ECE 543: Noise Reduction and Acceleration of Stochastic Gradient Descent for Least Square Regression Problem

Shu Chen*[1]

[1]Physics Department, UIUC

Abstract

In this project, the behavior of stochastic gradient descent (SGD) method for least square regression problem, including the noise reduction and acceleration, is studied and reviewed. To apply two theorems of the behavior of SGD with fixed stepsize and diminishing stepsize [1, 4], I first validate two assumptions for this least square regression problem, such as Lipschitz-continuous gradient assumption and upper bound assumption of stochastic gradient. And a numerical experiment is conducted to verify the conclusion of the theorem with fixed stepsize. Three main approaches to reduce noise in SGD are reviewed, and a following numerical result shows the effectiveness of iterate averaging method. Then the acceleration of SGD based on a recent paper [3] is studied. The concepts of condition number and statistical condition number are introduced, and their values in two examples are illustrated, such as a discrete distribution and a Gaussian distribution. Then the acceleration performance is numerically studied for a noise-free case and a noisy case with a preset high condition number.

1 Problem Setup and Introduction to SGD

Following reference [2, 3], least squares regression problem is considered:

$$\arg\min_{\mathbf{w}\in\mathbb{R}^d} F(\mathbf{w}),\ \ F(\mathbf{w}) \coloneqq \frac{1}{2}\mathbb{E}_P\left[(y-\langle \mathbf{w},\mathbf{x}\rangle)^2\right], \tag{1}$$

where each sample data $(\mathbf{x}, y)$ is drawn from the distribution P over $\mathbb{R}^d \times \mathbb{R}$, and d is the dimension of the problem, or the dimension of the feature space. From the statement of the problem, we can see that the prediction function is assumed to be linear functions with parameter as a real vector $\mathbf{w} \in \mathbb{R}^d$, and the loss function is quadratic. More specifically, prediction function class $\mathcal{H} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ and loss function $l : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is defined as:

$$\mathcal{H} \coloneqq \{h(\cdot;\mathbf{w}) = \langle \cdot, \mathbf{w}\rangle : \mathbf{w} \in \mathbb{R}^d\}, \quad l(h(\mathbf{x};\mathbf{w}), y) = \frac{1}{2}(y-\langle \mathbf{w},\mathbf{x}\rangle)^2, \tag{2}$$

where $\langle \mathbf{w}, \mathbf{x}\rangle$ denotes to the inner product between vectors $\mathbf{w}$ and $\mathbf{x}$. Noted that I adopt the notation from [4].

The optimal $\mathbf{w}^*$ can be approximated by the empirical risk minimizer (ERM). Given n i.i.d.samples $\{\mathbf{x}_i, y_i\}_{i=1}^n$ drawn from distribution P, ERM algorithm is defined as

$$\widehat{w}_n \coloneqq \arg\min_{\mathbf{w}\in\mathbb{R}^d} F_n(\mathbf{w}),\ \ F_n(\mathbf{w}) \coloneqq \frac{1}{n}\sum_{i=1}^n l(h(\mathbf{x}_i;\mathbf{w}), y_i) = \frac{1}{n}\sum_{i=1}^n \frac{1}{2}\left[(y_i-\langle \mathbf{w},\mathbf{x}_i\rangle)^2\right]. \tag{3}$$

*shuchen5@illinois.edu

Then the next task is to develop optimization algorithms for this minimization problem. Quoted from [4], there are two broad categories of optimization algorithm for this task: *stochastic* and *batch*. The batch gradient descent method is the traditional and well known optimization approach, which is defined by iteration:

$$\mathbf{w}_{k+1} \leftarrow \mathbf{w}_k - \alpha_k \nabla F_n(\mathbf{w}_k) = \mathbf{w}_k - \frac{\alpha_k}{n} \sum_{i=1}^{n} \nabla l_i(\mathbf{w}_k). \tag{4}$$

Here, $l_i(\mathbf{w}_k) = l(h(\mathbf{x}_i; \mathbf{w}_k), y_i)$ is used for simplicity, and its gradient is

$$\nabla l_i(\mathbf{w}_k) = -(y_i - \langle \mathbf{w}_k, \mathbf{x}_i \rangle)\ \mathbf{x}_i = (\langle \mathbf{w}_k, \mathbf{x}_i \rangle - y_i)\ \mathbf{x}_i. \tag{5}$$

