THEORETICAL GUARANTEES FOR WITHOUT-REPLACEMENT SAMPLING FOR STOCHASTIC GRADIENT DESCENT

PABLO BUSCH, GRANT GAMBETTA, AND IGNAT KULINKA

ABSTRACT. Machine learning methods usually involve the minimization of a set of convex functions. A common algorithm used in practice is stochastic gradient descent, that uses just one observation per iteration to approximate the gradient, making it computationally cheap. Theoretical guarantees have been provided for the with-replacement setting, but not much work has been done on the without-replacement case. This project summarizes the article of [Sha16] that provides convergence guarantees for the without-replacement case. Under the assumption that $f_i(\cdot)$ corresponds to convex Lipschitz functions, the algorithm has suboptimality on order of $\mathcal{O}(1/\sqrt{T})$, which is similar to with-replacement in that regime. For λ-strongly convex functions and smooth, the bound on suboptimality is on the order of $\mathcal{O}(1/\lambda T)$, again, similar to the with-replacement setting (but with-replacement does not need the smoothness condition). The article combine ideas from different fields such as stochastic optimization, adversarial online learning, and transductive learning.

1. INTRODUCTION

Machine learning problems usually involve solving a convex minimization problem over a set of loss/risk functions. As such, most machine learning algorithms rely on optimization algorithms to converge to the optimal solution after T iterations. While there's a number of algorithms available for specific use cases, a common first order method is gradient descent, which uses the geometric information about the function $f(\cdot)$ to iterate and move in the "steepest" direction to reach the minimum.

Among convex optimization algorithms, stochastic gradient methods are widely used and are the backbone of many popular statistical and machine learning models, ranging from least squares regression to deep neural networks. More specifically, most statistical and machine learning models that utilize stochastic gradient methods use with-replacement sampling. For this project, we aim to explore stochastic algorithms that use **without-replacement** sampling, which is a method that can have several benefits compared to with-replacement sampling. That leads us to the key question: why use without-replacement sampling?

In stochastic gradient descent algorithms, we use a sample (or a minibatch of samples) of the data points to compute the gradient and update the next step of the iteration. When using one sample at a time, we are simply approximating the expectation of the gradient, and the method works well in practice and converges to the optimum (under convex settings), and it reduces the computational burden as it only uses one observation per iteration. Usually the random sampling is done with replacement, so any observation can be draw in every iteration. But sampling without replacement can provide many benefits such as:

- Less computational resources as we simply shuffle the observations and at the beginning, and then we draw them sequentially. We notice that less computational resources could have a big impact in complex problems, as it reduces energy consumption and CO_2 emissions.

- It could be easier to implement in practice since after shuffling the observations, the algorithm can simply draw them sequentially. Note this is easier because sequential data access can be easier to implement compared to random data access.
- Could perform better empirically than with-replacement sampling, as it uses all observations thus giving each sample equal weight.

The theoretical background of without replacement sampling has some complications, as subsequent iterations are not statistically independent. Less theoretical work has been done on the properties of without-replacement algorithms. The **main contribution** of the article [Sha16] is to provide some convergence guarantees for stochastic gradient methods without replacement.

2. Algorithm Description

Most machine learning methods seek to minimize a set of loss functions. In the case where the functions $f_i(\cdot)$ are convex, gradient descent methods provide guarantees to reach these optimum after a certain number of iterations. Gradient descent and stochastic gradient descent rely on estimating the gradient of the function: $\nabla f_i(\mathbf{w}_t)$. Stochastic gradient methods use random sampling to draw observations that estimate the gradient in an unbiased manner, resulting in an algorithm that is computationally cheap.

