Statistical Models & Computing Methods

Lecture 2: Gradient Methods

Cheng Zhang

School of Mathematical Sciences, Peking University

October 15, 2020

- We now focus on numerical solutions for unconstrained optimization problems

$$\text{minimize} \quad f(x)$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is twice differentiable

- Descent method. We can set up a sequence

$$x^{(k+1)} = x^{(k)} + t^{(k)} \Delta x^{(k)}, \quad t^{(k)} > 0$$

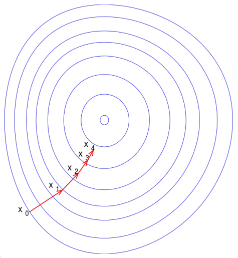
such that $f(x^{(k+1)}) < f(x^{(k)}), \quad k = 0, 1, \dots,$

- $\Delta x^{(k)}$ is called the search direction; $t^{(k)}$ is called the step size or learning rate in machine learning.

A reasonable choice for the search direction is the negative gradient, which leads to gradient descent methods

$$x^{(k+1)} = x^{(k)} - t^{(k)}\nabla f(x^{(k)}), \quad k = 0, 1, \dots$$

- step size $t^{(k)}$ can be constant or determined by line search
- every iteration is cheap, does not require second derivatives



- First-order Taylor expansion

$$f(x+v) \approx f(x) + \nabla f(x)^T v$$

- v is a descent direction iff $\nabla f(x)^T v < 0$
- Negative gradient is the steepest descent direction with respect to the Euclidean norm.

$$\frac{-\nabla f(x)}{\|\nabla f(x)\|_2} = \arg\min_v \{\nabla f(x)^T v \mid \|v\|_2 = 1\}$$

- Consider the second-order Taylor expansion of f at x,

$$\begin{aligned} f(x+v) &\approx f(x) + \nabla f(x)^T v + \frac{1}{2} v^T \nabla^2 f(x) v \\ &\triangleq \tilde{f}(x) \end{aligned}$$

- We find the optimal direction v by minimizing $\tilde{f}(x)$ with respect to v

$$v = -[\nabla^2 f(x)]^{-1} \nabla f(x)$$

- If $\nabla^2 f(x) \succeq 0$ (e.g., convex functions)

$$\nabla f(x)^T v = -\nabla f(x)^T [\nabla^2 f(x)]^{-1} \nabla f(x) < 0$$

when $\nabla f(x) \neq 0$

- The search direction in Newton's method can also be viewed as a steepest descent direction, but with a different metric
- In general, given a positive definite matrix P, we can define a quadratic norm

$$\|v\|_P = (v^T P v)^{1/2}$$

- Similarly, we can show that $-P^{-1}\nabla f(x)$ is the steepest descent direction w.r.t. the quadratic norm $\|\cdot\|_P$

$$\text{minimize } \nabla f(x)^T v, \quad \text{subject to } \|v\|_P = 1$$

- When P is the Hessian $\nabla^2 f(x)$, we get Newton's method

- Computing the Hessian and its inverse could be expensive, we can approximate it with another positive definite matrix $M \succ 0$ which is easier to use
- Update $M^{(k)}$ to learn about the curvature of f in the search direction and maintain a secant condition

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

- Rank-one update

$$\begin{aligned}
\Delta x^{(k)} &= x^{(k+1)} - x^{(k)} \\
y^{(k)} &= \nabla f(x^{(k+1)}) - \nabla f(x^{(k)}) \\
v^{(k)} &= y^{(k)} - M^{(k)}\Delta x^{(k)} \\
M^{(k+1)} &= M^{(k)} + \frac{v^{(k)}(v^{(k)})^T}{(v^{(k)})^T \Delta x^{(k)}}
\end{aligned}$$

- Easy to compute the inverse of matrices for low rank updates by **Sherman-Morrison-Woodbury formula**

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}$$

  where $A \in \mathbb{R}^{n\times n}, U \in \mathbb{R}^{n\times d}, C \in \mathbb{R}^{d\times d}, V \in \mathbb{R}^{d\times n}$

- Another popular rank-two update method: the BFGS (Broyden-Fletcher-Goldfarb-Shanno) method

$$M^{(k+1)} = M^{(k)} + \frac{y^{(k)}(y^{(k)})^T}{(y^{(k)})^T\Delta x^{(k)}} - \frac{M^{(k)}\Delta x^{(k)}(M^{(k)}\Delta x^{(k)})^T}{(\Delta x^{(k)})^T M^{(k)}\Delta x^{(k)}}$$

