Accelerated First-Order Optimization Algorithms for Machine Learning

Huan Li

Nankai University

Outline

- Optimization in Machine Learning
- Accelerated Gradient Descent for deterministic optimization
- Accelerated Gradient Descent for stochastic optimization
- Accelerated Gradient Descent for distributed optimization

Focus of This Talk

- First-Order Optimization Algorithms
  - Not higher-order algorithms, such as Newton's method
- Accelerated Algorithms
- Convex Optimization
  - Not nonconvex optimization
- Widely Used in Machine Learning
  - Not control, finance

Machine Learning

- Machine learning is one of the fastest-growing areas.
- Goal
  - Extract meaning from data: understand statistical properties, learn important features and fundamental structures in the data.
  - Use this knowledge to make decisions or predictions about other data.
- Optimization is at the heart of machine learning

$$\boxed{\text{Machine Learning} = \text{Representation} + \text{Optimization} + \text{Evaluation}}$$

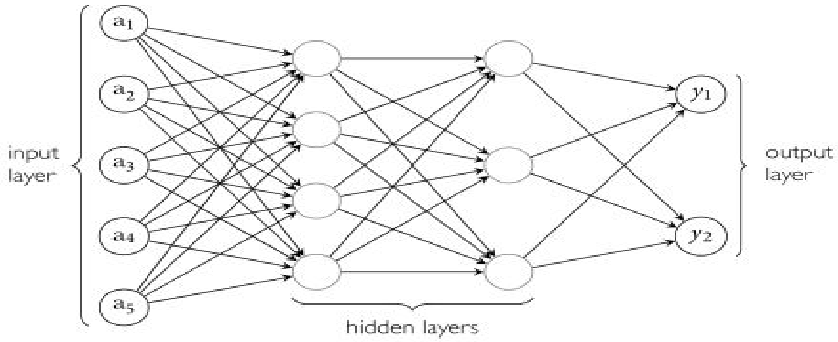
- Most of the machine learning problems are, in the end, optimization problems.

Typical Setup

- After cleaning and formatting, obtain a data set of n objects (a_j, y_j)
  - Vectors of input features: $a_j, j = 1,2, \dots, n$
  - Outcome y_j for each feature vector
- The outcomes y_j could be:
  - a real number: regression.
  - a label indicating that a_j lies in one of M classes (for $M \geq 2$): classification.
  - no labels (y_j is null), e.g., clustering: partition the a_j into a few clusters.

Fundamental Machine Learning Task

- Seek a function $\phi(\cdot, x)$ parametrized by x that
  - (training) approximately maps a_j to y_j for each j in the training set:
  - (testing) use the model to predict the output on new inputs.
- Example of prediction functions
  - Highly non-linear neural network



- Training: optimization comes into play.
  - Compute the parameter x which explains at best the data.

Training

- Typically, the training phase is formulated as an optimization problem

$$\boxed{\min_{x \in \mathbb{R}^p} \frac{1}{n} \sum_{j=1}^{n} \ell(\phi(a_j, x), y_j) + \lambda \Omega(x)}$$

- Loss function $l(z, y)$: measure of the mismatch.
- Interests of the regularization term Ω
  - avoid over-fitting on known data to better generalize to new data.
- Practical consequences for training

$$\boxed{\text{try to find (quickly) solutions.}}$$

Properties of Optimization Problems in Machine Learning

- Recall the optimization problem in machine learning

$$\boxed{\min_{x \in \mathbb{R}^p} \frac{1}{n} \sum_{j=1}^{n} \ell(\phi(a_j, x), y_j) + \lambda \Omega(x)}$$

- High dimension: p is large.
  - Millions of weights in deep neural network
  - First-order algorithms, not higher-order algorithms
- Large data: n is large
  - 1,281,167 training images in ImageNet
  - Stochastic algorithms and distributed algorithms

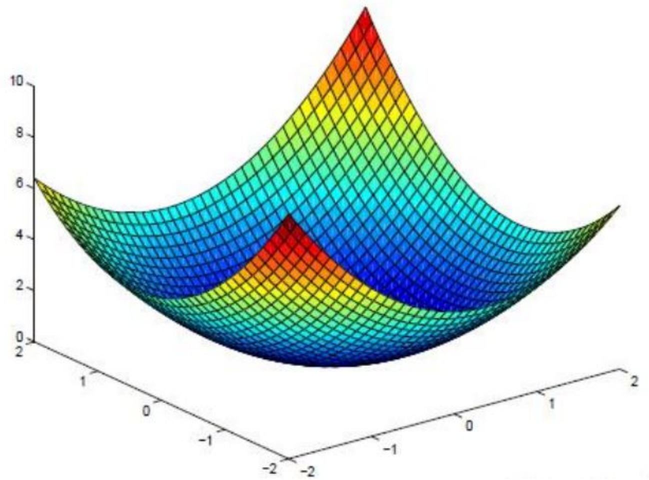
Basic Introduction to Convex Optimization

- Recall that training phase is formulated as an optimization problem

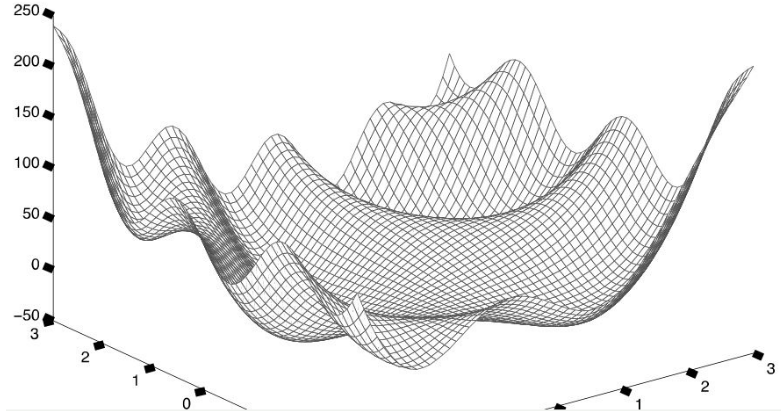
$$\min_{x \in \mathbb{R}^p} f(x)$$

- Convex formulations are often tractable and efficient in practice.
- Nonconvex formulations are more natural, but harder to solve and analyze.

Convex function



Non-convex function

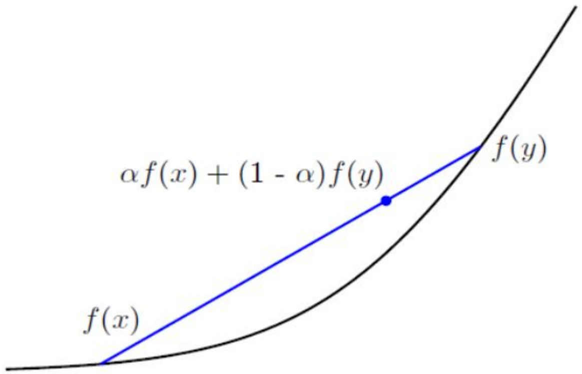


Basic Introduction to Convex Optimization

➢ Convex function

- A function $f(x)$ is convex if for any $x, y \in dom f$ and any $\alpha \in [0,1]$,

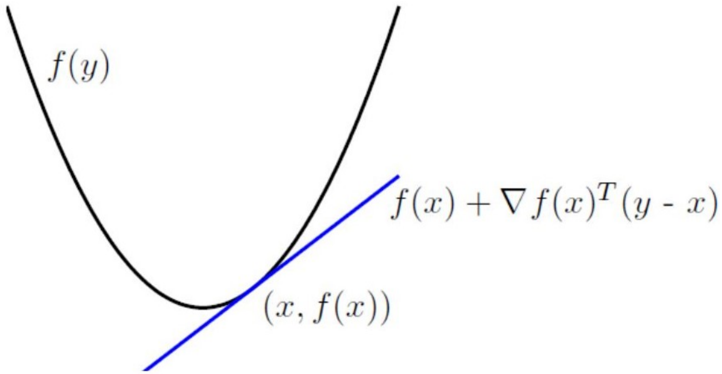
$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha) f(y)$$



- Property

$$\boxed{f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle}$$

We assume that $f(x)$ is differentiable such that $\nabla f(x)$ exists.