The main difference that this article [Sha16] proposes is to sample observations without replacement. The without replacement technique could be applied to many settings, such as stochastic gradient descent and least squares. A general sketch of the algorithm is:

Algorithm 1: General without-replacement Online Stochastic Gradient Descent.

```
Input: Initial vector $\mathbf{w}_0$, max number of iterations $t$, and stepsize $\eta_t$
Shuffle the $n$ observations randomly;
while $||\nabla f_i(\mathbf{w}_t)|| > \tau$ or current iteration $< t$ do
    Sequentially draw observation $i$ from the shuffled data;
    Compute $\nabla f_i(\mathbf{w}_t)$ using the single drawn observation;
    $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \nabla f_i(\mathbf{w}_t)$
end
return $\mathbf{w}_t$
```

The proposed method only requires to shuffle the data once, and then draw samples through multiple epochs (passes over the whole data set). It is noted that the algorithm could converge to an accurate solution even after one pass over the whole data.

3. Main Results

At a high level, the main results of the paper are convergence guarantees for without replacement sampling, with respect to three main types of algorithms: algorithms with online regret guarantees, stochastic gradient descent, and stochastic variance reduced gradient (SVRG). For this project, we focused on algorithms with online regret guarantees (convex Lipschitz functions) and stochastic gradient descent. Also, it is important to note that the paper proves these results under the circumstance that very few passes are made through the data and that reshuffling of the data is not needed. The end goal was to show that without replacement sampling will be no worse than with replacement sampling. With that being said, there are three main results that we will discuss.

First, the following two concepts are utilized to show the suboptimality of algorithms that make a single pass over a random permutation of m individual functions, given some convex functions on a convex domain $\mathcal{W}$. These two concepts are the **regret bound** that the algorithm attains in an adversarial online setting and the **transductive Rademacher Complexity** of $\mathcal{W}$, with respect to each individual function. An in-depth summary for these two concepts is given in section 4.

Next, the paper provides a key result using the above regret property, in the form of a convergence guarantee for convex Lipschitz functions. Specifically, consider a Lipschitz loss function $f_i(\cdot)$ on a convex domain $\mathcal{W}$ that involves an online algorithm which obtains a regret of $\mathcal{O}(\sqrt{T})$ on T functions. Then, it is proved that the suboptimality from using without replacement sampling is $\mathcal{O}(1/\sqrt{T})$. This convergence guarantee is the same as what is achieved from using with replacement sampling, up to constants.

The last main result that we examine is regarding a convergence guarantee for the stochastic gradient descent algorithm. Specifically, consider the stochastic gradient descent algorithm on some convex domain $\mathcal{W}$. When $f_i(\cdot)$ are Lipschitz smooth and the objective function $F(\cdot)$ is λ-strongly convex for $\lambda > 0$, the suboptimality is $\mathcal{O}(1/\lambda T)$ from without replacement sampling, which is on the order as for with replacement sampling.

As we see from these results, the paper is simply showing that without replacement sampling is no worse than with replacement sampling. However, considering the benefits of without replacement sampling as mentioned in section 1, these results provide evidence for why without replacement sampling should be used.

4. Key Proof Ideas

Here we present some of the key ideas that the article uses. The main novel approach that the article uses is to combine ideas from different fields such as stochastic optimization, adversarial online learning, and transductive learning. First, we will summarize the ideas behind the regret bound in online adversarial learning followed up a brief primer on transductive learning.

4.1. Regret Bound.

The first important concept is the regret bound in the adversarial online setting. To begin with, consider the convex and L-Lipschitz loss functions $f_1(\cdot), \ldots, f_m(\cdot)$ on a convex domain $\mathcal{W}$. The algorithm produces an iterate $\mathbf{w}_t \in \mathcal{W}$ by sequentially passing over a permuted ordering of the loss functions. Therefore, the regret bound in an adversarial online setting is defined as

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{w}) \leq R_T$$

for a sequence of T convex Lipschitz loss functions $f_1(\cdot), \ldots, f_T(\cdot)$ and any $\mathbf{w} \in \mathcal{W}$, where R_T scales sub-linearly in T. The $\sum_{t=1}^{T} f_t(\mathbf{w}_t)$ can be interpreted as the losses for one sample and $\sum_{t=1}^{T} f_t(\mathbf{w})$ is known as the loss on the entire data.