On the other hand, the stochastic gradient descent (SGD) method is to pick i_k randomly from $\{1, 2, \dots\}$ at time step k, and only compute the gradient of one loss function $l_{i_k}(\mathbf{w}_k)$. The iteration sequence is then defined as:

$$\mathbf{w}_{k+1} \leftarrow \mathbf{w}_k - \alpha_k \nabla l_{i_k}(\mathbf{w}_k). \tag{6}$$

One intuitive motivation for SGD [4] is that at each time step, we only need to compute the gradient for one sample, which is $1/n$ times cheaper than the batch method. Another reason is that with a large amount of training data, normally there is certain level of redundancy involved. Using all of the sample data to compute iteration steps could be inefficient. Another practical reason is that in some online or streaming applications, SGD can be performed one sample by one sample, while batch method (total batch) is not able to achieve this. However, it should be noted that a mini-batch method is a compromise of these two algorithms.

2 Behaviors of SGD in Least Square Regression

In our course and [4], the behavior of SGD for a strongly convex objective with both fixed and diminishing stepsize are studied and summarized as two theorems. I copy them here for convenience.

Theorem 1. *With $F_{inf} = F^*$, if the SGD method is run with a fixed stepsize, $\alpha_k = \bar{\alpha}$ for all $k \in \mathbb{N}$, satisfying*

$$0 < \bar{\alpha} \leq \frac{\mu}{LM_G}. \tag{7}$$

Then, the expected optimality gap satisfies the following inequality for all $k \in \mathbb{N}$:

$$\begin{aligned} \mathbb{E}\left[F(\mathbf{w}_k) - F^*\right] \leq & \frac{\bar{\alpha} LM}{2c\mu} + (1 - \bar{\alpha} c\mu)^{k-1} \left(F(\mathbf{w}_1) - F^* - \frac{\bar{\alpha} LM}{2c\mu} \right) \\ \xrightarrow{k \to \infty} & \frac{\bar{\alpha} LM}{2c\mu}. \end{aligned} \tag{8}$$

Theorem 2. *With $F_{inf} = F^*$, if the SGD method is run with a diminishing stepsize for $k \in \mathbb{N}$, satisfying*

$$\alpha_k = \frac{\beta}{\gamma + k} \quad \text{for some} \quad \beta > \frac{1}{c\mu} \quad \text{and} \quad \eta > 0 \quad \text{such that} \quad \alpha_1 \leq \frac{\mu}{LM_G} \tag{9}$$

Then, the expected optimality gap satisfies the following inequality for all $k \in \mathbb{N}$:

$$\begin{aligned} \mathbb{E}\left[F(\mathbf{w}_k) - F^*\right] \leq & \frac{\nu}{\gamma + k}, \\ \xrightarrow{k \to \infty} & 0, \end{aligned} \tag{10}$$

where

$$\nu \coloneqq \max\left\{\frac{\beta^2}{2(\beta c\mu - 1)}, (\gamma+1)(F(\mathbf{w}_1) - F^*)\right\} \tag{11}$$

In order to apply these two theorems, two assumptions needs to be satisfied. We examine them here for this least square regression problem stated in Section 1, and obtain parameters needed for these two theorems. First assumption is the Lipschitz-continuous objective gradients, namely the gradient of objective function $\nabla F(\mathbf{w})$ is Lipschitz-continuous with Lipschitz constant $L > 0$, i.e.,

$$\|\nabla F(\mathbf{w}_1) - \nabla F(\mathbf{w}_2)\| \leq L\|\mathbf{w}_1 - \mathbf{w}_2\| \quad \text{for all } \mathbf{w}_1, \mathbf{w}_2 \in \mathbb{R}^d. \tag{12}$$

In the least squares regression problem, we consider the expected loss function for samples and using Equation (5)

$$\text{LHS} = \|\mathbb{E}\left[\langle \mathbf{w}_1 - \mathbf{w}_2, \mathbf{x}\rangle \mathbf{x}\right]\| = \left\|\mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right] \cdot (\mathbf{w}_1 - \mathbf{w}_2)\right\| = \|\text{Cov}(\mathbf{x}) \cdot (\mathbf{w}_1 - \mathbf{w}_2)\| \tag{13}$$

where $\text{Cov}(\mathbf{x})$ is the covariance matrix of $\mathbf{x}$. Therefore, Lipschitz constant L can be chosen as the largest eigenvalue of $\mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right]$ or $\text{Cov}(\mathbf{x})$, and this assumption will be satisfied.

