Improving SGD via Mini-Batch and Adaptive Learning

Anurag Beniwal Xinzhou Guo

1. Introduction

Nowadays, many real optimization problems consider big data sets and streaming data in online learning problems. Both problems are intractable with common optimization methods. They usually need to evaluate the whole data set, which could be very slow for extremely big data sets or even not accessible for the data coming in streams.To overcome these difficulties, stochastic approximation is introduced, which applies random evaluation. Stochastic Gradient Descent (SGD) is a special case based on gradient descent and a widely used technique in large scale data analysis problems.

The key idea for SGD is to compute gradient on just one point (a sample from the data set) in a particular iteration of gradient descent instead of computing n (size of the data set) gradients in each iteration to make an update.Compared with the common optimization methods,SGD could significantly reduce per iteration time for large data set and be more suitable to the online learning tasks,i.e,the streaming data setting. However,SGD introduces noise in the computation of gradients due to the sampling and corresponding random evaluation.This,in turn,may lead to more iterations and time to achieve the best solution and may even converge to a sub optimal point in practice with common stop rules.

In this project,we will mainly study two methods to reduce the noise in the computation of gradients and improve SGD:(1) Sample b(b>1) data points instead of one (Mini-Batch).(2) Adaptively learn the best sampling distribution at each step instead of fixing on the uniform distribution on data (AW-SGD).We will apply the above two methods to least square problem in linear regression but this method could be easily extended to other problems in Machine learning with slight modifications.Through real data analysis,we study the following properties: (1) The best choice of batch size b in Mini-Batch (2) The convergence rate and the solution that AW-SGD achieves under different assumptions of the sampling distribution (3)The convergence rate and the solution that AW-SGD,SGD and Mini-Batch achieve on different size of the data. (4)the prediction error that AW-SGD and Mini-Batch achieve on real data.

We will first introduce the motivation and algorithm of Mini-Batch Stochastic Gradient Descent (Mini-Batch) and Adaptive Weighted Stochastic Gradient Descent (AW-SGD) in section 2.Then, we will illustrate and compare their performance under different choices of tuning parameters (b and sampling assumptions) and data setting (different size of the data and prediction) in section 3.At the end, we will summary and analyze the results in section 4.

2. Method

Stochastic Gradient Descent is used to minimize the function

$$\gamma(\omega) = E_{x\sim P}[f(x;\omega)] = \int_{\mathcal{X}} f(x;\omega)dP(x)$$

where P is a known distribution,f is the objective function and ω is the parameter. Most optimization problems in statistics,such as least square and logistic regression,can be written in this form when we set f as the corresponding log-likelihood and P as the empirical

distribution on $\mathcal{X}$.SGD approximate the gradient at steps t by: (1)sample $x_t \in \mathcal{X}$ according to P; (2)do gradient descent $w_{t+1} = w_t - \eta_t \nabla_w f(w_t; x_t)$, where η_t is the step size. The approximate gradient in SGD, $\nabla_w f(w_t; x_t)$ is unbiased estimator to the gradient, $\nabla_w \gamma(w_t)$ and the convergence properties of SGD are directly related to its variance. Usually, the variance would be very huge. To reduce the noise and improve the performance of SGD, Mini-Batch and AW-SGD are proposed.

2.1. **Mini-Batch Stochastic Gradient Decent.** Motivated by the fact that averaging can reduce variance,Mini-Batch Stochastic Gradient Decent (Mini-Batch) is proposed. Its key idea is to sample b points instead of 1, and then use the average of the gradients calculated on each point as the approximation to the gradient. The algorithm is showed below:

Mini-Batch:
Require: Batch size b , learning rate $\{\eta_t\}$
Require: Initialize w_0
for t = 0,1,...T-1 **do**
Pick b points from $\mathcal{X}$ according to P and denote B_t as the corresponding set
$$w_{t+1} = w_t - \eta_t \frac{\sum_{x_j \in B_t} \nabla_w f(w_t; x_j)}{b}$$
end for

Obviously,batch size b, $\{\eta_t\}$ are the tuning parameters for Mini-Batch.$\{\eta_t\}$ can be decided with the Lipschiz constant.However, how batch size affects the performance is still not clear.We will study these properties and see how Mini-Batch improve SGD with the best choice of the tuning parameters under different size of data set.

