Stochastic Gradient Method

Andersen Ang

Mathématique et recherche opérationnelle
UMONS, Belgium

manshun.ang@umons.ac.be Homepage: angms.science

First draft : MAy 17, 2019
Last update : May 18, 2019

Draw back of gradient descent for finite sum

Suppose we want to minimize a function that is a finite sum

$$\min_{\mathbf{x}} f(\mathbf{x}) = \sum_{i}^{n} f_i(\mathbf{x})$$

Assuming f and f_i are all differentiable, with suitable chosen step size α_k, the gradient descent approach to solve such problem is

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k \nabla f(\mathbf{x}^k),$$

where the full gradient here is also a finite sum

$$\nabla f(\mathbf{x}) = \sum_{i}^{n} \nabla f_i(\mathbf{x}).$$

Drawback of gradient descent : as $\nabla f(\mathbf{x})$ depends on n, the computation of the full gradient is expensive if n is very large.

Big data

Suppose we want to minimize a function that is a finite sum of objective function over many data points

$$\min_{\text{all } \mathbf{x}_i} \sum_{i}^{n} f(\mathbf{x}_i)$$

Note this is different from the last slide : in this slide, subscript i is attached to $\mathbf{x}$ not f, meaning that we are using the same objective function f on different data points $\mathbf{x}_i$.
In a more general case, we can have different objective function takes in different input :

$$\min_{\text{all } \mathbf{x}_i} \sum_{j} \sum_{i} f_j(\mathbf{x}_i)$$

However, in last slide : we have

$$\min_{\mathbf{x}} \sum_{i} f_i(\mathbf{x}),$$

the subscript i is attached to f, meaning that the objective function has many component f_i while they all take the same $\mathbf{x}$ as input.

The focus of this document : not mini-batch gradient

This document focus on $\sum_i f_i(\mathbf{x})$, not $\sum_i f(\mathbf{x}_i)$.

That is, we consider the case the expensiveness of computing full gradient comes from the objective function itself, rather than due to the large number of data points.

For the "big data problem" – having too many data points, one way to solve it is to use mini-batch gradient descent : in stead of minimizing $\sum_i^n f(\mathbf{x}_i)$ over $i = 1$ to n (i.e. the whole data set), we minimize with respect to an batch of data :

$$\min_{\mathbf{x}_i, i \in \mathcal{B}} \sum_{i \in \mathcal{B}} f(\mathbf{x}_i)$$

where $\mathcal{B}$ is a set holding the sampled indices from $\{1, 2, \ldots, n\}$.

However, mini-batch gradient is not the focus of this document.

Why mention these : many people mix them up.

The idea of stochastic gradient Methods

Situation : computing the full gradient is expensive (due to the objective function has too many components).

Then, instead of computing the true full gradient, we can compute an *approximate* version of it :

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k g^k$$

where $g^k \approx \nabla f(\mathbf{x}^k)$.

Important points on g :

- It is to be easier to compute than the full gradient $\nabla f(\mathbf{x})$
- It has to be an "accurate" approximation of the full gradient $\nabla f(\mathbf{x})$

One way to approximate $\nabla f(\mathbf{x})$: make sure $\mathbb{E}[g] = \nabla f(\mathbf{x})$.
i.e., g has to be an unbiased estimator of $\nabla f(\mathbf{x})$

Stochastic gradient (SG)

One simply way to approximate the full gradient $\nabla f(\mathbf{x}) = \sum_i^n \nabla f_i(\mathbf{x})$ is to take only one component of the full gradient :

$$g = \nabla f_i(\mathbf{x}),$$

where i is sampled over the index set $\{1, 2, \ldots, n\}$.

With g, we can then perform the update of $\mathbf{x}$ as

$$\begin{aligned} \mathbf{x}^{k+1} &= \mathbf{x}^k - \alpha_k g^k \\ &= \mathbf{x}^k - \alpha_k \nabla f_i(\mathbf{x}^k). \end{aligned}$$

Compared to the full gradient method

$$\text{(Full) gradient method} \quad \mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k \sum_{i=1^n} \nabla f_i(\mathbf{x}^k),$$

SG is independent of n. Hence when $n > 1$, SG is computationally cheaper than the full gradient method.

Stochastic gradient and other methods

Stochastic gradient update

$$\mathbf{x}^{k+1} \quad = \quad \mathbf{x}^k - \alpha_k \nabla f_i(\mathbf{x}^k).$$

Full gradient descent (GD) method

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k \sum_{i=1^n} \nabla f_i(\mathbf{x}^k),$$

We can see that SG is independent of n. Hence when $n > 1$, SG is computationally cheaper than the full gradient method.

However, SG is slower than GD, it can be shown the convergence rate of GD is higher than that of SG.