The second assumption is about the variance of stochastic gradient. More specifically, the first component require objective function $F(\mathbf{w})$ is bounded below by F_{inf}, which is obviously for this least square regression problem, and $F_{\text{inf}} = 0$. The second and third components of this assumption need some assumptions for the sample data. Followed from [3], we assume sample data are generated by

$$y = \langle \mathbf{w}^*, \mathbf{x}\rangle + \epsilon \tag{14}$$

where ϵ is an independent random variable with respect to $\mathbf{x}$. Therefore, for objective function $F(\mathbf{w})$ and its gradient defined in (1) can be written as

$$F(\mathbf{w}) = \frac{1}{2}\mathbb{E}_P\left[(\langle \mathbf{w} - \mathbf{w}^*, \mathbf{x}\rangle)^2\right] + \frac{1}{2}\mathbb{E}\left[\epsilon^2\right], \tag{15}$$

$$\nabla F(\mathbf{w}) = \mathbb{E}_P\left[\langle \mathbf{w} - \mathbf{w}^*, \mathbf{x}\rangle \mathbf{x}\right] = \text{Cov}(\mathbf{x}) \cdot (\mathbf{w} - \mathbf{w}^*). \tag{16}$$

Therefore, $F_{\text{inf}} = F^* = \frac{1}{2}\mathbb{E}\left[\epsilon^2\right]$. Then, for the below relations to be satisfied

$$\nabla F(\mathbf{w}_k)^T \mathbb{E}_{\xi_k}\left[\nabla l_{i_k}(\mathbf{w}_k)\right] \geq \mu\|\nabla F(\mathbf{w}_k)\|^2 \quad \text{and} \tag{17}$$

$$\|\mathbb{E}_{\xi_k}\left[\nabla l_{i_k}(\mathbf{w}_k)\right]\| \leq \mu_G\|\nabla F(\mathbf{w}_k)\|. \tag{18}$$

where $\xi_k = (\mathbf{x}_{i_k}, y_{i_k})$ is the random training sample. For the streaming problem, relation $\mathbb{E}_{\xi_k}\left[\nabla l_{i_k}(\mathbf{w}_k)\right] = \nabla F(\mathbf{w}_k)$ holds. So μ, μ_G can both be chosen as 1.

The third component of the second assumption reads as, there exist $M \geq 0$ and $M_V \geq 0$ such that, for all $k \in \mathbb{N}$,

$$\mathbb{V}_{\xi_k}\left[\nabla l_{i_k}(\mathbf{w}_k)\right] \leq M + M_V\|\nabla F(\mathbf{w}_k)\|^2. \tag{19}$$

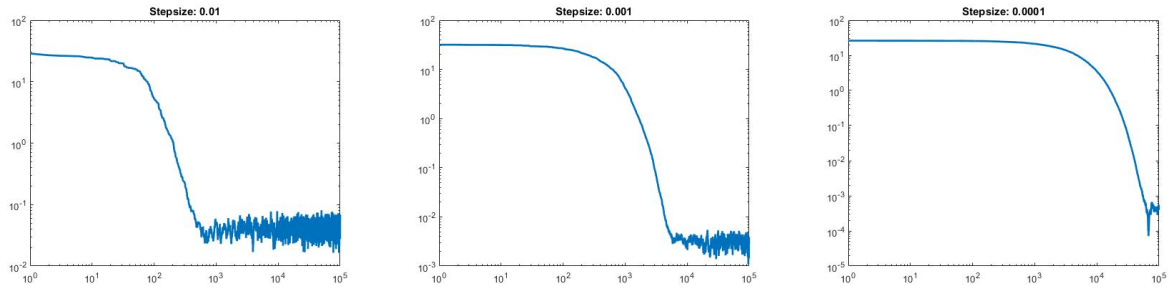
In fact, the left hand side of this relation can be written as

$$\begin{aligned}
\text{LHS} &= \mathbb{V}_{\xi_k}\left[\left(\langle \mathbf{w}_k - \mathbf{w}^*, \mathbf{x}_k\rangle - \epsilon\right)\mathbf{x}_k\right] \\
&= \mathbb{E}_{\xi_k}\left[\left(\langle \mathbf{w}_k - \mathbf{w}^*, \mathbf{x}_k\rangle^2 + \epsilon^2\right)\|\mathbf{x}_k\|^2\right] - \|\text{Cov}(\mathbf{x}) \cdot (\mathbf{w}_k - \mathbf{w}^*)\|^2 \\
&\leq \mathbb{E}[\epsilon^2]\sum_{i=1}^{d}\sigma_i^2 + M_V\|\nabla F(\mathbf{w}_k)\|^2
\end{aligned} \tag{20}$$

Here, M_V is related to the fourth moment of $\mathbf{x}_k$. Since $\|F(\mathbf{w}_k)\| \geq \lambda_{\min} \|\mathbf{w}_k - \mathbf{w}^*\|$ with $\lambda_{\min}$ being the smallest eigenvalue of the covariance matrix, a finite, positive M_V can always be obtained.

