EECS 127/227AT Optimization Models in Engineering
Spring 2020 Homework 12

This homework is due Saturday, April 25, 2020 at 23:00 (11pm).
Self grades are due Friday, April 31, 2020 at 23:00 (11pm).

This version was compiled on 2020-04-27 19:45.

Submission Format: Your homework submission should consist of a single PDF file that contains all of your answers (any handwritten answers should be scanned) as well as your IPython notebook with solutions saved as a PDF.

Note: This concepts covered in this homework are important for the final and you should still read and understand them despite the self-grade not being due till after the final.

1. Newton's Method, Coordinate Descent and Gradient Descent

In this question, we will compare three different optimization methods: Newton's method, coordinate descent and gradient descent. We will consider the simple set-up of unconstrained convex quadratic optimization; i.e we will consider the following problem:

$$\min_{\vec{x}\in\mathbb{R}^d} \vec{x}^\top A\vec{x} - 2\vec{b}^\top\vec{x} + c$$

where $A \succ 0$ and $\vec{b} \in \mathbb{R}^d$.

(a) How many steps does Newton's method take to converge to the optimal solution? Recall that the update rule for Newton's method is given by the equation:

$$\vec{x}_{t+1} = \vec{x}_t - (\nabla^2 f(\vec{x}_t))^{-1}\nabla f(\vec{x}_t).$$

when optimizing a function f.

Solution: Newton's method converges in a single step irrespective of the starting point. Let $\vec{x}_0$ by any starting point. We have:

$$\nabla^2 f(\vec{x}_0) = A \text{ and } \nabla f(\vec{x}_0) = 2(A\vec{x} - \vec{b}).$$

Therefore, we have:

$$\vec{x}_1 = \vec{x}_0 - A^{-1}(A\vec{x} - \vec{b}) = A^{-1}\vec{b}.$$

Note, that since this is an unconstrained convex quadratic optimization problem with A being full rank, we can find the optimum point by setting the derivative of the function to 0. Therefore, we have:

$$\nabla f(\vec{x}^*) = 2(A\vec{x}^* - \vec{b}) = 0 \implies \vec{x}^* = A^{-1}\vec{b}.$$

(b) Now, consider the simple two variable quadratic optimization problem for $\sigma > 0$:

$$\min_{\vec{x} \in \mathbb{R}^2} f(\vec{x}) = \sigma x_1^2 + x_2^2.$$

How many steps does coordinate descent take to converge on this problem? Assume that we start by updating the variable x_1 in the first step, x_2 in step two and so on; therefore, we will update x_1 and x_2 in odd and even iterations respectively:

$$(x_{t+1})_1 = \begin{cases} \text{argmin}_{x_1} f(x_1, (x_t)_2) & \text{for odd t} \\ (x_t)_1 & \text{otherwise} \end{cases} \text{ and } (x_{t+1})_2 = \begin{cases} \text{argmin}_{x_2} f((x_t)_1, x_2) & \text{for even t} \\ (x_t)_2. & \text{otherwise} \end{cases}$$

Here, $(x_t)_2$ represents x_2 at time t and so on.

Solution: On this problem, coordinate descent converges in 2 steps starting from any initialization point. Note that the optimal solution for each of the updates is 0, by setting the gradient to 0. Therefore, coordinate descent converges in two steps, one to update x_1 and the other to update x_2.

(c) We will now analyze the performance of coordinate descent on another quadratic optimization problem:

$$\min_{\vec{x} \in \mathbb{R}^2} f(\vec{x}) = \sigma(x_1 + x_2)^2 + (x_1 - x_2)^2.$$

where we have, as before, $\sigma > 0$. Note that $(0, 0)$ is the optimal solution to this problem. Now, starting from the point $(1, 1)$, how many steps does coordinate descent take to converge to $(0, 0)$. What happens when σ grows large? *Hint: First find the update rule for x_1, i.e. keep x_2 fixed and figure out how x_1 changes when t is odd. Then do the same for x_2 when x_1 is fixed.*

Solution: We first find the update rule for x_1. Note that we only update x_1 when t is odd. Now, by taking the gradient and setting it to 0, we get:

$$\sigma((x_{t+1})_1 + (x_t)_2) + ((x_{t+1})_1 - (x_t)_2) \implies (x_{(t+1)})_1 = \frac{(1-\sigma)}{(1+\sigma)}(x_t)_2.$$