4.2. Transductive Learning.

An important general concept employed in the paper is the idea behind transductive learning. Broadly put, transductive learning is using the training data to make a prediction on specific cases of test data. The idea is best understood through a comparison to inductive learning. In fact, this is the familiar regime where we have (for example) supervised learning model that is trained on test data and utilizes that data to create a general rule or model which is then applied to make a prediction on the test data. In this sense, transductive learning focuses on predicting the test data and foregoes creating a general rule.

To further illustrate the connection between transductive learning and the process of bounding without replacement stochastic gradient algorithm consider the following setup borrowed from [EYP09]. Let $S_{m+u} = (x_i, y_i)_i^{m+n}$ a fixed set with $m+n$ points x_i in some arbitrary space with labels y_i. The learner is provided with the unlabeled sample of $X_{m+n} = x_{i_i}^{m+n}$. A set of m points is selected from X_{m+n} uniformly at random among all subsets of size m. These m points together with their labels are given to the learner as the training set. After renumbering, we have unlabeled points $X_m = \{x_1, \cdots, x_m\}$, and labeled set $S_m = (x_i, y_i)_i^m$. The set of unlabeled points $X_u = \{(x_{m+1}, \cdots, x_{m+n}\} = X_{m+n} \setminus X_m$ is the test set. Lastly, the goal is to predict the labels of the test points in X_u based on $S_m X_u$. The choice of the set of m points can be seen as drawing m points from X_{m+n} uniformly without replacement or a random permutation of the full sample X_{m+n} and choosing the first m as the training set. Note that both of these interpretations are reminiscent of the stochastic gradient descent algorithm using with-replacement sampling. That is, in a single epoch or pass of the data, we randomly shuffle the samples and apply the loss function.

Theorem 2 as demonstrated mathematically below is able to utilize this structure to create an upper bound on the magnitude of the expected suboptimality to the expected difference $\mathbb{E}[F_{1:t-1}(\mathbf{w}_t) - F_{t:m}(\mathbf{w}_t)]$. In Corollary 2, we then plug in that bound and have the necessary tool to narrow down the scope to specific example of convex and L-Lipschitz loss functions with potential constant regularization.

4.3. **Main theorems.** We now apply the above general ideas to show suboptimality guarantees in the convex Lipschitz loss scenario. The main steps of this proof are presented as Theorem 1 and Theorem 2. Particularly, Theorem 1 shows us that we can use the regret bound in order to reduce the expected suboptimality to the expected difference $\mathbb{E}[F_{1:t-1}(\mathbf{w}_t) - F_{t:m}(\mathbf{w}_t)]$. Theorem 2 in turn allows us to use the tools of Rademacher Complexity, in order to bound this expected difference. Lastly, Corollary 2 walks us through how to apply the results bound to arrive at convergence guarantees. Next, we turn to the strongly convex Lipschitz loss regime. In this case we follow a similar structure to construct the upper bound as we used for convex case. With the key difference being that in order to get a faster rate (and take advantage of strongly convex properties), we instead apply the concentration results on the gradient inner products, that is $\langle \nabla F_{1:t-1}(\mathbf{w}_t) - F_{t:m}(\mathbf{w}^*), \mathbf{w}_t - \mathbf{w}^* \rangle$. Note that in this case $\mathbf{w}^*$ is the optimal solution. The proof is concluded by utilizing results of Rademacher Complexity tools. Note that a more complete treatment of these proofs is included in Section 5 below.

4.4. **Lemma 1.** Lemma 1 is key to prove many of the results outlined in the article, especially Theorem 1. The key idea behind Lemma 1 is to use the fact that if we choose a permutation over m elements randomly we can partitioned it in two sets: 1 to $t-1$, and t to m; and get an expression for the expected value of the error of the algorithm that depends on the first partition and second one independently. The proof of Lemma 1 is provided in the appendix.