Hence in general we should always use GD instead of SG if GD is doable : do not forget the main reason why SG is used — SG is an substitute of GD when n is very large (e.g. millions, billions)

Stochastic gradient and other methods

Stochastic gradient update

$$\mathbf{x}^{k+1} \quad = \quad \mathbf{x}^k - \alpha_k \nabla f_i(\mathbf{x}^k).$$

Randomized coordinate gradient (RCG, a kind of coordinate descent)

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k \mathbf{U}_{i_k} \nabla f(\mathbf{x}^k_{i_k})$$

The notation are the "same" but the setting and meaning are different :

- For SG, the objective function f is separable (as a finite sum).
  While in RCG, the variable $\mathbf{x}$ is separable.
- The update of $\mathbf{x}$ in SG applies to the whole vector $\mathbf{x}$.
  RCG only update part of $\mathbf{x}$: hence we have the subscript i_k and the matrix $\mathbf{U}$.

For more details on RCG, see here.

Stochastic gradient algorithm

Input : initial guess of $\mathbf{x}$
Output : $\mathbf{x}$ that approximately solve $\min f(\mathbf{x})$

FOR $k = 1, 2, \ldots$

- Sample $i \in \{1, 2, \ldots, n\}$
- Compute partial gradient $\nabla f_i(\mathbf{x}^k)$
- Update $\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k \nabla f_i(\mathbf{x}^k)$

END FOR

How to sample : one simple way is to sample i (with replacement) from the set $\{1, 2, \ldots, n\}$ by random under some distribution such as uniform distribution

$$\mathbb{P}(i = 1) = \mathbb{P}(i = 2) = \cdots = \mathbb{P}(i = n) = \frac{1}{n}.$$

Stochastic gradient is not a descent method

Full gradient method is a descent method : we always have $f(\mathbf{x}^{k+1}) \leq f(\mathbf{x}^k)$. That's why it is called "gradient descent".

However, stochastic gradient is not a descent method. It is possible for the objective function to increase in some iterations.

Why : look at the update equation of stochastic gradient method :

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k \nabla f_i(\mathbf{x}).$$

The key is that the direction $\nabla f_i(\mathbf{x})$ computed is the descent direction of $f_i(\mathbf{x})$, but not necessary the descent direction of $f(\mathbf{x})$. The update $\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k \nabla f_i(\mathbf{x})$ will guarantee decrease in $f_i(\mathbf{x})$ but not necessary $f(\mathbf{x})$.

Hence it is actually wrong for the term : stochastic gradient descent.

Simple example : two dimensional quadratic function

Consider $\underset{\mathbf{x}\in\mathbb{R}^2}{\arg\min} f(\mathbf{x}) = \frac{1}{2}\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2$ with $\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$, $\mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$.

Full gradient will be $\nabla f(\mathbf{x}) = \mathbf{A}^\top\mathbf{A}\mathbf{x} - \mathbf{A}^\top\mathbf{b}$. Gradient descent update will be

$$\mathbf{x}^{k+1} = x^k - \alpha_k \nabla f(\mathbf{x})$$

To express f as a finite sum, we have

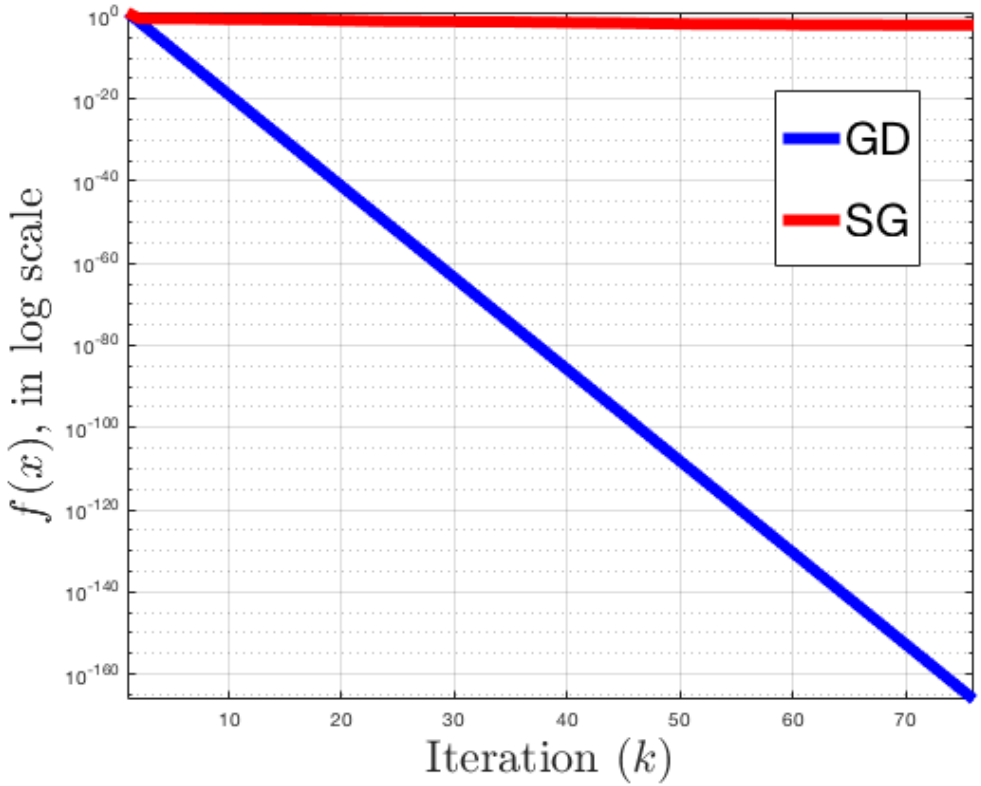
$$f(\mathbf{x}) = \underbrace{\frac{1}{2}\left(\left\langle \begin{pmatrix} a_{11} \\ a_{12} \end{pmatrix}, \mathbf{x} \right\rangle - b_1\right)^2}_{f_1} + \underbrace{\frac{1}{2}\left(\left\langle \begin{pmatrix} a_{21} \\ a_{22} \end{pmatrix}, \mathbf{x} \right\rangle - b_2\right)^2}_{f_2}$$

