Balancing Rates and Variance via Adaptive Batch-sizes in First-order Stochastic Optimization

Z. Gao, A. Koppel and A. Ribeiro

Dept. of Electrical and Systems Engineering
University of Pennsylvania

May 2020

Outline

- Motivation
- Two scale adaptive algorithm
- Convergence and sample complexity reduction
- Numerical simulation
- Conclusions

Outline

- **Motivation**
- Two scale adaptive algorithm
- Convergence and sample complexity reduction
- Numerical simulation
- Conclusions

Motivation

Stochastic optimization problem

$$\min_{\mathrm{x}} F(\mathrm{x}) = \min_{\mathrm{x}} \int_{\Omega} f(\mathrm{x}, \xi) p(d\xi) = \min_{\mathrm{x}} \mathbb{E}[f(\mathrm{x}, \xi)]. \quad (1)$$

random variable

objective function

probability distribution

decision vector

- Have wide applications in machine learning, control and signal processing tasks.

Q: The probability distribution p is unknown → The expectation $F(x)$ is not computable

One alternative solution:

$$\min_{\mathrm{x}} F(\mathrm{x}) = \min_{\mathrm{x}} \frac{1}{N} \sum_{i=1}^{N} f(\mathrm{x}, \xi_i) \quad (2)$$

> Draw N samples $\{\xi_i\}_{i=1}^{N}$ from the distribution p

> Solve the corresponding empirical risk minimization (ERM) problem

Motivation

Stochastic optimization problem

↓

Stochastic gradient descent (SGD)

$$\min_{\mathbf{x}} F(\mathbf{x}) = \min_{\mathbf{x}} \int_{\Omega} f(\mathbf{x}, \xi) p(d\xi) = \min_{\mathbf{x}} \mathbb{E}[f(\mathbf{x}, \xi)].$$

- The canonical tool for addressing stochastic optimization problems

> Approximate the true gradient $\nabla F(\mathbf{x})$ with a mini-batch gradient $\nabla f_S(\mathbf{x}) = \frac{1}{n} \sum_{i \in S} \nabla f(\mathbf{x}, \xi_i)$

> The update rule is:

At iteration k

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \nabla f_{S_k}(\mathbf{x}_k). \quad (3)$$

step size

> $S_k = \{\xi_1, \dots, \xi_{n_k}\}$ is the mini-batch of samples at iteration k with $n_k = |S_k|$ is the number of samples

> The SGD converges exactly or approximately → To be addressed

Motivation

Stochastic gradient descent (SGD)

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \nabla f_{S_k}(\mathbf{x}_k).$$

Step-size α_k

- Constant step-size has fast rates ⟷ Converge approximately
- Attenuating step-size converges exactly ⟷ Reducing the rate to null

Batch-size n_k

➢ Mini batch is used to reduce the variance of stochastic approximation error

↓

➢ Tighten asymptotic convergence radius

- Geometrically increasing batch-size converges exactly with constant step-size

↕

Computationally expensive with large sample complexity

Motivation: Allowing the batch-size grows as slow as possible while maintaining a fast rate with exact convergence

Outline

- Motivation
- **Two scale adaptive algorithm**
- Convergence and sample complexity reduction
- Numerical simulation
- Conclusions

Two scale adaptive algorithm

Preliminary

Characterize the convergence rate of SGD related to the batch-size n_k and step-size α_k

Three mild assumptions:

1. The gradient of expected objective function $\nabla F(x)$ is Lipschitz continuous: $\| \nabla F(\mathbf{x}) - \nabla F(\mathbf{y}) \|_2 \leq L \| \mathbf{x} - \mathbf{y} \|_2$
2. Objective functions $\{f(\boldsymbol{x}, \boldsymbol{\xi}_i)\}$ are differentiable and $F(x)$ is ℓ −strongly convex
3. There exists a constant ω such that $\| Var[\nabla f_i(\mathbf{x})] \|_1 \leq w$

