# Distributed and Scalable Variance-reduced Stochastic Gradient Descent

**Kelvin Kai Wing Ng**

The Chinese University of Hong Kong

## 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 stochastic gradient descent method.

## Objective

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. So, we have the following objectives:

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

## Contribution

1. Observations of the cause of the poor scalability of the existing mini-batch approach on VR method[2]
2. Improved performance of mini-batch approach on distributed setting by reducing the frequency of synchronization

## Methodology

The variance between updates will asymptotically go to zero[1]. Since mini-batch approach improves the result by reducing variance, VR method is not suitable for using mini-batch. To study the problem, the following propositions are made:

1. Mini-batch approach is useful at the first few iterations, but not useful afterwards
2. Frequency of synchronization $\downarrow \Rightarrow$ Variance $\uparrow \Rightarrow$ Effect of mini-batch $\uparrow$
3. Reducing frequency of synchronization does not affect the result seriously especially after many iterations

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

We used SVRG, one of the VR methods, for experiment. All VR methods have the same asymptotic behaviour. The dataset used is RCV1. The problem to be solved is logistic regression.

## Results

Figure 1 shows that:

1. Variance reduces with number of iteration passed
2. The effect of mini-batch reduces with variance as number of iteration passed increases

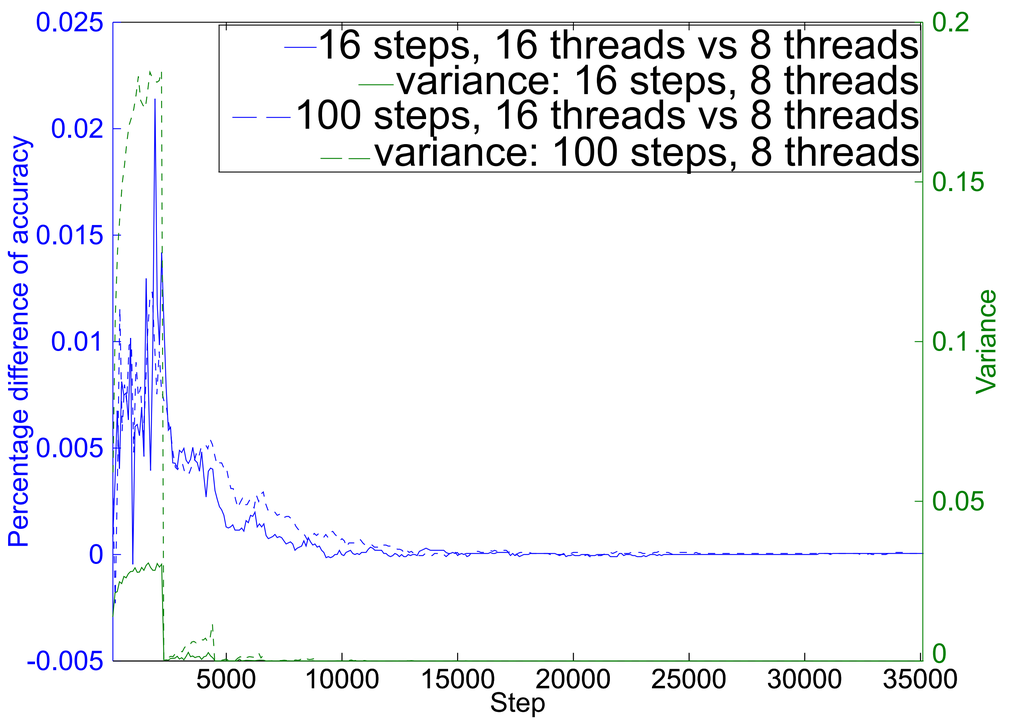


**Figure 1:** The percentage difference of accuracy changes with variance and number of step

Figure 2 shows the same result in another perspective.

1. Accuracy is low and variance is large => with more threads, number of iteration needed to achieve a certain accuracy is fewer
2. The algorithm scales well at the beginning: 2x of threads => 9x of speed

**The above observations verify the first proposition**.

Figure 1 also shows that mini-batch is more effective when the synchronization frequency is lower which **verifies the second proposition**.