Note that the function, f, is symmetric in the variables, x_1 and x_2. Therefore, the update rule for x_2 (when t is even) is given by:

$$(x_{(t+1)})_2 = \frac{(1-\sigma)}{(1+\sigma)}(x_t)_1.$$

Therefore, we get for all $t \geq 2$:

$$(x_t)_1 = \left(\frac{1-\sigma}{1+\sigma}\right)^{2\left\lfloor \frac{t}{2} \right\rfloor - 1} \text{ and } (x_t)_2 = \left(\frac{1-\sigma}{1+\sigma}\right)^{2\left\lfloor \frac{t-1}{2} \right\rfloor}.$$

When σ grows large, the $\frac{1-\sigma}{1+\sigma}$ goes to -1 and this results in slow convergence as the algorithm converges quickly when $\left|\frac{1-\sigma}{1+\sigma}\right|$ is small.

(d) Finally, for the objective function from the previous part, how long does gradient descent take to converge to $(0,0)$ starting from the point $(1,-1)$? Assume for this part that $\sigma > 1$ and reason about how many steps it takes for gradient descent to converge when σ grows large. *Hint: What is the step size for gradient descent? Also note that f is given by:*

$$f(\vec{x}) = \vec{x}^\top A\vec{x} \text{ where } A = 2\left(\sigma \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} + \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}\right).$$

Solution: We first note that f is given by:

$$f(\vec{x}) = \vec{x}^\top A\vec{x} \text{ where } A = 2\left(\sigma \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} + \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}\right).$$

Therefore, we have that the maximum singular value of A is 2σ and hence the step size for gradient descent is set to $1/(4\sigma)$. Now, we have that:

$$\nabla f((1,-1)) = \begin{bmatrix} 4 \\ -4 \end{bmatrix}.$$

Therefore, we have that:

$$\vec{x}_1 = \vec{x}_0 - \eta \nabla f((1,-1)) = \left(1 - \frac{1}{\sigma}\right)\begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

By iterating the above procedure we see that:

$$\vec{x}_t = \left(1 - \frac{1}{\sigma}\right)^t \begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

Therefore, when σ grows large, the convergence rate of gradient descent is really slow. However, Newton's method would find the optimum in one step.

2. Gradient Descent vs Newton Method

Run the jupyter-notebook 'Gradient_vs_Newton.ipynb' which demonstrates differences between gradient descent and Newton's method.

3. (Optional) Ridge Regression Classifier Vs. SVM

In this problem, we explore Ridge Regression as a classifier, and compare it to SVM. Recall Ridge Regression solves the problem

$$\min_{\vec{w}} \|X\vec{w} - \vec{y}\|_2^2 + \lambda\|\vec{w}\|_2^2,$$

where $X \in \mathbb{R}^{m,n}$, and $\vec{y} \in \mathbb{R}^n$

(a) Ridge Regression as is solves a regression problem. Given data $X \in \mathbb{R}^{m\times n}$ and labels $\vec{y} \in \{0,1\}^m$, explain how we might be able to train a Ridge Regression model and use it to classify a test point.

Solution: We first converts the labels from 0 and 1 to -1 and 1.

We have that the optimal $\vec{w}$ from solving ridge regression is given by

$$\vec{w}^* = (X^\top X + \lambda I)^{-1} X^\top \vec{y} \tag{1}$$

Hence given a new data point x_{test}, we look at $x_{\text{test}}^\top \vec{w}^*$ and if it is positive, we say $y_{\text{test}} = 1$ and otherwise we say $y_{\text{test}} = 0$.

(b) Complete the accompanying Jupyter Notebook to compare Ridge Regression and SVM.

Solution: See Jupyter Notebook for coding solution.

4. Soft-margin SVM

Consider the soft-margin SVM problem,

$$\begin{aligned} p^*(C) = \min_{\vec{w}\in\mathbb{R}^m, b\in\mathbb{R}, \vec{\xi}\in\mathbb{R}^n} \ & \frac{1}{2}\|\vec{w}\|_2^2 + C\sum_{i=1}^n \xi_i \\ \text{s.t. } & 1 - \xi_i - y_i(\vec{x}_i^\top \vec{w} - b) \le 0, \quad i = 1, 2, \ldots, n \\ & -\xi_i \le 0, \quad i = 1, 2, \ldots, n, \end{aligned} \tag{2}$$

where $\vec{x}_i \in \mathbb{R}^m$ refers to the i^{th} training data point, $y_i \in \{-1, 1\}$ is its label, and $C \in \mathbb{R}_+$ (i.e. $C > 0$) is a hyperparameter.