- x^* is the global minimizer of $f(x)$, if and only if

$$\nabla f(x^*) = 0$$

Basic Introduction to Convex Optimization

➢ Strongly convex function

- A function $f(x)$ is strongly convex if for any $x, y \in dom f$ and any $\alpha \in [0, 1]$,

$$\frac{\mu}{2}\alpha(1-\alpha)\|y-x\|^2 + f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha)f(y)$$

- Property

$$\boxed{f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}\|y-x\|^2}$$

- The minimizer is unique

Basic Introduction to Convex Optimization

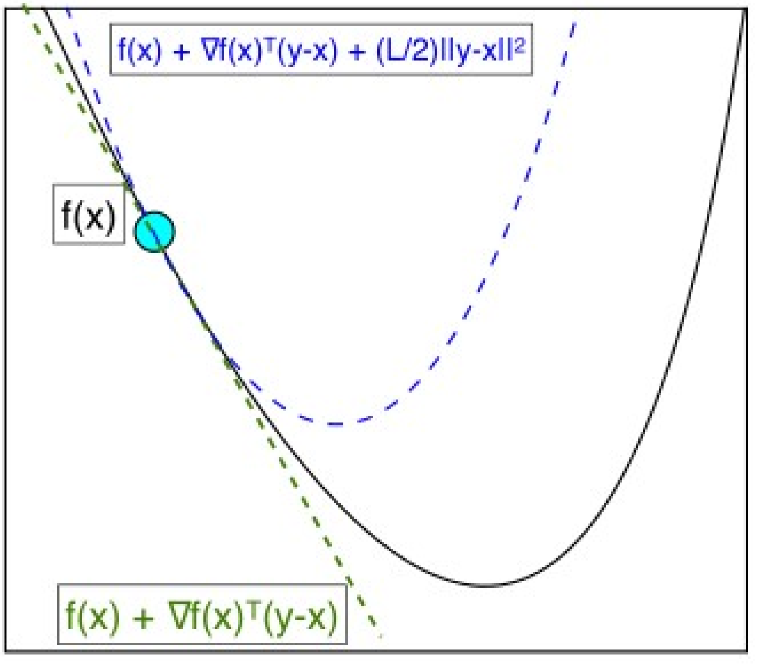
➢ Smoothness

- From Taylor's theorem, for some z we have

$$f(y) \le f(x) + \langle \nabla f(x), y - x \rangle + \frac{1}{2}(y-x)^T \nabla^2 f(z)(y-x)$$

- Using that $v^T \nabla^2 f(z) v \le L\|v\|^2$ for any v and z, we have

$$\boxed{f(y) \le f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|y-x\|^2}$$



- Global quadratic upper bound on function value

- Another form:

$$\boxed{\|\nabla f(y) - \nabla f(x)\| \le L\|y-x\|}$$

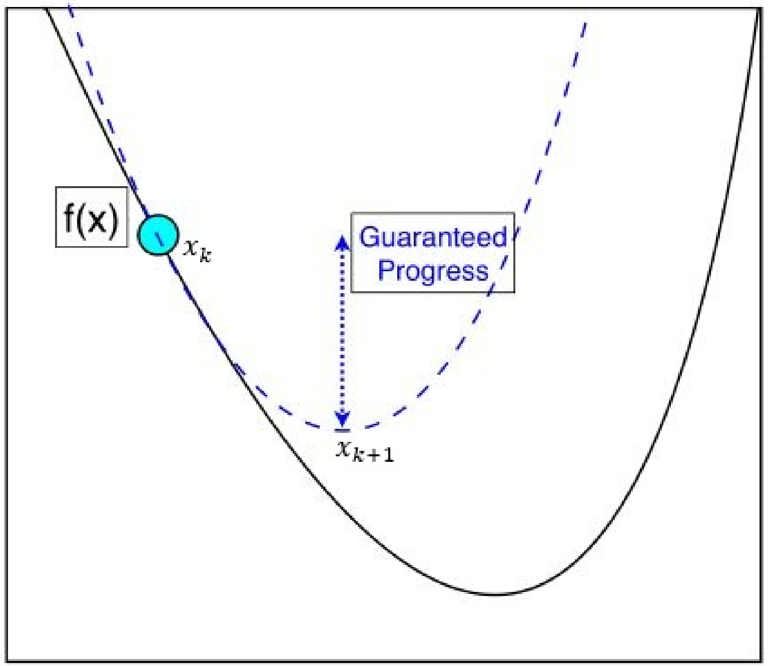
Gradient Descent

- Consider the basic problem $min_x\ f(x)$.
- We have the upper bound

$$
\begin{aligned}
f(y) &\le f(x_k) + \langle \nabla f(x_k), y - x_k \rangle + \frac{L}{2}\|y - x_k\|^2 \\
&= f(x_k) + \frac{L}{2}\left\| y - x_k + \frac{1}{L}\nabla f(x_k)\right\|^2 - \frac{1}{2L}\|\nabla f(x_k)\|^2
\end{aligned}
$$

- treating x_{k+1} as a variable that minimizing the right side gives

$$\boxed{x_{k+1} = x_k - \frac{1}{L}\nabla f(x_k)}$$



  - every iteration is inexpensive
  - does not require second derivatives

Convergence Rate of Gradient Descent

Theorem: Suppose that the function $f(x)$ is convex and smooth, then for GD we have

$$f(x_{k+1}) - f(x^*) \leq \frac{L\|x_0 - x^*\|^2}{2(k+1)} = O\left(\frac{1}{k}\right)$$

- We say the convergence rate of gradient descent is $O(\frac{1}{k})$ $\quad \frac{1}{k} = \epsilon \Rightarrow k = \frac{1}{\epsilon}$
- Equivalently, to find an ϵ accurate solution x such that $f(x) - f(x^*) \leq \epsilon$,

  we need $O(\frac{1}{\epsilon})$ iterations. We say the complexity of gradient descent is $O(\frac{1}{\epsilon})$

Proof of the Convergence Rate

Recall that we assume the convexity property of

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle \tag{1}$$

and the smoothness property of

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|y - x\|^2 \tag{2}$$

Gradient descent iterates as

$$x_{k+1} = x_k - \frac{1}{L}\nabla f(x_k) \tag{3}$$

Proof. It follows that

$$\begin{aligned}
f(x_{k+1}) &\overset{(2)}{\leq} f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k \rangle + \frac{L}{2}\|x_{k+1} - x_k\|^2 \\
&= f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \langle \nabla f(x_k), x_{k+1} - x \rangle + \frac{L}{2}\|x_{k+1} - x_k\|^2 \\
&\overset{(1)}{\leq} f(x) + \langle \nabla f(x_k), x_{k+1} - x \rangle + \frac{L}{2}\|x_{k+1} - x_k\|^2 \\
&\overset{(3)}{=} f(x) - L\langle x_{k+1} - x_k, x_{k+1} - x \rangle + \frac{L}{2}\|x_{k+1} - x_k\|^2 \\
&= f(x) + \frac{L}{2}\left(\|x_k - x\|^2 - \|x_{k+1} - x\|^2 - \|x_{k+1} - x_k\|^2\right) + \frac{L}{2}\|x_{k+1} - x_k\|^2 \\
&= f(x) + \frac{L}{2}\|x_k - x\|^2 - \frac{L}{2}\|x_{k+1} - x\|^2.
\end{aligned}$$

$$\begin{aligned}
&\|x_{k+1} - x\|^2 - 2\langle x_{k+1} - x, x_{k+1} - x_k \rangle + \|x_{k+1} - x_k\|^2 \\
&= \|(x_{k+1} - x) - (x_{k+1} - x_k)\|^2 = \|x_k - x\|^2
\end{aligned}$$

Proof of the Convergence Rate

Recall $f(x_{k+1}) \leq f(x) + \frac{L}{2}\|x_k - x\|^2 - \frac{L}{2}\|x_{k+1} - x\|^2$.

Letting $x = x_k$, we have $f(x_{k+1}) \leq f(x_k) - \frac{L}{2}\|x_{k+1} - x_k\|^2 \leq f(x_k)$. So we have

$$f(x_{k+1}) \leq f(x_k) \leq f(x_{k-1}) \leq \cdots \leq f(x_0). \tag{4}$$

Letting $x = x^*$, we have

$$f(x_{k+1}) - f(x^*) \leq \frac{L}{2}\|x_k - x^*\|^2 - \frac{L}{2}\|x_{k+1} - x^*\|^2.$$

Letting $k = 0, 1, \cdots, K$, we have

$$\begin{aligned}
f(x_{K+1}) - f(x^*) &\leq \frac{L}{2}\|x_K - x^*\|^2 - \frac{L}{2}\|x_{K+1} - x^*\|^2 \\
f(x_K) - f(x^*) &\leq \frac{L}{2}\|x_{K-1} - x^*\|^2 - \frac{L}{2}\|x_K - x^*\|^2 \\
f(x_{K-1}) - f(x^*) &\leq \frac{L}{2}\|x_{K-2} - x^*\|^2 - \frac{L}{2}\|x_{K-1} - x^*\|^2 \\
&\vdots \\
f(x_1) - f(x^*) &\leq \frac{L}{2}\|x_0 - x^*\|^2 - \frac{L}{2}\|x_1 - x^*\|^2
\end{aligned}$$

Summing up, we have

$$\sum_{k=0}^{K} f(x_{k+1}) - (K+1)f(x^*) \leq \frac{L}{2}\|x_0 - x^*\|^2 - \frac{L}{2}\|x_{K+1} - x^*\|^2 \leq \frac{L}{2}\|x_0 - x^*\|^2.$$

From (4), we have

$$(K+1)f(x_{K+1}) - (K+1)f(x^*) \leq \frac{L}{2}\|x_0 - x^*\|^2.$$

Dividing both sides by $K+1$, we have

$$f(x_{K+1}) - f(x^*) \leq \frac{L}{2(K+1)}\|x_0 - x^*\|^2.$$

Convergence Rate of Gradient Descent

Theorem: Suppose that the function $f(x)$ is strongly convex and smooth, then for GD, we have

$$f(x_{k+1}) - f(x^*) \leq \left(1 - \frac{\mu}{L}\right)^k \frac{L\|x_0 - x^*\|^2}{2} = O\left(\left(1 - \frac{\mu}{L}\right)^k\right)$$

