Distributed and Scalable Variance-reduced Stochastic Gradient Descent

Kelvin Kai Wing Ng

I. Introduction

This study employs the edge-cutting variance-reduced stochastic gradient descent method (VR method)[1] which provides linear convergence rate, together with modified mini-batch approach so as to improve parallelism and scalability of the variance-reduced gradient descent method.

II. Objective

This study aims at:

1) Improve the scalability of existing mini-batch VR method[2]
2) Reduce synchronization so as to improve the performance in distributed settings

III. Contribution

1) There exists a study on employing mini-batch approach on SVRG, one of the VR methods. It shows that the approach cannot scale well that there is no significant difference between using 16 threads and more[2]. This study observes the cause of the poor scalability of this existing mini-batch approach on VR method.
2) The performance of mini-batch approach on distributed setting is improved by reducing the frequency of synchronization without significantly affecting the result.

IV. Algorithm

Variance-reduced methods give better result over full gradient descent (FGD) and stochastic gradient descent (SGD) in general, but it is not easy to implement efficiently in distributed setting.

The variance between updates will asymptotically go to zero[1]. It means that at the beginning, the variance can be as high as SGD, but it is guaranteed that the variance will be reduced gradually, unlike SGD which has non-vanishing variance. Since mini-batch approach improves the result by reducing variance, the small variance of VR method makes it difficult to be scaled. In particular, we make the following propositions:

1) Mini-batch approach is useful at the beginning of VR method (when variance is still large), but not useful after many iterations (because the variance goes to zero)
2) The effect of mini-batch approach increases when frequency of synchronization decreases because of the increased variance
3) Reducing frequency of synchronization does not affect the result seriously as the variance is small (especially after many iterations)

The study aims at verifying the above propositions to give useful improvements on the existing method.

V. Results

We use SVRG, one of the VR methods, for experiment. All VR methods have the same asymptotic behaviour.

A. First Proposition: effect of mini-batch is large only when variance is large

Figure 1 shows the relationship between the effect mini-batch with variance and number of step. The effect of mini-batch is determined by the percentage difference of accuracy between having more threads and fewer threads. As expected, variance reduces with number of step. Also, the effect of mini-batch reduces with variance as number of step increases as expected.

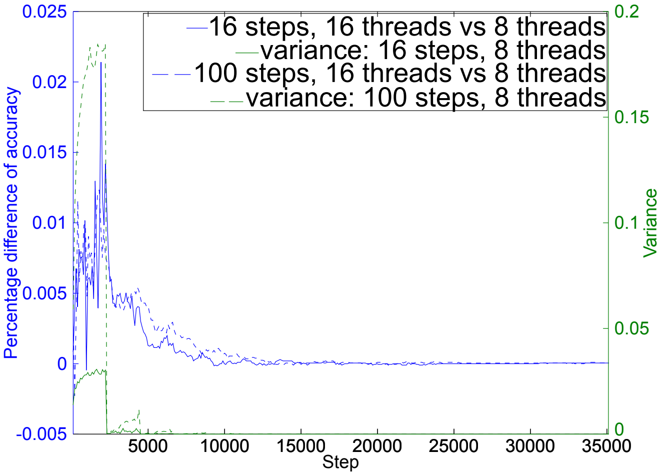

Fig. 1: The percentage difference of accuracy changes with variance and number of step

Figure 2 shows the same result in another perspective. It shows that when the accuracy is low and the variance is large, number of step needed to achieve a certain accuracy is fewer with more threads. A surprising result is that the algorithm scales well at the beginning that with a double of threads, the algorithm is accelerated by 8 times.

The above observations verify the first proposition.

B. Second proposition: effect of mini-batch is larger with smaller synchronization frequency

Figure 1 shows that mini-batch is more effective when the synchronization frequency is lower: the variance is higher and the percentage difference of accuracy is larger with smaller synchronization frequency. It verifies the second proposition.