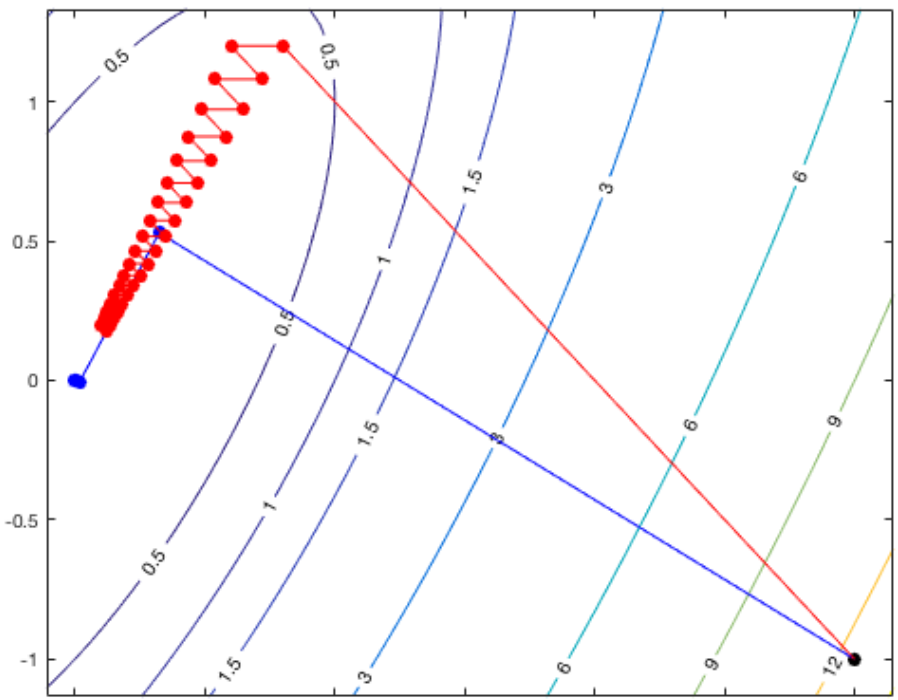
The partial gradients are

$$\begin{aligned} \nabla f_1(\mathbf{x}) &= \left(\left\langle \begin{pmatrix} a_{11} \\ a_{12} \end{pmatrix}, \mathbf{x} \right\rangle - b_1\right) \begin{pmatrix} a_{11} \\ a_{12} \end{pmatrix} \\ \nabla f_2(\mathbf{x}) &= \left(\left\langle \begin{pmatrix} a_{21} \\ a_{22} \end{pmatrix}, \mathbf{x} \right\rangle - b_2\right) \begin{pmatrix} a_{21} \\ a_{22} \end{pmatrix} \end{aligned}$$

Simple example : two dimensional quadratic function

The following shows SG is slow compared to GD on the two dimensional quadratic example (here optimal step size is used).
In general, SG is fast in beginning but slow asymptotically. It can be proved that the convergence rate of SG is slower than that of GD.





Last page - summary

Stochastic Gradient for the problem $\min f(\mathbf{x}) = \sum f_i(\mathbf{x})$

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k \nabla f_i(\mathbf{x}^k)$$

Not convered

- Convergence rate of SG
- How to improve SG :
  - Stochastic Average Gradient (SAG)
  - Stochastic Variance Reduced Gradient (SVRG)
  - Stochastic Average Gradient Améliore (SAGA)

End of document