SYS 6003: Optimization I **Fall 2016**

Homework 4

Handed Out: November 11^{th}*,* 2016 *Due: November* 21^{st}*,* 2016

- Feel free to talk to other students in the class when doing the homework. You should, however, write down your solution yourself. Please try to keep the solution brief and clear.
- You will write your solution in Latex, write the codes in Matlab, and submit the printed pdf file in class. You also need to submit the latex source files, Matlab files as a zipped package to UVa Collab.
- The first page of your submission should include the pledge followed by your signature.
- The homework is due at 3:30 PM (before the class) on the due date.

1. (10 points) Derive the conjugate of the Power function: $f(x) = x^p$ on $\mathbb{R}_{++}$ where $p > 1$. Repeat for $p < 0$.

2. (10 points) Show that the conjugate of $f(X) = \mathrm{tr}(X^{-1})$ with $\mathbf{dom}\ f = \mathbb{S}^n_{++}$ is given by
$$f^*(Y) = -2 \cdot \mathrm{tr}(-Y)^{1/2}, \quad \mathbf{dom}\ f = -\mathbb{S}^n_+.$$

3. (10 points) Young's inequality: Let $f : \mathbb{R} \to \mathbb{R}$ be an increasing function with $f(0) = 0$ and let g be its inverse. Define F and G as
$$F(x) = \int_0^x f(a)da, \quad G(y) = \int_0^y g(a)da.$$
Show that F and G are conjugates. Given a simple graphical interpretation of Young's inequality.
$$xy \leq F(x) + G(y).$$

4. (10 points) Conjugate of negative normalized entropy. Show that the conjugate of the negative normalized entropy
$$f(x) = \sum_{i=1}^n x_i \log(x_i / \mathbf{1}^\top \mathbf{x})$$
with $\mathbf{dom}\ f = \mathbb{R}^n_{++}$, is given by
$$f^*(y) = \begin{cases} 0 & , \sum_{i=1}^n e^{y_i} \leq 1 \\ \infty & , \text{otherwise.} \end{cases}$$

5. (10 points) Derive the proximal mapping of euclidean norm(ℓ_2 norm) by definition:
$$f(\mathbf{x}) = \|\mathbf{x}\|_2,$$
we have
$$\mathrm{Prox}_{tf}(\mathbf{x}) = \begin{cases} \left(1 - \frac{t}{\|\mathbf{x}\|_2}\right)\mathbf{x} & , \|\mathbf{x}\|_2 > t \\ 0 & , \text{otherwise} \end{cases}$$

6. (10 points) In this problem, we are going to analyze the convergence rate of the gradient descent method with *backtracking line search* for smooth and convex functions. Recall that in the gradient descent method that we learned in class, the stepsize during each iteration is either fixed ($\eta_t = 1/L$) or predetermined ($\eta_t = 1/\sqrt{t}$). Now, we want to choose stepsize dynamically such that in each iteration, the objective function f is approximately minimized along the descent direction. Backtracking line search is a simple and effective way to do that. The full details of backtracking line search is given in Algorithm 1. And the gradient descent algorithm with backtracking line search is demonstrated in Algorithm 2.

Algorithm 1 Backtracking Line Search

Input: constants $\beta \in (0, 1)$; current iterate $\mathbf{x}$, descent direction $\Delta\mathbf{x}$
Initialize: $\eta = \eta_0 = 1$
while $f(\mathbf{x} + \eta\Delta\mathbf{x}) > f(\mathbf{x}) + 1/2\eta\nabla f(\mathbf{x})^\top \Delta\mathbf{x}$ **do**
    $\eta = \beta\eta$
end while

Then we have the following gradient descent with backtracking line search algorithm.

Algorithm 2 Gradient Descent with Backtracking Line Search

Initialize: $\mathbf{x}_1 \in \operatorname{dom} f$
for $t = 1$ to $T - 1$ **do**
    $\Delta\mathbf{x} = -\nabla f(\mathbf{x})$
    determine η_t by backtracking line search in Algorithm 1
    $\mathbf{x}_{t+1} = \mathbf{x}_t + \eta_t\Delta\mathbf{x}$
end for

Now suppose that f is convex and L-smooth, prove that the gradient descent with backtracking line search satisfies

$$f(\mathbf{x}_t) - f(\mathbf{x}^*) \leq \frac{\|\mathbf{x}_0 - \mathbf{x}^*\|_2^2}{2t\eta_{\min}},$$

where $\eta_{\min} = \min(\eta_0, \beta/L)$.

Hint: Modify the proof of Theorem 1 in Lecture 14.

7. (20 points) In this problem and the next problem, you will be asked to implement proximal gradient descent method with *backtracking line search*. It is not the same as the backtracking line search introduced in problem 1. Consider the minimization problem as follows

$$\min_{\mathbf{x}} f(\mathbf{x}) + h(\mathbf{x}),$$

where $f(\mathbf{x})$ is smooth and convex, and $h(\mathbf{x})$ is convex and non-differentiable. The full details of backtracking line search is given in Algorithm 3. And the proximal gradient descent algorithm with backtracking line search is demonstrated in Algorithm 4.