- We say the convergence rate of gradient descent is $O\left(\left(1 - \frac{\mu}{L}\right)^k\right)$
- Equivalently, to find an ϵ accurate solution x such that $f(x) - f(x^*) \leq \epsilon$,

  we need $O\left(\frac{L}{\mu}\log\frac{1}{\epsilon}\right)$ iterations. We say the complexity of GD is $O\left(\frac{L}{\mu}\log\frac{1}{\epsilon}\right)$

$$\left(1 - \frac{\mu}{L}\right)^k = \epsilon \Rightarrow k\log\left(1 - \frac{\mu}{L}\right) = \log\epsilon \overset{\log(1-x)\approx -x}{\Rightarrow} -k\frac{\mu}{L} = \log\epsilon \Rightarrow k\frac{\mu}{L} = \log\frac{1}{\epsilon} \Rightarrow k = \frac{L}{\mu}\log\frac{1}{\epsilon}$$

Summary of the Complexity of Gradient Descent

Method	Strongly convex	Non-strongly convex
Gradient Descent	$O\left(\frac{L}{\mu}\log\frac{1}{\varepsilon}\right)$	$O\left(\frac{L}{\varepsilon}\right)$

- Can we hope to further accelerate convergence? Yes
  - Heavy ball method
  - Accelerated gradient method

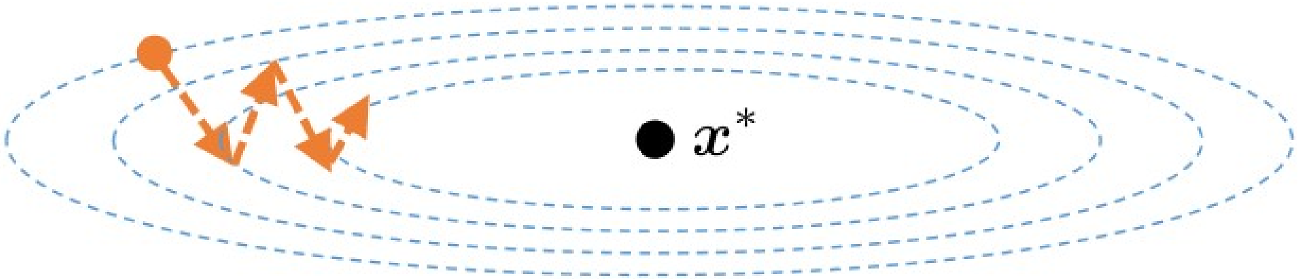
Heavy Ball Method with Momentum

- Fundamental idea:
  - Exploit information from the history (i.e. past iterates)
  - Use momentum to predict the trajectory
- The heavy ball method

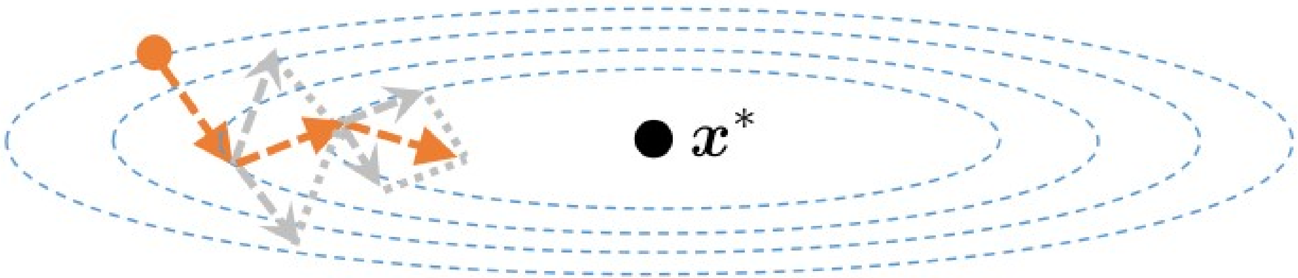
$$\boxed{x_{k+1} = x_k - \alpha \nabla f(x_k) + \underbrace{\beta(x_k - x_{k-1})}_{\text{momentum}}}$$

Search direction at iteration k depends on the latest gradient $\nabla F(x_k)$ and also the search direction at iteration $k-1$,

An Intuitive Comparison



gradient descent



heavy-ball method

Convergence Rate of the Heavy Ball Method

Theorem: Suppose that the function $f(x)$ is strongly convex and smooth, and moreover, it is twice continuously differentiable, then for HB, we have

$$f(x_{k+1}) - f(x^*) \leq O\left(\left(1 - \sqrt{\frac{\mu}{L}}\right)^k\right)$$

Theorem: Suppose that the function $f(x)$ is convex and smooth, then for HB, we have

$$f(x_{k+1}) - f(x^*) \leq O\left(\frac{1}{k}\right)$$

- HB Converges faster than GD only for strongly convex problems

Summary of the Complexity Comparisons

Method	Strongly convex	Non-strongly convex
Gradient Descent	$O\left(\frac{L}{\mu}\log\frac{1}{\varepsilon}\right)$	$O\left(\frac{L}{\varepsilon}\right)$
heavy-ball	$O\left(\sqrt{\frac{L}{\mu}}\log\frac{1}{\varepsilon}\right)$	$O\left(\frac{L}{\varepsilon}\right)$

- Can we further accelerate convergence for general convex problems? Yes
  - Accelerated gradient method

Accelerated Gradient Descent

- Also use the history information and momentum

$$y_k = x_k + \frac{\theta_k(1-\theta_{k-1})}{\theta_{k-1}}(x_k - x_{k-1})$$

$$x_{k+1} = y_k - \frac{1}{L}\nabla f(y_k)$$

where θ_k is computed by $\theta_k = \frac{\sqrt{\theta_{k-1}^4 + 4\theta_{k-1}^2} - \theta_{k-1}^2}{2}$,which is obtained from $\frac{1-\theta_k}{\theta_k^2} = \frac{1}{\theta_{k-1}^2}$ and $x_0 = x_{-1}$, $\theta_0 = 1$

- Recall the heavy ball iterations

$$x_{k+1} = x_k - \alpha\nabla f(x_k) + \beta(x_k - x_{k-1})$$

- Different momentum

Convergence Rate of Accelerated Gradient Descent

Theorem: Suppose that the function $f(x)$ is convex and smooth, then for AGD, we have

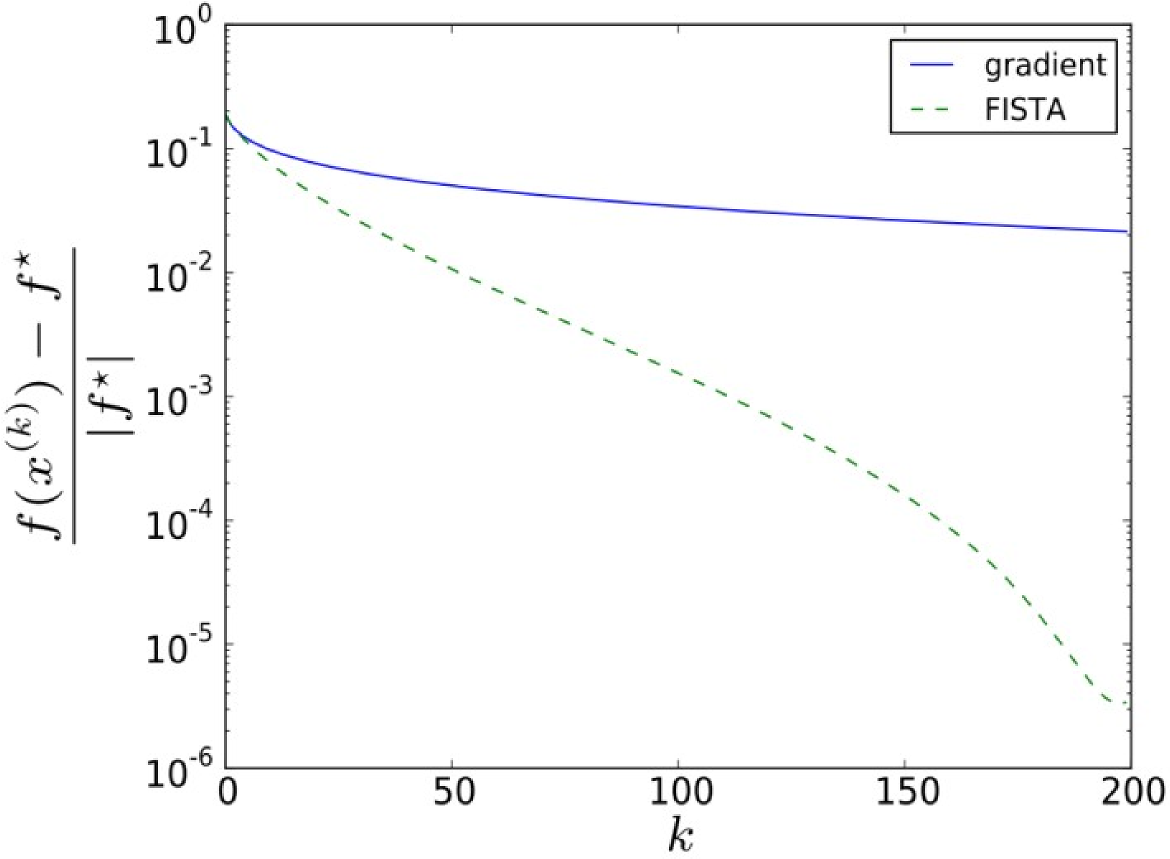
$$f(x_{k+1}) - f(x^*) \leq O\left(\frac{1}{k^2}\right)$$

$$\frac{1}{k^2} = \epsilon \Rightarrow k = \sqrt{\frac{1}{\epsilon}}$$

- The convergence rate of accelerated gradient descent is $O\left(\frac{1}{k^2}\right)$
- Equivalently, the complexity of accelerated gradient descent is $O\left(\sqrt{\frac{1}{\epsilon}}\right)$
- Recall that the convergence rate of gradient descent is $O\left(\frac{1}{k}\right)$ (or $O\left(\frac{1}{\epsilon}\right)$)

