# 10725/36725 Optimization
# Homework 4

Due November 27, 2012 at beginning of class

**Instructions:** There are four questions in this assignment. Please submit your homework as (up to) 4 separate sets of pages with your name and userid on each set. For the last question which involves coding, please print out your code and graphs and attach them to the written part of your homework and email your code to submission10725f12@gmail.com. Refer to the course webpage for policies regarding collaboration, due dates, and extensions.

**New:** For this problem set, you have a choice: you may submit answers for *any two* of the first three questions (questions 1, 2, and 3). You may choose to submit answers for all questions; if you do, we'll give you credit for the highest two scores out of the first three. All students must submit question 4 (the implementation question). (In the feedback survey, some students indicated that they like theoretical questions, and others indicated that they dislike theoretical questions—so, this two-of-three policy will let you choose which sorts of questions you prefer to work on. And of course it also helps to make the problem set shorter.)

## 1 Save our souls, dual cone [Shiva, 25 points]

Non-polyhedral dual cones are kind of tricky. Let's prove a tough-looking lemma about polynomials[1] by thinking about what the dual cone represents.

Let $V$ be the vector space of polynomials with a basis of monomials. For example, the polynomial $p(x) = x_1^3 + 2x_1^3x_2^2 + 9x_1x_2^2$ has coefficients $1, 2, 9$ in the coordinates corresponding to $x_1^3$, $x_1^3x_2^2$, and $x_1x_2^2$, respectively. The usual inner product between two polynomials $p$ and $q$ is $\langle p, q\rangle = \sum_\alpha p_\alpha q_\alpha$ where $\alpha$ indexes the monomials. The lemma is:

Lemma: if $\Sigma \subsetneq P$, then there is a $p \in P$ such that $Z(p) \neq Z(q)$ for any $q \in \Sigma$.

[1] This technical lemma is used in optimization to show that certain convex relaxations are tight.

Here, $P \subset V$ are nonnegative polynomials, i.e., $p \in P$ satisfies $p(x) \geq 0$ for all $x$. $\Sigma \subset P$ are sums-of-squares, i.e., $s \in \Sigma$ can be written as $s(x) = \sum_i p_i(x)^2$ for $p_i \in P$. Finally, the zero set of $p$ is $Z(p) = \{x : p(x) = 0\}$. So the lemma says: if there's a nonnegative polynomial that isn't a sum-of-squares, then there's a nonnegative polynomial whose zero set isn't shared by a sum-of-squares.

Before you crack open your analysis textbook, note that $P$ and $\Sigma$ are convex cones.

(a) [3 points] Prove it.

Recall the definition of the dual cone of $P$:

$$P^* = \{l : \langle l, p \rangle \geq 0 \ \forall p \in P\}$$

Also recall that a polynomial is nonnegative iff it is nonnegative on the unit sphere[2].

(b) [5 points] Interpret that fact by identifying a subset $B \subset P^*$ whose elements correspond to points on the unit sphere and whose conical hull is $P^*$.

Geometrically, $P^*$ consists of hyperplanes $l$ in which all of $P$ is on one side of the hyperplane. For some $l$, the $p$ on the boundary (i.e. the $p$ satisfying $\langle l, p \rangle = 0$) are said to be exposed by $l$.

(c) [7 points] If $R$ is exposed by $l$, what is the relationship among $\{Z(p) : p \in R\}$? Use $B$ (from the previous part of this question) in your proof.

(d) [5 points] Are there $p \in R, q \in P \backslash R$ such that $Z(p) = Z(q)$?

Let's assemble the proof. If $P$ is strictly larger than $\Sigma$, then there is an extreme ray of $P$ that is not fully contained in $\Sigma$. Since $P$ and $\Sigma$ are closed convex sets, we can strengthen the previous statement using Straszewicz's theorem: there is an exposed extreme ray $R$ of $P$ that is not in $\Sigma$.

(e) [5 points] Complete the proof in English.

