STA243 Final Project

Zhiwei Wang[*] Yuxiang Lin[†] Shan Gao[‡]

1 Introduction

1.1 Related work

To begin with, there exist classic convergence results for processing the individual functions in some determined order by sampling without replacement. However, these can be exponentially worse than those obtained using random without-replacement sampling, and this gap is inevitable. Then, Recht and Re shew that when the individual functions themselves are assumed to be generated i.i.d. from some distribution, without-replacement sampling is always better (or at least not substantially worse) than with replacement sampling on a given dataset. In a recent breakthrough, Gurbuzbalaban et al proved that under smoothness and strong convexity assumptions, without-replacement sampling is strictly better than with-replacement sampling, after sufficiently many passes over the data. In recent years, several algorithms have been introduced which could attain a high-accuracy solution after a small number of passes through the whole data set. Among them there is more sophisticated stochastic gradient approach called Stochastic Variance Reduction Gradient (SVRG).

1.2 Description of paper

Most machine learning problems aim to minimize a convex empirical risk

$$\min_{\mathbf{w}\in\mathcal{W}} F(\mathbf{w}) = \frac{1}{m}\sum_{i=1}^{m} f_i(\mathbf{w})$$

Stochastic gradient methods are widely used in large-scale applications. Update to w_t based on $\nabla f_i(w_t)$ is usually computationally cheap. Moreover, when the sampling is done independently and uniformly at random, $\nabla f_i(w_t)$ is an unbiased estimator of the true gradient $\nabla F_i(w_t)$, which allows for good convergence guarantees after a reasonable number of iterations. However, unlike with-replacement sampling, the functions of without-replacement sampling processed at every iteration are not statistically independent, and their correlations are difficult to analyze.The paper's aim is to show that stochastic gradient descent without replacement is not significantly worse than that under with-replacement framework.

In our understanding, the author proposes a novel framework for algorithms performing a single pass over a random permutation of m individual functions with convex functions on some convex domain $\mathcal{W}$. The algorithm's suboptimality can be characterized by a combination of two quantities, each from a different field. First, the regret which the algorithm can attain in the setting of adversarial online convex optimization. Second, the transductive Rademacher complexity

[*] wizwang@ucdavis.edu
[†] yuxlin@ucdavis.edu
[‡] angao@ucdavis.edu

of $\mathcal{W}$ with respect to the individual functions, a notion stemming from transductive learning theory.

During the theoretical derivation part, the author consider the convergence guarantee of without-replacement stochastic gradient descent algorithm on several different scenario. Firstly, if each individual loss function f_i corresponds to a convex Lipschitz loss of a linear predictor, and the algorithm belongs to the class of algorithms which in the online setting attain $\mathcal{O}(\sqrt{T})$ regret on T such functions (which includes, for example, stochastic gradient descent), then the suboptimality using without-replacement sampling, after processing T functions is the same as that obtained using with-replacement sampling up to constants. Secondly, if the objective function $F(\cdot)$ is λ-strongly convex, and the functions $f(\cdot)$ are also smooth, then the suboptimality bound becomes $\mathcal{O}(1/\lambda T)$, which matches the guarantee under with-replacement framework except the dependence on some parameters hidden in $\mathcal{O}(\cdot)$.

As an experiment for the theoretically derived convergence guarantee above, the author select Stochastic Variance Reduction Gradient Descent algorithm (SVRG) as the representative of fast stochastic algorithms. The result shows that, under the problem of regularized least squares, without-replacement SVRG has similar convergence guarantee with the with-replacement one. An interesting implication is also revealed that if the problem's condition number (μ/λ) is smaller than the data size, without-replacement SVRG can converge to an arbitrarily accurate solution (even up to machine precision), without the need to reshuffle the data.

A further application of the without-replacement SVRG result is in the field of distributed learning. By simulating without-replacement SVRG on data randomly partitioned between several machines, we get a nearly optimal algorithm for regularized least squares, in terms of communication and computational complexity, as long as the condition number is smaller than the data size per machine.

2 Methodology

2.1 Regret and Transductive Rademacher Complexity

The first methodological contribution of this paper is that the suboptimality of the algorithms performing a single pass over a random permutation of m individual functions with convex functions on some convex domain $\mathcal{W}$ can be characterized by a combination of two quantities:

1. Regret. It comes from the setting of adversarial online convex optimization. It measures the difference between the parameter we made and the optimal one.