5. Full Proof

We now provide the full proofs for Theorem 1 and Theorem 2. We also outline the key steps to prove Theorem 3.

5.1. **Theorem 1 Proof.**

Theorem 1 is one of the key important works developed in the paper, as it provides and upper bound to the expected suboptimality of the algorithm. The proof of Theorem 1 relies on Lemma 1 and on the assumption that the algorithm has a regret bound R_T, that is for any sequence of T convex loss functions we have:

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{w}) \leq R_T \tag{1}$$

Let's begin the proof by working with the expression that we want to upper bound:

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} F(\mathbf{w}_t) - F(\mathbf{w}^*)\right] \tag{2}$$

With $F(\mathbf{w}) = \frac{1}{m}\sum_{i=1}^{m} f_i(\mathbf{w})$. We can simply add and remove the term $f_{\sigma(t)}(\mathbf{w}_t)$ and split the expectation (linear property) so we get:

$$\begin{aligned}\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} F(\mathbf{w}_t) - F(\mathbf{w}^*)\right] &= \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} f_{\sigma(t)}(\mathbf{w}_t) - F(\mathbf{w}^*)\right] - \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} F(\mathbf{w}_t) - f_{\sigma(t)}(\mathbf{w}_t)\right] \\ &= \frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T} f_{\sigma(t)}(\mathbf{w}_t) - f_{\sigma(t)}(\mathbf{w}^*)\right] - \frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T} F(\mathbf{w}_t) - f_{\sigma(t)}(\mathbf{w}_t)\right]\end{aligned} \tag{3}$$

We can use the regret bound to provide an upper bound for the first term, so we have that:

$$\frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T} f_{\sigma(t)}(\mathbf{w}_t) - f_{\sigma(t)}(\mathbf{w}^*)\right] \leq \frac{R_T}{T} \tag{4}$$

Now we can use Lemma 1 (see appendix) for the second term to get:

$$\frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T} F(\mathbf{w}_t) - f_{\sigma(t)}(\mathbf{w}_t)\right] = \frac{1}{mT}\sum_{t=1}^{T}(t-1)\mathbb{E}\left[\frac{1}{t-1}\sum_{i=1}^{t-1} f_i(\mathbf{w}_t) - \frac{1}{m-t+1}\sum_{i=t}^{m} f_i(\mathbf{w}_t)\right] \tag{5}$$

Adding up both terms we get to our desired result, which provides the upper bound (note that we remove the term when t=1, as for Lemma 1 it is equal to zero):

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} F(\mathbf{w}_t) - F(\mathbf{w}^*)\right] \leq \frac{R_T}{T} + \frac{1}{mT}\sum_{t=2}^{T}(t-1)\mathbb{E}\left[\frac{1}{t-1}\sum_{i=1}^{t-1} f_i(\mathbf{w}_t) - \frac{1}{m-t+1}\sum_{i=t}^{m} f_i(\mathbf{w}_t)\right]$$

5.2. **Theorem 2 Proof.** The motivation behind the follow proof stems from understanding that the above theorem and lemma allowed us to reduce the expected suboptimality to the expected difference $\mathbb{E}[F_{1:t-1}(\mathbf{w}_t)-F_{t:m}(\mathbf{w}_t)]$. In the following discussion, we will use Corollary 2 in order to upper bound it using $\mathbb{E}[\sup_{\mathbf{w}\in\mathcal{W}}(F_{1:t-1}(\mathbf{w}_t)-F_{t:m}(\mathbf{w}_t))]$. Note that here we are looking to split the loss function F into two groups. First, evaluated with the first $t-1$ samples and then again with the rest of the data indexed from t to m. Here we are interested in the magnitude of this difference given any $\mathbf{w}\in\mathcal{W}$. This proof continues as follows

Let $\mathcal{V}=\{(f_1(\mathbf{w}),\cdots,f_m(\mathbf{w}))|\mathbf{w}\in\mathcal{W}\}$.

