Optimization Methods in Machine Learning
Lectures 11-12: Proximal and Accelerated Gradient Methods

Katya Scheinberg

Lehigh University

Spring 2016

Logistic Regression, Gradient and Hessian

We come back to the optimization problem

$$\min f(w) = \frac{1}{m}\sum log(1+e^{-y_i x_i^T w}) + \frac{\lambda}{2}\|w\|_2^2 \tag{1}$$

And we derive the gradient

$$\nabla f(w) = \frac{1}{m}\sum \frac{1}{e^{y_i x_i^T w}+1}(-y_i x_i) + \lambda w \tag{2}$$

And the Hessian

$$\nabla^2 f(w) = \frac{1}{m}\sum \frac{e^{y_i x_i^T w}}{(1+e^{y_i x_i^T w})^2}(y_i x_i)(y_i x_i)^T + \lambda I \tag{3}$$

Notice that the time complexity for computing the Hessian is $d^2 m$, size of Hessian is $d \times d$, the complexity for computing the inverse of Hessian is d^3. So computation can be very expensive for large Hessian!

Definition (Lipschitz Continuity)

Given an open set $B \subseteq \mathbb{R}^n$, we say that f has **Lipschitz continuous gradient** (Lipschitz smooth) on the open subset B if there exists a constant $L \geq 0$ such that

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|, \quad \forall x, y \in B.$$

L is called the **Lipschitz constant** of ∇f on B.

Example (10.1)

An example of functions with different Lipschitz constants for $\nabla f(x)$:

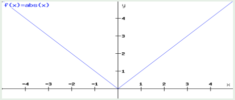


Figure: $L = +\infty$

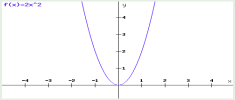


Figure: $L = 4$

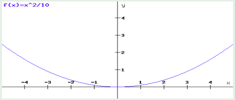


Figure: $L = 0.2$

Lipschitz smooth functions

- Recall that our goal is the minimization problem

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

  where the objective function f is convex and the gradient $\nabla f(x)$ has a Lipschitz constant L.

- By convexity we have the linear under-estimator $f(y) \geq f(x) + \nabla f(x)^T (y - x)$. Instead of using the linear model, we use a quadratic model as our subproblem:

$$Q(y) = f(x) + \nabla f(x)^T (y - x) + \frac{1}{2\mu} \|y - x\|^2,$$

  where if $\mu \leq \frac{1}{L}$, then $Q(y) \geq f(y)$.

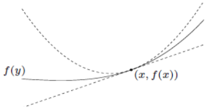
Over-approximation of a smooth function

- Linear lower approximation

$$f(y) \geq f(x) + \nabla f(x)^T (y - x)$$

- Quadratic upper-approximation

$$f(y) \leq f(x) + \nabla f(x)^T (y - x) + \frac{1}{2\mu}\|y - x\|^2 = Q(y), \quad \mu \leq \frac{1}{L}$$



- Rewrite

$$f(y) \leq f(x) + \frac{1}{2\mu}||x - \mu\nabla f(x) - y||^2 - \frac{1}{\mu^2}\|\nabla f(x^k)\|^2 = Q(y)$$

Minimizing $Q(y)$ gives $y = x - \mu\nabla f(x)$.

Basic Proximal Gradient Method

The proximal function of f is defined by:
$\mathbf{prox}_\mu(x^k) = \arg\min_x Q(x; x^k) = f(x^k) + \nabla f(x^k)^T(x - x^k) + \frac{1}{2\mu}\|x - x^k\|^2$.