- In the frequentist framework, we typically perform statistical inference by maximizing the log-likelihood $L(\theta)$, or equivalently minimizing negative log-likelihood, which is also known as the energy function
- Some notations we introduced before
  - Score function: $s(\theta) = \nabla_\theta L(\theta)$
  - Observed Fisher information: $J(\theta) = -\nabla_\theta^2 L(\theta)$
  - Fisher information: $\mathcal{I}(\theta) = \mathbb{E}(-\nabla_\theta^2 L(\theta))$
- Newton's method for MLE:

$$\theta^{(k+1)} = \theta^{(k)} + (J(\theta^{(k)}))^{-1} s(\theta^{(k)})$$

Fisher Scoring Algorithm

- If we use the Fisher information instead of the observed information, the resulting method is called the *Fisher scoring* algorithm

$$\theta^{(k+1)} = \theta^{(k)} + (\mathcal{I}(\theta^{(k)}))^{-1} s(\theta^{(k)})$$

- It seems that the Fisher scoring algorithm is less sensitive to the initial guess. On the other hand, the Newton's method tends to converge faster
- For exponential family models with natural parameters and generalized linear models (GLMs) with canonical links, the two methods are identical

- A generalized linear model (GLM) assumes a set of independent random variables $Y_1, \ldots, Y_n$ that follow exponential family distributions of the same form

$$p(y_i|\theta_i) = \exp\left(y_i b(\theta_i) + c(\theta_i) + d(y_i)\right)$$

- The parameters θ_i are typically not of direct interest. Instead, we usually assume that the expectation of Y_i can be related to a vector of parameters β via a transformation (link function)

$$E(Y_i) = \mu_i, \quad g(\mu_i) = x_i^T \beta$$

where x_i is the observed covariates for y_i.

- Using the link function, we can now write the score function in terms of β
- Let $g(\mu_i) = \eta_i$, we can show that for jth parameter

$$s(\beta_j) = \sum_{i=1}^{n} \frac{(y_i - \mu_i)x_{ij}}{\mathbb{V}\mathrm{ar}(Y_i)} \frac{\partial \mu_i}{\partial \eta_i}$$

  where $\partial \mu_i / \partial \eta_i$ depends on the link function we choose
- It is also easy to show that the Fisher information matrix is

$$\begin{aligned}\mathcal{I}(\beta_j, \beta_k) &= \mathbb{E}(s(\beta_j)s(\beta_k)) \\ &= \sum_{i=1}^{n} \frac{x_{ij}x_{ik}}{\mathbb{V}\mathrm{ar}(Y_i)} \left(\frac{\partial \mu_i}{\partial \eta_i}\right)^2\end{aligned}$$

- Note that the Fisher information matrix can be written as

$$\mathcal{I}(\beta) = X^T W X$$

where W is the $n \times n$ diagonal matrix with elements

$$w_{ii} = \frac{1}{\mathbb{V}\mathrm{ar}(Y_i)} \left(\frac{\partial \mu_i}{\partial \eta_i} \right)^2$$

- Rewriting Fisher scoring algorithm for updating β as

$$\mathcal{I}(\beta^{(k)})\beta^{(k+1)} = \mathcal{I}(\beta^{(k)})\beta^{(k)} + s(\beta^{(k)})$$

- After few simple steps, we have

$$X^T W^{(k)} X \beta^{(k+1)} = X^T W^{(k)} Z^{(k)}$$

where

$$z_i^{(k)} = \eta_i^{(k)} + (y_i - \mu_i^{(k)}) \frac{\partial \eta_i^{(k)}}{\partial \mu_i^{(k)}}$$

- Therefore, we can find the next estimate as follows

$$\beta^{(k+1)} = (X^T W^{(k)} X)^{-1} X^T W^{(k)} Z^{(k)}$$

- The above estimate is similar to the weighted least square estimate, except that the weights W and the response variable Z change from one iteration to another
- We iteratively estimate β until the algorithm converges

- Recall that the Log-likelihood for logistic regression is

$$L(Y|p) = \sum_{i=1}^{n} y_i \log \frac{p_i}{1-p_i} + \log(1-p_i)$$

- The natural parameters are $\theta_i = \log \frac{p_i}{1-p_i}$. We use $g(x) = \log \frac{x}{1-x}$ as the link function, $\theta_i = g(p_i) = x_i^T \beta$
- We now write the log-likelihood as follows

$$L(\beta) = Y^T X\beta - \sum_{i=1}^{n} \log(1 + \exp(x_i^T \beta))$$

- The score function is

$$s(\beta) = X^T(Y-p), \quad p = \frac{1}{1+\exp(-X\beta)}$$

Example: Logistic Regression

- The observed Fisher information matrix is

$$J(\beta) = X^T W X$$

where W is a diagonal matrix with elements

$$w_{ii} = p_i(1 - p_i)$$

- Note that $J(\beta)$ does not depend on Y, meaning that it is also the Fisher information matrix $\mathcal{I}(\beta) = J(\beta)$
- Newton's update

$$\beta^{(k+1)} = \beta^{(k)} + \left(X^T W^{(k)} X\right)^{-1} \left(X^T (Y - p^{(k)})\right)$$