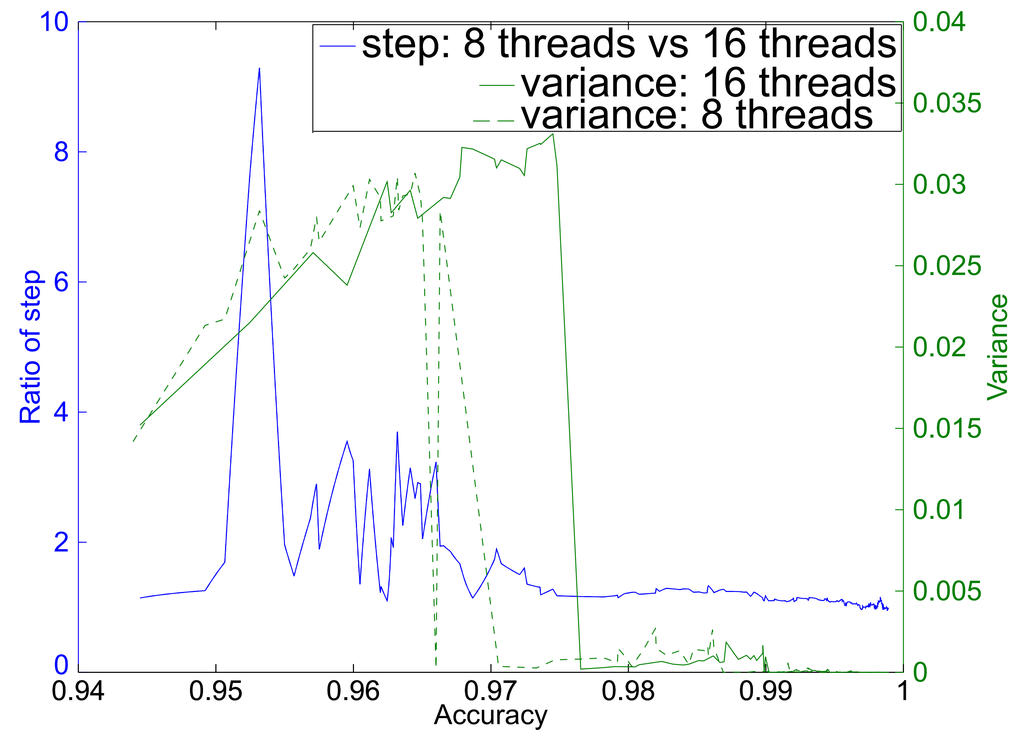


**Figure 2:** Ratio of step needed to achieve a certain accuracy between mini-batch with 8 threads and 16 threads

However, Figure 3 shows that the accuracy is not obviously affected when synchronization frequency is decreased. **Reducing synchronization frequency can significantly reduce running time: 8 steps: 5m48s $\rightarrow$ 16 steps: 4m14s, *27% decrease***.

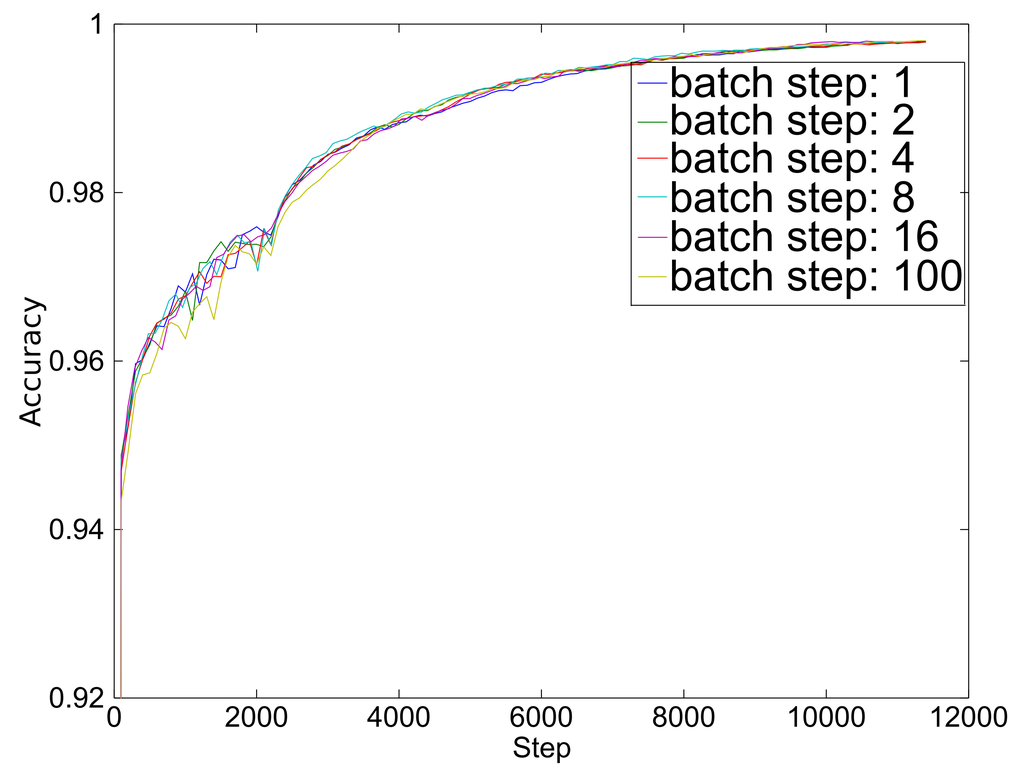


**Figure 3:** Comparison of the change of accuracy among different synchronization frequency

## Conclusion

1. The above propositions are verified
2. Practical use: reduce synchronization without affecting the result
3. Failed to accomplish the first objective
4. Future: Combine with Butterfly mixing[3]

## 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.