Here we turn to the result provided by Corollary 2. The setting is to suppose $\mathcal{V}\subseteq[-B,B]^m$ for some $B\langle 0$. Let σ be a permutation over $1,\cdots,m$ chosen uniformly at random, and define $\mathbf{v}_{1:t-1}=\frac{1}{t-1}\sum_{j=1}^{t-1}v_{\sigma(j)}$, $\mathbf{v}_{t:m}=\frac{1}{m-t+1}\sum_{j=t}^{m}v_{\sigma(j)}$. Then

$$\mathbb{E}\left[\sup_{\mathbf{v}\in\mathcal{V}}(\mathbf{v}_{1:t-1}-\mathbf{v}_{t:m})\right]\leq\mathcal{R}_{t-1:m-t+1}(\mathcal{V})+12B\left(\frac{1}{\sqrt{t-1}}+\frac{1}{\sqrt{m-t+1}}\right)$$

Note that the above result it exactly what we need in terms of allowing us to now put a supremum operator within the expectation, thus effectively upper bounding it with the right hand side. Now again, consider the expected difference and add the upper bound introduced in Corollary 2.

$$\begin{aligned}\mathbb{E}[F_{1:t-1}(\mathbf{w}_t)-F_{t:m}(\mathbf{w}_t)]&\leq\mathbb{E}\left[\sup_{\mathbf{v}\in\mathcal{V}}(\mathbf{v}_{1:t-1}-\mathbf{v}_{t:m})\right]\\&\leq\mathcal{R}_{t-1:m-t+1}(\mathcal{V})+12B\left(\frac{1}{\sqrt{t-1}}+\frac{1}{\sqrt{m-t+1}}\right)\end{aligned}$$

Now, we can return to the results from Theorem 1 and plug in the upper bound.

$$\mathbb{E}[F(\bar{\mathbf{w}}_T)-F(\mathbf{w}^*)]\leq\frac{R_T}{T}+\frac{1}{mT}\sum_{t=2}^{T}(t-1)\left(\mathcal{R}_{t-1:m-t+1}(\mathcal{V})+12B\left(\frac{1}{\sqrt{t-1}}+\frac{1}{\sqrt{m-t+1}}\right)\right)$$

At this time we turn to Lemma 6 which states that if T,m are positive integers, $T\leq m$, then

$$\frac{1}{mT}\sum_{t=2}^{T}(t-1)\left(\sqrt{\frac{1}{t-1}}+\sqrt{\frac{1}{m-t+1}}\right)\leq\frac{2}{\sqrt{m}}$$

Armed with the above inequality we now recover the statement for Theorem 2. Remember that the full setting is to suppose that each $\mathbf{w}_t$ is chosen from a fixed domain $\mathcal{W}$, that the algorithm enjoys a regret bound R_T, and that $\sup_{i,\mathbf{w}\in\mathcal{W}}|f_i(\mathbf{w})|\leq B$.

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}F(\mathbf{w}_t)-F(\mathbf{w}^*)\right]\leq\frac{R_T}{T}+\frac{1}{mT}\sum_{t=1}^{T}(t-1)\mathcal{R}_{t-1:m-t+1}(\mathcal{V})+\frac{24B}{\sqrt{m}}$$

where $\mathcal{V}=\{(f_1(\mathbf{w}),\cdots,f_m(\mathbf{w}))|\mathbf{w}\in\mathcal{W}\}$.