- While gradient descent is simple and intuitive, it has many problems as well.
  - Saddle-point problem
  - Not applicable to non-differential objectives
  - Could be slow
  - How to scale to big data problems
- In what follows, we will discuss some advanced techniques that can alleviate these problems

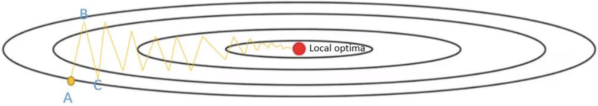
- Introduced in 1964 by Polyak, momentum method is a technique that can accelerate gradient descent by taking accounts of previous gradients in the update rule at each iteration.

$$
\begin{aligned}
m^{(k)} &= \mu m^{(k-1)} + (1-\mu)\nabla f(x^{(k)}) \\
x^{(k+1)} &= x^{(k)} - \alpha m^{(k)}
\end{aligned}
$$

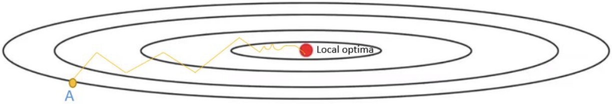
  where $0 \leq \mu < 1$
- When $\mu = 0$, gradient descent is recovered.

How Does Momentum Work?

- The vanilla gradient descent may suffer from oscillations when the magnitudes of gradient varies a lot across different directions.



- Using the exponential weighted gradient (momentum), those oscillations are more likely to be damped out, resulting in faster rate of convergence.



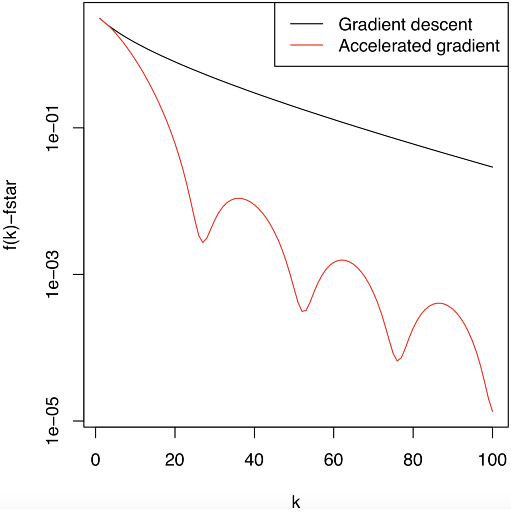
- Choose any initial $x^{(0)} = x^{(-1)}$, $\forall\, k = 1, 2, 3, \ldots$

$$y = x^{(k-1)} + \frac{k-2}{k+1}(x^{(k-1)} - x^{(k-2)})$$
$$x^{(k)} = y - t_k \nabla f(y)$$

- The first two steps are the usually gradient updates
- After that, $y = x^{(k-1)} + \frac{k-2}{k+1}(x^{(k-1)} - x^{(k-2)})$ carries some "momentum" from previous iterations, and $x^{(k)} = y - t_k \nabla f(y)$ uses *lookahead* gradient at y.

Example

Logistic regression



Assumptions

- f is convex and continuously differentiable on $\mathbb{R}^n$
- $\nabla f(x)$ is L-Lipschitz continuous w.r.t Euclidean norm: for any $x, y \in \mathbb{R}^n$

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$$

- optimal value $f^* = \inf_x f(x)$ is finite and attained at x^*.

Theorem: Gradient descent with $0 < t \leq 1/L$ satisfies

$$f(x^{(k)}) - f^* \leq \frac{1}{2kt}\|x^{(0)} - x^*\|^2$$

Some Useful Lemma and Strong Convexity

- If f is L-Lipschitz, then for any $x, y \in \mathbb{R}^n$

$$f(y) \leq f(x) + \nabla f(x)^T(y-x) + \frac{L}{2}\|y-x\|^2$$

- If f is differentiable and m-strongly convex, then

$$f(y) \geq f(x) + \nabla f(x)^T(y-x) + \frac{m}{2}\|y-x\|^2$$

  If $m = 0$, we cover the standard(weak) convexity
- In other words, f is *sandwiched* between two quadratic functions

Proof

- If $x^+ = x - t\nabla f(x)$ and $0 < t \leq 1/L$

$$\begin{aligned} f(x^+) &\leq f(x) - t\|\nabla f(x)\|^2 + \frac{t^2 L}{2}\|\nabla f(x)\|^2 \\ &\leq f(x) - \frac{t}{2}\|\nabla f(x)\|^2 \end{aligned}$$

- From convexity

$$f(x) \leq f^* + \nabla f(x)^T(x - x^*) - \frac{m}{2}\|x - x^*\|^2$$

- Add the above two inequalities

$$f(x^+) - f^* \leq \nabla f(x)^T(x - x^*) - \frac{t}{2}\|\nabla f(x)\|^2 - \frac{m}{2}\|x - x^*\|^2$$

Proof

- Continue ...