> Assumptions 1-3 are mild and common in optimization analysis

Proposition 1. With Assumptions 1-3, the SGD with constant step-size $\alpha_k = \alpha$ and batch-size $n_k = n$ satisfies

$$\mathbb{E}\left[F(\mathbf{x}_{k+1}) - F(\mathbf{x}^*)\right] \leq \underbrace{r(\alpha)^{k+1}\left(F(\mathbf{x}_0) - F(\mathbf{x}^*)\right)}_{:=Q_1} + \underbrace{\frac{\alpha L w}{2n(2\ell - L\ell\alpha)}}_{:=Q_2} \quad (4)$$

$r(\alpha) = 1 - 2\ell\alpha + L\ell\alpha^2$

Q_1: convergence rate term

Q_2: error neighborhood term

Two scale adaptive algorithm

Analysis of Proposition 1

$$\mathrm{E}[F(\mathrm{x}_{k+1}\text{-}\mathrm{x}^*)] \le \underbrace{r(\alpha)^{k+1}(F(\mathrm{x}_0) - F(\mathrm{x}^*))}_{Q_1} + \underbrace{\frac{\alpha L \omega}{2n(2\ell - L\ell\alpha)}}_{Q_2}$$

- The convergence rate term Q_1 decreases with iteration k provided that $r(\alpha) < 1$.
- The error neighborhood term Q_2 determines the limiting radius of convergence. ⟷ Inverse dependence on n

Two scale adaptive (TSA) algorithm exploits the structure of Q_1 and Q_2 to improve performance

---

Two scale adaptive algorithm

- Observe once Q_1 decays to be smaller than Q_2, SGD cannot converge to a tighter neighborhood than Q_2.
- Either reduce step-size α or increase batch-size n to further reduce Q_2.

➢ The TSA algorithm gives a strategy about when and how to make this change.

Two scale adaptive algorithm

$$E[F(x_{k+1}-x^*)] \le \underbrace{r(\alpha)^{k+1}(F(x_0)-F(x^*))}_{Q_1} + \underbrace{\frac{\alpha L\omega}{2n(2\ell - L\ell\alpha)}}_{Q_2}$$

TSA consists of two stages: the inner-scale stage performs SGD with constant step-size and batch-size, and the outer-scale stage tunes parameters to tighten the radius of convergence.

Initialization

With initial step-size α_0 and n_0, we have

$$Q_1^0 = r(\alpha_0)^{k+1}(F(x_0)-F(x^*)),\ \ Q_2^0 = \frac{\alpha_0 L\omega}{2n_0(2\ell - L\ell\alpha_0)} \qquad (5)$$

- Note the rate $r(\alpha_0)$ is a quadratic function of the step-size α_0

  ↓

  $\alpha_0 = 1/L$ is selected for an optimal decreasing rate → $r^* = r\left(\frac{1}{L}\right) = 1 - \frac{\ell}{L}$

- The corresponding $Q_2^0 = \frac{\omega}{2n_0\ell}$

➢ To ensure the fastest decreasing of Q_1, we fix the optimal step-size $\alpha = 1/L$ over all iterations and evolving the batch-size n to tighten Q_2.

Two scale adaptive algorithm

Two scale adaptive algorithm

$$E[F(x_{k+1}-x^*)] \le \underbrace{r(\alpha)^{k+1}(F(x_0)-F(x^*))}_{Q_1} + \underbrace{\frac{\alpha L\omega}{2n(2\ell - L\ell\alpha)}}_{Q_2}$$

Inner-scale stage

We have $\alpha_t = 1/L$, n_t, K as current step-size, batch-size and the beginning number of iteration at t-th inner scale stage.

passed number of iterations

$$Q_1^t = \left(1 - \frac{\ell}{L}\right)^{k_t} \mathbb{E}\left[F(\mathbf{x}_K) - F(\mathbf{x}^*)\right],$$
$$Q_2^t = \frac{\alpha_t L w}{2n_t(2\ell - L\ell\alpha_t)} = \frac{w}{2n_t\ell}, \quad (6)$$

- Then there exists K_t such that $K_t = \max_{k_t} \{Q_1^t \ge Q_2^t\}$

the largest iteration before Q_1 drops below Q_2

The duration of t-th inner scale stage

$\mathbb{E}\left[(F(\mathbf{x}_K) - F(\mathbf{x}^*))\right]$ in Q_1^t is unknown, this criterion cannot be used directly.