# 2 Projecting onto the L1-ball [Kevin, 25 points]

Some students indicated that they would prefer to try solving problems without us providing all of the intermediate steps, while others indicated that they prefer having the steps specified. If you would like additional direction for solving this problem, download `http://www.cs.cmu.edu/~ggordon/10725-F12/hws/hw4/q2full.pdf` and solve the expanded version of the problem written there *instead of* this version. The two versions of the problem are worth the same number of points. Please only submit the no-intermediate-steps version (below) if you are relatively confident that you have the entire answer close to correct: it's hard for us to

[2] Actually, this is true only for *homogenous* polynomials, whose constituent monomials all have the same degree. Every polynomial can be converted to a homogenous polynomial.

try to decipher an incorrect proof to award partial credit and we will be conservative about doing so.

Devise an efficient algorithm for projecting a vector $y \in \mathbb{R}^n$ onto the unit $L1$-ball, that is, to solve the optimization problem

$$x^* = \arg\min_x \ \|x - y\|_2^2/2, \quad \text{subject to:} \tag{1}$$

$$\|x\|_1 \leq 1. \tag{2}$$

Your algorithm must not be iterative, that is, it must exactly compute the projection up to machine precision. Be sure to prove that your algorithm is correct and describe its asymptotic complexity in $O$-notation.

# 3 Quadratically constrained quadratic programming (Wooyoung, 25 pts)

1 Consider the following optimization problem,

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\mathbf{x}^T P\mathbf{x} + \mathbf{q}^T\mathbf{x} + r \\ \text{subject to} \quad & \mathbf{x}^T\mathbf{x} \leq 1 \end{aligned}$$

where $\mathbf{x}, \mathbf{q} \in \mathcal{R}^n, P \in \mathcal{S}_{++}^n$

(a) [3 pts] Show using the KKT condition that $\mathbf{x}$ is optimal if and if only

$$\mathbf{x}^T\mathbf{x} < 1, \quad P\mathbf{x} + \mathbf{q} = 0 \tag{3}$$

or

$$\mathbf{x}^T\mathbf{x} = 1, \quad P\mathbf{x} + \mathbf{q} = -\lambda\mathbf{x} \,(\text{for some } \lambda \geq 0) \tag{4}$$

To solve the problem, let's start from $\mathbf{x} = -P^{-1}\mathbf{q}$. If the solution has norm less than or equal to one ($||P^{-1}\mathbf{q}||_2 \leq 1$), it's optimal according to Eq.(3). If the solution of $P\mathbf{x} = -\mathbf{q}$ has $L2$-norm greater than 1, that means the optimal solution should satisfy Eq.(4).

(b) [2 pts] Show that $P + \lambda I \succ 0$ for $\lambda \geq 0$.

Define

$$f(\lambda) = ||(P + \lambda I)^{-1}\mathbf{q}||_2^2$$

(c) [3 pts] Show that there exists a non-negative $\lambda$ such that $f(\lambda) = 1$.

(d) [2 pts] Express the optimal solution $\mathbf{x}$ as a function of $\lambda^\star$ which satisfies $f(\lambda^\star) = 1$.

(e) [2 pts] We covered two types of plausible optima's: one that lies on the norm ball (Eq.3) and the other that lies inside the ball (Eq.4). Discuss the geometric interpretation of each case.

2 Minimizing a linear function over an ellipsoid centered at the origin.

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}^T\mathbf{x} \\ \text{subject to} \quad & \mathbf{x}^T A\mathbf{x} \leq 1 \end{aligned}$$

where $A \in \mathcal{S}^n_{++}$ and $\mathbf{c} \neq 0$.

(a) [2 pts] Rewrite the constraint as a function of a new variable $\mathbf{y}$ to remove $A$ from the constraint. *Hint:* Use the fact that $A$ is a positive semidefinite matrix.

(b) [1 pts] Rewrite the objective function as a function of the new variable $\mathbf{y}$.