$$
\begin{aligned}
&\le \frac{1}{2t}(\|x - x^*\|^2 - \|x^+ - x^*\|^2) - \frac{m}{2}\|x - x^*\|^2 \\
&= \frac{1}{2t}\left((1 - mt)\|x - x^*\|^2 - \|x^+ - x^*\|^2\right) \qquad (1) \\
&\le \frac{1}{2t}(\|x - x^*\|^2 - \|x^+ - x^*\|^2) \qquad (2)
\end{aligned}
$$

- For gradient descent updates

$$
\begin{aligned}
\sum_{i=1}^{k}(f(x^{(i)}) - f^*) &\le \frac{1}{2t}\sum_{i=1}^{k}(\|x^{(i-1)} - x^*\|^2 - \|x^{(i)} - x^*\|^2) \\
&= \frac{1}{2t}(\|x^{(0)} - x^*\|^2 - \|x^{(k)} - x^*\|^2)
\end{aligned}
$$

Proof

- Since $f(x^{(i)})$ is non-increasing

$$f(x^{(k)}) - f^* \le \frac{1}{2kt}\|x^{(0)} - x^*\|^2$$

- If f is m-strongly convex, and $m > 0$, from (1)

$$\|x^{(i)} - x^*\|^2 \le (1 - mt)\|x^{(i-1)} - x^*\|^2, \quad \forall i = 1, 2, \ldots$$

- Therefore

$$\|x^{(k)} - x^*\|^2 \le (1 - mt)^k\|x^{(0)} - x^*\|^2$$

$i.e.$, linear convergence if f is strongly convex ($m > 0$)

- First order method: any iterative algorithm that selects $x^{(k+1)}$ in the set

$$x^{(0)} + \operatorname{span}\{\nabla f(x^{(0)}), \nabla f(x^{(1)}), \ldots, \nabla f(x^{(k)})\}$$

- **Theorem** (Nesterov): for every integer $k \leq (n-1)/2$ and every $x^{(0)}$, there exist functions that satisfy the assumptions such that for any first-order method

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

- Therefore, $1/k^2$ is the best convergence rate for all first-order methods.

- Accelerated gradient descent with fixed step size $t \leq 1/L$ satisfies
$$f(x^{(k)}) - f^* \leq \frac{2\|x^{(0)} - x^*\|^2}{t(k+1)^2}$$
- Nesterov's accelerated gradient (**NAG**) descent achieve the oracle convergence rate of first-order methods!

Reformulation of NAG

- Initialize $x^{(0)} = u^{(0)}$, and for $k = 1, 2, \dots$

$$
\begin{aligned}
y &= (1-\theta_k)x^{(k-1)} + \theta_k u^{(k-1)} \\
x^{(k)} &= y - t_k \nabla f(y) \\
u^{(k)} &= x^{(k-1)} + \frac{1}{\theta_k}(x^{(k)} - x^{(k-1)})
\end{aligned}
$$

  with $\theta_k = 2/(k+1)$.

- This is equivalent to the formulation of NAG presented earlier (slide 5), and makes convergence analysis easier

Proof

- If $y = (1-\theta)x + \theta u$, $x^+ = y - t\nabla f(y)$, and $0 < t \leq 1/L$

$$f(x^+) \leq f(y) + \nabla f(y)^T(x^+ - y) + \frac{1}{2t}\|x^+ - y\|^2$$

- From convexity, $\forall z \in \mathbb{R}^n$

$$f(y) \leq f(z) + \nabla f(y)^T(y - z)$$

- Add these together

$$f(x^+) \leq f(z) + \frac{1}{t}(x^+ - y)(z - x^+) + \frac{1}{2t}\|x^+ - y\|^2 \qquad (3)$$

Proof

- Let $u^+ = x + \frac{1}{\theta}(x^+ - x)$, using bound (3) at $z = x$ and $z = x^*$

$$
\begin{aligned}
f(x^+) &- f^* - (1-\theta)(f(x) - f^*) \\
&\leq \frac{1}{t}(x^+ - y)^T(\theta x^* + (1-\theta)x - x^+) + \frac{1}{2t}\|x^+ - y\|^2 \\
&= \frac{\theta^2}{2t}\left(\|u - x^*\|^2 - \|u^+ - x^*\|^2\right)
\end{aligned}
$$

- *i.e.*, at iteration k

$$
\begin{aligned}
&\frac{t}{\theta_k^2}(f(x^{(k)}) - f^*) + \frac{1}{2}\|u^{(k)} - x^*\|^2 \\
&\quad \leq \frac{(1-\theta_k)t}{\theta_k^2}(f(x^{(k-1)}) - f^*) + \frac{1}{2}\|u^{(k-1)} - x^*\|^2
\end{aligned}
$$

