Homework 2

Convex Optimization 10-725/36-725

Due Monday October 3 at 5:30pm
submitted to Christoph Dann in Gates 8013
(Remember to a submit separate writeup for each problem, with your name at the top)

Total: 75 points

1 Subgradients and Proximal Operators (16 pts) [Han]

(a) Recall that subgradient can be viewed as a generalization of gradient for general functions. Let f be a function from $\mathbb{R}^n$ to $\mathbb{R}$. The subdifferential of f at x is defined as $\partial f(x) = \{g \in \mathbb{R}^n : g \text{ is a subgradient of } f \text{ at } x\}$.

(i, 2 pts) Show that $\partial f(x)$ is a convex and closed set.

(ii, 2 pts) Let $f(x) = ||x||_2$. Show that

$$\partial f(x) = \begin{cases} \{x/||x||_2\}, & x \neq 0 \\ \{z : ||z||_2 \leq 1\}, & x = 0 \end{cases}$$

(iii, 2 pts) More generally, let $p, q > 0$ such that $\frac{1}{p} + \frac{1}{q} = 1$. Consider function $f(x) = ||x||_p$, where $||x||_p$ is defined as:

$$||x||_p = \max_{||z||_q \leq 1} z^T x$$

Based on the definition of $||x||_p$, show that $\forall x, y$:

$$x^T y \leq ||x||_p ||y||_q$$

The above inequality is known as Hölder's inequality.

(iv, 2 pts) Use Hölder's inequality to show that $\partial f(x) = \text{argmax}_{||z||_q \leq 1} z^T x$. (You are not allowed to use the rule for the subdifferential of a max of functions for this problem.)

(b) The proximal operator for function $h : \mathbb{R}^n \mapsto \mathbb{R}$ and $t > 0$ is defined as:

$$\text{prox}_{h,t}(x) = \underset{z}{\text{argmin}} \frac{1}{2}||z - x||_2^2 + th(z)$$

Compute the proximal operators $\text{prox}_{h,t}(x)$ for the following functions.

(i, 2 pts) $h(z) = \frac{1}{2}z^T Az + b^T z + c$, where $A \in \mathbb{S}^n_+$.

(ii, 2 pts) $h(z) = -\sum_{i=1}^n \log z_i$, where $z \in \mathbb{R}^n_{++}$.

(iii, 2 pts) $h(z) = ||z||_2$.

(iv, 2 pts) $h(z) = ||z||_0$, where $||z||_0$ is defined as $||z||_0 = |\{z_i : z_i \neq 0, i = 1, \ldots, n\}|$.

2 Properties of Proximal Mappings and Subgradients (16 points) [Christoph]

(a, 3pts) Prove one direction of the finite pointwise maximum rule for subdifferentials: The subdifferential of $f(x) = \max_{i=1,\ldots,n} f_i(x)$ satisfies

$$\partial f(x) \supseteq \operatorname{conv}\left(\bigcup_{i: f_i(x) = f(x)} \partial f_i(x) \right). \tag{1}$$

(Bonus: prove the other direction.)

(b, 3pts) Show that if $f : \mathbb{R}^n \to \mathbb{R}$ is a convex function the following property holds

$$(x-y)^\top (u-v) \geq 0 \qquad \forall x, y \in \mathbb{R}^n, u \in \partial f(x), v \in \partial f(y). \tag{2}$$

(c, 3pts) Let $f(x) = \mathbb{E}[g(x, Y)] = \int g(x, y) p(y) dy$ be the expectation of a random function g that takes as input a fix x and a random variable Y. Let $h(x, y)$ be a function such that for every y it holds that $h(x, y)$ is a subgradient of $g(\cdot, y)$ at x, that is $h(x, y) \in \partial_x g(x, y)$. Show that $\int h(x, y) p(y) dy \in \partial f(x)$.

(d, 3pts) Recall the definition of the proximal mapping: For a function h, the proximal mapping prox_t is defined as

$$\text{prox}_t(x) = \operatorname*{argmin}_u \frac{1}{2t} \|x - u\|_2^2 + h(u). \tag{3}$$

Show that $\text{prox}_t(x) = u \Leftrightarrow h(y) \geq h(u) + \frac{1}{t}(x-u)^\top (y-u) \quad \forall y$.

(e, 4pts) Prove that the prox_t mapping is non-expansive, that is,

$$\| \text{prox}_t(x) - \text{prox}_t(y) \|_2 \leq \|x - y\|_2 \quad \forall x, y. \tag{4}$$

3 Convergence Rate for Proximal Gradient Descent (20 points) [Alnur]

In this problem, you will show that the convergence rate for proximal gradient descent (also known as the proximal gradient method) is $O(1/k)$, where $k \geq 1$ is the number of iterations that the algorithm is run for, which was also presented in class. As a reminder, the setup for proximal gradient descent is as follows. We assume that the objective $f(x)$ can be written as $f(x) = g(x) + h(x)$ (more details on these functions are given below); then, we compute the iterates

$$x^{(i)} = \text{prox}_{t_i h}\left(x^{(i-1)} - t_i \nabla g(x^{(i-1)}) \right), \tag{5}$$

where $i \geq 1$ is an iteration counter, $x^{(0)}$ is the initial point, and the $t_i > 0$ are step sizes (chosen appropriately, during iteration i).