Numerical Comparisons

$$\text{minmize} \quad \log \sum_{i=1}^{m} \exp(a_i^T x + b_i)$$



Proof of the Convergence Rate

Recall that we assume the convexity property of

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle \tag{1}$$

and the smoothness property of

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|y - x\|^2 \tag{2}$$

Proof. It follows that

$$\begin{aligned}
f(x_{k+1}) &\overset{(2)}{\leq} f(y_k) + \langle \nabla f(y_k), x_{k+1} - y_k \rangle + \frac{L}{2}\|x_{k+1} - y_k\|^2 \\
&= f(y_k) + \langle \nabla f(y_k), x - y_k \rangle + \langle \nabla f(y_k), x_{k+1} - x \rangle + \frac{L}{2}\|x_{k+1} - y_k\|^2 \\
&\overset{(1)}{\leq} f(x) + \langle \nabla f(y_k), x_{k+1} - x \rangle + \frac{L}{2}\|x_{k+1} - y_k\|^2 \\
&= f(x) - L\langle x_{k+1} - y_k, x_{k+1} - x \rangle + \frac{L}{2}\|x_{k+1} - y_k\|^2 \\
&= f(x) + L\langle x_{k+1} - y_k, x - y_k \rangle - L\langle x_{k+1} - y_k, x_{k+1} - y_k \rangle + \frac{L}{2}\|x_{k+1} - y_k\|^2 \\
&= f(x) + L\langle x_{k+1} - y_k, x - y_k \rangle - \frac{L}{2}\|x_{k+1} - y_k\|^2.
\end{aligned}$$

Letting $x = x_k$, we have

$$f(x_{k+1}) \leq f(x_k) + L\langle x_{k+1} - y_k, x_k - y_k \rangle - \frac{L}{2}\|x_{k+1} - y_k\|^2.$$

Letting $x = x^*$, we have

$$f(x_{k+1}) \leq f(x^*) + L\langle x_{k+1} - y_k, x^* - y_k \rangle - \frac{L}{2}\|x_{k+1} - y_k\|^2.$$

Proof of the Convergence Rate

Multiplying the first inequality by $1-\theta_k$, multiplying the second by θ_k, adding them together, we have

$$f(x_{k+1}) \le f(x_k) + L\langle x_{k+1}-y_k, x_k-y_k\rangle - \frac{L}{2}\|x_{k+1}-y_k\|^2$$
$$f(x_{k+1}) \le f(x^*) + L\langle x_{k+1}-y_k, x^*-y_k\rangle - \frac{L}{2}\|x_{k+1}-y_k\|^2.$$

$$\begin{aligned}
& f(x_{k+1}) - (1-\theta_k)f(x_k) - \theta_k f(x^*) \\
\le & L\left\langle x_{k+1}-y_k, (1-\theta_k)x_k + \theta_k x^* - y_k\right\rangle - \frac{L}{2}\|x_{k+1}-y_k\|^2 \\
= & \frac{L}{2}\left(\|y_k-(1-\theta_k)x_k-\theta_k x^*\|^2 - \|x_{k+1}-(1-\theta_k)x_k-\theta_k x^*\|^2 + \|x_{k+1}-y_k\|^2\right) - \frac{L}{2}\|x_{k+1}-y_k\|^2 \\
= & \frac{L\theta_k^2}{2}\left\|\frac{y_k}{\theta_k} - \frac{1-\theta_k}{\theta_k}x_k - x^*\right\|^2 - \frac{L\theta_k^2}{2}\left\|\frac{x_{k+1}}{\theta_k} - \frac{1-\theta_k}{\theta_k}x_k - x^*\right\|^2.
\end{aligned}$$

$$\begin{aligned}
& \|x_{k+1}-y_k\|^2 \\
& + 2\left\langle x_{k+1}-y_k, y_k-(1-\theta_k)x_k-\theta_k x^*\right\rangle \\
& + \|y_k-(1-\theta_k)x_k-\theta_k x^*\|^2 \\
= & \|x_{k+1}-(1-\theta_k)x_k-\theta_k x^*\|^2
\end{aligned}$$

Denoting $z_{k+1} = \frac{x_{k+1}}{\theta_k} - \frac{1-\theta_k}{\theta_k}x_k$, from $y_k = x_k + \frac{\theta_k(1-\theta_{k-1})}{\theta_{k-1}}(x_k - x_{k-1})$, we have

$$\begin{aligned}
\frac{y_k}{\theta_k} - \frac{1-\theta_k}{\theta_k}x_k = & \frac{x_k}{\theta_k} + \frac{1-\theta_{k-1}}{\theta_{k-1}}(x_k - x_{k-1}) - \frac{1-\theta_k}{\theta_k}x_k \\
= & \left(\frac{1}{\theta_k} + \frac{1-\theta_{k-1}}{\theta_{k-1}} - \frac{1-\theta_k}{\theta_k}\right)x_k - \frac{1-\theta_{k-1}}{\theta_{k-1}}x_{k-1} \\
= & \frac{x_k}{\theta_{k-1}} - \frac{1-\theta_{k-1}}{\theta_{k-1}}x_{k-1} = z_k.
\end{aligned}$$

So we have

$$\begin{aligned}
& f(x_{k+1}) - f(x^*) - (1-\theta_k)\left(f(x_k) - f(x^*)\right) \\
= & f(x_{k+1}) - (1-\theta_k)f(x_k) - \theta_k f(x^*) \\
\le & \frac{L\theta_k^2}{2}\|z_k - x^*\|^2 - \frac{L\theta_k^2}{2}\|z_{k+1}-x^*\|^2.
\end{aligned}$$

Proof of the Convergence Rate

Dividing both sides by θ_k^2 and using $\frac{1-\theta_k}{\theta_k^2} = \frac{1}{\theta_{k-1}^2}$, we have

$$\begin{aligned}\frac{f(x_{k+1}) - f(x^*)}{\theta_k^2} - \frac{f(x_k) - f(x^*)}{\theta_{k-1}^2} &= \frac{f(x_{k+1}) - f(x^*)}{\theta_k^2} - \frac{(1-\theta_k)(f(x_k) - f(x^*))}{\theta_k^2} \\ &\leq \frac{L}{2}\|z_k - x^*\|^2 - \frac{L}{2}\|z_{k+1} - x^*\|^2 .\end{aligned}$$

Summing over $k = 0, 1, \cdots, K$ and using $\frac{1}{\theta_{-1}^2} = \frac{1-\theta_0}{\theta_0^2} = 0$ with $\theta_0 = 1$, we have

$$\frac{f(x_{K+1}) - f(x^*)}{\theta_K^2} = \frac{f(x_{K+1}) - f(x^*)}{\theta_K^2} - \frac{f(x_0) - f(x^*)}{\theta_{-1}^2} \leq \frac{L}{2}\|z_0 - x^*\|^2 .$$

On the other hand, from $\frac{1-\theta_k}{\theta_k^2} = \frac{1}{\theta_{k-1}^2}$ and $\theta_0 = 1$, we have

$$\begin{aligned}&\frac{1}{\theta_{k-1}^2} = \frac{1-\theta_k}{\theta_k^2} \leq \frac{1}{\theta_k^2} - \frac{1}{\theta_k} + \frac{1}{4} = \left(\frac{1}{\theta_k} - \frac{1}{2}\right)^2 \\ &\Rightarrow \frac{1}{\theta_{k-1}} \leq \frac{1}{\theta_k} - \frac{1}{2} \Rightarrow \frac{K}{2} + \frac{1}{\theta_0} \leq \frac{1}{\theta_K} \Rightarrow \theta_K \leq \frac{2}{K+2}\end{aligned}$$

