Homework 2

Convex Optimization 10-725/36-725

Due Tuesday February 17 at 4:00pm
submitted to Mallory Deptola in GHC 8001
(make sure to submit each problem separately)

1 Gradient Descent and Strong Convexity [Mattia]

We proved in class that for a convex differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ with Lipschitz gradient, the convergence rate for gradient descent is $O(1/k)$ where k is the number of iterations. More precisely, we proved that

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

where $0 < t < 1/L$ is a fixed step size, L is the Lipschitz constant of the gradient of f, $x^{(0)}$ is the starting point, $x^* \in \arg\min_{x \in \mathbb{R}_n} f(x)$, $f^* = f(x^*)$, and k is the number of iterations. We also mentioned that if f is also strongly convex – i.e. there exists some $m > 0$ such that $g(x) = f(x) - \frac{m}{2}\|x\|_2^2$ is still a convex function – then the rate of convergence for gradient ascent is $O(c^k)$ for some $c \in (0, 1)$ as soon as the step size satisfies $t \leq 2/(m + L)$. In this exercise you will prove this result.

(a) Throughout this exercise the step size t is assumed to be a fixed number (across the gradient descent iterations) and $0 < t \leq 2/(m + L)$. Start by showing that

$$\|x^{(k)} - x^*\|_2^2 \leq \left(1 - \frac{2tmL}{m + L}\right)^k \|x^{(0)} - x^*\|_2^2.$$

Hint 1: use the gradient descent update rule to write an expression for $\|x^{(k+1)} - x^*\|_2^2$ and apply the following Theorem.

Theorem 1. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a strongly convex function with Lipschitz gradient. Let L and m be respectively the Lipschitz constant of the gradient of f and the strong convexity constant of f. Then, for any $x, y \in \mathbb{R}^n$,*

$$\langle \nabla f(x) - \nabla f(y), x - y \rangle \geq \frac{mL}{m + L}\|x - y\|_2^2 + \frac{1}{m + L}\|\nabla f(x) - \nabla f(y)\|_2^2.$$

Hint 2: remember that $\nabla f(x^*) = 0$.

(b) Use the result of part (a) to prove that

$$f(x^{(k)}) - f^* \leq \frac{L}{2}\left(1 - \frac{2tmL}{m + L}\right)^k \|x^{(0)} - x^*\|_2^2.$$

Hint 3: if $f : \mathbb{R}^n \to \mathbb{R}$ is a convex differentiable function with Lipschitz gradient and Lipschitz constant L, then for any $x, y \in \mathbb{R}^n$

$$0 \le f(y) - f(x) - \langle \nabla f(x), y - x \rangle \le \frac{L}{2}\|x - y\|_2^2.$$

Hint 4: remember Hint 2! Also, you may find the following characterization of a convex differentiable function useful: a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if, for any $x, y \in \mathbb{R}^n$, $\langle \nabla f(x) - \nabla f(y), x - y \rangle \ge 0$.

(c) Verify that $0 < \left(1 - \frac{2tmL}{m+L}\right) < 1$ for $0 < t \le 2/(m + L)$. In particular, show that

$$\min_{0<t\le 2/(m+L)} \left(1 - \frac{2tmL}{m+L}\right) = \left(\frac{Q-1}{Q+1}\right)^2 = c(Q)$$

where $Q = L/m$ is the condition number of f.

(d) Compute $\lim_{Q\to\infty} c(Q)$. Based on the limit that you just computed, how does the condition number affect the rate of convergence of part (b)?

2 Subgradients and Prox Operators [Yu-Xiang]

(a) We learned subgradients in class as generalization of gradients. For a function, the subdifferential at a point $x \in \mathrm{dom}(f)$ (denoted by $\partial f(x)$) is the set that collects all subgradients at x. Characterize the subdifferential of the following functions.

i. Finite pointwise maximum: $f(x) = \max_{i=1,\dots,m} g_i(x)$, where $g_1, \dots g_m$ are convex. State the solution as learned in lecture, and then prove the "$\supseteq$" direction of the subdifferential characterization; the "$\subseteq$" direction is left as a bonus.