To be clear, we are assuming the following conditions here.

(A1) g is convex, differentiable, and $\text{dom}(g) = \mathbb{R}^n$.

(A2) ∇g is Lipschitz, with constant $L > 0$.

(A3) h is convex, not necessarily differentiable, and we take $\text{dom}(h) = \mathbb{R}^n$ for simplicity.

(A4) The step sizes t_i are either taken to be constant, i.e., $t_i = t = 1/L$, or chosen by backtracking line search; either way, the following inequality holds:

$$g(x^{(i)}) \leq g(x^{(i-1)}) - t\nabla g(x^{(i-1)})^T G_t(x^{(i-1)}) + (t/2)\|G_t(x^{(i-1)})\|_2^2, \tag{6}$$

where t is the step size at any iteration of the algorithm, and we define

$$G_t(x^{(i-1)}) = (1/t)\left(x^{(i-1)} - x^{(i)}\right).$$

(In case you are wondering, this inequality follows from assumption (A2), but you can just take it to be true for this problem.)

Now, finally, for the problem. Assume, for all parts of this problem except the last one, that the step size is fixed, i.e., $t_i = t = 1/L$.

(a, 2pts) Show that

$$s = G_t(x^{(i-1)}) - \nabla g(x^{(i-1)})$$

is a subgradient of h evaluated at $x^{(i)}$. (As a reminder, h is the potentially nondifferentiable function in our decomposition of the objective f; see above for details.)

(Hint: Look back at what you did in Q2 part (d).)

(b, 4pts) Derive the following (helpful) inequality:

$$f(x^{(i)}) \leq f(z) + G_t(x^{(i-1)})^T(x^{(i-1)} - z) - (t/2)\|G_t(x^{(i-1)})\|_2^2, \quad z \in \mathbb{R}^n.$$

(c, 4pts) Show that the sequence of objective function evaluations $\{f(x^{(i)})\}$, $i = 0, \ldots, k$, is nonincreasing (don't worry about the case when $x^{(i)}$ is a minimizer of f). (By the way, this result basically says that proximal gradient descent is a "descent method".)

(d, 4pts) Derive the following (helpful) inequality:

$$f(x^{(i)}) - f(x^*) \leq \frac{1}{2t}\left(\|x^{(i-1)} - x^*\|_2^2 - \|x^{(i)} - x^*\|_2^2\right),$$

where x^* is a minimizer of f (we assume $f(x^*)$ is finite). (By the way, this result, taken together with what you showed in part (c), implies that we move closer to the optimal point(s) on each iteration of proximal gradient descent.)

(e, 4pts) Now, show that after k iterations, the accuracy that proximal gradient descent (with a fixed step size of $1/L$) obtains is $O(1/k)$, i.e.,

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

in other words, the convergence rate for proximal gradient descent is $O(1/k)$. (Put differently, if you desire ε-level accuracy, roughly speaking, then you must run proximal gradient descent for $O(1/\varepsilon)$ iterations.)

(f, 2pts) Establish the analogous convergence rate result when the step sizes are chosen according to backtracking line search, i.e.,

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

where $t_{\min} = \min_{i=1,\ldots,k} t_i$.

4 Proximal Gradient Descent for Group Lasso (23 points) [Justin & Mariya]

Suppose predictors (columns of the design matrix $X \in \mathbb{R}^{n\times(p+1)}$) in a regression problem split up into J groups:

$$X = \begin{bmatrix}\mathbf{1}\; X_{(1)}\; X_{(2)}\; \dots\; X_{(J)}\end{bmatrix} \tag{7}$$

where $\mathbf{1} = (1\;1\;\cdots\;1) \in \mathbb{R}^n$. To achieve sparsity over non-overlapping groups rather than individual predictors, we may write $\beta = (\beta_0, \beta_{(1)}, \dots, \beta_{(J)})$, where β_0 is an intercept term and each $\beta_{(j)}$ is an appropriate coefficient block of β corresponding to $X_{(j)}$, and solve the group lasso problem:

$$\min_{\beta\in\mathbb{R}^{p+1}} \; g(\beta) + \lambda \sum_{j=1}^{J} w_j \|\beta_{(j)}\|_2 \tag{8}$$

A common choice for weights on groups w_j is $\sqrt{p_j}$, where p_j is number of predictors that belong to the jth group, to adjust for the group sizes.

(a) (3 pts) Derive the proximal operator $\text{prox}_{h,t}(x)$ for the nonsmooth component $h(\beta) = \lambda \sum_{j=1}^{J} w_j \|\beta_{(j)}\|_2$.