2.2. **Adaptive Weighted-Stochastic Gradient Decent.** As discussed in the previous section, SGD estimate,though computationally efficient, has a high variance and makes the objective function fluctuate.Different from Mini-Batch, Adaptively Weighted-Stochastic Gradient Descent (AW-SGD) reduces the variance by adaptively learning the best sampling distribution of the points on which gradient is computed.

The motivation for AW-SGD is the importance sampling. Considering a family distribution $\{Q_\tau\}$,where τ is the parameter,and assuming that they are absolutely continuous with respect to P, denoted by $q(x;\tau) = \frac{dQ_\tau(x)}{dP(x)}$. If we sample x according to Q_τ, then $\frac{f(x;\omega)}{q(x;\tau)}$ is also a reasonable and unbiased approximation to $\nabla_w \gamma(w)$.It is very natural to minimize the variance $\sum(\omega_t, \tau) := Var_\tau[\frac{f(x;\omega_t)}{q(x;\tau)}]$.Since for any reasonable gradient descent methods and optimization problems,after some iterations, ω_t will be very close to the solution, ω^*, so we only need to minimize $\sum(\omega^*, \tau)$. We can achieve that by approximately calculating the gradient $\sum(\omega^*, \tau)$ with respect to τ and add one step gradient descent to update τ in each loop.Also,AW-SGD can be combined with mini batch.The algorithm is given by:

AW-SGD:
Require: Initial target and sampling parameter vectors w_0 and τ_0
Require: Learning rates $\{\rho_t\}$ and $\{\eta_t\}$
Require: batch size b
for t = 0,1,...T-1 **do**
Sample b points from Q_{τ_t} and denote B_t as the corresponding set
$d_t = (\sum_{x_j \in B_t} \frac{\nabla_w f(x_j; w_t)}{q(x_j; \tau_t)})/b$
$w_{t+1} \leftarrow w_t - \rho_t d_t$
$\tau_{t+1} = \tau_t + \eta_t \|d_t\|^2 \sum_{j \in B_t} \nabla_\tau log\{q(x_j; \tau_t)\}/b$
end for

$\{\rho_t\}$, $\{\eta_t\}$, batch size b and sampling assumption $\{Q_\tau\}$ are tuning parameters for AW-SGD. $\{\rho_t\}$ can be chosen with Lipschiz constant and $\{\eta_t\}$ has some suggested value at the existing papers. We will study batch size b and the choice of $\{Q_\tau\}$ later and compare AW-SGD with Mini-Batch and SGD with the corresponding best choice of the tuning parameter under different size of the data set.

Applications:
1) **Large Scale Learning:** In large scale settings,where number of parameters and data size are large (for ex. Neural Networks),the computation of gradient is very expensive in each iteration but SGD also could get very slow.In such cases,it is important to sample observations that have real signal and do not have redundant information.AW-SGD can help in sampling those observation where features have non redundant signals.Curriculum learning in Neural Nets is a special case of AW-SGD ,AW-SGD is however more general.
2) **Class Imbalance problems:** Importance sampling based gradient estimator could be more useful in classification problems where their is high class imbalance.AW-SGD will give importance to classes that have less number of observations and will make learning faster.
3) **Online learning:** In Online learning problems, Importance sampling together with stochastic gradient descent could give estimators with less bias and could potentially be helpful in achieving lower regret bound.

3. Real Data Analysis

In order to study the properties and the choice of the tuning parameters and compare the performance of the above methods, we run the algorithms on the Boston housing prices dataset.The data has 506 rows and 14 columns, denoting $x_i = (y_i, z_i)$, where i=1...506, y_i is the price and z_i is the covariates at the ith row. To model their linear relationship and estimate the coefficients, we use least square, which can be easily changed to the optimization problems SGD targeted on.

First, we would introduce the concrete form of the formulations in the above algorithms under least square. It is easy to see that

$$\nabla_\omega f(x; \omega) = -2z^T(y - z^T\beta)$$

For the sampling distribution,we will consider that Q_τ, where $\tau = (\tau_1, ..., \tau_{506})$ and the probability sampling the i^{th} sample is $\frac{e^{\tau_i}}{\sum_j e^{\tau_j}}$.This is the full sampling distribution family which includes all the possible sampling possibilities for $\mathcal{X}$. The corresponding derivative is

$$\nabla_\tau log\{q(x_i; \tau)\} = e_i - q_1(\tau)$$

where e_i is the vectors with 1 at index i and $q_1(\tau)$ is the sampling probability vector under Q_τ

Second, we standardize the data beforehand. Since SGD evaluate the gradient randomly, any results showed below would be the average result of 3 runs in 1 minute. Last, we fix some tuning parameters that have been studied before. For η_t, we will choose 0.02, which is decided by calculating Lipschiz Constant. For ρ_t, we will choose 0.005, which is suggested by the existing papers.