Note that the above represents a generalized result. That is the above needs to be instantiated to a specific subset of loss functions and other entered specifications. We follow Corollary 1, and examine the above bound when considering loss functions to be convex and L-Lipschitz. We also consider a regularization parameter. Formally, the above can be expressed to suppose $\mathcal{W} \subseteq \{\mathbf{w} : ||\mathbf{w}|| \leq \bar{B}\}$ and each loss function f_i has the form $\ell_i(\langle \mathbf{w}, \mathbf{x}_i \rangle) + r(\mathbf{w})$ for some L-Lipschitz ℓ_i, $||\mathbf{x_i}|| \leq \mathbf{1}$, and fixed function r. Then:

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} F(\mathbf{w}_t) - F(\mathbf{w}^*)\right] \leq \frac{R_T}{T} + \frac{2(12+\sqrt{2})\bar{B}L}{\sqrt{m}}$$

Now we are almost ready to interpret and compare this bound to with-replacement sampling. But first we need to plug in the regret bound. In this case the typical regret bound is on the order of $\mathcal{O}(\bar{B}L\sqrt{T})$. By plugging that into the above equation we get:

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} F(\mathbf{w}_t) - F(\mathbf{w}^*)\right] \leq \frac{\bar{B}L}{\sqrt{T}} + \frac{2(12+\sqrt{2})\bar{B}L}{\sqrt{m}}$$

Note that in the above setting $T \leq m$ thus the above has the expected suboptimality on the order of $\mathcal{O}(\frac{\bar{B}L}{\sqrt{T}})$.

5.3. **Theorem 3 Proof.** Suppose $\mathcal{W}$ has a diameter $\mathcal{B}$, and that $F(\cdot)$, is λ-strongly convex on $\mathcal{W}$. Assume that $f_i(\mathbf{w}) = \ell_i(\langle \mathbf{w}, \mathbf{x} \rangle) + r(\mathbf{w})$ where $||\mathbf{x}_i|| \leq 1$, $r(\cdot)$ is a potential regularization and each ℓ_i is L-Lipshitz and μ-smooth on $\{z : z = \langle \mathbf{w}, \mathbf{b} \rangle, \mathbf{w} \in \mathcal{W}, ||\mathbf{x}|| \leq 1\}$. Also suppose that $\sup_{\mathbf{w}\in\mathcal{W}} ||\nabla f_i(\mathbf{w})|| \leq G$. Then for any $1 < T \leq m$, if we run SGD for T iterations with step size $\eta_t = \frac{2}{\lambda t}$ and c is a universal positive constant.

$$\begin{aligned}\mathbb{E}[F(\bar{\mathbf{w}}_T) - F(\mathbf{w}^*)] &\leq c \cdot \frac{(L+\mu B)^2\left(1+\log\left(\frac{m}{m-T+1}\right)\right) + G^2\log(T)}{\lambda T}\\ &\leq c \cdot \frac{((L+\mu B)^2 + G^2)\log(T)}{\lambda T}\end{aligned}$$

Note that Remark 1 tells us that the $\log(T)$ factor can be removed by instead of considering $\frac{1}{T}\sum_{t=1}^{T} F(\mathbf{w}_t)$ but instead an average over a suffix of the iterates $(\mathbf{w}_{\epsilon T}, \mathbf{w}_{\epsilon T+1}, \cdots, \mathbf{w}_{T+1})$ for some fixed $\epsilon \in (0,1)$, or by weighted averaging scheme.

Note that the proof for the above theorem is technically challenging and minor details and calculations are omitted in favor of a concise road map, rather than a step by step approach. For a better understanding, it is helpful to compare the proof for Theorem 2 given above. In Theorem 2, we used the transductive Rademacher Complexity to reduce the suboptimality of the algorithm to difference $F_{1:t-1}(\mathbf{w}_t) - F_{t:m}(\mathbf{w}_t)$. For the proof of Theorem 3, we instead apply the concentration results on the gradient inner products, that is $\langle \nabla F_{1:t-1}(\mathbf{w}_t) - F_{t:m}(\mathbf{w}^*), \mathbf{w}_t - \mathbf{w}^* \rangle$. Note that in this case, $\mathbf{w}^*$ is the optimal solution. Next, we apply transductive Rademacher Complexity tools so we can upper bound the inner product to roughly $\frac{\sqrt{\mathbb{E}[||\mathbf{w}_t - \mathbf{w}^*||^2]}}{\sqrt{t}}$. Lastly, remembering that in the strongly convex case, $||\mathbf{w}_t - \mathbf{w}^*||$ will decrease to zero with t, allowing us to get the suboptimality on the order of $\mathcal{O}(1/T)$ in the strongly convex case compared to $\mathcal{O}(1/\sqrt{T})$.