So we have

$$f(x_{K+1}) - f(x^*) \leq \frac{L\theta_K^2}{2}\|z_0 - x^*\|^2 \leq \frac{2L}{(K+2)^2}\|z_0 - x^*\|^2.$$

Accelerated Gradient Descent

- For strongly convex problems,

$$y_k = x_k + \frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}}(x_k - x_{k-1})$$

$$x_{k+1} = y_k - \frac{1}{L}\nabla f(y_k)$$

- Recall the iterations for nonstrongly convex problems

$$y_k = x_k + \frac{\theta_k(1 - \theta_{k-1})}{\theta_{k-1}}(x_k - x_{k-1})$$

$$x_{k+1} = y_k - \frac{1}{L}\nabla f(y_k)$$

Convergence Rate of Accelerated Gradient Descent

Theorem: Suppose that the function $f(x)$ is strongly convex and smooth, then for AGD, we have

$$f(x_{k+1}) - f(x^*) \leq O\left(\left(1 - \sqrt{\frac{\mu}{L}}\right)^k\right)$$

- The convergence rate of accelerated gradient descent is $O\left(\left(1 - \sqrt{\frac{\mu}{L}}\right)^k\right)$
- Equivalently, the complexity of accelerated gradient descent is $O\left(\sqrt{\frac{L}{\mu}} \log \frac{1}{\epsilon}\right)$
- Recall that the convergence rate of gradient descent is $O\left(\left(1 - \frac{\mu}{L}\right)^k\right)$ (or $O\left(\frac{L}{\mu} \log \frac{1}{\epsilon}\right)$ complexity)

Summary of the Complexity Comparisons

Method	Strongly convex	Non-strongly convex
Gradient Descent	$O\left(\frac{L}{\mu}\log\frac{1}{\varepsilon}\right)$	$O\left(\frac{L}{\varepsilon}\right)$
heavy-ball	$O\left(\sqrt{\frac{L}{\mu}}\log\frac{1}{\varepsilon}\right)$	$O\left(\frac{L}{\varepsilon}\right)$
Accelerated Gradient Descent	$O\left(\sqrt{\frac{L}{\mu}}\log\frac{1}{\varepsilon}\right)$	$O\left(\sqrt{\frac{L}{\varepsilon}}\right)$

Another Accelerated Gradient Descent

- Algorithm iterations:

$$y_k = (1-\theta_k)x_k + \theta_k z_k$$

$$z_{k+1} = z_k - \frac{1}{L\theta_k}\nabla f(y_k)$$

$$x_{k+1} = (1-\theta_k)x_k + \theta_k z_{k+1}$$

- Equivalent to the previous one:

$$y_k = x_k + \frac{\theta_k(1-\theta_{k-1})}{\theta_{k-1}}(x_k - x_{k-1})$$

$$x_{k+1} = y_k - \frac{1}{L}\nabla f(y_k)$$

- Useful in the extension to composite optimization, stochastic optimization, and distributed optimization

$$\boxed{\begin{aligned}
x_{k+1} =& (1-\theta_k)x_k + \theta_k z_{k+1} \\
=& (1-\theta_k)x_k + \theta_k z_k - \frac{1}{L}\nabla f(y_k) \\
=& y_k - \frac{1}{L}\nabla f(y_k) \\
y_k =& (1-\theta_k)x_k + \theta_k z_k \\
=& (1-\theta_k)x_k + \theta_k \frac{x_k - (1-\theta_{k-1})x_{k-1}}{\theta_{k-1}} \\
=& x_k + \frac{\theta_k(1-\theta_{k-1})}{\theta_{k-1}}x_k - \frac{\theta_k(1-\theta_{k-1})}{\theta_{k-1}}x_{k-1}
\end{aligned}}$$

Another Accelerated Gradient Descent

- For strongly convex problems

$$
\begin{aligned}
y_k &= (1-\theta_k)x_k + \theta_k z_k \\
z_{k+1} &= \frac{1}{1+\frac{\mu\alpha}{\theta_k}}\left(z_k + \frac{\mu\alpha}{\theta_k} y_k - \frac{\alpha}{\theta_k}\nabla f(y_k)\right) \\
x_{k+1} &= (1-\theta_k)x_k + \theta_k z_{k+1}
\end{aligned}
$$

- Not equivalent to the previous one

$$
\begin{aligned}
y_k &= x_k + \frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}(x_k - x_{k-1}) \\
x_{k+1} &= y_k - \frac{1}{L}\nabla f(y_k)
\end{aligned}
$$

Lower Bounds

- Give the possible fastest convergence rate among all first-order algorithms
- No first-order algorithm can be faster than the lower bound
- An algorithm is optimal if its convergence rate equals to the lower bound

Theorem: There exists a special convex and smooth function $f(x)$ such that for any first-order algorithms satisfying

$$x_k \in \mathrm{Span}\left\{x_0, \nabla f(x_0), x_1, \nabla f(x_1), \cdots, x_{k-1}, \nabla f(x_{k-1})\right\}$$

we have

$$f(x_K) - f(x^*) \geq \frac{3L}{32(K+1)^2} \|x_0 - x^*\|^2$$

- Recall the upper bound of AGD:

$$f(x_{k+1}) - f(x^*) \leq O\left(\frac{1}{k^2}\right)$$

Lower Bounds

- Give the possible fastest convergence rate among all first-order algorithms
- No first-order algorithm can be faster than the lower bound
- An algorithm is optimal if its convergence rate equals to the lower bound

Theorem: There exists a special convex and smooth function $f(x)$ such that for any first-order algorithms satisfying

$$x_k \in \text{Span}\{x_0, \nabla f(x_0), x_1, \nabla f(x_1), \cdots, x_{k-1}, \nabla f(x_{k-1})\}$$

we have

$$f(x_K) - f(x^*) \geq \frac{3L}{32(K+1)^2}\|x_0 - x^*\|^2$$

If we can find an algorithm such that

$$f(x_k) - f(x^*) \leq \frac{C}{k^3}$$

for any convex and smooth function f. Then for the special f in the Theorem, we have

$$\frac{3L\|x_0 - x^*\|^2}{32(k+1)^2} \leq f(x_k) - f(x^*) \leq \frac{C}{k^3}$$

- Recall the upper bound of AGD:

$$f(x_{k+1}) - f(x^*) \leq O\left(\frac{1}{k^2}\right)$$

Lower Bounds

- Give the possible fastest convergence rate among all first-order algorithms
- No first-order algorithm can be faster than the lower bound
- An algorithm is optimal if its convergence rate equals to the lower bound

Theorem: There exists a special convex and smooth function $f(x)$ such that for any first-order algorithms satisfying

$$x_k \in \text{Span}\{x_0, \nabla f(x_0), x_1, \nabla f(x_1), \cdots, x_{k-1}, \nabla f(x_{k-1})\}$$

we have

$$f(x_K) - f(x^*) \geq \frac{3L}{32(K+1)^2}\|x_0 - x^*\|^2$$

- Recall the upper bound of AGD:

$$f(x_{k+1}) - f(x^*) \leq O\left(\frac{1}{k^2}\right)$$

$$\begin{aligned}
x_1 &= x_0 - \alpha \nabla f(x_0) \\
x_2 &= x_1 - \alpha \nabla f(x_1) + \beta(x_1 - x_0) \\
&\vdots \\
x_k &= x_{k-1} - \alpha \nabla f(x_{k-1}) + \beta(x_{k-1} - x_{k-2})
\end{aligned}$$

Lower Bounds

- For strongly convex problems:

Theorem: There exists a special strongly convex and smooth function $f(x)$ such that for any first-order algorithms satisfying

$$x_k \in \text{Span}\left\{x_0, \nabla f(x_0), x_1, \nabla f(x_1), \cdots, x_{k-1}, \nabla f(x_{k-1})\right\}$$

we have

$$\begin{aligned} f(x_K) - f(x^*) \geq & \frac{\mu}{2}\left(\frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}\right)^{2K}\|x_0 - x^*\|^2 \\ = & \frac{\mu}{2}\left(1 - \frac{2\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}\right)^{2K}\|x_0 - x^*\|^2 \geq \frac{\mu}{2}\left(1 - 2\sqrt{\frac{\mu}{L}}\right)^{2K}\|x_0 - x^*\|^2 \end{aligned}$$

- Recall the convergence rate of AGD:

$$f(x_{k+1}) - f(x^*) \leq O\left(\left(1 - \sqrt{\frac{\mu}{L}}\right)^k\right) \leq O\left(\left(1 - \frac{1}{2}\sqrt{\frac{\mu}{L}}\right)^{2k}\right)$$

Summary of the Complexity Comparisons

Method	Strongly convex	Non-strongly convex
Gradient Descent	$O\left(\frac{L}{\mu}\log\frac{1}{\varepsilon}\right)$	$O\left(\frac{L}{\varepsilon}\right)$
heavy-ball	$O\left(\sqrt{\frac{L}{\mu}}\log\frac{1}{\varepsilon}\right)$	$O\left(\frac{L}{\varepsilon}\right)$
Accelerated Gradient Descent	$O\left(\sqrt{\frac{L}{\mu}}\log\frac{1}{\varepsilon}\right)$	$O\left(\sqrt{\frac{L}{\varepsilon}}\right)$
Lower Bounds	$O\left(\sqrt{\frac{L}{\mu}}\log\frac{1}{\varepsilon}\right)$	$O\left(\sqrt{\frac{L}{\varepsilon}}\right)$

- The complexities of accelerated gradient descent match the lower bounds
- Accelerated gradient descent is the optimal first-order method

It cannot be improved!