- We then search for an alternative criterion for implementation

Two scale adaptive algorithm

Two scale adaptive algorithm

$$E[F(x_{k+1}-x^*)] \le \underbrace{r(\alpha)^{k+1}(F(x_0)-F(x^*))}_{Q_1} + \underbrace{\frac{\alpha L \omega}{2n(2\ell - L\ell\alpha)}}_{Q_2}$$

Inner-scale stage

- Let $\{n_0, \dots, n_{t-1}\}$ and $\{K_0, \dots, K_{t-1}\}$ be batch-sizes and durations of previous inner-scale stages such that $K = \sum_{i=1}^{t-1} K_i$
- We then have

$$\mathbb{E}\left[(F(\mathbf{x}_K) - F(\mathbf{x}^*))\right] = \mathbb{E}\left[\left(F(\mathbf{x}_{\sum_{i=0}^{t-1} K_i}) - F(\mathbf{x}^*)\right)\right]$$

$$\le \left(1 - \frac{\ell}{L}\right)^{K_{t-1}} \mathbb{E}\left[F(\mathbf{x}_{\sum_{i=0}^{t-2} K_i}) - F(\mathbf{x}^*)\right] + Q_2^{t-1} \quad \text{Proposition 1}$$

$$\le 2\left(1 - \frac{\ell}{L}\right)^{K_{t-1}} \mathbb{E}\left[F(\mathbf{x}_{\sum_{i=0}^{t-2} K_i}) - F(\mathbf{x}^*)\right]. \quad \text{Definition of } K_{t-1} \tag{7}$$

- By recursively applying this property, we get

$$Q_1^t \le 2^t \left(1 - \frac{\ell}{L}\right)^{\sum_{i=0}^{t-1} K_i + k_t} (F(\mathbf{x}_0) - F(\mathbf{x}^*)) \tag{8}$$

The alternative criterion:

$$K_t = \max_{k_t}\left\{2^t\left(1 - \frac{\ell}{L}\right)^{\sum_{i=0}^{t-1} K_i + k_t} (F(\mathbf{x}_0) - F(\mathbf{x}^*)) \ge \frac{w}{2n_t\ell}\right\} \tag{9}$$

Two scale adaptive algorithm

Two scale adaptive algorithm

$$E[F(x_{k+1}\text{-}x^*)] \le \underbrace{r(\alpha)^{k+1}(F(x_0)-F(x^*))}_{Q_1} + \underbrace{\frac{\alpha L\omega}{2n(2\ell - L\ell\alpha)}}_{Q_2}$$

Outer-scale stage

- Evolve the step-size and the batch-size to reduce the error neighborhood term Q_2

Slow down the convergence rate $r(\alpha_t)$

Increases the sample computational complexity

- Fix the step-size to maintain the fastest decreasing of Q_1

- Increase the batch-size in one of two ways

Additive way: $n_{t+1} = n_t + \beta_t, \quad \beta_t \ge 1,$

Multiplicative way: $n_{t+1} = m_t n_t, \quad m_t > 1,$

Outline

- Motivation
- Two scale adaptive algorithm
- **Convergence and sample complexity reduction**
- Numerical simulation
- Conclusions

Convergence

Exact convergence of TSA algorithm

- The sequence of objective values $F(\mathbf{x}_k)$ generated by the TSA converges to the optimal value $F(\mathbf{x}^*)$ exactly

Theorem 1. Consider the TSA scheme. If Assumptions 1-3 hold, we have

$$\lim_{k\to\infty} \mathbb{E}\left[F(\mathbf{x}_k) - F(\mathbf{x}^*)\right] = 0,$$

$$\lim_{k\to\infty} \mathbb{E}\left[\| \mathbf{x}_k - \mathbf{x}^* \|_2\right] = 0.$$

- The TSA scheme inherits the asymptotic convergence behavior of SGD with attenuating step-size selection.