Proof

- Using $(1-\theta_i)/\theta_i^2 \le 1/\theta_{i-1}^2$, and iterating this inequlity

$$\begin{aligned}
&\frac{t}{\theta_k^2}(f(x^{(k)}) - f^*) + \frac{1}{2}\|u^{(k)} - x^*\|^2 \\
&\quad \le \frac{(1-\theta_1)t}{\theta_1^2}(f(x^{(0)}) - f^*) + \frac{1}{2}\|u^{(0)} - x^*\|^2 \\
&\quad = \frac{1}{2}\|x^{(0)} - x^*\|^2
\end{aligned}$$

- Therefore

$$f(x^{(k)}) - f^* \le \frac{\theta_k^2}{2t}\|x^{(0)} - x^*\|^2 = \frac{2}{t(k+1)^2}\|x^{(0)} - x^*\|^2$$

- Although the algebraic manipulations of the proof is beautiful, the acceleration effect in NAG has been mysterious and hard to understand
- Recent works reinterpreted the NAG algorithm from different point of views, including Zhu et al (2017) and Su et al (2014)
- Here we introduce the ODE explanation from Su et al (2014)

- Su et al (2014) proposed an ODE based explanation where NAG can be viewed as a discretization of the following ordinary differential equation

$$\ddot{X} + \frac{3}{t}\dot{X} + \nabla f(X) = 0, \quad t > 0 \tag{4}$$

with initial conditions $X(0) = x^{(0)}, \dot{X}(0) = 0$.

- **Theorem** (Su et al): For any $f \in \mathcal{F}_\infty \triangleq \cap_{L>0}\mathcal{F}_L$ and any $x^{(0)} \in \mathbb{R}^n$, the ODE (4) with initial conditions $X(0) = x^{(0)}, \dot{X}(0) = 0$ has a unique global solution $X \in C^2((0,\infty);\mathbb{R}^n) \cap C^1([0,\infty);\mathbb{R}^n)$.

- **Theorem** (Su et al): For any $f \in \mathcal{F}_\infty$, let $X(t)$ be the unique global solution to (4) with initial conditions $X(0) = x^{(0)}, \dot{X}(0) = 0$. For any $t > 0$,

$$f(X(t)) - f^* \leq \frac{2\|x^{(0)} - x^*\|^2}{t^2}$$

- Consider the energy functional defined as

$$\mathcal{E}(t) \triangleq t^2(f(X(t)) - f^*) + 2\|X + \frac{t}{2}\dot{X} - x^*\|^2$$

- The derivative of the energy function is

$$\dot{\mathcal{E}} = 2t(f(X) - f^*) + t^2\langle \nabla f, \dot{X}\rangle + 4\langle X + \frac{t}{2}\dot{X} - x^*, \frac{3}{2}\dot{X} + \frac{t}{2}\ddot{X}\rangle$$

- Substituting $3\dot{X}/2 + t\ddot{X}/2$ with $-t\nabla f(X)/2$

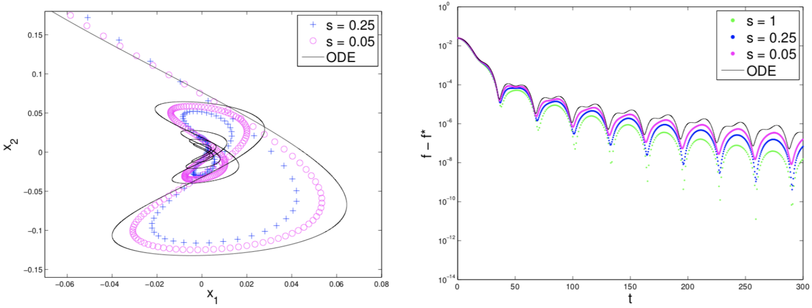
$$\begin{aligned}\dot{\mathcal{E}} &= 2t(f(X) - f^*) + 4\langle X - x^*, -\frac{t}{2}\nabla f(X)\rangle \\ &= 2t(f(X) - f^*) - 2t\langle X - x^*, \nabla f(X)\rangle \\ &\leq 0\end{aligned}$$

  where the last inequality follows from the convexity of f.
- Therefore,

$$f(X(t) - f^*) \leq \mathcal{E}(t)/t^2 \leq \mathcal{E}(0)/t^2 = \frac{2\|x^{(0)} - x^*\|^2}{t^2}$$

Example

$$f(x) = 0.02x_1^2 + 0.005x_2^2, \quad x^{(0)} = (1,1)$$



The objective in many unconstrained optimization problems can be split in two components

$$\text{minimize } f(x) = g(x) + h(x)$$

- g is convex and differentiable on $\mathbb{R}^n$
- h is convex and simple, but may be non-differentiable

Examples

- Indicator function of closed convex set C

$$h(x) = \mathbf{1}_C(x) = \begin{cases} 0, & x \in C \\ +\infty, & x \notin C \end{cases}$$