Algorithm 1 Basic Proximal Gradient Method

Initialization: start with x^0.

for $k = 0$ to maximum iterations **do**
- **repeat**
  - $z \leftarrow \mathbf{prox}_\mu(x^k - \mu \nabla f(x^k))$, and decrease μ,
- **until** $f(z) \leq Q(z; x^k)$.
- $x^{k+1} \leftarrow z$.

end for

Theorem

If $f(x^{i+1}) \leq Q(x^{i+1}; x^i)$ *for all* $i = 1, \ldots, k$, *then*

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

Chosing $\mu = 1/L$ ensures that $f(x^{i+1}) \leq Q(x^{i+1}; x^i)$. By setting $\|x^0 - x^*\| = 1$ and assuming the precision to be ϵ, the computational complexity is estimated to be $k \approx \frac{1}{2\mu\epsilon} \sim \mathcal{O}(\frac{L}{\epsilon})$. Then when $\epsilon \approx 10^{-3}$, $k \approx 1000$ if $L \approx 1$. Compare this to $O(log(1/\epsilon))$.

Proof

- From $f(x^{i+1}) \leq Q(x^{i+1}; x^i)$, it can be shown (skipped here)

$$f(x^{i+1}) - f(x^*) \leq \frac{1}{2\mu}(\|x^i - x^*\|^2 - \|x^{i+1} - x^*\|^2).$$

- By summing the above equations from $i = 0$ to $(k-1)$, and using $f(x^k) \leq f(x^i)$, $\forall i \leq k$

$$k(f(x^k) - f(x^*)) \leq \sum_{i=0}^{k-1} f(x^{i+1}) - kf(x^*) \leq \frac{1}{2\mu}(\|x^0 - x^*\|^2 - \|x^k - x^*\|$$

$$\leq \frac{1}{2\mu}\|x^0 - x^*\|^2.$$

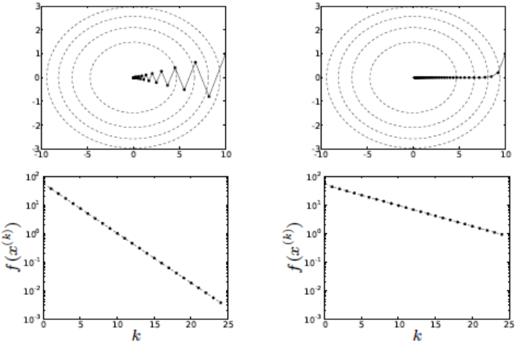
- Hence $f(x^k) - f(x^*) \leq \frac{1}{2\mu k}\|x^0 - x^*\|^2$, which proves the algorithm convergence **sub-linearly**.

Convergence depending on choice of μ

Slides from L. Vandenberghe http://www.ee.ucla.edu/ vandenbe/ee236c.html

Quadratic example

$f(x_1, x_2) = (x_1^2 + Lx_2^2)/2$; left: $t_k = 1.8/L$; right: $t_k = 0.8/L$



Accelerated Gradient Method

Algorithm 2 Accelerated Generalized Gradient Method

Initialization: start with x^0 and $y^1 = x^0$.

for $k = 0$ to maximum iterations **do**

  repeat

    $z \leftarrow \mathbf{prox}_\mu(y^k - \mu \nabla f(y^k))$, and decrease μ,

  until $f(z) \leq Q(z; x^k)$.

  $x^k \leftarrow z$,

  $y^{k+1} \leftarrow x^k + \frac{k-1}{k+2}(x^k - x^{k-1})$.

end for

FISTA (fast iterative shrinkage-thresholding algorithm)

Algorithm 3 FISTA (fast iterative shrinkage-thresholding algorithm)

Initialization: start with x^0, $y^1 = x^0$ and $t^1 = 1$.

for $k = 0$ to maximum iterations **do**

  repeat

    $z \leftarrow \mathbf{prox}_\mu(y^k - \mu \nabla f(y^k))$, and decrease μ,

  until $f(z) \leq q(z; x^k)$.

  $x^k \leftarrow z$,

  $t^{k+1} = (1 + \sqrt{1 + 4(t^k)^2})/2$,

  $y^{k+1} \leftarrow x^k + \frac{t^k - 1}{t^{k+1}}(x^k - x^{k-1})$.

end for

Complexity Result for Algorithms 2 & 3

- For Algorithms 2 & 3, the convergence result for smooth function is

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

  If we pick $\mu = 1/L$ then $f(x^k) \leq f(x^*) \leq \epsilon$ when $k \geq \mathcal{O}(\sqrt{\frac{L}{\epsilon}})$.