Let α_i denote the dual variable corresponding to the inequality $1 - \xi_i - y_i(\vec{x}_i^\top \vec{w} - b) \le 0$ and let β_i denote the dual variable corresponding to the inequality $-\xi_i \le 0$.

The Lagrangian is then given by

$$\mathcal{L}(\vec{w}, b, \vec{\xi}, \vec{\alpha}, \beta) = \frac{1}{2}\|\vec{w}\|_2^2 + C\sum_{i=1}^n \xi_i + \sum_{i=1}^n \alpha_i(1 - \xi_i - y_i(\vec{x}_i^\top \vec{w} - b)) - \sum_{i=1}^n \beta_i \xi_i.$$

Suppose $\vec{w}^*, b^*, \vec{\xi}^*, \vec{\alpha}^*, \beta^*$ satisfy the KKT conditions.

Classify the following statements as true or false and justify your answers mathematically.

(a) Suppose the optimal solution $\vec{w}^*, b^*$ changes when the training point $\vec{x}_i$ is removed. Then originally, we necessarily have $y_i(\vec{x}_i^\top \vec{w}^* - b^*) = 1 - \xi_i^*$.

Solution: True. Since optimal $\vec{w}^*$ changes if we remove point $\vec{x}_i$ we have $\alpha_i^* \neq 0$. By complementary slackness we have,

$$\alpha_i^*(1 - \xi_i^* - y_i(\vec{x}_i^\top \vec{w}^* - b^*)) = 0,$$

which gives,

$$\begin{aligned} & 1 - \xi_i^* - y_i(\vec{x}_i^\top \vec{w}^* - b^*) = 0 \\ \implies & y_i(\vec{x}_i^\top \vec{w}^* - b^*) = 1 - \xi_i^*. \end{aligned}$$

(b) Suppose the optimal solution $\vec{w}^*, b^*$ changes when the training point $\vec{x}_i$ is removed. Then originally, we necessarily have $\alpha_i^* > 0$.

Solution: True. Since optimal $\vec{w}^*$ changes if we remove point $\vec{x}_i$ we have $\alpha_i^* \neq 0$. Further by dual feasibility we have $\alpha_i^* \ge 0$ which together gives $\alpha_i^* > 0$.

(c) Suppose the data points are strictly linearly separable, i.e. there exist $\vec{w}$ and $\tilde{b}$ such that for all i,

$$y_i(\vec{x}_i^\top \vec{w} - \tilde{b}) > 0.$$

Then $p^*(C) \to \infty$ as $C \to \infty$.

Solution: False.

Since

$$y_i(\vec{x}_i^\top \vec{w} - \tilde{b}) > 0.$$

we have for sufficiently small $\epsilon > 0$,

$$\begin{aligned} & y_i(\vec{x}_i^\top \vec{w} - \tilde{b}) \geq \epsilon \\ \implies & y_i \left(\vec{x}_i^\top \frac{\vec{w}}{\epsilon} - \frac{\tilde{b}}{\epsilon} \right) \geq 1. \end{aligned}$$

Thus, $\vec{w} = \frac{\vec{w}}{\epsilon}$, $\bar{b} = \frac{\tilde{b}}{\epsilon}, \vec{\xi} = 0$ is a feasible point with objective value $\frac{1}{2} \left\| \vec{w} \right\|_2^2 < \infty$ irrespective of value of C.

5. Linear Quadratic Regulator

In this question, we will derive the Riccati equation for the LQR model studied in class. We first recall the statement of the LQR problem:

$$\begin{aligned} \min_{\vec{x}_t, \vec{u}_t} \sum_{t=0}^{N-1} \frac{1}{2} \left(\vec{x}_t^\top Q \vec{x}_t + \vec{u}_t^\top R \vec{u}_t \right) + \frac{1}{2} \vec{x}_N^\top Q \vec{x}_N \\ \text{s.t } \vec{x}_{t+1} = A\vec{x}_t + B\vec{u}_t \\ \vec{x}_0 = \vec{x}_{\text{init}} \end{aligned}$$

where $\vec{x}_t$ is thought of as the state of the system and $\vec{u}_t$ is the control input at time t and the matrices A and B define the dynamics of the system. While the problem can be solved as a quadratic program, we will now take a slightly different approach. We start by defining the functions, J_k for $0 \leq k \leq N$, as follows:

$$\begin{aligned} J_k(\vec{x}) = \min_{\{\vec{u}_t\}_{t=k}^{N-1}} \sum_{t=k}^{N-1} \frac{1}{2} \left(\vec{x}_t^\top Q \vec{x}_t + \vec{u}_t^\top R \vec{u}_t \right) + \frac{1}{2} \vec{x}_N^\top Q \vec{x}_N \\ \text{s.t } \vec{x}_{t+1} = A\vec{x}_t + B\vec{u}_t \\ \vec{x}_k = \vec{x}. \end{aligned}$$