- L_1 regularization (LASSO): $h(x) = \|x\|_1$

The **proximal mapping** (or **proximal-operator**) of a convex function h is defined as

$$\mathrm{prox}_h(x) = \arg\min_u \left(h(u) + \frac{1}{2}\|u - x\|_2^2 \right)$$

Examples

- $h(x) = 0$: $\mathrm{prox}_h(x) = x$
- $h(x) = \mathbf{1}_C(x)$: prox_h is projection on C

$$\mathrm{prox}_h(x) = \arg\min_{u \in C} \|u - x\|_2^2 = P_C(x)$$

- $h(x) = \|x\|_1$: prox_h is the "soft-threshold" (shrinkage) operation

$$\mathrm{prox}_h(x)_i = \begin{cases} x_i - 1 & x_i \geq 1 \\ 0 & |x_i| \leq 1 \\ x_i + 1 & x_i \leq -1 \end{cases}$$

- **Proximal gradient algorithm**

$$x^{(k+1)} = \mathrm{prox}_{t_k h}(x^{(k)} - t_k \nabla g(x^{(k)})), \quad k = 0, 1, \ldots$$

- Interpretation. If $x^+ = \mathrm{prox}_{th}(x - t\nabla g(x))$, from the definition of proximal mapping

$$\begin{aligned} x^+ &= \arg\min_u \left(h(u) + \frac{1}{2t}\|u - x + t\nabla g(x)\|_2^2 \right) \\ &= \arg\min_u \left(h(u) + g(x) + \nabla g(x)^T (u - x) + \frac{1}{2t}\|u - x\|_2^2 \right) \end{aligned}$$

- x^+ minimizes $h(u)$ plus a simple quadratic local approximation of $g(u)$ around x

- **Gradient Descent**: special case with $h(x) = 0$

$$x^+ = x - t\nabla g(x)$$

- **Projected Gradient Descent**: special case with $h(x) = \mathbf{1}_C(x)$

$$x^+ = P_C(x - t\nabla g(x))$$

- **ISTA** (Iterative Shrinkage-Thresholding Algorithm): special case with $h(x) = \|x\|_1$

$$x^+ = \mathcal{S}_t(x - t\nabla g(x))$$

where

$$\mathcal{S}_t(u) = (|u| - t)_+ \mathrm{sign}(u)$$

- If h is convex and closed,

$$\mathrm{prox}_h(x) = \arg\min_u \left(h(u) + \frac{1}{2}\|u - x\|_2^2 \right)$$

exists and is unique for all x. Moreover, it has the following useful properties

$$\begin{aligned} u = \mathrm{prox}_h(x) &\iff x - u \in \partial h(u) \\ &\iff h(z) \geq h(u) + (x-u)^T(z-u),\ \forall z \end{aligned}$$

- Proximal gradient descent has the same convergence rate as gradient descent when $0 < t \leq 1/L$

$$f(x^{(k)}) - f^* \leq \frac{1}{2kt}\|x^{(0)} - x^*\|_2^2$$

- Similarly, we can apply Nesterov's acceleration for proximal gradient descent. Choose any initial $x^{(0)} = x^{(-1)}$, $\forall\ k = 1, \ldots$

$$y = x^{(k-1)} + \frac{k-2}{k+1}(x^{(k-1)} - x^{(k-2)})$$
$$x^{(k)} = \mathrm{prox}_{t_k h}(y - t_k \nabla g(y))$$

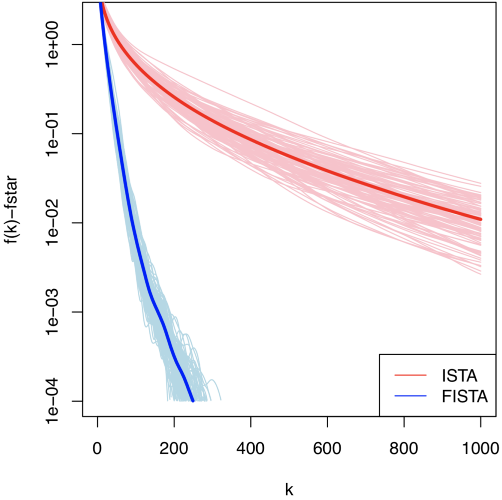
- Convergence rate is the same with NAG if $0 < t \leq 1/L$

$$f(x^{(k)}) - f^* \leq \frac{2\|x^{(0)} - x^*\|^2}{t(k+1)^2}$$

- When applied to LASSO, this is called **FISTA** (Fast Iterative Shrinkage-Thresholding Algorithm)

Example: ISTA vs FISTA

LASSO Logistic regression: 100 instances



Consider the following stochastic optimization problem

$$\min_x \; f(x) = \mathbb{E}_\xi(F(x,\xi)) = \int F(x,\xi)p(\xi)d\xi$$

- ξ is a random variable
- The challenge: evaluation of the expectation/integration