The first step of the proof is to express the stochastic gradient descent algorithm update as $\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}(\mathbf{w}_t - \eta_t f_{\sigma(t)}(\mathbf{w}_t))$. Where $\eta_t \geq 0$ are fixed step sizes, $\Pi_{\mathcal{W}}$ is a project on $\mathcal{W}$. Note that the algorithm is invariant to shifting the coordinates or shifting all loss functions by a constant. Thus the objective function $F(\cdot)$ is minimized at 0 and $F(\mathbf{0}) = 0$. Using this definition and convexity of $\mathcal{W}$ we have

$$\begin{aligned}\mathbb{E}[||\mathbf{w}_{t+1}||^2] &= \mathbb{E}\Big[||\prod_{\mathcal{W}}(\mathbf{w}_t - \eta_t \nabla f_{\sigma(t)}(\mathbf{w}_t))||^2\Big] \leq \mathbb{E}\Big[||\mathbf{w}_t - \eta_t \nabla f_{\sigma(t)}(\mathbf{w}_t)||^2\Big] \\ &\leq \mathbb{E}\Big[||\mathbf{w}_t||^2\Big] - 2\eta_t \mathbb{E}\big[\langle \nabla f_{\sigma(t)}(\mathbf{w}_t), \mathbf{w}_t\rangle\big] + \eta_t^2 G^2 \\ &= \mathbb{E}\Big[||\mathbf{w}_t||^2\Big] - 2\eta_t \mathbb{E}\big[\langle \nabla F(\mathbf{w}_t), \mathbf{w}_t\rangle\big] + 2\eta_t \mathbb{E}[\langle \nabla F(\mathbf{w}_t) - \nabla f_{\sigma(t)}(\mathbf{w}_t), \mathbf{w}_t\rangle] + \eta_t^2 G^2\end{aligned}$$

Now we can look at the individual terms in the above equation's right hand side. We utilize the definition of strong convexity, and since $F(\cdot)$ is λ-strongly convex to get the following $\langle F(\mathbf{w}_t), \mathbf{w}_t\rangle \geq F(\mathbf{w}_t) + \frac{\lambda}{2}||\mathbf{w}||^2$. Plugging in the above we get

$$\mathbb{E}[F(\mathbf{w}_t)] \leq \left(\frac{1}{2\eta_t} - \frac{\lambda}{2}\right)\mathbb{E}\big[||\mathbf{w}_t||^2\big] - \frac{1}{2\eta_t}\mathbb{E}\big[||\mathbf{w}_{t+1}||^2\big] + \mathbb{E}[\langle \nabla F(\mathbf{w}_t) - \nabla f_{\sigma(t)}(\mathbf{w}_t), \mathbf{w}_t\rangle] + \frac{\eta_t}{2}G^2$$

Then we turn to the third term in the right hand side above. Since $\mathbf{w}_t$ depends only on $\sigma(1), \cdots, \sigma(t-1)$, we use Lemma 1 and Cauchy-Schwartz inequality. We finally arrive at

$$\mathbb{E}[F(\mathbf{w}_t)] \leq \left(\frac{1}{2\eta_t} - \frac{\lambda}{4}\right)\mathbb{E}\big[||\mathbf{w}_t||^2\big] - \frac{1}{2\eta_t}\mathbb{E}\big[||\mathbf{w}_{t+1}||^2\big] + \frac{2(19L + 2\mu B)^2}{\lambda m^2}\left(t - 1 + \frac{(t-1)^2}{m - t + 1}\right) + \frac{\eta_t}{2}G^2$$