Full Gradient: Does It Make Sense?

- The methods above, based on full gradients.
- Recall that in machine learning, the optimization problem is often

$$\min_{x \in \mathbb{R}^p} f(x) = \frac{1}{n} \sum_{j=1}^{n} \ell(\phi(a_j, x), y_j) + \lambda \Omega(x) = \frac{1}{n} \sum_{j=1}^{n} f_j(x)$$

- They are less appealing when n is large. To calculate

$$\nabla f(x) = \frac{1}{n} \sum_{j=1}^{n} \nabla f_j(x)$$

generally need to make a full pass through the data.

Stochastic Gradient Descent

- Fundamental idea:
  - Sample, in each iteration, one or several gradients as an estimator of the full gradient
- Stochastic gradient iterations:

$$\boxed{\begin{array}{l}\text{Choose } j_k \in \{1, 2, \cdots, n\} \text{ uniformally at random} \\ x_{k+1} = x_k - \alpha_k \nabla f_{j_k}(x_k)\end{array}}$$

  - Step size:

$$\alpha_k = \begin{cases} O(\frac{1}{k}) & \text{for strongly convex problems} \\ O(\frac{1}{k^{0.5+\varepsilon}}) & \text{for nonstrongly convex problems} \end{cases}$$

- Compare with gradient descent

$$x_{k+1} = x_k - \alpha \nabla f(x_k) = x_k - \frac{\alpha}{n} \sum_{j=1}^{n} \nabla f_j(x_k)$$

Stochastic Gradient Descent

- $\nabla f_{j_k}(x_k)$ is a approximation for $\nabla f(x_k)$
  - Unbiased:

$$\mathbf{E}_{j_k}[\nabla f_{j_k}(x_k)] = \frac{1}{n}\sum_{j=1}^{n} \nabla f_j(x_k) = \nabla f(x_k)$$

  - The variance will never go to zero even if $x_k \to x^*$

$$\mathbf{E}_{j_k}\left[\|\nabla f_{j_k}(x_k) - \nabla f(x_k)\|^2\right] \le \sigma^2$$

- Slow convergence rate due to the variance

$$\mathbf{E}[f(x_k)] - f(x^*) \le \begin{cases} O(\frac{1}{k}) & \text{for strongly convex problems} \\ O(\frac{1}{\sqrt{k}}) & \text{for nonstrongly convex problems} \end{cases}$$

Stochastic Gradient Descent

- Compare between gradient descent (GD) and stochastic gradient descent (SGD)

Method	Iteration complexity	Per-iteration cost	Total computation cost
GD	$O\left(\frac{L}{\mu}\log\frac{1}{\epsilon}\right)$	n	$O\left(\frac{nL}{\mu}\log\frac{1}{\epsilon}\right)$
SGD	$O\left(\frac{\sigma^2}{\mu\epsilon}\right)$	1	$O\left(\frac{\sigma^2}{\mu\epsilon}\right)$

  - SGD is more appealing for large n

- Can we expect faster convergence rate?

Yes, by variance reduction

Variance Reduction

- Fundamental idea:
  - Keep a snapvector w after every n SGD iterations, and use

$$\nabla_k = \nabla f_{j_k}(x_k) - \nabla f_{j_k}(w) + \frac{1}{n}\sum_{j=1}^{n} \nabla f_j(w)$$

as the descent direction:

$$x_{k+1} = x_k - \alpha \nabla_k$$

  - In each iteration, we only compute $\nabla f_{j_k}(x_k)$ and $\nabla f_{j_k}(w)$. $\frac{1}{n}\sum_{j=1}^{n} \nabla f_j(w)$ is computed after every n SGD iterations. So the cost in each iteration is the same with SGD

Variance Reduction

- ∇_k is an approximation of the full gradient
  - Unbiased:

$$\mathbf{E}_{j_k}[\nabla_k] = \frac{1}{n}\sum_{j_k=1}^{n}\left(\nabla f_{j_k}(x_k) - \nabla f_{j_k}(w) + \frac{1}{n}\sum_{j=1}^{n}\nabla f_j(w)\right) = \frac{1}{n}\sum_{j=1}^{n}\nabla f_j(x_k) = \nabla f(x_k)$$

  - The variance reduces to zero

$$\begin{aligned}
&\mathbf{E}_{j_k}\left[\|\nabla_k - \nabla f(x_k)\|^2\right] \\
=&\mathbf{E}_{j_k}\left\|\nabla f_{j_k}(x_k) - \nabla f_{j_k}(w) + \frac{1}{n}\sum_{j=1}^{n}\nabla f_j(w) - \frac{1}{n}\sum_{j=1}^{n}\nabla f_j(x_k)\right\|^2 \\
\leq&\mathbf{E}_{j_k}\|\nabla f_{j_k}(x_k) - \nabla f_{j_k}(w)\|^2 \leq \mathbf{E}_{j_k} L^2\|x_k - w\|^2 = L^2\|x_k - w\|^2
\end{aligned}$$

$\nabla_k \to \nabla f(x_k)$

when

$x_k \to w$

where we use the following inequality in the third step

$$\mathbf{E}[\|a - \mathbf{E}[a]\|^2] = \mathbf{E}[\|a\|^2 + \|\mathbf{E}[a]\|^2 - 2\langle a, \mathbf{E}[a]\rangle] = \mathbf{E}[\|a\|^2] - \|\mathbf{E}[a]\|^2 \leq \mathbf{E}[\|a\|^2]$$

Variance Reduction

$$\frac{1}{n}\sum_{j_k=1}^{n}\left(\frac{1}{n}\sum_{j=1}^{n}\nabla f_j(w)\right) = \frac{1}{n}\sum_{j=1}^{n}\nabla f_j(w)$$

- ∇_k is an approximation of the full gradient
  - Unbiased:

$$\mathbf{E}_{j_k}[\nabla_k] = \frac{1}{n}\sum_{j_k=1}^{n}\left(\nabla f_{j_k}(x_k) - \nabla f_{j_k}(w) + \frac{1}{n}\sum_{j=1}^{n}\nabla f_j(w)\right) = \frac{1}{n}\sum_{j=1}^{n}\nabla f_j(x_k) = \nabla f(x_k)$$

  - The variance reduces to zero

$$\begin{aligned}
&\mathbf{E}_{j_k}\left[\|\nabla_k - \nabla f(x_k)\|^2\right] \\
=&\mathbf{E}_{j_k}\left\|\nabla f_{j_k}(x_k) - \nabla f_{j_k}(w) + \frac{1}{n}\sum_{j=1}^{n}\nabla f_j(w) - \frac{1}{n}\sum_{j=1}^{n}\nabla f_j(x_k)\right\|^2 \\
\le&\mathbf{E}_{j_k}\|\nabla f_{j_k}(x_k) - \nabla f_{j_k}(w)\|^2 \le \mathbf{E}_{j_k} L^2\|x_k - w\|^2 = L^2\|x_k - w\|^2
\end{aligned}$$

$\nabla_k \to \nabla f(x_k)$

when

$x_k \to w$

where we use the following inequality in the third step

$$\mathbf{E}[\|a - \mathbf{E}[a]\|^2] = \mathbf{E}[\|a\|^2 + \|\mathbf{E}[a]\|^2 - 2\langle a, \mathbf{E}[a]\rangle] = \mathbf{E}[\|a\|^2] - \|\mathbf{E}[a]\|^2 \le \mathbf{E}[\|a\|^2]$$

Variance Reduction

- ∇_k is an approximation of the full gradient
  - Unbiased:

$$\mathbf{E}_{j_k}[\nabla_k] = \frac{1}{n}\sum_{j_k=1}^{n}\left(\nabla f_{j_k}(x_k) - \nabla f_{j_k}(w) + \frac{1}{n}\sum_{j=1}^{n}\nabla f_j(w)\right) = \frac{1}{n}\sum_{j=1}^{n}\nabla f_j(x_k) = \nabla f(x_k)$$

  - The variance reduces to zero

$$\begin{aligned}
&\mathbf{E}_{j_k}\left[\|\nabla_k - \nabla f(x_k)\|^2\right] \\
=&\mathbf{E}_{j_k}\left\|\nabla f_{j_k}(x_k) - \nabla f_{j_k}(w) + \frac{1}{n}\sum_{j=1}^{n}\nabla f_j(w) - \frac{1}{n}\sum_{j=1}^{n}\nabla f_j(x_k)\right\|^2 \\
\leq&\mathbf{E}_{j_k}\|\nabla f_{j_k}(x_k) - \nabla f_{j_k}(w)\|^2 \leq \mathbf{E}_{j_k} L^2\|x_k - w\|^2 = L^2\|x_k - w\|^2
\end{aligned}$$