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

2. Transductive Rademacher Complexity.It stemmed from transductive learning theory.

$$\mathcal{R}_{s,u}(\mathcal{V}) = (\frac{1}{s} + \frac{1}{v}) \cdot \mathbb{E}_{r_1,\cdots,r_m}[\sup_{\mathbf{v}\in\mathcal{V}} \sum_{i=1}^{m} r_i v_i]$$

Here $\mathcal{V}$ is a set of vectors $\mathbf{v} = (v_1, \cdots, v_m)$ in $\mathbb{R}^m$. s and u are positive intergers whose sum is m. This quantity is an important measurement for the richness of the set $\mathcal{V}$.

2.2 without-replacement SVRG:

The SVRG algorithm using without-replacement sampling on a dataset of size m is composed of multiple epochs, each involving one gradient computation on the entire data set. In the paper, the author has proved that without-replacement SVRG has similar convergence guarantee compared to the with-replacement one.

This result provides an implication in empirical work that as long as the condition number is smaller than the data size, SVRG can be used to obtain a high-accuracy solution, without the need to reshuffle the data: Only a single shuffle at the beginning suffices, and the algorithm terminates after a small number of sequential passes (logarithmic in the required accuracy).

2.3 Without-Replacement SVRG to distributed learning:

Findings from without-replacement SVRG intuitively leads to the application related to distributed learning. An important variant of the problems we discussed so far is when the training data is split between different machines, who need to communicate in order to reach a good solution. One existing problem in with-replacement SVRG distributed learning is that the communication between machines is efficient only when $1/\lambda$ is relatively small (usually between $\sqrt{m}$ and m). However, the situation immediately improves if we can use a without-replacement version of SVRG, which can easily be simulated with randomly partitioned data. No data points need to be sent across machines, even if $1/\lambda$ is large.

The procedure of distributed learning of without-replacement SVRG is described below:

(a) The data is assigned to each machine by sampling without replacement from $f_1(\cdot), ..., f_m(\cdot)$.

(b) Then, each machine splits its data to batches of size T.

(c) initialize $\tilde{\mathbf{w}}_1$ at 0; create a matrix to store all $\mathbf{w}_0, ..., \mathbf{w}_{T+1}$

(d) for each batch s:
The initial $\mathbf{w}$ is set as $\tilde{\mathbf{w}}_s$.
With the dataset of size m, gradient $\tilde{n}$ is calculated as

$$\nabla F_i(\tilde{\mathbf{w}}_s) = \frac{1}{m}\sum_{i=1}^{m} \nabla f_i(\tilde{\mathbf{w}}_s)$$

where $f_i(\mathbf{w}) = \frac{1}{2}(\langle \mathbf{w}, \mathbf{x}_i \rangle - y_i)^2 + \frac{\hat{\lambda}}{2}\left|\left|\mathbf{w}\right|\right|^2$

(e) For each batch of each machine, $t = 1, ..., T$, we update the values of $\mathbf{w}$ based on the gradient we just calculate with the formula:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta(\nabla f_{\sigma(t)}(\mathbf{w}_t) - \nabla f_{\sigma(t)}(\tilde{\mathbf{w}}_s) + \tilde{n})$$

(f) We move to update $\tilde{\mathbf{w}}$. One machine compute gradients with respect to one of its batches.

$$\tilde{\mathbf{w}}_{s+1} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{w}_t$$

(g) $\tilde{\mathbf{w}}_{s+1}$ is distributed to all machines. When we run out all batches of a machine, we move to the next machine and repeat the process above.

3 Theoretical Results

3.1 Convergence Guarantee of Without-replacement under Convex Lipschitz Loss

The paper shows that for linear predictor, if each function $f_i(\cdot)$ corresponds to a convex Lipschitz loss and the algorithm belongs to the class of algorithms which in the online setting attain $\mathcal{O}(\sqrt{T})$ regret on T such functions, then without replacement sampling method will yield an suboptimality of $\mathcal{O}(\frac{1}{\sqrt{T}})$ on processing T functions.