Therefore, the least square regression problem stated satisfies all two assumptions that theorem 1 and 2 are based on. Then, we can perform several numerical experiments with SGD to verify these two theorems.

I considered a sample space with dimension $d = 50$, and elements of vector $\mathbf{x}$ are all independent to each other. The value of y is generated by a normal random vector $\mathbf{w}^*$ and a Gaussian random noise ϵ as in Equation (14). Therefore, the covariance matrix is diagonal. The covariance along each axis is randomly chosen to be in the range $[0.1, 10]$. Different fixed stepsize $\bar{\alpha} = 0.01, 0.001, 0.0001$ are chosen for 10^5 time steps, the results are shown below.



(a) (b) (c)

Figure 1: The excess error with respect to time steps with different stepsize.

The numerical result in 1 coincides with theorem 1 excellently. First, the remaining error is proportional to the stepsize as shown in the first term of Equation (8), as the stepsize is decreased, the remaining error also decreases. Second, with a smaller stepsize, the decaying rate is slower. In logarithm manner, when the second term dominates, we have

$$\log \mathbb{E}\left[F(\mathbf{w}_k) - F^*\right] \leq (k-1) \log\left(1 - \bar{\alpha} c \mu\right) + \log\left(F(\mathbf{w}_1) - F^* - \frac{\bar{\alpha} L M}{2c\mu}\right) \tag{21}$$

When two terms in (8) become comparable, the error starts to drop dramatically.

3 Noise Reduction Method

From theorem 1 and 2, we can see that with a fixed stepsize, the error converges much faster than the diminishing stepsize algorithm. However, the remaining error for the fixed stepsize case could be ignorable. There are three main approaches for noise reduction of the SGD [4], such as *dynamic sampling, gradient aggregation,* and *iterate averaging.* They can all be understood within the framework of two theorems mentioned.

For the dynamic sampling method, the iteration is modified as

$$\mathbf{w}_{k+1} \leftarrow \mathbf{w}_k - \bar{\alpha} g_k(\mathbf{w}_k) = \mathbf{w}_k - \frac{\alpha_k}{n_k} \sum_{i \in \mathcal{S}_k} \nabla l_i(\mathbf{w}_k), \quad \text{with} \quad n_k := |\mathcal{S}_k| = \lceil \tau^{k-1} \rceil. \tag{22}$$

Therefore, obviously, the M in (8) is controlled to decrease geometrically, and

$$\mathbb{V}\left[g_k(\mathbf{w}_k)\right] \leq \frac{\mathbb{V}\left[\nabla l_i(\mathbf{w}_k)\right]}{n_k} \leq \frac{M}{n_k}. \tag{23}$$

Therefore, both two terms in (8) are decreasing geometrically. Therefore, with some constant $\omega, \rho > 0$,

$$\mathbb{E}\left[F(\mathbf{w}_k) - F^*\right] \leq \omega \rho^{k-1} \tag{24}$$

Different from dynamic sampling, gradient aggregation reuses previously computed information to reduce noise. For example, the *stochastic variance reduced gradient* (SVRG) method has each iteration as

$$\widetilde{\mathbf{w}}_{j+1} \leftarrow \widetilde{\mathbf{w}}_j - \alpha g_j(\widetilde{\mathbf{w}}_j, \mathbf{w}_k) = \mathbf{w}_k - \alpha \left\{\nabla l_j(\widetilde{\mathbf{w}}_j) - [\nabla l_j(\mathbf{w}_k) - \nabla R_n(\mathbf{w}_k)]\right\}, \tag{25}$$