ii. Nuclear norm (or trace norm): $f(X) = \|X\|_{\mathrm{tr}} := \sum_{i=1}^{r} \sigma_i(X)$, where $X \in \mathbb{R}^{m\times n}$, rank r, and $\sigma_i(X)$, $i = 1, \dots r$ are its singular values. Recall that we claimed in class that

$$\partial\|X\|_{\mathrm{tr}} = \{UV^T + W : \|W\|_{\mathrm{op}} \le 1,\ U^TW = 0,\ WV = 0\},$$

where $X = U\Sigma V^T$ is the singular value decomposition of X, and $\|W\|_{\mathrm{op}} = \sigma_1(X)$ is the operator norm (largest singular value) of X. Prove "$\supseteq$" in the above equality; the other direction "$\subseteq$" is left as a bonus.

iii. Two nonconvex examples: $f(x) = \sin(x)$ and $f(x) = x^3$.

(b) Assume that $\mathrm{dom}(f) = \mathbb{R}^n$ for simplicity. Prove that $g \in \partial f(x)$ iff the vector $(g, -1) \in \mathbb{R}^{n+1}$ defines a supporting hyperplane to $\mathrm{epi}(f) = \{(x, z) \in \mathbb{R}^{n+1} : f(x) \le z\}$ at the point x. (Note: by "defines" here, we mean that $(g, -1)$ is the normal vector to the supporting hyperplane.)

(c) In proximal gradient descent, every iteration applies what is called a proximal map (or prox operator). Given a function f, its proximal map is defined, for $t \ge 0$, as

$$\mathrm{prox}_{f,t}(y) = \operatorname*{argmin}_x \ \frac{1}{2}\|y - x\|_2^2 + tf(x).$$

For instance, when $f(x) = I_C(x)$, the proximal map is the Euclidean projection onto the set C. Find the proximal map $\mathrm{prox}_{f,t}(y)$ for the following functions.

i. Quadratic function: $f(x) = \frac{1}{2}x^TAx + b^Tx + c$ with $A \succeq 0$.

ii. Logarithmic barrier: $f(x) = -\sum_{i=1}^{n} \log x_i$.

iii. ℓ_0 (pseudo-)norm: $f(x) = \|x\|_0 = \sum_{i=1}^{n} 1\{x_i \neq 0\}$.

Hint: x is an optimal solution to a (possibly nonconvex) optimization problem $\min_x f(x)$ if and only if $0 \in \partial f(x)$. Also if $\ell(x)$ is differentiable and $f(x)$ is not, you can use the identity:

$$\partial(\ell(x) + f(x)) = \nabla\ell(x) + \partial f(x) = \{g + \nabla\ell(x) \mid g \in \partial f(x)\}.$$

(d) Is the proximal map uniquely defined when $f(x)$ is convex? How about when $f(x)$ is nonconvex? Prove or disprove by constructing a counterexample.

3 ISTA and FISTA for LASSO [Junier]

As covered in class, ℓ_1 regularization is an important and widely used tool for achieving sparse solutions to optimization problems. Below we shall use ℓ_1 regularization in the LASSO to find sparse solutions to a least squares regression problem. Recall that the LASSO optimizes:

$$\min_{w \in \mathbb{R}^p} \frac{1}{2}\|y - Xw\|_2^2 + \lambda\|w\|_1, \tag{1}$$

where $y \in \mathbb{R}^n$ is a vector of responses, $X \in \mathbb{R}^{n \times p}$ is a matrix of covariates, and $\lambda \in \mathbb{R}_{++}$ is a penalty parameter.

(a) First we shall code Iterative Shrinkage-Thresholding Algorithm (ISTA), which uses proximal gradient descent to optimize the LASSO. Please write a function with the following interface:

```
ws = ista(X, y, lambda, w0, iters);
```

That is, the function `ista` should take arguments of the covariate matrix X, response vector y, and λ penalty parameter, as well as `w0`, the vector to begin optimizing from, and `iters`, the number of iterations to run ISTA for. `ista` should return a $p \times$`iters` matrix of the values of the weight vector w through the iterations; e.g., in MATLAB notation, `ws(:,10)` should contain the value of w after 10 iterations of ISTA. Make sure to implement back-tracking for step sizes in your function, with $\beta = .9$ and $t_0 = 1$ (see slide 11 of `http://www.stat.cmu.edu/~ryantibs/convexopt/lectures/08-prox-grad.pdf`).

