PERUI: A General Framework of Reduced Variance Stochastic Gradient Gradient and the Hybrid Implementation

Yawei Zhao
School of Computer
National University
of Defense Technology
Changsha, China, 410073

Yuewei Ming
School of Computer
National University
of Defense Technology
Changsha, China, 410073

Jianping Yin
School of Computer
National University
of Defense Technology
Changsha, China, 410073

Algorithm 1 The general framework of variance reduced SGD: PERUI

Require: $\omega^0 \in \mathbb{R}^d$. $\forall i \in [n]$, and $[n]$ represents $1, 2, ...n$.

Probability: $[i_t] \leftarrow \mathcal{P}([n])$ where $i_t \in 1, 2, ..., n$. t is a positive integer;
Epoch: the sequence of the epoch size $\{m^0, m^1, ..., m^S\} \leftarrow \mathcal{E}([i_t])$;
for $s = 0, 1, 2, ..., S$ **do**
  $\omega_0^s = \tilde{\omega}^s$;
  $g = \frac{1}{n}\sum\limits_{i=1}^{n} \nabla f_i(\tilde{\omega}^s)$;
  the size of the next epoch: m^s;
  for $t < m^s$ **do**
    Reduced variance: $v = \mathcal{R}(\nabla f_{i_t}(\omega_{i_t}^t) - \nabla f_{i_t}(\tilde{\omega}^s))$;
    $\gamma_t^s = v + g$;
    Update: $\omega_{t+1}^s = \mathcal{U}(\eta_t, \omega_t^s, \gamma_t^s)$;
  Identificaton: $\tilde{\omega}^{s+1} \leftarrow \mathcal{I}([w_j^s])$ with $j \in \{1, 2, ..., m^s\}$;
return $\tilde{\omega}^S$;

Introduction

Related work

The general hybrid framework of SGD

Conventionally, each $f_i(\omega)$ in the optimization problem **??** is a L-smooth function. That is, $\exists$ a non-negative L, and $\forall$ a and b, the following inequality holds.

$$f_i(a) \leq f_i(b) + \nabla f_i(b)^{\mathrm{T}}(a-b) + \frac{L}{2} \parallel a - b \parallel^2 \quad (1)$$

Besides, the loss function $F(\omega)$ in the optimization problem **??** is γ-strongly convex, which means that $\exists$ a non-negative γ, and $\forall$ a and b, the following inequality holds.

$$F(a) \geq F(b) + \nabla F(b)^{\mathrm{T}}(a-b) + \frac{\mu}{2} \parallel a - b \parallel^2 \quad (2)$$

Example: HybridSVRG

Convergence analysis

Optimization

Constant learning rate with an acceleration factor

Adaptive update sharing strategy

Discussion

Performance evaluation

In this section, we evaluate the performance of HybridSVRG by using a l_2-regularized logistic regression on four datasets, namely, dna[4], epsilon[5], SUSY[6], KDDCup2010[7].

$$\min \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + exp(-y_i\omega^{\mathrm{T}}x_i)\right) + \lambda \parallel \omega \parallel^2 \quad (3)$$

. Here, n is the size of the training data.

The following algorithms will be used for comparison.

- **DownpourSGD:** An asynchronous version of SGD which is used to train neural network (Dean et al. 2012).
- **PetuumSGD:** The distributed version of SGD is implemented by using the asynchronous communication protocol, i.e., SSP (Xing et al. 2015). The learning rate in PetuumSGD is decayed with a fixed factor 0.95 at the end of an epoch.
- **SSGD:** It is the state-of-the-art distributed version of SGD, which adopts the variance reduction technique (Zhang, 0004, and Kwok 2015). The update rule in the SSGD has a variable θ which is used to update the parameters asynchronously. The details of SSGD can be referred in (Zhang, 0004, and Kwok 2015). Here, we set $\theta = 0.5$.
- **HSAG:** A hybrid SGD which is proposed in (Reddi et al. 2015).
- **KroMagnon:** A lock-free version of SGD which adopts variance reduction technique, and is proposed in (Mania et al. 2015).