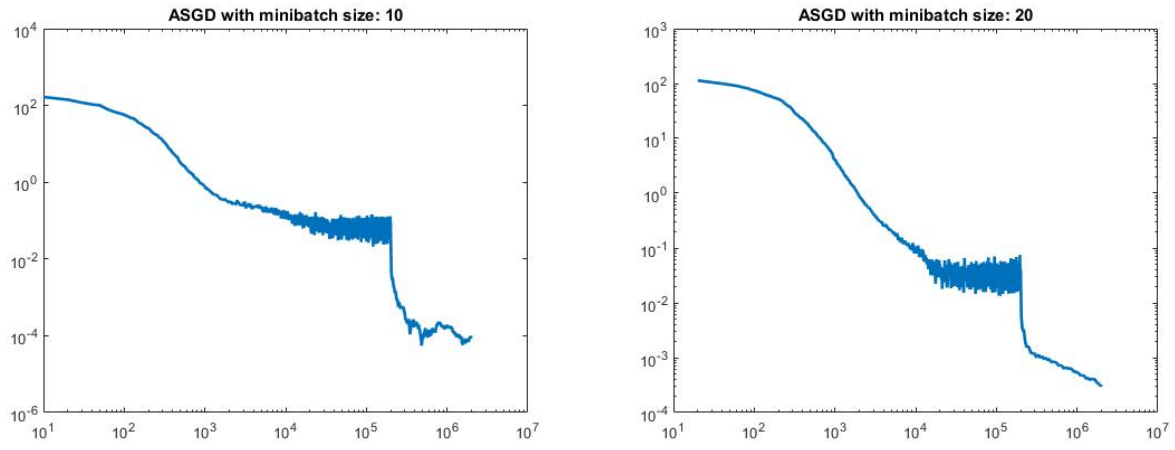
where ∇R_n is a batch gradient computed with $\mathbf{w}_k$. Therefore, the spirit of this method is reuse the stored batch gradient, and apply the difference of ∇l_j and this batch gradient as the noise generated by random samples.

The last noise reduction method we will examine here is much simpler than previous two, and its iteration reads as

$$\mathbf{w}_{k+1} \leftarrow \mathbf{w}_k - \alpha_k \nabla l_{i_k}(\mathbf{w}_k), \quad \text{and} \quad \widetilde{\mathbf{w}}_{k+1} \leftarrow \frac{1}{k+1} \sum_{j=1}^{k+1} \mathbf{w}_j \tag{26}$$

Noted that although the averaged sequence $\{\widetilde{\mathbf{w}}_k\}$ is the optimal parameter vector we are seeking, but it does not affect the normal SGD iteration sequence. This method is based on the argument that with a fixed stepsize, the remaining error is nonzero is because the parameter vector oscillates around the minimum randomly due to the noisy gradient. And taking an average of these oscillating parameter vectors can cancel the noise. Since the initial *bias* error introduced by $\mathbf{w}_1$ could be large, we may prefer taking average after certain number of steps. Trail averaging method can be developed based on this thought [2]. And the iteration in (26) is modified to be

$$\begin{aligned} &\mathbf{w}_{k+1} \leftarrow \mathbf{w}_k - \alpha_k \nabla l_{i_k}(\mathbf{w}_k), \\ \text{and} \quad &\widetilde{\mathbf{w}}_{k+1} \leftarrow \frac{1}{k-s+1} \sum_{j=s+1}^{k+1} \mathbf{w}_j = \frac{1}{k-s+1} \mathbf{w}_{k+1} + \frac{k-s}{k-s+1} \widetilde{\mathbf{w}}_k. \end{aligned} \tag{27}$$



(a) (b)

Figure 2: The excess error with respect to time steps with different mini-batch size. Total $n = 2\times10^6$ sample data are considered. Averaging starts from $s = 2 \times 10^5$.

A numerical experiment is performed using this trail averaging SGD, and mini-batch is also incorporated. The same training sample space considered is the same as in Section 2, and different mini-batch size is considered. From Figure 2, we can see that before averaging, the error behaves the same as normal SGD in Figure 1. And a larger batch leads to smaller variance of noise, or M value, which leads to a smaller remaining error. After averaging starting from $s = 2 \times 10^5$, a kink curve appears for both figures in Figure 2, and the excess error drops dramatically. It should be noted that the error will converge to $\sigma^2 d/n$ [2], where σ^2 denotes the noise level.