J_k can be thought of as the minimum cost that we would incur from time k assuming that we start at state $\vec{x}_k = \vec{x}$. We can now decompose J_k for $0 \leq k \leq N-1$ further as follows:

$$\begin{aligned} J_k(\vec{x}) = \min_{\vec{u}_k} \frac{1}{2} \left(\vec{x}_k^\top Q \vec{x}_k + \vec{u}_k^\top R \vec{u}_k \right) + \min_{\{\vec{u}_t\}_{t=k+1}^{N-1}} \sum_{t=k+1}^{N-1} \frac{1}{2} \left(\vec{x}_t^\top Q \vec{x}_t + \vec{u}_t^\top R \vec{u}_t \right) + \frac{1}{2} \vec{x}_N^\top Q \vec{x}_N \\ \text{s.t } \vec{x}_{t+1} = A\vec{x}_t + B\vec{u}_t \\ \vec{x}_k = \vec{x}. \end{aligned}$$

Note that in particular, the first constraint implies that $\vec{x}_{k+1} = A\vec{x}_k + B\vec{u}_k$. Therefore, the above characterization gives the following decomposition:

$$J_k(\vec{x}) = \min_{\vec{u}} \frac{1}{2}\left(\vec{x}^\top Q\vec{x} + \vec{u}^\top R\vec{u}\right) + J_{k+1}(A\vec{x} + B\vec{u}). \tag{3}$$

We will see that the functions, J_k, are all in fact quadratic functions in $\vec{x}$ and this will give us convenient ways to derive the optimal control inputs at each time.

(a) First, we will show by reverse induction that each of the functions J_k for $0 \leq k \leq N$ are convex quadratics. In particular, prove that $J_k(\vec{x}) = \vec{x}^\top \frac{Q_k}{2}\vec{x}$ for some $Q_k \succ 0$ and determine the value of Q_k in terms of Q_{k+1}.
Hint 1: $J_N(\vec{x}) = \frac{1}{2}\vec{x}^\top Q\vec{x}$. Therefore, $Q_N = Q$. Also can use (3) above, and substitute $J_{k+1}(A\vec{x} + B\vec{u}) = (A\vec{x} + B\vec{u})^\top Q_{k+1}(A\vec{x} + B\vec{u})^\top$. Then solve the resulting QP to find the optimal $\vec{u}$.
Hint 2: You should get the following recursion for Q_k:

$$Q_k = Q + A^\top Q_{k+1}A - A^\top Q_{k+1}B(R + B^\top Q_{k+1}B)^{-1}B^\top Q_{k+1}A.$$

Solution: We will prove the statement by reverse induction starting with $k = N$ and proving that J_k is a quadratic in the given form for $k = l$ given that it is true for $k = l + 1$. Note that the statement is true for $k = N$ as $J_N(\vec{x}) = \frac{1}{2}\vec{x}^\top Q\vec{x}$. Now, assume that the statement is true for $k = l + 1$, we will show that it holds for $k = l$. Observe that we can use our recursive definition of J_k to obtain:

$$\begin{aligned}
J_l(\vec{x}) &= \min_{\vec{u}} \frac{1}{2}\left(\vec{x}^\top Q\vec{x} + \vec{u}^\top R\vec{u}\right) + J_{l+1}(A\vec{x} + B\vec{u}) \\
&= \min_{\vec{u}} \frac{1}{2}\left(\vec{x}^\top Q\vec{x} + \vec{u}^\top R\vec{u} + (A\vec{x} + B\vec{u})^\top Q_{l+1}(A\vec{x} + B\vec{u})^\top\right) \\
&= \min_{\vec{u}} \frac{1}{2}\left(\vec{x}^\top Q\vec{x} + \vec{u}^\top R\vec{u} + \vec{x}^\top A^\top Q_{l+1}A\vec{x} + \vec{u}^\top B^\top Q_{l+1}B\vec{u} + 2\vec{u}^\top B^\top Q_{l+1}A\vec{x}\right) \\
&= \min_{\vec{u}} \frac{1}{2}\left(\vec{x}^\top (Q + A^\top Q_{l+1}A)\vec{x} + \vec{u}^\top (R + B^\top Q_{l+1}B)\vec{u} + 2\vec{u}^\top B^\top Q_{l+1}A\vec{x}\right).
\end{aligned}$$

