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

EE381K: Large Scale Optimization — Fall 2016

Problem Set One

Constantine Caramanis Due: Wednesday, September 14, 2016.

Submission. This problem set should be submitted electronically through Canvas, including any code submission, or printouts of plots, etc.

About this problem set. On the computational/algorithmic side: this problem set gets us started with thinking about two of the key thrusts in this class – modeling, and algorithms. On the written side, the exercise on over-complete and under-complete regression is a prelude to the work we'll do on projection, and characterizations of optimality. Overall, the written exercises also provide much practice with linear algebra. *Note that not all problems need to be turned in: those marked with a "?" need not be turned in.*

Reading Assignments

1. (?) Reading: Boyd & Vandenberghe: Chapters 1 and 2, and Chapters 9.1 & 9.2. (This book is available on line: `http://stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf`)

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. This has been posted on Canvas.

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

1. (?) Set up CVX. CVX is a Matlab add-on that provides an extremely easy syntax for solving small-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. As you will see in this problem set, it does not scale well to larger problems. See here for directions, source code, etc: `http://cvxr.com/cvx/download/` For Python users, you may consider CVXOPT `http://cvxopt.org/`.

2. 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 noisy extension of the problem discussed in class on Thursday: Sparse Recovery. The set-up is as follows. There is an *unknown* vector $\beta \in \mathbb{R}^p$. 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 in many cases it is possible to recover β from fewer measurements than dimensions. [1] In this problem you will explore solving this problem when β is sparse.

   Download the file: `ps1A` from Canvas. This contains a file that will generate the data for three sparse-recovery problems of different sizes (see the ReadMe file). 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: $\|X_{\text{test}}\beta - y_{\text{test}}\|_2$.
   - (?) **Side Note**: As you will compute in the written exercises below, this problem has a closed form solution: $\hat{\beta}_{\text{LS}} = (X^\top X)^{-1} X^\top y$. Use the `tic` and `toc` commands to clock how long it takes to compute the value of this closed form equation (much faster than CVX!). Note that Matlab has a special implementation for finding the least-squares solution of a system $X\beta = y$, given by $\hat{\beta}_{\text{LS}} = X \setminus y$. Again use the `tic` and `toc` commands to see how much faster this is. This has to do with the particular implementation Matlab uses, and relies on the LU matrix factorization. We will not have time to cover these aspects of numerical linear algebra, but when appropriate we will give pointers. As you can see from this simple example, not only is the higher level algorithm very important, but also the specific implementation details can have a significant impact on running time.
   - Algorithm 2: Sparse Recovery via an optimization-based algorithm called LASSO.

[1] We will understand the reason for this much better later in the course, though we will not cover the statistical results that guarantee recovery.

Instead of minimizing the squared error, the idea is to "encourage" the solution to be sparse, by minimizing a *penalized* version of the squared error – the squared error plus a small multiple of the ℓ^1-norm of the solution:

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

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 well suited 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.

3. (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, and thereby see that if the right algorithm is used, even the largest of the three problems is quite easy. *It will be useful to do the written problem on Least Squares before doing this computational problem.*

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

```
Initialize:  $\mathcal{I} = \emptyset$, $y_{\text{residual}} = y$
  Step 1 :   Set $\mathcal{I} = \mathcal{I} \cup \operatorname{argmax}_i \langle y_{\text{residual}}, X_i \rangle$, where $X_i$ is the $i^{th}$ column of $X$.
  Step 2 :   Let $\hat{\beta}_{\mathcal{I}} \in \mathbb{R}^{|\mathcal{I}|}$ be the optimal solution to the over-determined least-squares problem:
             min : $\|X_{\mathcal{I}}\beta - y\|$ where $X_{\mathcal{I}}$ denotes the $n \times |\mathcal{I}|$ sub-matrix obtained by selecting the
             columns of $X$ corresponding to elements in $\mathcal{I}$.
  Step 3 :   Let $y_{\text{residual}} = y - X_{\mathcal{I}}\hat{\beta}_{\mathcal{I}}$.
  Step 4 :   Repeat $k$ times, where $k$ is the sparsity of $\beta^*$. For us, $k = 5$.
```

Once you have found the support, the final solution $\hat{\beta}_{\mathcal{I}}$ is inserted into that support, and thus we obtain the final output of OMP.

Implement this algorithm 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.

4. This problem illustrates how the gradient descent algorithm behaves in different levels of strong convexity. To begin with, download the file: `ps1B` from Canvas, which contains a file that will generate the data for the problem.

We have a simple unconstrained optimization problem:

$$\min_{\beta \in \mathbb{R}^n} \quad f(\beta) \triangleq \frac{1}{2}\beta^T X \beta$$

where $X \in \mathbb{R}^{n\times n}$ is a symmetric and positive definite matrix. In the file, you can find three matrices for the problem, in which (a) all eigenvalues are one, (b) a half of the eigenvalues are one and the other half of them are very small, (c) all other than a few very large eigenvalues are one.

We want to run the gradient descent algorithm which iteratively computes

$$\beta^{(n+1)} = \beta^{(n)} - \gamma \nabla f(\beta^{(n)})$$

where γ is a constant step size. The initial $\beta^{(0)}$ is the all-ones vector.

For each matrix, find the range of γ that the solution converges to zero and the range of γ that the algorithm diverges, and explain why. Take example values of γ to illustrate the two behaviors, convergence to zero and divergence. Plot $f(\beta^{(n)})$ over n for the two of your values.