Averaging both sides over $t = 1, \cdots, T$, and using Jensen's inequality, we have:

$$\begin{aligned}\mathbb{E}\Big[\frac{1}{T}\sum_{t=1}^{T} F(\mathbf{w}_t)\Big] \leq \frac{1}{2T}\sum_{t=1}^{T}\mathbb{E}\big[||\mathbf{w}_t||^2\big]\left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} - \frac{\lambda}{2}\right) + \\ + \frac{2(19L + 2\mu B)^2}{\lambda m^2 T}\sum_{t=1}^{T}\left(t - 1 + \frac{(t-1)^2}{m - t + 1}\right) + \frac{G^2}{2T}\sum_{t=1}^{T}\eta_t\end{aligned}$$

Lastly, setting $\eta_t = 2/\lambda t$ and considering the fact that $\sum_{t=1}^{T}\frac{1}{t} \leq \log(T) + 1$, we obtain

$$\mathbb{E}\big[F(\bar{\mathbf{w}}_T)\big] \leq \frac{2(19L + 2\mu B)^2(\frac{3}{2} + \log(\frac{m}{m-T+1}))}{\lambda T} + \frac{2G^2(\log(T) + 1)}{\lambda T}$$

Under the prior assumption that $F(w^*) = F(0) = 0$ we got the full derivation of Theorem 3, which finalizes the proof.

References

[EYP09] Ran El-Yaniv and Dmitry Pechyony, *Transductive rademacher complexity and its applications*, J. Artif. Int. Res. **35** (2009), no. 1, 193–234.

[Sha16] Ohad Shamir, *Without-replacement sampling for stochastic gradient methods*, Advances in Neural Information Processing Systems **29** (2016).

Appendix A. Additional Details of the Proofs

A.1. **Lemma 1 Proof.** The conditions for the Lemma are the following: let σ be a uniformly random permutation over 1 to m. Let s_1,...,s_m be scalar random variables that conditioned on $\sigma(1), ..., \sigma(t-1)$ are independent of $\sigma(t), ..., \sigma(m)$. The key idea of the proof is that if σ is chosen randomly uniform and conditioned over $\sigma(1), ..., \sigma(t-1)$, it will have a uniform distribution over $\{t, ..m\}$. Now we have that:

$$\mathbb{E}\left[\frac{1}{m}\sum_{i=1}^{m} s_i - s_{\sigma(t)}\right] = \mathbb{E}\left[\frac{1}{m}\sum_{i=1}^{m} s_{\sigma(i)} - \frac{1}{m-t+1}\sum_{i=t}^{m} s_{\sigma(i)}\right] \tag{6}$$

If we split the first summation into the sum of 1 to t-1 and t to m we have:

$$\begin{aligned}\mathbb{E}\left[\frac{1}{m}\sum_{i=1}^{m} s_i - s_{\sigma(t)}\right] &= \mathbb{E}\left[\frac{1}{m}\sum_{i=1}^{t-1} s_{\sigma(i)} - (\frac{1}{m-t+1} - \frac{1}{m})\sum_{i=t}^{m} s_{\sigma(i)}\right] \\ &= \mathbb{E}\left[\frac{1}{m}\sum_{i=1}^{t-1} s_{\sigma(i)} - \frac{t-1}{m(m-t+1)}\sum_{i=t}^{m} s_{\sigma(i)}\right] \\ &= \frac{t-1}{m}\mathbb{E}\left[\frac{1}{t-1}\sum_{i=1}^{t-1} s_{\sigma(i)} - \frac{1}{m-t+1}\sum_{i=t}^{m} s_{\sigma(i)}\right]\end{aligned} \tag{7}$$

Which proves the desired Lemma. We note that for the case of t=1 we have that $\mathbb{E}\left[\frac{1}{m}\sum_{i=1}^{m} s_i - s_{\sigma(t)}\right] = 0$.

Email address: pmbusch@ucdavis.edu

Email address: gkgambetta@ucdavis.edu

Email address: ikulinka@ucdavis.edu