There are several basic assumptions below:

- Loss function $f_1(\cdot), f_2(\cdot), \cdots, f_m(\cdot)$ are convex and L-Lipschitz over convex domain $\mathcal{W}$.
- The algorithm sequentially goes over some permuted ordering of the losses and produces an iterate $\mathbf{w}_t \in \mathcal{W}$ before processing the t-th loss function.
- There is a R_T regret bound in the adversarial online setting:

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

First, given the definition of Regret bound:

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

Then, we apply Transductive Rademacher Complexity here by setting $\mathcal{V} = \{(f_1(\mathbf{w}), \cdots, f_m(\mathbf{w}))|\mathbf{w} \in \mathcal{W}\}$. The author proves that the right hand side is at most:

$$\frac{R_T}{T} + \frac{1}{mT}\sum_{t=2}^{T}(t-1)\mathcal{R}_{(t-1):(m-t+1)}(\mathcal{V}) + \frac{24B}{\sqrt{m}}$$

As the final result, the paper shows as follows:

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

Take online gradient descent as the example. Since the regret bound are on the order of $\mathcal{O}(\bar{B}L\sqrt{T})$ for online gradient descent, so the expected suboptimality is $\mathcal{O}(\frac{\bar{B}L}{\sqrt{T}})$ which is similar to that of with-replacement one up to some constants.

3.2 Convergence Guarantee of Without-replacement under λ-Strong Convex Loss

In this section, the author considers a strong assumption on the loss functions:

Each $f_i(\mathbf{w})$ is a Lipschitz and smooth loss of a linear predictor w, possibly with some regularization and the objective function $F(\cdot)$ is λ-strongly convex and L-smooth.

In this scenario, if we apply stochastic gradient descent with-replacement algorithm, previous work shows that their averaged results in expected suboptimality is $\tilde{\mathcal{O}}(G^2/\lambda T)$, where G^2 is an upper bound on the expected squared norm of g_t. The author studied the similar condition for stochastic gradient descent without replacement with the further assumption:

When the iterations run for T times with step size $2/\lambda T$, we can derive the an upper bound for the expected optimality of a single $\mathbf{w}_t$ (similar methodology as section 3.1):

$$\mathbb{E}(\frac{1}{T}\sum_{t=1}^{T} F(\mathbf{w}_t) - F(\mathbf{w}^*)) \leq c\frac{((L+\mu B)^2 + G^2)log(T)}{\lambda T}$$

This result is similar to the with-replacement case, up to numerical constants and the additional $(L+\mu B)^2$ term in the numerator. This extra constant could be unnecessary because in some cases G^2 is dominant.

Through the work above, the author has proved that, in strongly-convex objective function, without-replacement stochastic gradient descent algorithm could increase its converging rate from $\mathcal{O}(1/\sqrt{T})$ to $\mathcal{O}(1/T)$. One thing needs attention is that this rate is constrained to the case up to $T = m$.

3.3 Without-Replacement SVRG for Least Squares

In this part, the author takes SVRG as a representative of stochastic gradient descent algorithms as the analysis target because, under specific algorithm framework, we could derive more specific conclusion on the convergence guarantee difference between with-replacement and without-replacement setting. Here we consider Regularized Least Square probelms.

Considering the regularized least mean squares problem which satisfies the assumption of strong convexity and smoothness. For convenience, we also assume:

$$\|\mathbf{x}_i\| \leq 1, \quad |y_i| \leq 1, \quad \lambda \leq 1$$

This assumption does not lose generality because we can always re-scale the loss functions by an appropriate factor. The author proves that if the learning rate η, number of stochastic iterations T and the data size m satisfies:

$$\eta = \frac{1}{c}, \quad T \geq \frac{9}{\eta\lambda}, \quad m \geq clog^2(\frac{64dmB^2}{\lambda\epsilon})T$$

Under this condition the algorithm attains an o-accurate solution after $\mathcal{O}(\frac{log(1/\epsilon)}{\lambda})$ stochastic iterations of without-replacement sampling and $\mathcal{O}(log(1/\epsilon))$ sequential passes over the data. This result is similar to that of with-replacement SVRG.

4 Experimental Details

The first experiment is to show that without-replacement SVRG will not be significantly worse (in a worst-case sense) than with-replacement SVRG. The second experiment is to confirm if distributed-learning without-replacement SVRG could get a nearly optimal result as without-replacement SVRG when all settings are appropriate.

