The University of Texas at Austin
Department of Electrical and Computer Engineering

EE381V-11: Large Scale Optimization — Fall 2012

Problem Set Zero

Caramanis/Sanghavi — Due: Thursday, September 6, 2012.

This problem set is intended to get the semester off to a good start (!) and to help you refresh your memory about basic concepts of linear algebra. It is also intended to get us started with Matlab, and to provide some connections and motivation for what we will see later in the course. Any problems marked by a ? do not need to be turned in, and are just additional review problems, or reading assignments.

Reading Assignments

1. (?) Reading: Boyd & Vandenberghe: Chapters 1 and 2.
2. (?) Linear Algebra is probably the single most important technical tool used in this class. The course text by Boyd and Vandenberghe has a good review in Appendix A. An excellent and quite in depth review of the most relevant topics from Linear Algebra and Analysis can be found in Appendix A of the Lecture Notes by Ben-Tal and Nemirovski. These have been posted on Blackboard.

   For example, concepts that will be used repeatedly in this class include:

   (a) Matrices, linear operators, vector spaces.

   (b) Independence, range, null space, etc.

   (c) Eigenvalues/eigenvectors, symmetric matrices, spectral theorem, singular values and singular value decomposition (SVD),

   (d) etc...

   If these topics are not fresh, spend some time learning/reminding yourself of the basic notions.

Matlab and Computational Assignments. Please provide a printout of the Matlab code you wrote to generate the solutions to the problems below.

1. (?) Figure out how to use Matlab. If you have not used this before, see here for a tutorial: `http://www.math.ufl.edu/help/matlab-tutorial/`
2. (?) Set up CVX. CVX is a Matlab add-on that provides an extremely easy syntax for solving small and medium-scale optimization problems. You will need this for a problem in this problem set, and in general it is extremely useful for quickly setting up and solving smallish convex optimization problems. See here for directions, source code, etc: `http://cvxr.com/cvx/download/`

3. This problem illustrates the power of convex optimization. At the same time, it suggests the flexibility but also the limitations of generic optimization algorithms not tailored for the problem at hand.

   Consider the problem discussed in class on Thursday: Sparse Recovery. The set-up is as follows. There is an *unknown* matrix $\beta \in \mathbb{R}^p$. We get n noisy linear measurements of β: $y_i = \langle x_i, \beta \rangle + e_i$, $i = 1, \ldots, n$. We write this in matrix notation:

$$y = X\beta + e.$$

   Here we have $n < p$. Typically, this means that the problem is *under-determined*: there are more unknowns than constraints. However, it turns out that if β is sparse, then it is possible to solve this.[1] In this problem you will explore solving this problem when β is sparse.

   Download the file: `http://users.ece.utexas.edu/~cmcaram/EE381V_2012F/ps0_matlab.zip`. This contains a matlab file that will generate the data for three sparse-recovery problems of different sizes. For each problem, we provide: (X, y) — these data specify the problem, and you will use them to compute β. We also provide *testing* data, $(X_{\text{test}}, y_{\text{test}})$. Once you have computed β, you will use the testing data to see how well you did, by computing:

$$\|X_{\text{test}}\beta - y_{\text{test}}\|_2.$$

   The files provided also give the parameter λ which is used by the optimization problem specified below as Algorithm 2 (Lasso).

   - Algorithm 1: Least Squares.
     Compute a least-squares solution to the problem by solving:

$$\text{minimize :} \qquad \|X\beta - y\|_2$$

     Figure out how to solve this problem using CVX. Ask CVX to solve each of the three problems, and report: (a) Did CVX succeed? (b) If so, how long did it take to solve each instance? (c) Report the Regression error of the solution computed: $\|X\beta^* - y\|_2$ and also the Testing error: $\|X_{\text{test}}\beta - y_{\text{test}}\|_2$.
   - Algorithm 2: Sparse Recovery via an optimization-based algorithm called LASSO.

$$\text{minimize :} \qquad \|X\beta - y\|_2 + \lambda\|\beta\|_1$$

     Ask CVX to solve each of the three problems, and report: (a) Did CVX succeed? (b) If so, how long did it take to solve each instance? (c) Report the Regression error of the solution computed: $\|X\beta^* - y\|_2$ and also the Testing error: $\|X_{\text{test}}\beta - y_{\text{test}}\|_2$. (d) What is the support of β? That is, what are the non-zero coefficients of β.

