Homework 5
ADMM, Primal-dual interior point
Dual Theory, Dual ascent

CMU 10-725/36-725: Convex Optimization (Fall 2017)

OUT: Nov 4
DUE: **Nov 18, 11:59 PM**

START HERE: Instructions

- **Collaboration policy:** Collaboration on solving the homework is allowed, after you have thought about the problems on your own. It is also OK to get clarification (but not solutions) from books or online resources, again after you have thought about the problems on your own. There are two requirements: first, cite your collaborators fully and completely (e.g., "Jane explained to me what is asked in Question 3.4"). Second, write your solution independently: close the book and all of your notes, and send collaborators out of the room, so that the solution comes from you only.
- **Submitting your work:** Assignments should be submitted as PDFs using Gradescope unless explicitly stated otherwise. Each derivation/proof should be completed on a separate page. Submissions can be handwritten, but should be labeled and clearly legible. Else, submissions can be written in LaTeX. Upon submission, label each question using the template provided by Gradescope. Please refer to Piazza for detailed instruction for joining Gradescope and submitting your homework.
- **Programming**: All programming portions of the assignments should be submitted to Gradescope as well. We will not be using this for autograding, meaning you may use any language which you like to submit.

1 Primal-Dual Interior Points Methods (20pts) [Devendra]

A network is described as a directed graph with m arcs or links. The network supports n flows, with nonnegative rates $x_1, ..., x_n$. Each flow moves along a fixed, or pre-determined, path or route in the network, from a source node to a destination node. Each link can support multiple flows, and the total traffic on a link is the sum of the rates of the flows that travel over it. The total traffic on link i can be expressed as $(Ax)_i$, where $A \in R^{m \times n}$ is the flow-link incidence matrix defined as

$$A_{ij} = \begin{cases} 1 & \text{when flow } j \text{ passes through link } i \\ 0 & \text{otherwise} \end{cases}$$

Usually each path passes through only a small fraction of the total number of links, so the matrix A is sparse. Each link has a positive capacity, which is the maximum total traffic it can handle. These link capacity constraints can be expressed as $Ax \leq b$, where b_i is the capacity of link i. We consider the network rate optimization problem (in which matrix A is given, so that we are not optimizing over where each flow passes through)

$$\begin{gathered} \text{maximize } f_1(x_1) + \cdots + f_n(x_n) \\ \text{subject to } Ax \leq b \\ x \geq 0 \end{gathered}$$

where

$$f_k(x_k) = \begin{cases} x_k & x_k < c_k \\ (x_k + c_k)/2 & x_k \geq c_k \end{cases}$$

and $c_k > 0$ is given. In this problem we choose feasible flow rates x_k that maximize a utility function $\sum_k f_k(x_k)$.

(a) **[6pt]** Express the network rate optimization problem as a linear program in inequality form.

(b) **[14pt]**Write a custom implementation of the primal-dual interior point method for the linear program in part (a). Give an efficient method for solving the linear equations that arise in each iteration of the algorithm. Justify briefly your method, assuming that m and n are large, and that the matrix $A^T A$ is sparse.

2 Duality for Support Vector Regression (Yichong; 25 pts)

In the last homework, we used barrier methods to solve support vector regression. This time we look at SVR from a different point of view. Recall that for a set of data points $(X_1, Y_1), ..., (X_n, Y_n)$ with $X_i \in \mathbb{R}^d$ and $Y_i \in \mathbb{R}$ for all i,an ε-SVR solves the convex optimization problem

$$\begin{aligned} \min_{w \in \mathbb{R}^{d+1}, \xi \in \mathbb{R}^n} f(w, \xi) &= \frac{1}{2}\|w_{1:d}\|_2^2 + C\sum_{i=1}^{n} \xi_i \\ \text{s.t. } Y_i - w^T X_i &\leq \varepsilon + \xi_i, \\ w^T X_i - Y_i &\leq \varepsilon + \xi_i, \\ \xi_i \geq 0, i &= 1, 2, ..., n. \end{aligned}$$

For simplicity, we append a 1 to the end of each X_i so that we don't need to consider bias. Here ε and C are parameters of SVR, and $w_{1:d}$ is the first d dimension of w.