3.1. **Different Batch Size for Mini-Batch and AW-SGD.** The first tuning parameter we want to study is the batch size,for Mini-Batch as well as AW-SGD.Accuracy increases with higher batch sizes up to an extent with a minor increment after that point.However,larger batch size will also cost more time for one iteration. Thus,given this trade off,we should find that sweet spot where we see a significant increase in accuracy without bearing significant computational cost.The simulation for Mini-Batch is showed below:

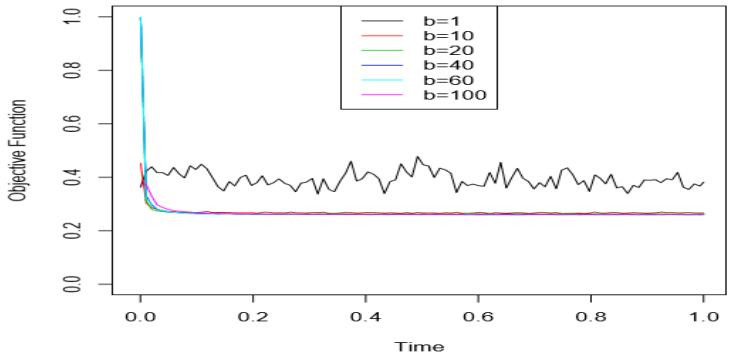

FIGURE 1. The convergent rate for Mini-Batch under different batch size

From Figure 1,we can notice that SGD(b=1) performs very bad as it is very noisy and does not converge in 1 min. b=100 is slightly worse than the other four,which can be expected from the above analysis that with too large batch size, it may cost too much time for one iteration and slow down the convergence.For b=10, 20, 40 and 60, the performance are very similar, which shows that the choice of the batch size is quite robust for Mini-Batch. For later analysis, we will consider b=10 (2 percent batch ratio) as the best choice for Mini-Batch. For AW-SGD, we also do similar analysis and get the result below:

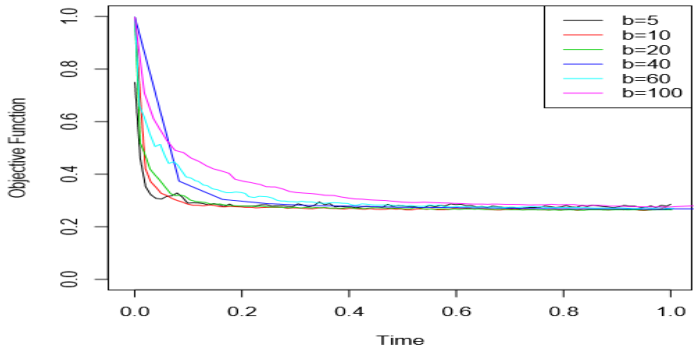

FIGURE 2. The convergence rate for AW-SGD with different batch sizes

From Figure 2, we can notice that batch size affect the convergent rate of AW-SGD significantly, so the choice of batch size is not robust in this case. In general, similar to Mini-Batch, too large b will slow down the convergence and We would consider b=10 as the best batch size for AW-SGD. Also, it is obvious that both Mini-Batch and AW-SGD do improve SGD a lot.

Notice that AW-SGD is based on Mini-Batch,a natural questions is that with the same batch size, will AW-SGD have some advantages? we compare them in Figure 3 and Figure 4. We can see that with smaller batch size, the AW-SGD will improve Mini-Batch more by reducing the noise. The reason is that with larger batch size, the sampling distribution become less and less important. For example, if we sample 506 points without replacement, then for any sampling distribution, we would get the same results.