(c) [2 pts] Solve the new optimization function with the new constraint and the new objective function. Show your work and the optimal solution.

Let's now consider a more general case where $A$ is symmetric but not necessarily positive definite.

(d) [2 pts] Express A in terms of its eigen-decomposition, calling its eigenvalues $\lambda_i$ and eigenvectors $\mathbf{q}_i, (i = 1, \cdots, n)$.

(e) [1 pts] Define a new variable $\mathbf{z} = Q^T\mathbf{x}$, such that each column of Q corresponds to $\mathbf{q}_i$. Rewrite the constraint as a function of $\mathbf{z}$ and $\lambda_i \, (i = 1, \cdots, n)$.

(f) [1 pts] Define a new vector $\mathbf{b} = Q^T\mathbf{c}$. Rewrite your original objective function as a function of $\mathbf{b}$ and $\mathbf{z}$.

(g) [2 pts] Show that the optimization problem is unbounded below if one of the eigenvalues of $A$, $\lambda_n$ is negative, $\lambda_n < 0$.

(h) [2 pts] Show that the optimization problem is unbounded below if $\lambda_i = 0$ $b_i \neq 0$ for some $i$.

# 4 AADI-MM [40 points]

There are many subtleties involved when trying to use ADMM to solve the lasso, as discussed in class ("Lasso Using Repeated Ridge"). When do we terminate? How sensitive is convergence to the penalty parameter $\rho$? What's the fastest way to perform inversion in update steps? Is it easy to calculate the regularization path for multiple $\lambda$ values?

**Data and Problem Setup** Download the file "A.txt" to get a dense $1500 \times 5000$ (short and fat) matrix $A$ and the file "b.txt" for the output vector of size 1500. Assume that $b = Ax + e$ where $e$ is some independent random noise and $x$ is an unknown sparse vector. We are going to solve for $x$ using the lasso setup: $\min_x (1/2)\|Ax - b\|_2^2 + \lambda \|x\|_1$.

**[3 points]** In 2 lines, argue that for the lasso problem, a choice of $\lambda = \|A^\top b\|_\infty$ will lead to the solution $x^* = 0$.

**ADMM setup** We reframe the lasso as $\min_{x,z} (1/2)\|Ax - b\|_2^2 + \lambda \|z\|_1$ such that $x = z$. We then use a dual variable $\rho u$ to set up the augmented Lagrangian $L_\rho(x, z, u) = (1/2)\|Ax - b\|_2^2 + \lambda \|z\|_1 + \rho u^\top (x - z) + (\rho/2)\|x - z\|_2^2$.

**[5 points]** In 3 lines, derive the ADMM updates for $x, z, u$, whose final answer is on the ADMM slides. (So, you will be graded on the derivation rather than the final answer.)

**Fast Inverses** Notice that the update step for $x$ looks like the result of a ridge regression, and the inverse is computed at every iteration. Assume that we have to calculate $(\rho I + A^\top A)^{-1} y_t$ at every step $t$. We shall discuss three methods to do this.

**Naive, Cool, Awesome** The first is just to use a direct matrix inverse (called inv in MATLAB) and store it, so that we can do a matrix-vector multiplication at every step. The second is again to store the inverse and use a matrix-vector multiply at every step, but to use the Woodbury matrix identity to calculate the inverse $(\rho I + A^\top A)^{-1} = (1/\rho) I - (1/\rho^2) A^\top (I + (1/\rho) A A^\top)^{-1} A$ so that we have to calculate the inverse of a smaller matrix. The third is to use the Woodbury matrix identity and cache the Cholesky decomposition of the positive definite matrix $(I + (1/\rho) A A^\top)$ so that at every iteration, multiplying by $y$ is solving a linear system twice and two matrix-vector multiplications (implemented by "\" or linsolve in MATLAB).

