STATS 608A, Fall 2015
Homework 1

Ambuj Tewari

Assigned: Nov 4, 2015. Due: Nov 11, 2015.

1 Lipschitz continuous gradients and strong convexity (5 points)

For this problem assume that all convex functions we talk about are at least twice differentiable. Recall that convexity has three equivalent definitions. A function $f : \mathbb{R}^p \to \mathbb{R}$ is convex iff:

- For any $x, y \in \mathbb{R}^p, \theta \in [0,1]$, $f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y)$
- For any $x, y \in \mathbb{R}^p$, $f(y) \geq f(x) + (\nabla f(x))^\top (y - x)$
- For any x, $\nabla^2 f(x) \succeq \mathbf{0}$ ($\mathbf{0}$ = zero matrix)

Recall that the notation $M \succeq N$ means that the matrix $M - N$ is positive semidefinite. We want to now consider Lipschitz continuity of gradients and strong convexity.

First, prove that the following are equivalent for any convex $f : \mathbb{R}^p \to \mathbb{R}$.

1. ∇f is Lipschitz continuous in ℓ_2 norm with constant L
2. $\nabla^2 f(x) \preceq L \cdot \mathbf{I}$ ($\mathbf{I}$ = identity matrix)
3. For any $x, y \in \mathbb{R}^p$, $f(y) \leq f(x) + (\nabla f(x))^\top (y - x) + \frac{L}{2}\|y - x\|_2^2$

Second, prove that the following are equivalent for any convex $f : \mathbb{R}^p \to \mathbb{R}$.

1. $f(x) - \frac{m}{2}\|x\|_2^2$ is convex
2. $\forall x, y \in \mathbb{R}^p, \|\nabla f(x) - \nabla f(y)\|_2 \geq m\|x - y\|_2$
3. $\nabla^2 f(x) \succeq m \cdot \mathbf{I}$ ($\mathbf{I}$ = identity matrix)
4. For any $x, y \in \mathbb{R}^p$, $f(y) \geq f(x) + (\nabla f(x))^\top (y - x) + \frac{m}{2}\|y - x\|_2^2$

2 Convergence analysis for exact line search (5 points)

Consider the exact line search algorithm that makes the update

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

where the step size t is chosen by solving

$$t = \operatorname*{argmin}_{s \geq 0} f(x - t\nabla f(x)).$$

For a convex function with Lipschitz continuous gradient (constant L), show that exact line search enjoys the convergence guarantee

$$f(x^{(k)}) - f^\star \leq \frac{L\,\|x^{(0)} - x^\star\|_2^2}{2k},$$

after k iterations starting from the point $x^{(0)}$.

3 Lipschitz constant for a logistic regression problem (5 points)

Consider the following objective function:

$$f(\beta) = \frac{1}{n}\sum_{i=1}^{n} \log(1 + \exp(-y_i \beta^\top x_i)).$$

Establish the best possible upper bound you can for the Lipschitz constant of the gradient $\nabla f(\beta)$ in terms of properties of the design matrix $X \in \mathbb{R}^{n\times p}$ that has the n p-dimensional vectors x_i's as its rows. Note that this problem is about logistic regression in a binary classification context, i.e., the label $y_i \in \{-1, +1\}$.

4 Subdifferentials of the group lasso and elastic net penalties (5 points)

Consider a kp-dimensional vector x partitioned into k p-dimensional blocks: $x = (x_1, x_2, \ldots, x_k)$. The group lasso penalty is defined as

$$r_1(x) = \sum_{b=1}^{k} \|x_b\|_2$$

as the sum of the ℓ_2 norms of the k blocks. Note that when $p = 1$, this is simply the Lasso penalty. Compute its subdifferential $\partial r_1(x)$

The elastic net penalty is $r_2(x) = \lambda\|x\|_2^2 + \mu\|x\|_1$. Compute its subdifferential $\partial r_2(x)$.

5 Computational Problem (5 points)

We will test different versions of gradient descent on a least squares objective

$$f(\beta) = \frac{1}{n}\|Y - X\beta\|_2^2$$

Don't forget the $1/n$ scaling above! The data set (that will give us the design matrix $X \in \mathbb{R}^{n\times p}$ and response vector $Y \in \mathbb{R}^n$) we will use is the Wine-Quality data set:
`http://mlr.cs.umass.edu/ml/datasets/Wine+Quality`
Note that $n = 4898$ and $p = 11$ while the 12th attribute (quality score between 0 and 10) is our Y.

Implement gradient descent with: (i) constant step-size (use $t = 1/L$ where L is the Lipschitz constant of the gradient ∇f), (ii) back-tracking line search $\alpha = \beta = 1/2$ and (iii) exact line search (use your favorite univariate function minimizer in your favorite programming language to implement the exact line search).

Note added later: Standardize the data before minimizing the least squares objective. That is, center and scale all features so that they have mean zero and variance one.

Stopping rule: stop your optimization once the gradient norm $\|\nabla f(\beta^{(k)})\|_2$ falls below 10^{-3}.

Submit 2 plots. In plot 1, show the objective function vs. number of iterations for all 3 methods. In plot 2, show the objective function vs. time taken for all 3 methods.

Answer the following questions:

1. What L did you use?
2. What was the average number of backtracking steps per iteration in the backtracking implementation?
3. What is the starting value of the objective function (start every method at $\mathbf{0}$)?
4. What is the ending value of the objective function for each method?
5. How do the methods compare in terms of the number of iterations required to reach a certain accuracy? In terms of time?