$\mathbf{E}_{j_k}\left[\|\nabla_k - \nabla f(x_k)\|^2\right] \to 0$

when

$x_k \to w$

where we use the following inequality in the third step

$$\mathbf{E}[\|a - \mathbf{E}[a]\|^2] = \mathbf{E}[\|a\|^2 + \|\mathbf{E}[a]\|^2 - 2\langle a, \mathbf{E}[a]\rangle] = \mathbf{E}[\|a\|^2] - \|\mathbf{E}[a]\|^2 \leq \mathbf{E}[\|a\|^2]$$

Stochastic Variance Reduction Gradient

- SVRG iterations:

$$\begin{aligned}
&\text{Choose } j_k \in \{1, 2, \cdots, n\} \text{ uniformally at random} \\
&\nabla_k = \nabla f_{j_k}(x_k) - \nabla f_{j_k}(w_k) + \nabla f(w_k) \\
&x_{k+1} = x_k - \alpha \nabla_k \\
&w_{k+1} = \begin{cases} x_k & \text{with probability } \frac{1}{n} \\ w_k & \text{with probability } 1 - \frac{1}{n} \end{cases}
\end{aligned}$$

Theorem: Suppose that the each $f_j(x)$ is convex and smooth, $f(x)$ is strongly convex, then for SVRG, we need

$$O\left(\left(n + \frac{L}{\mu}\right) \log \frac{1}{\epsilon}\right)$$

Iterations such that $\mathbf{E}[\|x_k - x^*\|^2] \leq \epsilon$

Stochastic Variance Reduction Gradient

- Complexity comparisons:

Method	Iteration complexity	Per-iteration cost	Total computation cost
GD	$O\left(\frac{L}{\mu}\log\frac{1}{\epsilon}\right)$	n	$O\left(\frac{nL}{\mu}\log\frac{1}{\epsilon}\right)$
SGD	$O\left(\frac{\sigma^2}{\mu\epsilon}\right)$	1	$O\left(\frac{\sigma^2}{\mu\epsilon}\right)$
SVRG	$O\left(\left(n+\frac{L}{\mu}\right)\log\frac{1}{\epsilon}\right)$	1	$O\left(\left(n+\frac{L}{\mu}\right)\log\frac{1}{\epsilon}\right)$

- SVRG Combines the advantages of GD and SGD
  - The same convergence rate with GD when $n \leq \dfrac{L}{\mu}$
  - The same cost per iteration with SGD
  - Lower total cost than both GD and SGD
- Other VR methods
  - Stochastic Average Gradient (SAG), Stochastic Dual Coordinate Ascent (SDCA), SAGA

Accelerated Stochastic Variance Reduction Gradient

- Combines SVRG with accelerated gradient descent

$$
\begin{aligned}
&\text{Choose } j_k \in \{1,2,\cdots,n\} \text{ uniformally at random}\\
&\nabla_k = \nabla f_{j_k}(x_k) - \nabla f_{j_k}(w_k) + \nabla f(w_k)\\
&x_{k+1} = x_k - \alpha \nabla_k\\
&w_{k+1} = \begin{cases} x_k & \text{with probability } \frac{1}{n} \\ w_k & \text{with probability } 1-\frac{1}{n} \end{cases}
\end{aligned}
$$

$+$

$$
\begin{aligned}
&y_k = (1-\theta_k)x_k + \theta_k z_k\\
&z_{k+1} = \frac{1}{1+\frac{\mu\alpha}{\theta_k}}\left(z_k + \frac{\mu\alpha}{\theta_k} y_k - \frac{\alpha}{\theta_k}\nabla f(y_k)\right)\\
&x_{k+1} = (1-\theta_k)x_k + \theta_k z_{k+1}
\end{aligned}
$$

$\Downarrow$

$$
\boxed{
\begin{aligned}
&y_k = \theta_1 z_k + \theta_2 w_k + (1-\theta_1-\theta_2)x_k\\
&\text{Choose } j_k \in \{1,2,\cdots,n\} \text{ uniformally at random}\\
&\nabla_k = \nabla f_{j_k}(y_k) - \nabla f_{j_k}(w_k) + \nabla f(w_k)\\
&z_{k+1} = \frac{1}{1+\frac{\alpha\mu}{\theta_1}}\left(\frac{\alpha\mu}{\theta_1} y_k + z_k - \frac{\alpha}{\theta_1}\nabla_k\right)\\
&x_{k+1} = \theta_1 z_{k+1} + \theta_2 w_k + (1-\theta_1-\theta_2)x_k\\
&w_{k+1} = \begin{cases} x_k & \text{with probability } \frac{1}{n} \\ w_k & \text{with probability } 1-\frac{1}{n} \end{cases}
\end{aligned}
}
$$

Accelerated Stochastic Variance Reduction Gradient

Theorem: Suppose that the each $f_j(x)$ is convex and smooth, $f(x)$ is strongly convex, then for accelerated SVRG, we need

$$O\left(\left(n+\sqrt{\frac{nL}{\mu}}\right)\log\frac{1}{\epsilon}\right)$$

Iterations such that $\mathbf{E}[\|x_k - x^*\|^2] \le \epsilon$

- Recall that the complexity of SVRG is $O\left(\left(n+\frac{L}{\mu}\right)\log\frac{1}{\epsilon}\right)$

$$2\sqrt{\frac{nL}{\mu}} \le n + \frac{L}{\mu}$$

- We always have $\left(n+\sqrt{\frac{nL}{\mu}}\right)\log\frac{1}{\epsilon} \le \left(n+\frac{L}{\mu}\right)\log\frac{1}{\epsilon}$, so the accelerated SVRG is always not worse than SVRG. The strict inequality holds when $n < \frac{L}{\mu}$
- When $n \ge \frac{L}{\mu}$, we have $\left(n+\sqrt{\frac{nL}{\mu}}\right)\log\frac{1}{\epsilon} = n\log\frac{1}{\epsilon} = \left(n+\frac{L}{\mu}\right)\log\frac{1}{\epsilon}$, acceleration takes no effect

Accelerated Stochastic Variance Reduction Gradient

- Complexity comparisons:

Method	Iteration complexity	Per-iteration cost	Total computation cost
GD	$O\left(\frac{L}{\mu}\log\frac{1}{\epsilon}\right)$	n	$O\left(\frac{nL}{\mu}\log\frac{1}{\epsilon}\right)$
SGD	$O\left(\frac{\sigma^2}{\mu\epsilon}\right)$	1	$O\left(\frac{\sigma^2}{\mu\epsilon}\right)$
SVRG	$O\left(\left(n+\frac{L}{\mu}\right)\log\frac{1}{\epsilon}\right)$	1	$O\left(\left(n+\frac{L}{\mu}\right)\log\frac{1}{\epsilon}\right)$
Acc-SVRG	$O\left(\left(n+\sqrt{\frac{nL}{\mu}}\right)\log\frac{1}{\epsilon}\right)$	1	$O\left(\left(n+\sqrt{\frac{nL}{\mu}}\right)\log\frac{1}{\epsilon}\right)$
Lower bound	\	\	$O\left(\left(n+\sqrt{\frac{nL}{\mu}}\right)\log\frac{1}{\epsilon}\right)$

- The iteration complexity of accelerated SVRG matches the lower bound. So it is optimal
- Acceleration has no help to improve SGD

Accelerated Stochastic Variance Reduction Gradient

- Other accelerated algorithms for stochastic optimization
  - Accelerated Stochastic Coordinate Descent
    - Accelerated Stochastic Dual Coordinate Ascent
  - Accelerated Stochastic Primal–Dual Method
  - A Universal Catalyst Acceleration Framework

Distributed Optimization

- Distributed optimization has broad applications in machine learning
  - Large scale training data distributed among a group of servers
  - Data are generated and stored by the mobile users
- Typical setup
  - Consider problem $$\min_{x \in \mathbb{R}} f(x) = \frac{1}{m} \sum_{i=1}^{m} f_i(x)$$
  - The local function $f_i(x)$ represents the data on node i. It is only available to node i.
  - The nodes are connected by a network

Distributed Optimization

➢ Communication network

- Directed or undirected. We only consider undirected network here
- The network is described by a mixing matrix $W \in \mathbb{R}^{m \times m}$ to characterize the connectivity and the weight of the communication edges

  1. $W_{i,j} > 0$ *if and only if nodes* i *and* j *are connected or* $i = j$*. Otherwise,* $W_{i,j} = 0$.

  2. $W\mathbf{1} = \mathbf{1}$ *and* $\mathbf{1}^T W = \mathbf{1}^T$.