[4] ftp://largescale.ml.tu-berlin.de/largescale
[5] http://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/binary.html#epsilon
[6] http://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/binary.html#SUSY
[7] https://pslcdatashop.web.cmu.edu/KDDCup/downloads.jsp

- **HybridSVRG-lock:**
- **HybridSVRG:**

Convergence

Speedup

Wait time

Parallel threads

Conclusion

References

Dean, J.; Corrado, G.; Monga, R.; 0010, K. C.; Devin, M.; Le, Q. V.; Mao, M. Z.; Ranzato, M.; Senior, A. W.; Tucker, P. A.; Yang, K.; and Ng, A. Y. 2012. Large Scale Distributed Deep Networks. NIPS 1232–1240.

Mania, H.; Pan, X.; Papailiopoulos, D.; Recht, B.; Ramchandran, K.; and Jordan, M. I. 2015. Perturbed Iterate Analysis for Asynchronous Stochastic Optimization. CoRR abs/1511.08486 stat.ML.

Reddi, S. J.; Hefny, A.; Sra, S.; Póczos, B.; and Smola, A. 2015. On Variance Reduction in Stochastic Gradient Descent and its Asynchronous Variants. arXiv.

Xing, E. P.; Yu, Y.; Ho, Q.; Dai, W.; Kim, J. K.; Wei, J.; Lee, S.; Zheng, X.; Xie, P.; and Kumar, A. 2015. Petuum: A New Platform for Distributed Machine Learning on Big Data . In SIGKDD.

Zhang, R.; 0004, S. Z.; and Kwok, J. T. 2015. Asynchronous Distributed Semi-Stochastic Gradient Optimization. CoRR abs/1508.01633.

Table 1: Design details

Name	Strategy	Return	Algorithm								
			SVRG	S2GD	mS2GD	EMGD	SVR-GHT	Prox-SVRG	SVRG^{++}	Katyusha	synthetic
$\mathcal{P}$	uniformly	$i_t \sim \mathbb{U}$, namely, $P(i_t) = \frac{1}{n}$	✓	✓	✓	✓	✓	✓	✓	✓	✓
	non-uniformly[1]	$i_t \sim \mathbb{N}$						✓			
$\mathcal{E}$	random	m^s is picked from $\{1, 2, \ldots, C\}$ randomly			✓						
	constant	$m^s = C$	✓			✓	✓	✓		✓	✓
	ascent	$2^s C$							✓		
	descent	$P(m^s) = \frac{(1-\tilde{\mu}\eta)^{\mathrm{m^s}-t}}{\sum_{t=1}^{M}(1-\tilde{\mu}\eta)^{M-t}}$		✓							
$\mathcal{R}$	single	$\frac{1}{nP(i_t)}\left(\nabla f_{i_t}(\omega^s) - \nabla f_{i_t}(\omega^t_{i_t})\right)$	✓	✓		✓	✓	✓	✓	✓	✓
	mini-batch[2]	$\frac{1}{b}\sum_{j=1}^{b}\left(\nabla f_{i_t}(\omega^s) - \nabla f_{i_t}(\omega^t_{i_t})\right)$			✓						
$\mathcal{U}$	steepest descent	$\omega^s_t - \eta_t * \gamma^s_t$	✓	✓	✓			✓	✓	✓	✓
	steepest descent, shrinking domain	$\omega^s_t - \mathbb{B}_{\Delta_t}(\eta_s * \gamma^s_t)$ with $\Delta_s = \frac{C}{2^s}$ and $\parallel \omega^s_t - \omega^s_{t-1} \parallel \leq \Delta_s$				✓					
	steepest descent, sparse[3]	$\mathbb{O}_k(\omega^s_t - \eta_t * \gamma^s_t)$					✓				
$\mathcal{I}$	random	pick w^s_j from $\{1, 2, \ldots, m^s\}$ randomly	✓								
	the last one	$w^{m^s}_j$	✓	✓	✓	✓	✓				✓
	average	$\sum_{j=1}^{m^s} w^s_j P(j)$						✓	✓		
	negative momentum	$\begin{pmatrix} \tau_1 \\ \tau_2 \\ 1-\tau_1-\tau_2 \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} z_0 - \alpha \sum_{i=1}^{m^s} \tilde{\gamma}^s_t \\ w^{s-1} \\ x_0 - \frac{1}{3L}\sum_{i=1}^{m^s} \tilde{\gamma}^s_t \end{pmatrix}$								✓	