Algorithm 3 Backtracking Line Search for Proximal Gradient Descent

Input: constant $\beta \in (0,1)$; current iterate $\mathbf{x}$, descent direction $\Delta\mathbf{x}$
Initialize: $\eta = 1$
$\mathbf{x}^+ = \text{Prox}_{\eta \cdot h}(\mathbf{x} + \eta\Delta\mathbf{x})$
while $f(\mathbf{x}^+) > f(\mathbf{x}) + \nabla f(\mathbf{x})^\top(\mathbf{x}^+ - \mathbf{x}) + 1/(2\eta)\|\mathbf{x}^+ - \mathbf{x}\|_2^2$ **do**
    $\eta = \beta\eta$
end while

Algorithm 4 Proximal Gradient Descent with Backtracking Line Search

Initialize: $\mathbf{x}_1 \in \text{dom}\, f$
for $t = 1$ to $T - 1$ **do**
    $\Delta\mathbf{x} = -\nabla f(\mathbf{x})$
    determine η_t by backtracking line search in Algorithm 3
    $\mathbf{x}_{t+1} = \text{Prox}_{\eta_t \cdot h}(\mathbf{x}_t + \eta_t\Delta\mathbf{x})$
end for

Now let's consider the Lasso problem as follows:

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^d} \frac{1}{2n}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1,$$

where $\mathbf{y} \in \mathbb{R}^n, \mathbf{X} \in \mathbb{R}^{n\times d}$ and $\lambda > 0$.

(a) (5 points) Implement gradient descent algorithm with backtracking line search (see Algorithm 1) in *LassoGD_LS.m*. Set $\beta = 0.9$. (posted on Collab).

(b) (5 points) What is the generalized gradient mapping in the proximal gradient descent from $\mathbf{x}_t$ to $\mathbf{x}_{t+1}$?

(c) (5 points) Implement proximal gradient descent algorithm with backtracking line search (see Algorithm 4) in *LassoPGD_LS.m*. Set $\beta = 0.9$. (posted on Collab).

(d) (5 points) Write a Matlab script *test_Lasso.m* in Matlab to call *LassoGD_LS.m*, *LassoPGD_LS.m*, *LassoGD.m* and *LassoCVX.m* (Remind that *LassoGD.m* and *LassoCVX.m* have been implemented in homework 3), and apply them to the data stored in *LassoData.mat*, which contains $\mathbf{X}, \mathbf{y}$ and the true parameter $\boldsymbol{\beta}^*$. Set the penalty parameter $\lambda = 0.1$. Set the total number of iterations $T = 200$. For *LassoGD.m*, the step size is set to $\eta = 0.1$. Plot $\|\boldsymbol{\beta}_t - \widehat{\boldsymbol{\beta}}\|_2$ vs. t for gradient descent, gradient descent with backtracking line search and proximal gradient descent with backtracking line search in one figure, where $\boldsymbol{\beta}^t$ is the t-th iterate in gradient descent algorithm. Make another plot of $|f(\boldsymbol{\beta}_t) - f(\widehat{\boldsymbol{\beta}})|$ versus t for the three algorithms in another figure.
Show the two figures in your homework.

Note: the Matlab files you need to submit for this problem is *LassoGD_LS.m, LassoPGD_LS.m, test_Lasso.m*.

8. (20 points) Consider the regularized problem for logistic regression:

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n}\big[-y_i\cdot\mathbf{x}_i^\top\boldsymbol{\beta}+\log\big(1+\exp(\mathbf{x}_i^\top\boldsymbol{\beta})\big)\big]+\lambda\|\boldsymbol{\beta}\|_1,$$

where $\mathbf{x}_i\in\mathbb{R}^d, y_i\in\{0,1\}$ and $\lambda>0$.

(a) (5 points) Write a function *LogisticGD_LS.m* to implement gradient descent algorithm with backtracking line search (see Algorithm 1). Set $\beta=0.9$.

(b) (5 points) What is the generalized gradient mapping in the proximal gradient descent from $\mathbf{x}_t$ to $\mathbf{x}_{t+1}$?

(c) (5 points) Write a function *LogisticPGD_LS.m* to implement proximal gradient descent algorithm with backtracking line search (see Algorithm 4) . Set $\beta=0.9$.

(d) (5 points) Write a Matlab script *test_Logistic.m* in Matlab to call *LogisticGD_LS.m*, *LogisticPGD_LS.m*, *LogisticGD.m* and *LogisticCVX.m* (Remind that *LogisticGD.m* and *LogisticCVX.m* have been implemented in homework 3), and apply them to the data stored in *LogisticData.mat*, which contains $\mathbf{X},\mathbf{y}$ and the true parameter $\boldsymbol{\beta}^*$. Set the penalty parameter $\lambda=0.1$. Set the total number of iterations $T=500$. For *LogisticGD.m*, the step size is set to $\eta=0.1$. Plot $\|\boldsymbol{\beta}_t-\widehat{\boldsymbol{\beta}}\|_2$ vs. t for gradient descent, gradient descent with backtracking line search and proximal gradient descent with backtracking line search in one figure, where $\boldsymbol{\beta}^t$ is the t-th iterate in gradient descent algorithm. Make another plot of $|f(\boldsymbol{\beta}_t)-f(\widehat{\boldsymbol{\beta}})|$ versus t for these three algorithms in another figure.
Show the two figures in your homework.

Note: the Matlab files you need to submit for this problem is *LogisticGD_LS.m, LogisticPGD_LS.m, test_Logistic.m.*