4 Acceleration of SGD

It was proposed in [3] that, a SGD algorithm could be accelerated within certain distribution of the training samples. It has been proved in [2] that the error of an averaged SGD method is converging with rate

$$\mathcal{O}\left(\exp\left(\frac{-n}{\kappa}\cdot\left(F(\mathbf{w}_1)-F(\mathbf{w}^*)\right)\right)+\frac{\sigma^2 d}{n}\right) \tag{28}$$

And the proposed accelerated SGD (ASGD) could reach a rate as

$$\mathcal{O}^*\left(\exp\left(\frac{-n}{\sqrt{\kappa\widetilde{\kappa}}}\cdot\left(F(\mathbf{w}_1)-F(\mathbf{w}^*)\right)\right)+\frac{\sigma^2 d}{n}\right) \tag{29}$$

Here, $\kappa, \widetilde{\kappa}$ are condition number and statistical condition number, which will be introduced below. Noted that this acceleration method is effective only if $\widetilde{\kappa}$ is much smaller than κ, which is correct for certain distributions.

We first introduce the second moment matrix $\mathbf{H}$, and it is the same with the Hessian $\nabla^2 F(\mathbf{w})$ in this least square regression problem.

$$\mathbf{H} \coloneqq \mathbb{E}_P\left[\mathbf{x}\mathbf{x}^T\right] = \mathrm{Cov}(\mathbf{x}) = \nabla^2 F(\mathbf{w}) \tag{30}$$

The noise level is defined as:

$$\sigma^2 = \mathbb{E}\left[\epsilon^2\right] \tag{31}$$

where ϵ is defined in (14). Define μ as the smallest eigenvalue of $\mathbf{H}$:

$$\mu \coloneqq \lambda_{\min}(\mathbf{H}) \tag{32}$$

And R^2 is the smallest positive number satisfying:

$$\mathbb{E}\left[\|\mathbf{x}\|^2\,\mathbf{x}\mathbf{x}^T\right] \preceq R^2\mathbf{H} \tag{33}$$

Then condition number κ is deduced as:

$$\kappa \coloneqq \frac{R^2}{\mu} \tag{34}$$

And statistical condition number $\widetilde{\kappa}$ is defined as the smallest positive number s.t.

$$\mathbb{E}\left[\|\mathbf{x}\|^2_{\mathbf{H}^{-1}}\,\mathbf{x}\mathbf{x}^T\right] \preceq \widetilde{\kappa}\mathbf{H} \tag{35}$$

where $\|x\|^2_{\mathbf{S}} = \mathbf{x}^T\mathbf{S}\mathbf{x}$.

Relation $\widetilde{\kappa} \preceq \kappa$ can be deduced since $\mathbb{E}\left[\|\mathbf{x}\|^2_{\mathbf{H}^{-1}}\,\mathbf{x}\mathbf{x}^T\right] \preceq \frac{1}{\mu}\mathbb{E}\left[\|x\|^2\,\mathbf{x}\mathbf{x}^T\right] \preceq \kappa\mathbf{H}$.

Next, I will examine the value of these two condition numbers for two different distributions. First is a discrete distribution described in [3], the sample vector $\mathbf{x}$ can only be unit basis vectors along each axis with probability p_i. Then matrix $\mathbf{H}$ can be written as:

$$\mathbf{H} = \begin{bmatrix} p_1 & & \\ & \ddots & \\ & & p_d \end{bmatrix}. \tag{36}$$

Therefore, $\mu = \min_i p_i$, and since $\|\mathbf{x}\|^2 = 1$, $R = 1$ is easily deduced. Then,

$$\kappa = \frac{1}{\min_i p_i} \tag{37}$$

For the statistical condition number,

$$\mathbb{E}\left[\|\mathbf{x}\|^2_{\mathbf{H}^{-1}}\,\mathbf{x}\mathbf{x}^T\right] = \sum_i^d p_i \frac{1}{p_i}\mathbf{e}_i\mathbf{e}_i^T = \mathbf{I} \tag{38}$$

where $\mathbf{e}_i$ is the unit basis vector along i-th axis, and $\mathbf{I}$ is the $d \times d$ identity matrix. Therefore, $\widetilde{\kappa}$ also equal to $1/\min_i pi$.