With constant step-size → Increase the convergence rate

Sample complexity reduction

- One critical benefit of TSA is the sample complexity reduction compared with SGD

  ↓

  Require less sample computation to achieve an $\varepsilon -$ suboptimality

- For a clear comparison, we assume: 1. SGD uses the same optimal step-size and constant batch-size

  2. The TSA uses the multiplicative way to increase batch-size with $m_t = m$

Theorem 2. Consider the TSA scheme with initial batch-size $n_0 = 1$ and the SGD with step-size $\alpha = 1/L$ and batch-size n. Let $D = F(\mathbf{x}_0) - F(\mathbf{x}^*)$ be the initial error. To achieve an $\varepsilon -$ suboptimality, the ratio between the number of training samples required for TSA and SGD is

$$\gamma \leq \frac{m\left\lceil \log_{1-\frac{\ell}{L}} \frac{L-\ell}{2mL} \right\rceil}{(m-1)\left\lceil \log_{1-\frac{\ell}{L}} \frac{\epsilon}{2D} \right\rceil} + \mathcal{O}(\epsilon), \quad \mathbf{(10)}$$

Convergence

Sample complexity reduction

$$\gamma \leq \frac{m \left\lceil \log_{1-\frac{\ell}{L}} \frac{L-\ell}{2mL} \right\rceil}{(m-1) \left\lceil \log_{1-\frac{\ell}{L}} \frac{\epsilon}{2D} \right\rceil} + \mathcal{O}(\epsilon),$$

- The ratio is approximately proportional to $\mathcal{O}(-1/\log\epsilon) + \mathcal{O}(\epsilon)$.

> For accurate solutions, i.e., ε is close to null, a significant sample complexity reduction is achieved

- For special case when $m = 2$, we refer that

$$\epsilon \leq D(1-\ell/L)^2/8 \quad \longleftrightarrow \quad \gamma < 1 \qquad (11)$$

This is almost always true unless the initial point is very close to the optimizer

- Overall, the TSA only increase the batch-size when necessary and saves sample complexity as much as possible

Outline

- Motivation
- Two scale adaptive algorithm
- Convergence and sample complexity reduction
- **Numerical simulation**
- Conclusions