  For example, when $\epsilon \approx 10^{-3}$, and $L \approx 1$ then the complexity is approximately $\frac{1}{\sqrt{10^{-3}}} \approx 2^5 = 32$. Compare this to $\mathcal{O}(\frac{L}{\epsilon})$.
- The complexity of standard sub gradient method for non-smooth functions is $\mathcal{O}(\frac{1}{\epsilon^2})$.

  For example, when $\epsilon \approx 10^{-3}$, the complexity is approximately $\frac{1}{(10^{-3})^2} \approx 10^6$.
- We can use accelerated algorithm to have an algorithm for non smooth functions with complexity $\mathcal{O}(\frac{1}{\epsilon})$.

Dealing with the non-smooth function

- Our strategy to reduce the computational complexity is to approximate the function with a smooth function.
- For a non-smooth function, consider a smooth approximation function with the Lipschitz constant of the gradient $L \approx \frac{1}{\epsilon}$. Then $\mu \sim \frac{1}{L} \approx \epsilon$. From the previous slide, $\frac{L}{2k^2} \sim \epsilon$, so $k \sim \frac{1}{\epsilon}$, which means that we can reduce the complexity for non-smooth case from $\mathcal{O}(\frac{1}{\epsilon^2})$ to $\mathcal{O}(\frac{1}{\epsilon})$.

Dealing with the non-smooth function

Example (10.2)

An example of approximation for non-smooth function $f(x) = \frac{1}{\mu}|x|$ is the smooth function:

$$\phi_\mu(x) = \begin{cases} \frac{x^2}{2\mu} & \text{if } |x| \leq \mu \\ |x| - \frac{\mu}{2} & \text{if } |x| > \mu \end{cases}$$

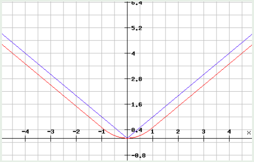


Figure: Non-smooth function and corresponding smooth function with $\mu = 1$

$$|x| - \mu \leq \phi_\mu(x) \leq |x|, \quad \phi''_\mu(x) = \frac{1}{\mu}$$

Unconstrained SVM

Given a training set $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, $x_i \in R^d$, $y \in \{+1, -1\}$

$$\min_w f(w) = \frac{\lambda}{2}\|w\|^2 + \frac{1}{n}\sum_{i=1}^{n} \ell(w, (x_i, y_i))$$

where

$$\ell(w, (x, y)) = \max\{0, 1 - y(w^T x)\}$$

We want to find $f(w) \leq f(w^*) + \epsilon$ - ϵ-optimal solution.

Smoothed SVM

Given a training set $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, $x_i \in R^d$, $y \in \{+1, -1\}$

$$\min_w f_\mu(w) = \frac{\lambda}{2}\|w\|^2 + \frac{1}{n}\sum_{i=1}^{n} \phi_\mu(w, (x_i, y_i))$$

where

$$\phi_\mu(w, (x, y)) = \begin{cases} 0 & y(w^\top x) \geq 1 \\ \frac{(y(w^\top x)-1)^2}{2\mu} & 1-\mu < y_i(w^\top x) < 1 \\ 1 - y(w^\top x) - \frac{\mu}{2} & y(w^\top x) \leq 1-\mu \end{cases}$$

Find $f(w) \leq f(w^*) + \epsilon$ - ϵ-optimal solution. Set $\mu = \epsilon/2$ and find $f_\mu(w) \leq f_\mu(w^*) + \epsilon/2$ with an accelerated method. The L for $f_\mu(w)$ is $2/\epsilon$, hence we find the solution in $O(\sqrt{\frac{4}{\epsilon^2}})$ iterations.