   What you should have found through this exercise is that: (a) CVX is extremely simple to use, and works very well for problems of small or even medium size, but is not good for bigger problems. (b) Optimization can be used to solve quite interesting problems, but it must be used correctly. Note that the solution to least squares formulation does nothing to help us find β^*, and the solution it finds performs terribly on the *testing* data. Meanwhile, the solution to LASSO finds β^*, and that solution has a much better performance on the *testing* data.

[1] We will understand the reason for this much better later in the course, and even better in the second part of this two-course sequence in the Spring.

4. (OMP – Orthogonal Matching Pursuit). In the last problem, you used an optimization-based algorithm to solve the sparse inverse problem. You found that it broke down for the largest of the three problems. Here you will implement a different, greedy algorithm, in Matlab, and thereby see that if the right algorithm is used, even the largest of the three problems is extremely easy.

   The algorithm is called *Orthogonal Matching Pursuit*, and it greedily computes the support of β^* as follows:

Initialize:	$\mathcal{I} = \emptyset$.
Step 1 :	Set $\mathcal{I} = \mathrm{argmax}_i \langle y, X_i \rangle$, where X_i is the i^{th} column of X.
Step 2 :	Let r be the perpendicular complement of y to X_i.[2]
Step 3 :	Repeat Step 1 and augment $\mathcal{I}$ by the maximizer of $\langle r, X_i \rangle$.
Step 4 :	Repeat Steps 2 and 3 k times, where k is the sparsity of β^*. For us, $k = 5$.

   Once you have found the support, solve the corresponding standard least squares regression problem (this should have only 5 variables for all three cases) in order to obtain the value of the coefficients of β.

   Implement this algorithm in Matlab, and solve the three problems from the previous problem. Report (a) what is the sparsity pattern found; (b) how long does the solution take; (c) the regression and testing errors, as in the previous problem.

Linear Algebra Review

1. (?) Vector Spaces: For the following examples, state whether or not they are in fact vector spaces.

   - The set of polynomials in one variable, of degree at most d.
   - The set of continuous functions mapping $[0, 1]$ to $[0, 1]$, such that $f(0) = 0$.
   - The set of continuous functions mapping $[0, 1]$ to $[0, 1]$, such that $f(1) = 1$.

2. (?) Recall that a linear operator $T : V \to W$ is a map that satisfies:

$$T(a\boldsymbol{v}_1 + b\boldsymbol{v}_2) = aT\boldsymbol{v}_1 + bT\boldsymbol{v}_2,$$

   for every $\boldsymbol{v}_1, \boldsymbol{v}_2 \in V$.

   Show which of the following maps are linear operators:

   - $T : V \to V$ given by the identity map: $\boldsymbol{v} \mapsto \boldsymbol{v}$.
   - $T : V \to W$ given by the constant map: $\boldsymbol{v} \mapsto \boldsymbol{w}_0$ for every $\boldsymbol{v} \in V$. Does your answer change depending on what $\boldsymbol{w}_0$ is?
   - Let V be the vector space of polynomials of degree at most d. Let $T : V \to V$ be the map defined by the derivative: $p(x) \mapsto p'(x)$.
   - For V as above, let T be given by:

$$T(p) = \int_0^1 p(x)\, dx.$$

- What about

$$T(p) = \int_0^1 p(x)x^3\,dx.$$

3. (?) Independence:

   - If $\boldsymbol{v}_1, \dots, \boldsymbol{v}_m \in V$ are independent, and $T : V \to W$ is a linear operator, is it true that $T\boldsymbol{v}_1, \dots, T\boldsymbol{v}_m \in W$ are independent?
   - If $\boldsymbol{v}_1, \dots, \boldsymbol{v}_m \in V$ are dependent, and $T : V \to W$ is a linear operator, is it true that $T\boldsymbol{v}_1, \dots, T\boldsymbol{v}_m \in W$ are dependent?

4. (?) True or False: If vectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3$ are elements of a vector space V, and $\{\boldsymbol{v}_1, \boldsymbol{v}_2\}$, $\{\boldsymbol{v}_2, \boldsymbol{v}_3\}$, and $\{\boldsymbol{v}_1, \boldsymbol{v}_3\}$ are independent, then the set $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3\}$ is also linearly independent.

5. (?) Range and Nullspace of Matrices: Recall the definition of the null space and the range of a linear transformation, $T : V \to W$:

$$\begin{aligned} \text{null}(T) &= \{\boldsymbol{v} \in V \,:\, T\boldsymbol{v} = 0\} \\ \text{range}(T) &= \{T\boldsymbol{v} \in W \,:\, \boldsymbol{v} \in V\} \end{aligned}$$

   Remind yourselves of the Rank-Nullity Theorem.