(a) **[5pt]** Derive the KKT conditions for SVR. Use $u_i, i = 1, 2, ..., n$ for the first set of constraints, and u'_i, v_i for the second and third one.

(b) **[5pt]** Derive the dual program for SVR. Simplify your program so that v_i does not appear in the program.

(c) **[5pt]** Describe how to derive the primal solution w using the dual solution $u_i^*, u_i'^*$. Remember also to show how to calculate the intercept (the $d+1$-th dimension of w).
Hint: It is possible that you cannot obtain a unique solution for the intercept.

(d) **[10pt]** Without a proof, figure out what is the location of (X_i, Y_i) with respect to the margin $\{(x, y) : |y - w^T X_i| \leq \varepsilon\}$ in the following cases, when w is the optimal solution?

i) $u_i^* = u_i'^* = 0$;

ii) $u_i^* \in (0, C)$;

iii) $u_i'^* \in (0, C)$;

iv) $u_i^* = C$;

v) $u_i'^* = C$.

3 Dual Ascent [Hongyang; 25 pts]

The key to the success of dual ascent method is the so-called strong duality, namely, the primal problem and the dual problem have exactly the same optimal objective function value. Let us see a simple example:

$$\min_x x^2 + 1, \quad \text{s.t.} \quad (x-2)(x-4) \leq 0,$$

with variable $x \in \mathbb{R}$.

(a) **[5pt]** *Analysis of primal problem.* Give the feasible set, the optimal value, and the optimal solution.

(b) **[15pt]**Lagrangian and dual function.

i) Plot the objective $x^2 + 1$ versus x. On the same plot, show the feasible set, optimal point and value, and plot the Lagrangian $L(x, \lambda)$ versus x for a few positive values of λ.

ii) Verify the lower bound property ($p^* \geq \inf_x L(x, \lambda)$ for $\lambda \geq 0$).

iii) Derive and sketch the Lagrange dual function g.

(c) **[5pt]** Lagrange dual problem. State the dual problem, and verify that it is a concave maximization problem. Find the dual optimal value and dual optimal solution λ^*. Does strong duality hold?

4 Quantile regression and ADMM [Yifeng; 30 pts]

Quantile regression is a way to estimate the conditional α-quantile, for some $\alpha \in [0, 1]$ fixed by the implementer, of some response variable Y given some predictors $X_1, \ldots, X_p$. (Just to remind you: the α-quantile of $Y|X_1, \ldots, X_p$ is given by $\inf_t \{t : \text{Prob}(Y \leq t|X_1, \ldots, X_p) \geq \alpha\}$.) It turns out that we can estimate the α-quantile by solving the following (convex) optimization problem:

$$\min_{\beta \in \mathbb{R}^p} \quad \sum_{i=1}^n \ell_i^{(\alpha)} (y - X\beta), \tag{1}$$

where $y \in \mathbb{R}^n$ is a response vector, $X \in \mathbb{R}^{n \times p}$ is a data matrix, and the map $\ell^{(\alpha)} : \mathbb{R}^n \to \mathbb{R}^n$ is the *quantile loss*, defined as

$$\ell_i^{(\alpha)}(a) = \max\{\alpha a_i, (\alpha - 1)a_i\}, \quad i = 1, \ldots, n,$$

for some $a \in \mathbb{R}^n$. (It is not too hard to show why this is the case, but you can just assume it is true for this question.) In Figure 1, we plot some estimated α-quantiles, for a few values of α, obtained by solving problem (1) on some synthetic data.