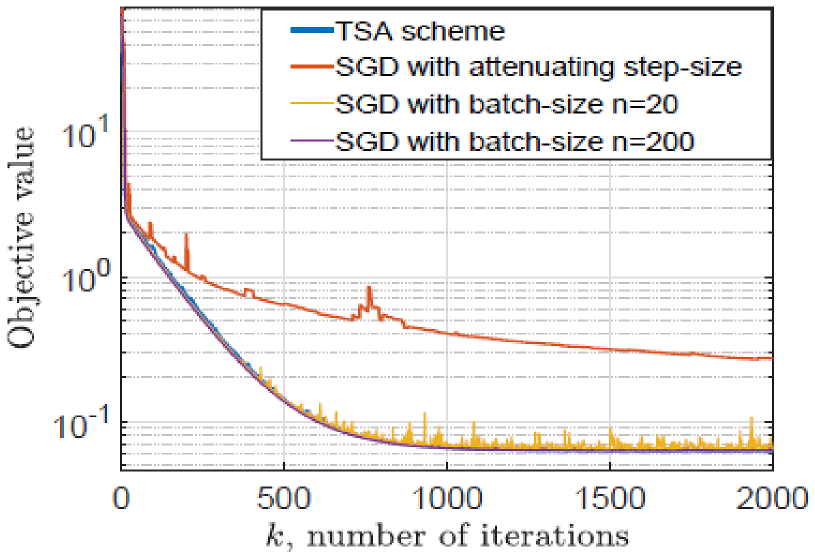
Numerical simulation

Hand-written digits classification

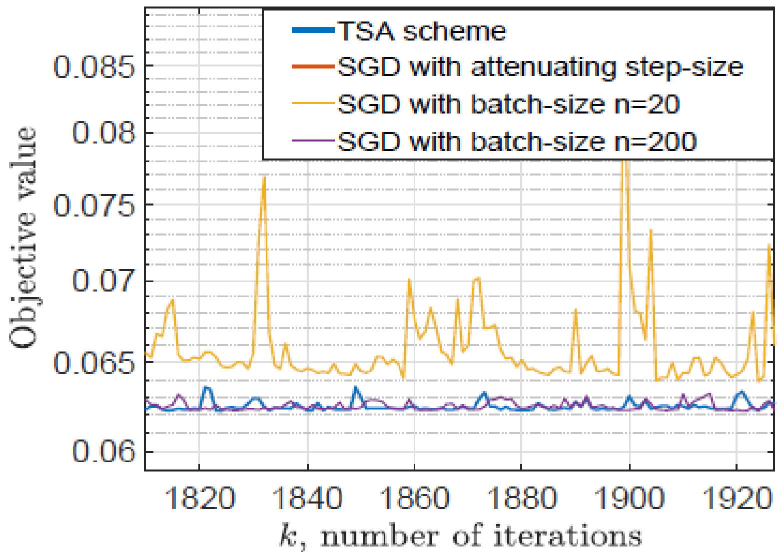
- MNIST dataset

✓ Formulate the problem as a logistic regression to train a hand-written digit classifier

- Performance comparison between TSA and SGD schemes



(a) Objective vs. iteration (overall figure)



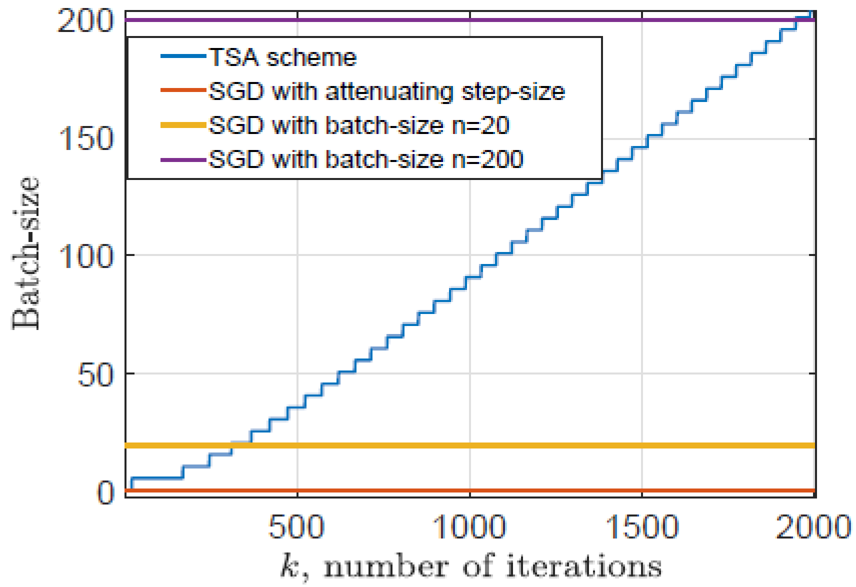
(b) Objective vs. iteration (larger figure)

- TSA has a comparable performance with SGD of n=200

Numerical simulation

Hand-written digits classification

- Sample complexity comparison between TSA and SGD schemes



(c) Batch-size vs. iteration

- TSA saves almost half of sample complexity compared with SGD of n=200

Table 1: Number of training samples required to reduce the objective below 0.0622 for three algorithms: TSA, SGD with $n = 200$ and SGD with $n = 20$.

	Number of required samples
TSA	55651
SGD with $n = 200$	111500
SGD with $n = 20$	∞

- For an $\epsilon = 0.0622$-suboptimality, TSA saves more than a half samples compared with SGD of n=200.
- SGD of n=20 can never achieve this accuracy due to the large error neighborhood term Q_2

Outline

- Motivation
- Two scale adaptive algorithm
- Convergence and sample complexity reduction
- Numerical simulation
- **Conclusions**

Conclusion

- Propose the two scale adaptive algorithm that balances the rate and variance in the stochastic optimization problem.

- The exact convergence and the sample complexity is obtained for the TSA scheme

- Numerical simulations are performed to show strong performance of TSA compared with the SGD.

Thank you !