4.1 Setup

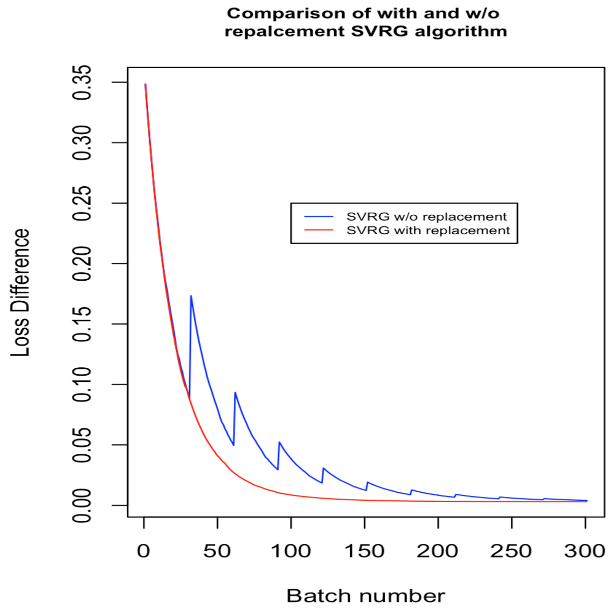
We generate 600 samples with dimension 3. Every element in X takes a uniform value from $(\frac{-1}{\sqrt{3}}, \frac{1}{\sqrt{3}})$ and we set $y = X\beta + \epsilon$ where every element in ϵ is generated from (0, 0.1) normal distribution. The true value for β is [1,1,1] (in order to satisfy the assumptions in 3.3). Since it is a regularized least mean squares problem, we use Ridge regression model to estimate $\hat{\lambda}$, which is 0.0008.

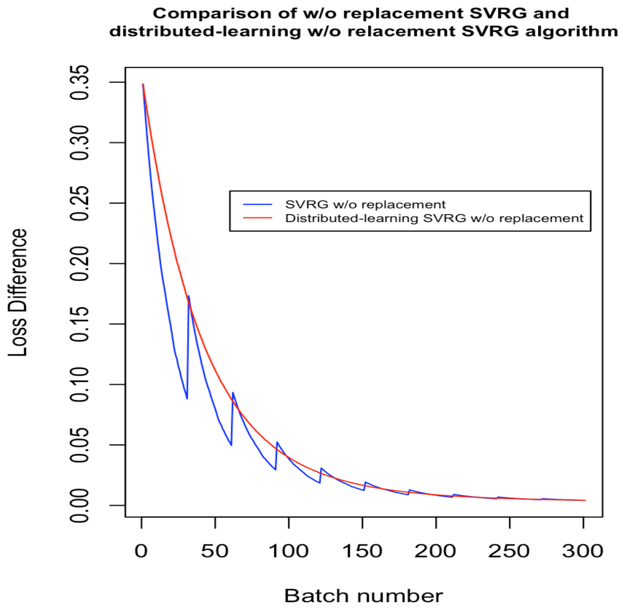
(a) without-replacement SVRG:
The algorithm using without-replacement sampling on a dataset of size m is composed of multiple epochs, each involving one gradient computation on the entire data set.
The number of epochs S are set 10 and learning rate η=0.01. We updated T=600 times within each epoch.
The same parameters are set for with-replacement SVRG.

(b) Without-Replacement SVRG to distributed learning:
Here we split training data between different machines. For each machine, we need to separate data into several mini-batches. Then β is updated continuously.
We assigned 3 machines with 10 batches for each. The number of passes S is 10 and the learning rate η is 0.01. Within each batch, the number of data is T=20.

4.2 Results

The final result for without-replacement SVRG is [0.8752778, 0.8800923, 0.8782932].
The final result for with-replacement SVRG is [0.8904606, 0.8954379, 0.8915736].
The final result for without-replacement distributed-learning SVRG is [0.8728255, 0.8787120, 0.8771726].





As we can see from the result, without and with replacement SVRG framework have close performances when iteration batches exceed 100. Also, the result of distributed-learning for without-replacement SVRG is simialr to without-replacement SVRG which implies that distributed-learning of SVRG could attain nearly optimal result. In conclusion, the results of our experiments are consistent with the paper's claims.