- One example

$$W_{ij} = \begin{cases} \frac{1}{1+\max\{\text{degree}(i),\text{degree}(j)\}} & \text{if } i \text{ and } j \text{ are connected and } i \neq j \\ 0 & \text{if } i \text{ and } j \text{ are not connected} \\ 1 - \sum_r W_{ir} & \text{if } i = j \end{cases}$$

- The largest singular value of W: $\sigma_1 = 1$; The second largest singular value: $\sigma_2 < 1$

Distributed Gradient Descent

- Each node keeps an auxiliary variable $x(i)$ and updates it by local computations on $\nabla f_i(x(i))$ and local communications with its neighbors

$$x(i)_{k+1} = \sum_{j \in \mathcal{N}_i} W_{ij} x(j)_k - \alpha_k \nabla f_i(x(i)_k)$$

- Compact form

$$\boxed{\mathbf{x}_{k+1} = W\mathbf{x}_k - \alpha_k \nabla f(\mathbf{x}_k)}$$

by letting

$$\mathbf{x} = \begin{bmatrix} x(1) \\ \vdots \\ x(i) \\ \vdots \\ x(m) \end{bmatrix}, \qquad \nabla f(\mathbf{x}) = \begin{bmatrix} \nabla f_1(x(1)) \\ \vdots \\ \nabla f_i(x(i)) \\ \vdots \\ \nabla f_m(x(m)) \end{bmatrix}$$

Slow Convergence of Distributed Gradient Descent

- Assume $x(i)_k \to x^*$, then we have

$$x(i)_{k+1} = \sum_{j \in \mathcal{N}_i} W_{ij} x(j)_k - \alpha_k \nabla f_i(x(i)_k)$$

$$\Rightarrow x^* = x^* - \alpha_k \nabla f_i(x^*)$$

At the minimum, we have $\nabla f(x^*) = \sum_{i=1}^{m} \nabla f_i(x^*) = 0$. However, we often have $\nabla f_i(x^*) \neq 0$

So we should let $\alpha_k \to 0$

- Slow $O(\frac{1}{k})$ convergence rate due to the diminishing stepsize, even for smooth and strongly convex problems. The same with SGD
- Can we expect faster convergence rate? Yes, by gradient tracking

Gradient Tracking

- Each node keeps an auxiliary variable $s(i)_k$ as the descent direction

$$\sum_{i=1}^{m} s(i)_k - \sum_{i=1}^{m} \nabla f_i(x(i)_k) = \sum_{i=1}^{m} \sum_{j\in\mathcal{N}_i} W_{ij} s(j)_{k-1} - \sum_{i=1}^{m} \nabla f_i(x(i)_{k-1})$$
$$= \sum_{i=1}^{m} \sum_{j=1}^{m} W_{ij} s(j)_{k-1} - \sum_{i=1}^{m} \nabla f_i(x(i)_{k-1})$$
$$= \sum_{j=1}^{m} s(j)_{k-1} \sum_{i=1}^{m} W_{ij} - \sum_{i=1}^{m} \nabla f_i(x(i)_{k-1})$$
$$= \sum_{j=1}^{m} s(j)_{k-1} - \sum_{i=1}^{m} \nabla f_i(x(i)_{k-1})$$
$$\sum_{i=1}^{m} s(i)_0 = \sum_{i=1}^{m} \nabla f_i(x(i)_0) \Rightarrow \sum_{i=1}^{m} s(i)_k = \sum_{i=1}^{m} \nabla f_i(x(i)_k)$$

$$s(i)_k = \sum_{j\in\mathcal{N}_i} W_{ij} s(j)_{k-1} + \nabla f_i(x(i)_k) - \nabla f_i(x(i)_{k-1})$$
$$x(i)_{k+1} = \sum_{j\in\mathcal{N}_i} W_{ij} x(j)_k - \alpha s(i)_k$$

The first step gives $\sum_{j=1}^{m} s(j)_k = \sum_{j=1}^{m} \nabla f_j(x(j)_k)$, so if we have $s(i)_k - s(j)_k \to 0$ and $x(i)_k - x(j)_k \to 0$, then we have $s(i)_k \to \frac{1}{m}\sum_{j=1}^{m} \nabla f_j(x(i)_k)$, so the second step approximates gradient descent

- Compact form

$$\boxed{\begin{aligned} \mathbf{s}_k &= W\mathbf{s}_{k-1} + \nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}_{k-1}) \\ \mathbf{x}_{k+1} &= W\mathbf{x}_k - \alpha \mathbf{s}_k \end{aligned}}$$

Gradient Tracking

Theorem: Suppose that each $f_j(x)$ is convex and smooth, then for GT we need

$$O\left(\frac{L}{\epsilon(1-\sigma_2)^2}\right)$$

Iterations to find x such that $f(x) - f(x^*) \leq \epsilon$

Theorem: Suppose that each $f_j(x)$ is strongly convex and smooth, then for GT we need

$$O\left(\left(\frac{L}{\mu} + \frac{1}{(1-\sigma_2)^2}\right)\log\frac{1}{\epsilon}\right)$$

Iterations to find x such that $f(x) - f(x^*) \leq \epsilon$

Accelerated Gradient Tracking

- Combines gradient tracking with accelerated gradient descent

$$\mathbf{s}_k = W\mathbf{s}_{k-1} + \nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}_{k-1})$$
$$\mathbf{x}_{k+1} = W\mathbf{x}_k - \alpha\mathbf{s}_k$$

$+$

$$y_k = (1-\theta_k)x_k + \theta_k z_k$$
$$z_{k+1} = \frac{1}{1+\frac{\mu\alpha}{\theta_k}}\left(z_k + \frac{\mu\alpha}{\theta_k}y_k - \frac{\alpha}{\theta_k}\nabla f(y_k)\right)$$
$$x_{k+1} = (1-\theta_k)x_k + \theta_k z_{k+1}$$

$\Downarrow$

$$\boxed{\begin{aligned}
\mathbf{y}_k &= \theta_k\mathbf{z}_k + (1-\theta_k)\mathbf{x}_k \\
\mathbf{s}_k &= W\mathbf{s}_{k-1} + \nabla f(\mathbf{y}_k) - \nabla f(\mathbf{y}_{k-1}) \\
\mathbf{z}_{k+1} &= \frac{1}{1+\frac{\mu\alpha}{\theta_k}}\left(W\left(\frac{\mu\alpha}{\theta_k}\mathbf{y}_k + \mathbf{z}_k\right) - \frac{\alpha}{\theta_k}\mathbf{s}_k\right) \\
\mathbf{x}_{k+1} &= \theta_k\mathbf{z}_{k+1} + (1-\theta_k)W\mathbf{x}_k
\end{aligned}}$$

Accelerated Gradient Tracking

Theorem: Suppose that each $f_j(x)$ is convex and smooth, then for Acc-GT we need

$$O\left(\frac{1}{(1-\sigma_2)^2}\sqrt{\frac{L}{\epsilon}}\right)$$

Iterations to find x such that $f(x) - f(x^*) \leq \epsilon$

Theorem: Suppose that each $f_j(x)$ is strongly convex and smooth, then for Acc-GT we need

$$O\left(\frac{1}{(1-\sigma_2)^{1.5}}\sqrt{\frac{L}{\mu}}\log\frac{1}{\epsilon}\right)$$

Iterations to find x such that $f(x) - f(x^*) \leq \epsilon$

Accelerated Gradient Tracking

- Complexity comparisons:

Method	Strongly convex	Non-strongly convex
Gradient Tracking	$O\left(\left(\frac{L}{\mu}+\frac{1}{(1-\sigma_2)^2}\right)\log\frac{1}{\epsilon}\right)$	$O\left(\frac{L}{\epsilon(1-\sigma_2)^2}\right)$
Accelerated Gradient Tracking	$O\left(\frac{1}{(1-\sigma_2)^{1.5}}\sqrt{\frac{L}{\mu}}\log\frac{1}{\epsilon}\right)$	$O\left(\frac{1}{(1-\sigma_2)^2}\sqrt{\frac{L}{\epsilon}}\right)$
Accelerated Gradient Tracking+ Chebyshev acceleration	$O\left(\sqrt{\frac{L}{\mu(1-\sigma_2)}}\log\frac{1}{\epsilon}\right)$	$O\left(\sqrt{\frac{L}{\epsilon(1-\sigma_2)}}\right)$
Communication Lower Bounds	$O\left(\sqrt{\frac{L}{\mu(1-\sigma_2)}}\log\frac{1}{\epsilon}\right)$	$O\left(\sqrt{\frac{L}{\epsilon(1-\sigma_2)}}\right)$

- The iteration complexity of accelerated gradient tracking combined with Chebyshev acceleration matches the lower bound. So it is optimal

Accelerated Gradient Tracking

- Other accelerated algorithms for distributed optimization

  - Accelerated Dual Ascent

  - Accelerated Primal-Dual Method

Conclusions and Take Home Messages

- Accelerated gradient descent is the theoretical fastest first-order algorithm for unconstrained convex optimization
- Accelerated gradient descent has been successfully extended to stochastic optimization and distributed optimization
- Accelerated algorithms always perform much faster than non-accelerated algorithms in practice. Just use it.

Reference:

- Huan Li, Cong Fang, and Zhouchen Lin, *Accelerated First-Order Optimization Algorithms for Machine Learning*. Proceedings of the IEEE, 108(11):2067-2082, 2020.
- Zhouchen Lin, Huan Li, and Cong Fang, *Accelerated Optimization in Machine Learning: First-Order Algorithms*. Springer 2020.

Thanks for your attention!