Example

- Supervised Learning

$$\min_w f(w) = \mathbb{E}_{(x,y)\sim D(x,y)}(\ell(h_w(x),y))$$

where $D(x,y)$ is the data distribution, $\ell(\cdot,\cdot)$ is certain loss, w is the model parameter

- Gradient descent with stochastic approximation (SA)

$$x^{(k+1)} = x^{(k)} - t_k g(x^{(k)})$$

where $\mathbb{E}(g(x)) = \nabla f(x),\ \forall x$

- Example. Consider supervised learning with observations $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$

$$\min_w f(w) = \frac{1}{N} \sum_{i=1}^{N} \ell(h_w(x^{(i)}, y^{(i)}))$$

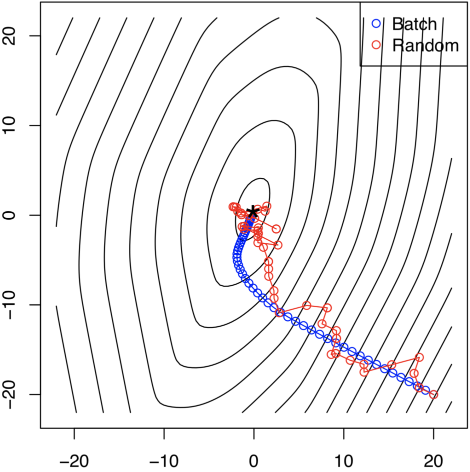
SGD

$$w^{(k+1)} = w^{(k)} - t_k \nabla \ell(h_w(x^{(i_k)}, y^{(i_k)}))$$

where $i_k \in \{1, \ldots, m\}$ is some chosen index at iteration k.

Example

Stochastic logistic regression



- Assume that $\mathbb{E}(\|g(x)\|^2) \le M^2$ and $f(x)$ is convex

$$\mathbb{E}f(\tilde{x}^{[0:k]}) - f^* \le \frac{\|x^{(0)} - x^*\|_2^2 + M^2 \sum_{j=0}^{k} t_j^2}{2\sum_{j=0}^{k} t_k}$$

where $\tilde{x}^{[0:k]} = \sum_{j=1}^{k} t_j x^{(j)} / \sum_{j=1}^{k} t_j$

- Fix the number of iterations K and constant step sizes $t_j = \frac{\|x^{(0)} - x^*\|}{M\sqrt{K}}$, $j = 0, 1, \ldots, K$, we have

$$\mathbb{E}(f(\bar{x}^K)) - f^* \le \frac{\|x^{(0)} - x^*\| M}{\sqrt{K}}$$

where $\bar{x}^K = \frac{1}{K+1} \sum_{j=0}^{K} x^{(j)}$

Proof

By convexity, we have $f(x^{(k)}) - f^* \leq \nabla f(x^{(k)})^T (x^{(k)} - x^*)$

$$
\begin{aligned}
t_k \mathbb{E}(f(x^{(k)})) - t_k f^* &\leq t_k \mathbb{E}(g(x^{(k)})^T (x^{(k)} - x^*)) \\
&= \frac{1}{2}(\mathbb{E}\|x^{(k)} - x^*\|_2^2 - \mathbb{E}\|x^{(k+1)} - x^*\|_2^2) + \frac{1}{2} t_k^2 \mathbb{E}\|g(x^{(k)})\|_2^2 \\
&\leq \frac{1}{2}(\mathbb{E}\|x^{(k)} - x^*\|_2^2 - \mathbb{E}\|x^{(k+1)} - x^*\|_2^2) + \frac{1}{2} t_k^2 M^2
\end{aligned}
$$

$\forall k \geq 0$. Therefore

$$
\sum_{j=0}^{k} t_j \mathbb{E}(f(x^{(j)})) - \sum_{j=0}^{k} t_j f^* \leq \frac{1}{2}\|x^{(0)} - x^*\|_2^2 + \frac{M^2}{2} \sum_{j=0}^{k} t_j^2
$$

Dividing both size with $\sum_{j=0}^{k} t_j$ together with convexity complete the proof

Pros and Cons of Vanilla SGD

What We Love About SGD

- Efficient in computation and memory usage, naturally scalable for big data problems
- Less likely to be trapped at local modes

What Needs to Be Improved

- In general, vanilla SGD is slow to converge (only $1/k$ even with strong convexity). Variance reduction seems to be a good remedy, see algorithms like SVRG, SAGA, etc.
- Choosing a proper learning rate can be difficult, require much effort in hyperparameter tuning to get good results
- The same learning rate applies to all parameter updates

Inspiration From Fisher Scoring

- Assume that f can be related to a probabilistic model, *i.e.*

$$f(\theta) = -\mathbb{E}_{y\sim P_{data}} L(y|\theta) = -\mathbb{E}_{y\sim P_{data}} \log p(y|\theta)$$

