\gamma_k}\mathbb{E}_t\left[\left(\ell'(\langle \mathbf{w}_k^t, \mathbf{x}_k^t\rangle, y_t)[\mathbf{x}_k^t]_i\right)^2\right] \\ & \leq \frac{2\|\mathbf{w}_k^t - \mathbf{w}_*\|_\infty}{\gamma_k}\mathbb{E}_t\left[\left(\ell'(\langle \mathbf{w}_k^t, \mathbf{x}_k^t\rangle, y_t)\right)^2\right] \end{aligned}$$

Using Lemma 12 to upper bound the right hand side, we further simplify the above bound for e_k^t as

$$\begin{aligned} e_k^t & \leq \frac{8\beta\|\mathbf{w}_k^t - \mathbf{w}_*\|_\infty}{\gamma_k}\mathbb{E}_t\left[\ell\left(\langle \mathbf{w}_k^t, \mathbf{x}_k^t\rangle, y_t\right)\right] \\ & = \frac{8\beta\|\mathbf{w}_k^t - \mathbf{w}_*\|_\infty}{\gamma_k}\mathcal{L}(\mathbf{w}_k^t) \\ & \leq \frac{8\beta\Delta_k}{\gamma_k}\mathcal{L}(\mathbf{w}_k^t) \\ & = \frac{4}{\xi}\mathcal{L}(\mathbf{w}_k^t) \end{aligned}$$

where the second inequality follows from $\|\mathbf{w}_k^t - \mathbf{w}_*\|_\infty \leq \|\mathbf{w}_k^t - \mathbf{w}_*\| \leq \Delta_k$. Therefore we obtain

$$\begin{aligned} E_k = \sum_{t=1}^{T_1} e_k^t \leq \frac{4}{\xi}\sum_{t=1}^{T_1}\mathcal{L}(\mathbf{w}_k^t) & = \frac{4}{\xi}\sum_{t=1}^{T_1}\mathcal{L}(\mathbf{w}_*) + \frac{4}{\xi}\sum_{t=1}^{T_1}\mathcal{L}(\mathbf{w}_k^t) - \mathcal{L}(\mathbf{w}_*) \\ & \leq \frac{4T_1}{\xi}\mathcal{L}(\mathbf{w}_*) + \frac{4\beta}{\xi}\sum_{t=1}^{T_1}\|\mathbf{w}_k^t - \mathbf{w}_*\|^2 \\ & = \frac{4T_1}{\xi}\mathcal{L}(\mathbf{w}_*) + \frac{4\beta}{\xi}W_k, \end{aligned}$$

where the second inequality follows from the smoothness assumption of $\mathcal{L}(\mathbf{w})$. ■