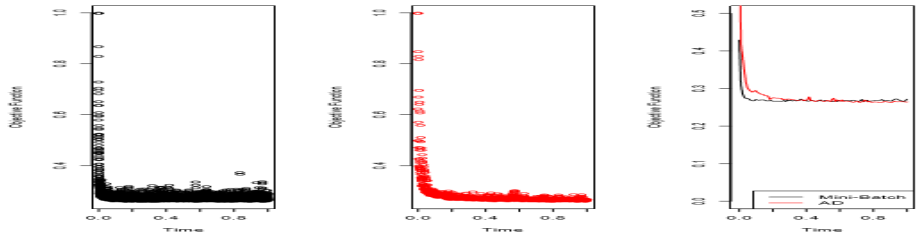

FIGURE 3. The performance for AW-SGD and Mini-Batch with batch-size 10

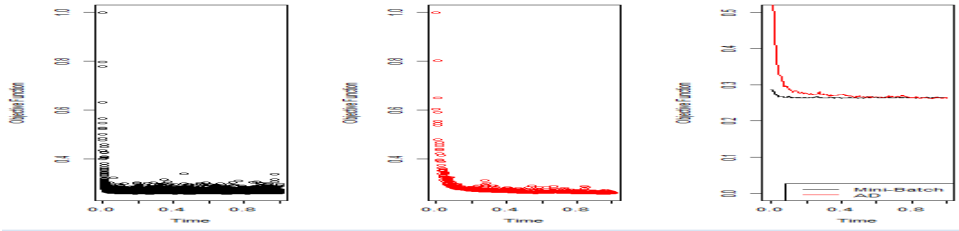

FIGURE 4. The performance for AW-SGD and Mini-Batch with batch-size 30

3.2. **Different Sampling Assumption for AW-SGD.** Notice that we use full sampling with 506 parameters, the convergence to the best sampling distribution may be very slow. A natural idea to improve AW-SGD is to use simple sampling family. Here, we consider that dividing the data into two parts $y_i > median(y)$ and $y_i \leq median(y)$. The probability for the first group is that $\frac{e^{\tau_1}}{n_1 * (e^{\tau_1} + e^{\tau_2})}$, n_1 is the size of the first group. Similar probability will be applied to the other group. Now, this family only contains 2 parameters. We choose batch size as 10 and compare their performance in Figure 5. We can notice that with simple family, the objective function will go down faster than full sampling at the beginning, but in the end, it is worse. The reason is that with fewer parameters, the best distribution in the 2 parameter family can be achieved faster. However, in the end, the best distribution for full sampling would have advantageous. Actually, if the data have some special structure, like the classification data, we can expect that the best distribution would be in the simple family and the simple sampling may improve the performance. For the rest of the part, we will still use full sampling.

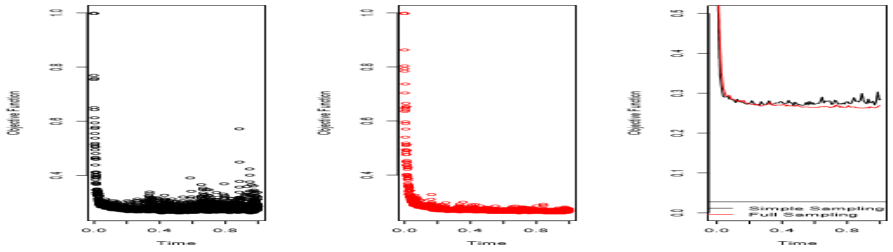

FIGURE 5. The performance for AW-SGD under different sampling

3.3. **Different Data Size for Mini-Batch and AW-SGD.** Since we have known the best choice of the tuning parameter for Mini-Batch and AW-SGD. Now, with these choices, we would like to compare their performance under different size of the data. We consider two size here. One is the original data, the large data size, and the other is the smaller data set. We generate the smaller data set by randomly pick 100 points from the original data. Knowing the best ratio is about 2 percent, we will use batch size 10 for large data set while 3 for the smaller and full sampling for AW-SGD. The result is showed in Figure 6 and 7.