So according to (28) and (29), this acceleration method is not applicable to this discrete distribution. Next we consider the Gaussian sample data with diagonal covariance matrix, which is just the second moment matrix $\mathbf{H}$:

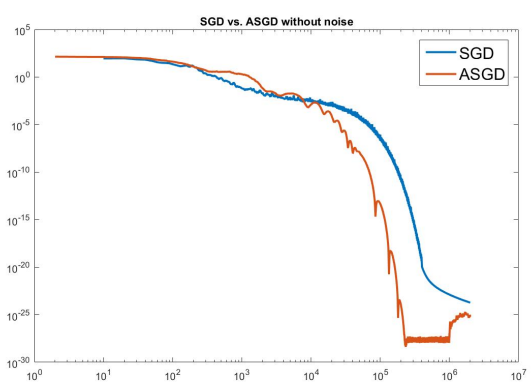
$$\mathrm{Cov}(\mathbf{x}) = \begin{bmatrix} \sigma_1^2 & & \\ & \ddots & \\ & & \sigma_d^2 \end{bmatrix} = \mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right] = \mathbf{H} \tag{39}$$

Then, $\mu = \min_i \sigma_i^2$, and from (33) $R^2 \approx \sum_i^d \sigma_i^2$. Therefore, κ is $\mathcal{O}(\sum_i^d \sigma_i^2/\mu)$, and it could be very large. The statistical condition number $\widetilde{\kappa}$ with this distribution can be considered as:

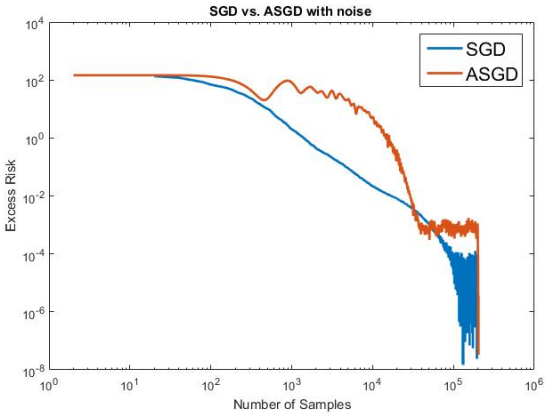
$$\mathbb{E}\left[\|\mathbf{x}\|^2_{\mathbf{H}^{-1}}\,\mathbf{x}\mathbf{x}^T\right] = \mathbb{E}\left[\left(\sum_i^d \frac{x_i^2}{\sigma_i^2}\right)\mathbf{x}\mathbf{x}^T\right] \approx d\mathbf{H} \tag{40}$$

Therefore, $\widetilde{\kappa}$ is only comparable to dimension d. So with a considerable large κ, according to (28) and (29), the SGD can be greatly accelerated by the algorithm proposed in [3].

I also performed numerical study of this accelerated algorithm. I generated the data the same way as before. But it should be noted that to guarantee a large κ to compare the acceleration performance, I forced the $\min_i \sigma_i^2 = 0.01$, and other variance are randomly chosen from $[0.01, 10]$. Therefore the κ could be as high as 2×10^4 according to what I mentioned earlier. Meanwhile, $\widetilde{\kappa}$ is only comparable to $d = 50$. Based on (28) and (29), this algorithm could be 20 times faster than the normal averaged SGD. We performed the calculation for both with noise and without noise cases. The numerical results are shown below: A numerical experiment is performed using this trail averaging SGD, and mini-batch is also incorporated. The same training sample space considered is the same as in Section 2, and different mini-batch size is considered.



(a)



(b)

Figure 3: The excess error with respect to time steps with noise and without noise. Total $n = 2 \times 10^6$ sample data are considered.

Shown in Figure 3, we can see the acceleration did perform much better than the normal averaged SGD without noise. However, in the noisy case, it is not as expected and some part of the noisy case is lost due to the noise level is possibly too high. One thing is that the acceleration may not be as appealing as 20 times as predicted, because the superscript in (29) means that the outside constant parameter also includes κ, which is different from (28). Another key point is that the κ has to be set high enough to see the difference. However, another issue is that with certain level of noise, the element of $\mathbf{x}$ with very small variance may not be even seen, which will bring in a considerable error.

References

[1] B. Hajek, Leture Notes of ECE 543, 2017.

[2] P. Jain, S. M. Kakade, R. Kidambi, P. Netrapalli, and A. Sidford, "Parallelizing stochastic approximation through mini-batching and tail-averaging," *arXiv preprint arXiv:1610.03774*, 2016.

[3] P. Jain, S. M. Kakade, R. Kidambi, P. Netrapalli, and A. Sidford, "Accelerating stochastic gradient descent," *arXiv preprint arXiv:1704.08227*, 2017.

[4] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *arXiv preprint arXiv:1606.04838*, 2016.