(b) (10 pts) Download the birthweight data set `birthwt.zip` from the course website. This data contains 189 observations, 16 predictors (in `X.csv`), and an outcome, birthweight (in `y.csv`). The data were collected at Baystate Medical Center, Springfield, Mass during 1986. The 16 columns in the predictor matrix have groupings according to their names (e.g. `mygroupname3`), and each represent the following:

- `age1,age2,age3`: Orthogonal polynomials of first, second, and third degree representing mothers age in years
- `lwt1,lwt2,lwt3`: Orthogonal polynomials of first, second, and third degree representing mothers weight in pounds at last menstrual period
- `white,black`: Indicator functions for mothers race; "other" is reference group.
- `smoke`: Smoking status during pregnancy
- `ptl1,ptl2m`: Indicator functions for one or for two or more previous premature labors, respectively. No previous premature labors is the reference category.
- `ht`: History of hypertension
- `ui`: Presence of uterine irritability
- `ftv1,ftv2,ftv3m`: Indicator functions for one, for two, or for three or more physician visits during the first trimester, respectively. No visits is the reference category.

✍ *Did you know?* A reference category/group is the *baseline* level in categorical data when it is coded as dummy variables. For instance, if a categorical variable has 3 levels (say, three drugs administered to patients), then a baseline category may be the first drug, and two dummy variables may be created each with ith entry is coded as 1 if the ith person was treated with that drug. Why do this? This allows for the model's fitted coefficient for the dummy variables measure the average difference between the response level between the second or third category and the first category (adjusting for the effect of all other variables).

(i) Let $g(\beta) = \|y - X\beta\|_2^2$, in which case the problem above is called the least squares group lasso problem. Derive the gradient of g in this case.

(ii) Use proximal gradient descent on the least squares group lasso problem on with the birthweight data set. Set $\lambda = 4$, using a fixed step size $t = 0.002$ and 1000 steps.

Now, implement accelerated proximal gradient descent with fixed step size. Use the same λ, t, and number of iterations as before.

For both methods, plot $f^{(k)} - f^\star$ versus k (i.e., $f^{(k)} - f^\star$ is on the y-axis in log scale, and k on the x-axis), where $f^{(k)}$ denotes the objective value at iteration k, and the optimal objective value is $f^\star = 84.6952.$

(iii) Print the components of the solutions numerically from the two methods in part (ii) to see that they are close. What are the selected groups?

(iv) Now implement the lasso (hint: you shouldn't have to do any additional coding), with fixed step size with $\lambda = 0.35$, and compare the lasso solution with your group lasso solutions.

(c) (10 pts) In this problem, we'll use logistic group lasso to classify a person's age group from his movie ratings. The movie ratings can be categorized into groups according to a movie's genre (e.g. all ratings for action movies can be grouped together). Our data does not contain ratings for movies from multiple genre (i.e. has no overlapping groups). Similarly to part (b), we'll use proximal gradient descent to solve the group lasso problem.

We formulate the problem as a binary classification with output label $y \in \{0, 1\}$, corresponding to whether a person's age is under 40, and input features $X \in \mathbb{R}^{n \times p}$. We model each $y_i|x_i$ with the probabilistic model

$$\log\left(\frac{p_\beta(y_i = 1|x_i)}{1 - p_\beta(y = 1|x_i)}\right) = (X\beta)_i,$$

$i = 1, \ldots, n$. The logistic group lasso estimator is given by solving the minimization problem in (8) with

$$g(\beta) = -\sum_{i=1}^n y_i(X\beta)_i + \sum_{i=1}^n \log(1 + \exp\{(X\beta)_i\}),$$

the negative log-likelihood under the logistic probability model.

(i) Derive the gradient of g in this case.

(ii) Implement proximal gradient descent to solve the logistic group lasso problem. Fit the model parameters on the training data (`moviesTrain.mat` available on the class website). The features have already been arranged into groups and you can find information about the labels of each group in `moviesGroups.mat`. Use regularization parameter $\lambda = 5$ for 1000 iterations with fixed step size $t = 10^{-4}$.

Now, implement accelerated proximal gradient descent with fixed step size. Use the same λ, t, and number of iterations as before.

Lastly, implement proximal gradient descent with backtracking line search (and no acceleration). Here you can set the step-size shrinking parameter β to 0.1. Use the same λ as before, but only 400 outer iterations.

For each of the three methods, plot $f^{(k)} - f^\star$ versus k, where $f^{(k)}$ denotes the objective value at iteration k, and now the optimal objective value is $f^\star = 336.207$ on a semi-log scale (i.e. where the y-axis is in log scale). For backtracking line search, count the inner iterations towards the iteration number, in order to make a fair comparison.

(iii) Finally, we will use the accelerated proximal gradient descent from part (*ii*) to make predictions on the test set, available in `moviesTest.mat`. What is the classification error? What movie genre are important for classifying whether a viewer is under 40 years old?