**[10 points]** Download a new file "y.txt" having a vector of size 5000, and try all three methods for the given $A$ and $y$, assuming $\rho = 1$. Report the time taken for each of the three methods (for example, use tic and toc in MATLAB).

**Termination** Define the primal residual at step $t$ to be $r_t = \|x_t - z_t\|_2$ and the dual residual at step $t$ to be $s_t = \rho \|z_{t+1} - z_t\|_2$. We shall terminate when both residuals are smaller than some tolerance parameter $\epsilon$. (This condition was not discussed in class, but is a result of the proof of convergence of ADMM.)

**[4 points]** Create a function "ADMMLasso($A, b, x_0, \lambda, \rho$)" where $x_0$ is the initial guess and implement the above derived ADMM algorithm. Choose $\lambda = 0.1 * \|A^\top b\|_\infty$, $\rho = 1$, $x_0 = \vec{0}$ and run the algorithm for $T = 100$ steps (irrespective of $r_t, s_t$). Submit your code for this part.

**[10 points]** Plot one graph with both the residuals $\log_{10}(r_t)$ (in blue) and $\log_{10}(s_t)$ (in red) on the y-axis, against the iteration/step number on the x-axis. Fix the scales to be $[-8, 2]$ for the y-axis and $[1, 100]$ for the x-axis. Now, repeat this procedure for $\rho = 0.1$ and $\rho = 10$. Which residual converges faster for large $\rho$ and which one for small $\rho$? This should match your intuition about giving a larger or smaller penalty to the $\|x - z\|_2^2$ term. Useful Matlab commands are plot, legend, xlabel, ylabel, axis. Please submit three graphs, with the same scaling, with the same color scheme as mentioned.

**Regularization Path** We will now generate a regularization path for $x^*$. The regularization path for a particular element $x_i^*$ is a graph with the $x$-axis having increasing $\lambda$ values and the $y$-axis having the value of $x_i^*$ for the lasso solution solved at the different $\lambda$s. We shall let the $\lambda$ range from $\lambda_{min}$ to $\lambda_{max}$ in small steps, thus giving a path for $x_i^*$ from a possibly non-zero value (at small $\lambda$) to a possibly zero value (at higher $\lambda$).

**[5 points]** Define a function RegPath($A, b, \lambda_{min}, \lambda_{max}, \rho, \epsilon$) which iteratively runs subroutines of ADMMLasso for different values of $\lambda$ and warm-starting $x_0$ for new $\lambda$ with the solution $x^*$ for the previous $\lambda$. Let $\lambda_{min} = 0.001$ and $\lambda_{max} = 0.99\|A^\top b\|_\infty$ with twenty logarithmic steps (i.e. $\lambda_t = \lambda_{min} m^t$ where $\lambda_{max} = \lambda_{min} m^{20}$). Each subroutine of ADMMLasso should be stopped when both $r_t$ and $s_t$ are smaller than $\epsilon = 0.001$ (hence not for exactly 100 steps). On the same graph, plot the regularization path of all 5000 coordinates of $x^*$ (most of them will remain zero anyway). Hence, report one graph for this part.

**[3 points]** On a new graph, plot the number of non-zero coordinates in your solution on the y-axis against the lambdas (the same range from the previous part) on the x-axis. What is the maximum number of non-zero coordinates you found? We learned a theorem in class that if $A$ is drawn from a continuous probability distribution on $\mathbb{R}^{m \times n}$, then $x \in \mathbb{R}^n$ will have at most $\min\{m, n\}$ non-zeros with probability one. Submit one graph for this part.

**DISCLAIMER:** These termination criteria (formulae for primal and dual residuals) are for the Lasso problem only. For termination criteria for more general problems, refer to Boyd's paper (referenced on the slides).

**NOTE:** The Cholesky and matrix inverse method is quite general. For the specific case of ridge regression, one might want to use the kernel regression trick $(X^\top X + \lambda I)^{-1} X^\top y = X^\top (X X^\top + \lambda I)^{-1} y$.