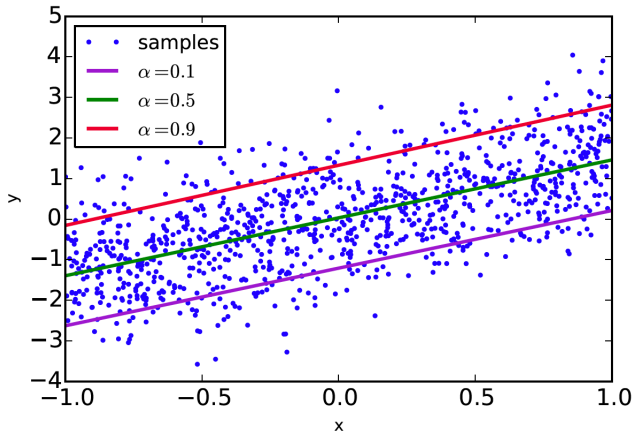

Figure 1: Various quantiles estimated from synthetic data.

We often want to estimate multiple quantiles simultaneously, but would like them to not intersect: e.g., in Figure 1, we would like to *avoid* the situation where the 0.1-quantile line (maybe eventually) lies above the 0.9-quantile line, since this doesn't make any sense. To do so, we can instead solve the following (again convex) optimization problem, referred to as the *multiple quantile regression problem with non-crossing constraints*:

$$\begin{array}{ll} \min_{B \in \mathbb{R}^{p \times r}} & \sum_{i=1}^n \sum_{j=1}^r \ell_{ij}^{(\mathcal{A})} \left(y 1^T - XB\right) \\ s.t. & XBD^T \geq 0. \end{array} \tag{2}$$

Here, r is the number of α's (i.e., quantile levels); $\mathcal{A} = \{\alpha_1, \ldots, \alpha_r\}$ is a set of user-defined quantile levels; y, X are as before; 1 is the r-dimensional all-ones vector; and $B \in \mathbb{R}^{p \times r}$ is now a parameter *matrix*, structured as follows:

$$B = \begin{bmatrix} | & & | \\ \beta_1 & \cdots & \beta_r \\ | & & | \end{bmatrix},$$

for $\beta_i \in \mathbb{R}^p$, $i = 1, \ldots, r$. Here, we have also extended the definition of the quantile loss: it is now a map $\ell^{(\mathcal{A})} : \mathbb{R}^{n \times r} \to \mathbb{R}^{n \times r}$, defined as

$$\ell_{ij}^{(\mathcal{A})}(A) = \max\left\{\alpha_j A_{ij}, (\alpha_j - 1) A_{ij}\right\}, \quad i = 1, \ldots, n, \; j = 1, \ldots, r.$$

Lastly, $D \in \mathbb{R}^{(r-1) \times r}$ is the first-order finite differencing operator, given by

$$D = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 \end{bmatrix}.$$

Hence, you can convince yourself that the constraints in (2) prevent the estimated quantiles from crossing.

Now, finally, for the problem.

(a, 5pts) Derive the prox operator for the quantile loss $\ell_{ij}^{(\mathcal{A})}$ (defined above). In other words, write down an expression for $\text{prox}_{\ell_{ij}^{(\mathcal{A})}, \lambda}(A)$, where $\lambda > 0$ is a user-defined constant, and $A \in \mathbb{R}^{m \times n}$ is the point at which the prox operator is evaluated.

(b, 10pts) We can put problem (2) into "ADMM form" as follows:

$$\begin{aligned} \min_{\substack{B\in\mathbb{R}^{p\times r},\\ Z_1,Z_2\in\mathbb{R}^{n\times r},\\ Z_3\in\mathbb{R}^{n\times(r-1)}}} \quad & \sum_{i=1}^{n}\sum_{j=1}^{r} \ell_{ij}^{(\mathcal{A})}(Z_1) + I_+(Z_3) \\ s.t. \quad & Z_1 = y1^T - XB, \quad Z_2 = XB, \quad Z_3 = Z_2 D^T \end{aligned} \tag{3}$$

where we introduced the variables Z_1, Z_2, Z_3, and $I_+(\cdot)$ is the projection map onto the nonnegative orthant.