(b) Next, we will code Fast Iterative Shrinkage-Thresholding Algorithm (FISTA), which uses accelerated proximal gradient descent to optimize the LASSO. Write your function implementing FISTA as:

```
ws = fista(X, y, lambda, w0, iters);
```

The arguments and outputs are as in (a). As before, be sure to implement back-tracking for step sizes in your function with $\beta = .9$ and $t_0 = 1$ (see slide 26 of `http://www.stat.cmu.edu/~ryantibs/convexopt/lectures/08-prox-grad.pdf`).

(c) We compare the performance of `ista` and `fista`. On the course website, you will find the data files `homework2_q3_data.mat` and `homework2_q3_data.RData`. They contain following variables:

- `lambda`;
- `Xs`, a $100 \times 500 \times 100$ matrix (this is $n \times p \times$ trials);

- `ys`, a 100×100 matrix;
- `w_stars`, a 500×100 vector.

Note that in MATLAB notation, for trial `t`, `Xs(:,:,t)` is the predictor matrix, `ys(:,t)` is the response vector, and `w_stars(:,t)` is the corresponding optimal weight vector. The variable `lambda` contains the penalization value to use in all optimizations.

For each `t` between 1 and 100, solve the corresponding LASSO problem with ISTA and FISTA, both initialized at 0. Record the difference in achieved versus optimal criterion value across 1000 iterations. Then produce a figure that draws all of these curves on the same plot. Your figure should look something like (but necessarily exactly like) Figure 1 below.

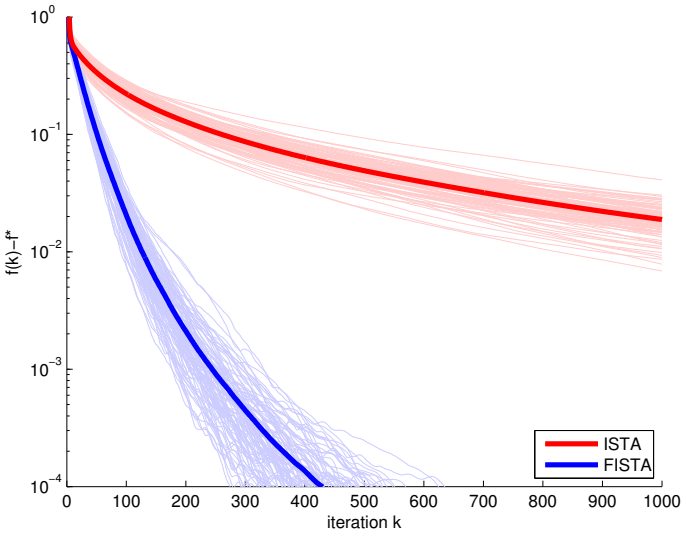

Figure 1: Note the log scale on the y-axis, and and the bold mean curve (over trials) of objective differences for ISTA and FISTA.

(e) Simply describe (do not implement): what would you have to change in your code if we asked you to solve an ℓ_1-penalized logistic regression problem, instead of the ℓ_1-penalized Gaussian regression in (1)?

4 Warm Starts and Stagewise Regression [Nicole]

In general, it is good practice to be able to generate your own synthetic data sets so that you can debug, compare, and evaluate your implementations of various algorithms.

For this problem, generate the following synthetic data set, with 100 samples and 500 features. The true weight vector is 10 1s followed by 490 0s. Generate the predictor matrix $X \in \mathbb{R}^{100 \times 500}$ to have with i.i.d. $\mathcal{N}(0, 1)$ entries. Then let the repsonse vector be

$$y = Xw_{true} + \epsilon,$$

where $\epsilon \in \mathbb{R}^{100}$ has i.i.d. entries also drawn from a standard Gaussian. Using the same weight vector, generate another predictor matrix and response vector, in the same manner as above, to form your hold-out set.

(a) In the previous question you solved the LASSO problem for a fixed value of the hyperparameter λ. Since λ governs the degree of sparsity in the solution, in general we will want to find an appropriate value of λ for any given problem. Run FISTA for 100 different values of λ, log spaced between 10^4 and $10^{0.015}$, in that order (e.g. use MATLAB's logspace(4, 0.015, 100)). For each λ, use the solved w from the previous λ as your starting value, as opposed to starting from 0 each time.