- Recall that Fisher information is defined as

$$\mathcal{I}(\theta) = \mathbb{E}_{y\sim p(y|\theta)} (\nabla L(y|\theta)(\nabla L(y|\theta))^T) \tag{5}$$

- We can use Fisher information to adapt the learning rate according to the local curvature. (5) inspire us to use some average of $g(\theta^{(t)})(g(\theta^{(t)}))^T$

- Previously, we performed an update for all parameters using the same learning rate
- Duchi et al (2011) proposed an improved version of SGD, AdaGrad, that adapts the learning rate to the parameters, according to the frequencies of their associated features
- Denote the vector of parameters as θ and the gradient at iteration t as g_t. Let η be the usual learning rate for SGD. AdaGrad's update rule:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{G_t + \epsilon}} \odot g_t$$

  where G_t is a diagonal matrix where each diagonal element is the sum of the squares of the corresponding gradients up to time step t

RMSprop

- A potential weakness about AdaGrad is its accumulation of the squared gradients in G_t, which in turn cause the learning rate to shrink and eventually become very small
- RMSprop (Geoff Hinton): resolve AdaGrad's diminishing learning rate via the exponentially decaying average

$$\mathbb{E}(g^2)_t = 0.9\mathbb{E}(g^2)_{t-1} + 0.1 g_t^2$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\mathbb{E}(g^2)_t + \epsilon}} g_t$$

- Presumably the most popular stochastic gradient methods in machine learning, proposed by D.P. Kingma et al (2014).
- In addition to the squared gradients, Adam also keeps an exponentially decaying average of the past gradients

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t, \quad v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$$

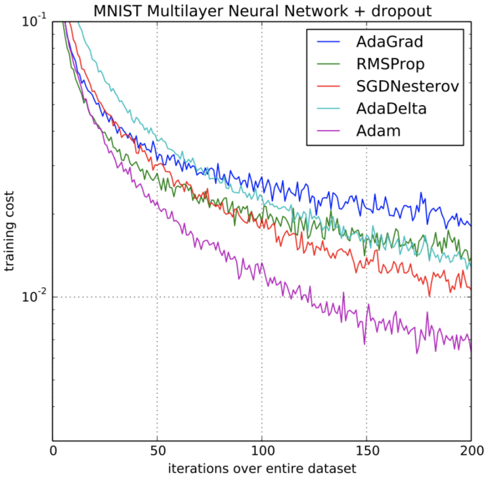
- Bias correction for zero initialization

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1-\beta_2^t}$$

- Adam uses the same update rule

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \hat{m}_t$$

Test on MNIST Images



Pros and Cons for Adaptive Methods

Pros

- Faster training speed and smoother learning curve
- Easier to choose hyperparameters
- Better when data are very sparse

Cons

- Worse performance on unseen data (Wilson et al., 2017)
- Convergence issue: non-decreasing learning rates, extreme learning rates

Some recent proposals for improvement: AMSGrad (Reddi et al., 2018), AdaBound (Luo et al., 2019), etc.

- Polyak, B.T. Some methods of speeding up the convergence of iteration methods. USSR Computational Mathematics and Mathematical Physics, 4(5):1–17, 1964.
- Yurii Nesterov. A method of solving a convex programming problem with convergence rate o (1/k2). Soviet Mathematics Doklady, 27:372–376, 1983.
- Yurii Nesterov. Introductory Lectures on Convex Optimization, volume 87. Springer Science & Business Media, 2004.
- Weijie Su, Stephen Boyd, and Emmanuel J Candes. A differential equation for modeling Nesterov's accelerated gradient method: theory and insights. Journal of Machine Learning Research, 17 (153):1–43, 2016.

References

- A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," SIAM Journal on Imaging Sciences, vol. 2, no. 1, pp. 183–202, 2009.
- A. Nemirovski and A. Juditsky and G. Lan and A. Shapiro (2009), "Robust stochastic optimization approach to stochastic programming"
- R. Johnson and T. Zhang (2013), "Accelerating stochastic gradient descent using predictive variance reduction"
- Kingma, D. P., & Ba, J. L. (2015). Adam: a Method for Stochastic Optimization. International Conference on Learning Representations, 1–13

- Zeyuan Allen-Zhu and Lorenzo Orecchia. Linear Coupling: An Ultimate Unification of Gra- dient and Mirror Descent. In Proceedings of the 8th Innovations in Theoretical Computer Science, ITCS '17, 2017.
- Ashia C Wilson, Rebecca Roelofs, Mitchell Stern, Nati Srebro, and Benjamin Recht. The marginal value of adaptive gradient methods in machine learning. In Advances in Neural Information Processing Systems 30 (NIPS), pp. 4148–4158, 2017.
- Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In International Conference on Learning Representations (ICLR), 2018.

- Liangchen Luo, Yuanhao Xiong, Yan Liu, and Xu Sun. 2019. Adaptive gradient methods with dynamic bound of learning rate. arXiv preprint arXiv:1902.09843 (2019).