The augmented Lagrangian associated with problem (3) is then (in "scaled" form)

$$\begin{aligned} L_\rho(B, Z_1, Z_2, Z_3, U_1, U_2, U_3) = \sum_{i=1}^{n}\sum_{j=1}^{r} \ell_{ij}^{(\mathcal{A})}(Z_1) + I_+(Z_3) + \frac{\rho}{2}\Big(\|y1^T - XB - Z_1 + U_1\|_F^2 + \\ \|XB - Z_2 + U_2\|_F^2 + \|Z_2 D^T - Z_3 + U_3\|_F^2 - \|U_1\|_F^2 - \|U_2\|_F^2 - \|U_3\|_F^2\Big), \end{aligned}$$

where U_1, U_2, U_3 are dual variables.

Here are the updates steps of ADMM in our question:

For $k = 0, 1, 2, ...,$

$$\begin{aligned} B^{(k+1)} &= \operatorname*{argmin}_B L_\rho(B, Z_1^{(k)}, Z_2^{(k)}, Z_3^{(k)}, U_1^{(k)}, U_2^{(k)}, U_3^{(k)}); \\ Z_1^{(k+1)} &= \operatorname*{argmin}_{Z_1} L_\rho(B^{(k+1)}, Z_1, Z_2^{(k)}, Z_3^{(k)}, U_1^{(k)}, U_2^{(k)}, U_3^{(k)}); \\ Z_2^{(k+1)} &= \operatorname*{argmin}_{Z_2} L_\rho(B^{(k+1)}, Z_1^{(k+1)}, Z_2, Z_3^{(k)}, U_1^{(k)}, U_2^{(k)}, U_3^{(k)}); \\ Z_3^{(k+1)} &= \operatorname*{argmin}_{Z_3} L_\rho(B^{(k+1)}, Z_1^{(k+1)}, Z_2^{(k+1)}, Z_3, U_1^{(k)}, U_2^{(k)}, U_3^{(k)}); \\ U_1^{(k+1)} &= U_1^{(k)} + y1^T - XB^{(k+1)} - Z_1^{(k+1)}; \\ U_2^{(k+1)} &= U_2^{(k)} + XB^{(k+1)} - Z_2^{(k+1)}; \\ U_3^{(k+1)} &= U_3^{(k)} + Z_2^{(k+1)} D^T - Z_3^{(k+1)}. \end{aligned}$$

Please provide the explicit update formula for $B^{(k+1)}, Z_1^{(k+1)}, Z_2^{(k+1)}, Z_3^{(k+1)}$. You can assume that the quantity $X^T X$ is invertible.

(c, 15 pts) Implement your ADMM algorithm from part (b) for the quantiles $\mathcal{A} = \{0.1, 0.5, 0.9\}$, using synthetic data from hw5q4.zip which contain a predictor matrix `X`, response vector `y`, and in csv files. Note that the predictor matrix in this example data is 1000×2, with the 2nd column being the column of all 1s, hence providing an intercept term in the quantile regression.

Set initial values for all primal and dual variables as zero matrices of appropriate size, use an augmented Lagrangian parameter of $\rho = 1$, and run ADMM for 50 steps. Report your optimal criterion ($\sum_{i=1}^{n}\sum_{j=1}^{r} \ell_{ij}^{(\mathcal{A})}(Z_1)$) value after 50 steps, and plot the optimal criterion value across the iterations. Recreate the plot similar to Figure 1, plotting data as points and overlaying as lines the *conditional quantile* predictions for the quantiles in $\mathcal{A}$. (Hint: recall that B is a matrix whose r columns are coefficient estimates for each quantile level $\alpha_1, \cdots, \alpha_r$, so that $X_{new}^T B$ at a new data point $X_{\text{new}} = (x_{\text{new}}, 1)$ is the fitted estimates for each quantile in $\mathcal{A}$.)