6. More Range and Nullspace.

   - Suppose A is a 10-by-10 matrix of rank 5, and B is also a 10-by-10 matrix of rank 5. What is the **smallest** and **largest** the rank the matrix $C = AB$ could be?
   - Now suppose A is a 10-by-15 matrix of rank 7, and B is a 15-by-11 matrix of rank 8. What is the **largest** that the rank of matrix $C = AB$ can be?

7. Riesz Representation Theorem: Consider the standard basis for $\mathbb{R}^n$: $e_1 = (1, 0, \dots, 0), e_2 = (0, 1, 0, \dots, 0)$, etc. Recall that the inner-product of two vectors $\boldsymbol{w}_1 = (\alpha_1, \dots, \alpha_n), \boldsymbol{w}_2 = (\beta_1, \dots, \beta_n) \in \mathbb{R}^n$, is given by:

$$\langle \boldsymbol{w}_1, \boldsymbol{w}_2 \rangle = \sum_{i=1}^{n} \alpha_i \beta_i.$$

   Let $f : \mathbb{R}^n \to \mathbb{R}$ be a linear map. Show that there exists a vector $\boldsymbol{x} \in \mathbb{R}^n$, such that

$$f(\boldsymbol{w}) = \langle \boldsymbol{x}, \boldsymbol{w} \rangle,$$

   for any $\boldsymbol{w} \in \mathbb{R}^n$.

   Remark: It turns out that this result is true in much more generality. For example, consider the vector space of square-integrable functions (something we will see much more later in the course). Let F denote a linear map from square integrable functions to $\mathbb{R}$. Then, as a consequence similar to the finite dimensional exercise here, there exists a square integrable function, g, such that:

$$F(f) = \int fg.$$

8. Let V be the vector space of (univariate) polynomials of degree at most d. Consider the mapping $T : V \to V$ given by:

$$Tp = a_0 p(t) + a_1 t p^{(1)}(t) + a_2 t^2 p^{(2)}(t) + \cdots + a_d t^d p^{(d)}(t),$$

where $p^{(r)}(t)$ denotes the r^{th} derivative of the polynomial p.

- True or False: if $Tp = 2p(t) - tp'(t)$, then for every polynomial $q \in V$, there exists a polynomial $p \in V$, with $Tp = q$.
- What about for T given by $Tp = 2p(t) - 3tp'(t)$?
- Provide a characterization of the set of coefficients $(a_0, a_1, \ldots, a_d)$, such that the operator T they define has the property that for every polynomial $q \in V$, there exists a polynomial $p \in V$, with $Tp = q$.

9. Recall the definition of rank, and show the following.

- For A an $m \times n$ matrix, $\text{rank}A \leq \min\{m, n\}$.
- For A an $m \times k$ matrix and B a $k \times n$ matrix,

$$\text{rank}(A) + \text{rank}(B) - k \leq \text{rank}(AB) \leq \min\{\text{rank}(A), \text{rank}(B)\}.$$

- For A and B $m \times n$ matrices,

$$\text{rank}(A + B) \leq \text{rank}(A) + \text{rank}(B).$$

- For A an $m \times k$ matrix, B a $k \times p$ matrix, and C a $p \times n$ matrix, then

$$\text{rank}(AB) + \text{rank}(BC) \leq \text{rank}(B) + \text{rank}(ABC)$$

10. (?) Consider a mapping $T : V \to V$. If the vector space V is finite dimensional, then if $\text{null}T = \{0\}$, T is surjective (also known as onto), that is, for any $\boldsymbol{v} \in V$, there exists $\hat{\boldsymbol{v}}$ such that $T\hat{\boldsymbol{v}} = \boldsymbol{v}$. Conversely, if T is surjective, then $\text{null}T = \{0\}$, and $T\boldsymbol{v} = 0$ implies $\boldsymbol{v} = 0$.

- Give an example of an infinite dimensional vector space, V, and a linear operator $T : V \to V$, such that T is surjective, but $\text{null}T \neq \{0\}$.
- Give an example of an infinite dimensional vector space, V, and a linear operator $T : V \to V$, such that $\text{null}T = \{0\}$, but T is not surjective.

[Hint: consider the space of polynomials of arbitrary degree.]