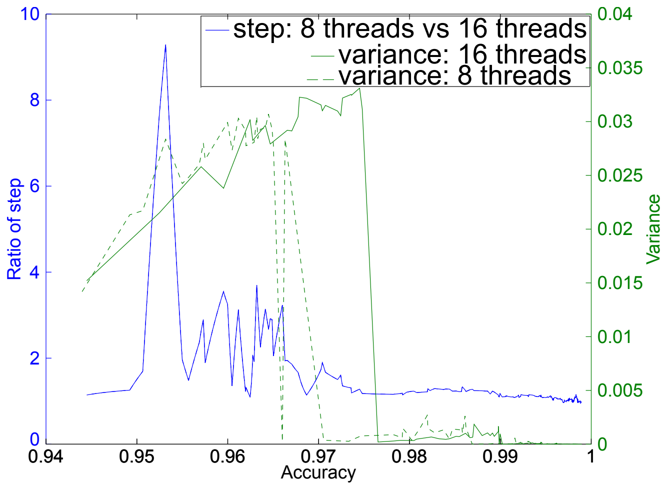

Fig. 2: Ratio of step needed to achieve a certain accuracy between mini-batch with 8 threads and 16 threads

C. Third proportion: reducing synchronization frequency does not affect the result

Figure 3 shows that the accuracy is not obviously affected when the threads synchronize per 2, 4, 8, 16 steps instead of every step. It directly verifies the third proposition. The accuracy is slightly decreased at first when the threads synchronize per 100 steps, but there are no effects afterwards. It is caused by that the increased effect of mini-batch as discussed in section V-B cancels with the worse performance of each thread. The implication is that we can improve the speed of the algorithm in distributed setting by reducing synchronization as the synchronization time is the most time-consuming job in the distributed algorithm.

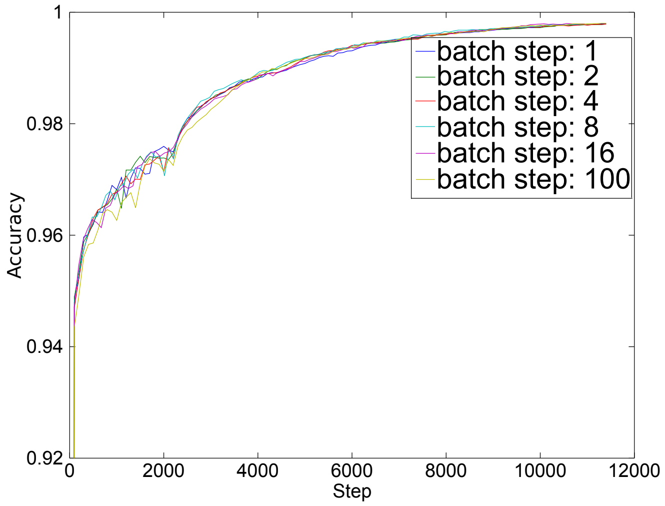

Fig. 3: Comparison of the change of accuracy among different synchronization frequency

VI. Conclusion

This study has verified the above propositions. The practical use of the result is that the mini-batch VR algorithm can be accelerated by reducing synchronization frequency without affecting the result. Unfortunately, the first objective is not accomplish. Instead, the cause of the problem is thoroughly analysed.

The future research direction is to combine the result with Butterfly mixing[3] to further improve Butterfly mixing by reducing synchronization frequency and hopefully does not significantly affect the result.

References

[1] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," in *NIPS*, pp. 315–323, 2013.

[2] J. Konecný, J. Liu, P. Richtárik, and M. Takác, "Mini-batch semi-stochastic gradient descent in the proximal setting," *JSTSP*, vol. 10, no. 2, pp. 242–255, 2016.

[3] J. Canny and H. Zhao, "Butterfly mixing: Accelerating incremental-update algorithms on clusters," in *SIAM*, pp. 785–793, 2013.