Note that since the above equation is a convex quadratic optimization problem in $\vec{u}$ for a fixed $\vec{x}$. Therefore, we can compute its derivative with respect to $\vec{u}$ and set it to 0 to explicitly minimize over $\vec{u}$ to obtain:

$$(R + B^\top Q_{l+1}B)\vec{u}^* + B^\top Q_{l+1}A\vec{x} = 0 \implies \vec{u}^* = -(R + B^\top Q_{l+1}B)^{-1}B^\top Q_{l+1}A\vec{x}.$$

By substituting $\vec{u}^*$ we see that:

$$\begin{aligned}
&(\vec{u}^*)^\top (R + B^\top Q_{l+1}B)\vec{u}^* + 2(\vec{u}^*)^\top B^\top Q_{l+1}A\vec{x} \\
&\quad = \vec{x}^\top A^\top Q_{l+1}B(R + B^\top Q_{l+1}B)^{-1}(R + B^\top Q_{l+1}B)(R + B^\top Q_{l+1}B)^{-1}B^\top Q_{l+1}A\vec{x} \\
&\qquad - 2\vec{x}^\top A^\top Q_{l+1}B(R + B^\top Q_{l+1}B)^{-1}B^\top Q_{l+1}A\vec{x} \\
&\quad = -\vec{x}^\top A^\top Q_{l+1}B(R + B^\top Q_{l+1}B)^{-1}B^\top Q_{l+1}A\vec{x}.
\end{aligned}$$

Therefore, we by substituting this in the above equation, we get the following:

$$J_l(\vec{x}) = \vec{x}^\top \frac{Q_l}{2} \vec{x}$$
$$Q_l = Q + A^\top Q_{l+1} A - A^\top Q_{l+1} B (R + B^\top Q_{l+1} B)^{-1} B^\top Q_{l+1} A.$$

The fact that $Q_l \succ 0$ follows from the fact that Q_l is the sum of $Q \succ 0$ and a PSD matrix.

(b) **(Optional)** Now, show that the expression for Q_l is equivalent for the expression obtained by using the Lagrangian. That is, show that Q_l from the previous part is the same as:

$$Q_l = Q + A^\top (Q_{l+1}^{-1} + B R^{-1} B^\top)^{-1} A.$$

You may find useful the Sherman-Morrison-Woodbury matrix identity:

$$(M + UWV)^{-1} = M^{-1} - M^{-1} U (W^{-1} + V M^{-1} U) V M^{-1}.$$

Solution: We start by manipulating the middle expression in the definition of Q_l found in part (a):

$$\begin{aligned} &A^\top Q_{l+1} A - A^\top Q_{l+1} B (R + B^\top Q_{l+1} B)^{-1} B^\top Q_{l+1} A \\ &\quad = A^\top (Q_{l+1} - Q_{l+1} B (R + B^\top Q_{l+1} B)^{-1} B^\top Q_{l+1}) A \\ &\quad = A^\top (Q_{l+1}^{-1} + B R^{-1} B^\top)^{-1} A \end{aligned}$$

where the last equality follows from the Sherman-Morrison-Woodbury identity by substituting $M = Q_{l+1}^{-1}$, $W = R^{-1}$, $U = B$ and $V = B^\top$. By substituting this expression in the definition of Q_l from Part (a), we get the desired conclusion:

$$Q_{l+1} = Q + A^\top (Q_{l+1}^{-1} + B R^{-1} B^\top)^{-1} A.$$

Finally, notice that we can further factorize the above expression as follows:

$$A^\top (Q_{l+1}^{-1} + B R^{-1} B^\top)^{-1} A = A^\top Q_{l+1}^{1/2} (I + Q_{l+1}^{1/2} B R^{-1} B^\top Q_{l+1}^{1/2})^{-1} Q_{l+1}^{1/2} A$$

to obtain

$$Q_{l+1} = Q + A^\top Q_{l+1}^{1/2} (I + Q_{l+1}^{1/2} B R^{-1} B^\top Q_{l+1}^{1/2})^{-1} Q_{l+1}^{1/2} A.$$

6. Homework process

Whom did you work with on this homework? List the names and SIDs of your group members.