Modify your code from the last question to include a stopping rule for FISTA. Stop when the absolute relative difference in criterion values across iterations (current criterion value minus previous, divided by previous) is less than 10^{-4}. At each λ, you will run FISTA until this is met, or for a maximum of 500 iterations.

In your writeup, include the following four plots:

i MSE on the held-out set vs $\log(\lambda)$;

ii MSE on the held-out set vs $||w||_1$;

iii a plot of the values of w vs $||w||_1$; one line per component;

iv the number of iterations to complete vs $\log(\lambda)$.

Which λ would you choose? To what value of $||w||_1$ does this correspond?

(b) In part (a) you used "warm starts," that is, you started with a large value for λ and then used that solution to guide the next. In terms of the types of convergence rates we have covered in class, why might this be a good idea, i.e., why would it make solving the problem for subsequent values of λ faster? And what is this assuming?

(c) Implement forward stagewise regression to solve this problem. For this, it is convenient to reformulate the LASSO problem in bound form:

$$\min_{w \in \mathbb{R}^p} \frac{1}{2}||y - Xw||_2^2 \quad \text{s.t.} \quad ||w||_1 \leq s, \tag{2}$$

where now $s \geq 0$ is the tuning parameter. As you probably already know, problems (1) and (2) are equivalent in the sense that, for each λ in (1) there is an s in (2) that yields the same solution, and vice versa. (We will learn this more precisely later in the course when we study duality.)

In stagewise regression, we start with $s = 0$ and $w^{(0)} = 0$. Then, writing $X_1, \ldots X_p \in \mathbb{R}^n$ for the columns of X, we repeat the following steps for $k = 1, 2, 3, \ldots$:

$$w^{(k)} = w^{(k-1)} + \epsilon \cdot \text{sign}\big(X_i^T(y - Xw^{(k-1)})\big) \cdot e_i,$$
$$\text{where } i = \underset{j=1,\ldots p}{\text{argmax}} |X_j^T(y - Xw^{(k-1)})|.$$

In the above, e_i is the ith standard basis vector, and $\epsilon > 0$ is a small fixed constant.

Run stagewise regression on the same data set as in part (a), and record the estimates $w^{(k)}$ across steps k of the algorithm. (Note: the interpretation of an iteration k here is different than in ISTA or FISTA, because each iteration produces for us a new estimate of the regression coefficients.) Record the value of $||w^{(k)}||_1$ across iterations k, and run for 1500 iterations. Use the step size $\epsilon = 10^{-2}$.

In your writeup, produce plots ii and iii as in part (a) but for stagewise regression. What do you feel are the key differences in the performance of each algorithm? To what do you attribute these differences?

Did you settle on a different value of $||w||_1$ for stagewise? What is the corresponding MSE on the held-out set, and how does it compare to the error of the best LASSO solution?

(d) Why did we use a held-out portion of our data set (as opposed to training data) to compute the MSE when we were trying to gauge performance for various values of λ (values of $\|w\|_1$)?

Bonus Problem 1: Strong Convexity Key Fact [Mattia]

A key technical device used in Question 1 was Theorem 1. Prove it!

Bonus Problem 2: Complete Characterizations of Subdifferentials [Yu-Xiang]

Prove the reverse directions for the subdifferential characterizations in Question 2 part (a) subpart i., and Question 2 part (b).

That is, for $f(x) = \max_{i=1\ldots m} g_i(x)$, where $g_1, \ldots g_m$ are convex, prove that

$$\partial f(x) \subseteq \text{conv}\Big(\bigcup_{i:g_i(x)=f(x)} \partial g_i(x) \Big).$$

You may assume that each g_i has full domain, and is continuous. Hint: start with the simple case of just two functions, g_1 and g_2.

Also, prove that for $f(x) = \|X\|_{\text{tr}}$,

$$\partial \|X\|_{\text{tr}} \subseteq \{UV^T + W : \|W\|_{\text{op}} \leq 1,\ U^T W = 0,\ WV = 0\}.$$