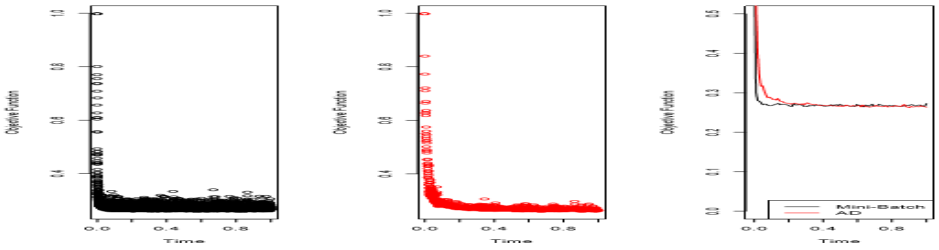

FIGURE 6. The performance for AW-SGD and Mini-Batch for 506 points

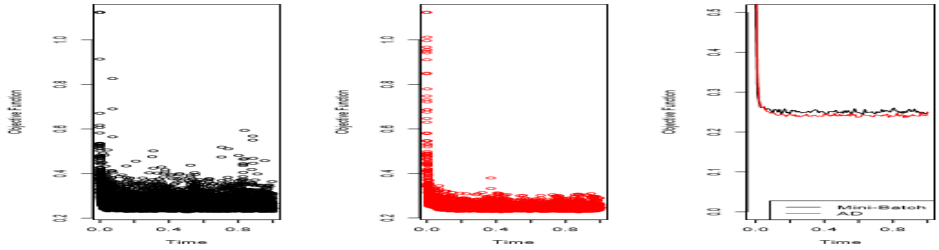

FIGURE 7. The performance for AW-SGD and Mini-Batch for 100 points

From Figure 6 and 7, we can notice that AW-SGD have some advantages in both case, but improve more in smaller data set. One of the possible reasons is that the best sampling distribution can be achieved faster for smaller data set due to less parameters.

3.4. **Prediction.** One of the statistical goals that we solve the optimization problem of least square is to predict. Thus, we do some prediction by cross-validation. With the best choice of the tuning parameter, We randomly draw 400 points for training and the other points for testing. We repeat it for 5 times and get Figure 8. The performance are similar, but, again, AW-SGD has some advantageous

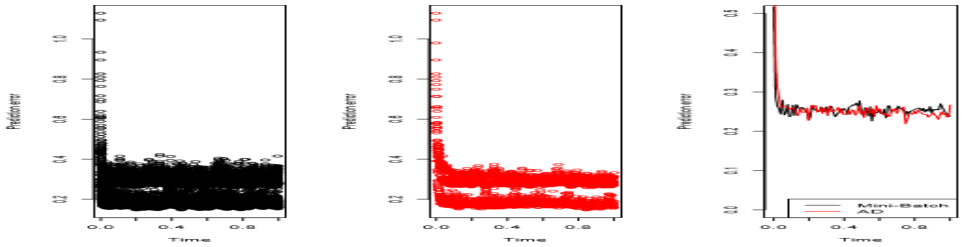

FIGURE 8. The prediction error for AW-SGD and Mini-Batch

4. SUMMARY

From the real data analysis above, we can draw the conclusions that under similar data set and for least square problem: (1)The best choice of batch size for Mini-Batch and AW-SGD are both 10 (2 percent). Either too large or too small batch size will cause problems. The choice is relative robust for Mini-Batch, while it is not for AW-SGD; (2) For any reasonable choice of batch size, both Mini-Batch and AW-SGD will improve SGD significantly. AW-SGD can improve Mini-Batch under the same batch size by reducing more noise and will improve more for smaller batch size. (3) The simple sampling will perform better at the beginning, but worse than full sampling at the end due to different best sampling distribution and the corresponding convergent rates for different sampling family. (4)With the best choice of the tuning parameters, AW-SGD will be better than Mini-Batch for both large data set and smaller data set, but more significant for smaller data set.

Doing this project, we also notice some possible problems for future work: (1) Doing real data analysis, we notice that the algorithm is very sensitive to the choice of the step size for updating the sampling distribution, though there are some suggestions have been made. If we can adaptively learn the step size, it would be very helpful. (2) In our view, the benefit of AW-SGD is not only the overall convergent trend which we primarily focus on in this report. We believe the reduced variance of the point estimate would be directly useful in setting the stop rule. (3) As we have mentioned in sampling assumption analysis part, it may be interesting to see when simple family would have benefits and decide some specific simple family for some specific type of data. (4) The data set we focus on is still too small. Actually, for this problem, we can use common gradient decent or closed form solution to solve the problems quicker. It would be meaningful to study their performance on extremely big data set or even streaming data setting.

Reference

[1]Bouchard, Guillaume, et al. "Accelerating stochastic gradient descent via online learning to sample." arXiv preprint arXiv:1506.09016 (2015).

[2]Lange, Kenneth. Optimization. 2nd ed. 2013. New York, NY: Springer New York, 2013.