5. Take $\gamma = 1$, and plot $f(\beta^{(n)})$ over n for the second matrix (b) of the above three. Explain the convergence behavior of the solution based on the plot.

Written Problems

1. Over- and Under-determined Least Squares

   - Consider finding an approximate solution to the system of equations $y = X\beta$ where X is a $m \times p$ matrix with $m > p$. This means that there are more equations than variables (degrees of freedom), and hence there may not be a solution β that satisfies all the equations. In this case, we must consider in what sense we wish to find an "approximate" solution. For now, let us consider finding the so-called least-squares solution, i.e., the solution to the problem:

     $$\min_{\beta} : \sum_{i=1}^{n} (\langle x_i, \beta \rangle - y_i)^2 = \|X\beta - y\|_2^2.$$

     – Expand the expression $\|X\beta - y\|_2^2$, and show that it is convex as a function of β; then use the local optimality condition $\nabla f(x) = 0$ to find the optimal solution, β_{LS}.

     – Now we will use ideas of projection to obtain the same solution. Another way to write the least squares problem is as follows:

     $$\begin{aligned} \min : &\quad \|z - y\|_2^2 \\ \text{subject to} : &\quad z \in \text{Range}(X) = \{X\beta \ : \ \beta \in \mathbb{R}^p\} \end{aligned}$$

     That is, the solution is the point β_{proj} such that $X\beta_{\text{proj}}$ is the orthogonal projection of y onto the range of X. Since the range of X is a subspace, draw the picture to convince yourself that the orthogonal projection $\Pi_M(y)$ of a point y onto a subspace M satisfies: $\langle y - \Pi_M(y), x \rangle = 0$ for all $x \in M$. Use this fact for the specific problem at hand, to rederive your answer from above.

- Now consider the setting where we seek to find a solution to $y = X\beta$, where X is $m \times p$ but $m < p$. This is the so-called under-determined setting (more variables than equations) and unless some of the equations are contradictory, there is an affine subspace of solutions: if there is any solution β_0 with $y = X\beta_0$, then for any $z \in \text{Nullspace}(X)$, $\beta_0 + z$ is again a solution. One common choice from this subspace of solutions, is to find the one that is smallest, with respect to some norm. Here we choose the 2-norm: $\|\beta\|_2 = \sqrt{\sum \beta_i^2}$, and hence we wish to solve:

$$\begin{aligned} \text{min} : \quad & \|\beta\|_2^2 \\ \text{subjectto} : \quad & X\beta = y. \end{aligned}$$

  We will again use ideas from projection to find the solution.

  - Let β_0 be a solution that satisfies $y = X\beta$, and is perpendicular to the Nullspace of X, i.e., $\beta_0 \perp z$ for any $z \in \text{Nullspace}(X)$. Evidently, any other solution has the form $\beta = \beta_0 + z$, for $z \in \text{Nullspace}(X)$ (show this, if you are not quite sure why any other solution must satisfy this property). Use this to show that β_0 is the minimum norm solution, i.e., the solution to the above optimization problem.
  - Now using the fact that $y = X\beta_0$ and $\beta_0 \perp z$ for any $z \in \text{Nullspace}(X)$, compute β_0. (Hint: show first that $\beta_0 \perp z$ for any $z \in \text{Nullspace}(X)$ is equivalent to $\beta_0 \in \text{span} x_1, \ldots, x_n$, i.e., β_0 is in the span of the rows of X).

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 $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m \in V$ are independent, and $T : V \to W$ is a linear operator, is it true that $T\boldsymbol{v}_1, \ldots, T\boldsymbol{v}_m \in W$ are independent?
   - If $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m \in V$ are dependent, and $T : V \to W$ is a linear operator, is it true that $T\boldsymbol{v}_1, \ldots, T\boldsymbol{v}_m \in W$ 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 $\mathbb{R}^n$: $e_1 = (1, 0, \ldots, 0), e_2 = (0, 1, 0, \ldots, 0)$, etc. Recall that the inner-product of two vectors $\boldsymbol{w}_1 = (\alpha_1, \ldots, \alpha_n), \boldsymbol{w}_2 = (\beta_1, \ldots, \beta_n) \in \mathbb{R}^n$, 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, $\mathrm{rank} A \leq \min\{m, n\}$.
- For A an $m \times k$ matrix and B a $k \times n$ matrix,

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

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

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

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

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

10. (?) Consider a mapping $T : V \to V$. If the vector space V is finite dimensional, then if $\mathrm{null} T = \{0\}$, 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 $\mathrm{null} T = \{0\}$, and $T\boldsymbol{v} = 0$ implies $\boldsymbol{v} = 0$.

11. (?) Various properties of orthogonal subspaces: Let V be a finite dimensional vector space with an inner produce, and let $U \subseteq V$ be a subspace. Recall that the space $U^\perp$ is defined as:

$$U^\perp = \{v \in V \,:\, \langle v \,,\, u \rangle = 0,\ \forall u \in U\}.$$

(a) Show that if U is a subspace, then so is $U^\perp$.

(b) Show that $(U^\perp)^\perp = U$.

(c) Show that if $U, W \subseteq V$ are subspaces of V, then

$$U \subseteq W \ \Leftrightarrow U^\perp \supseteq W^\perp.$$

(d) Suppose now that $X \subseteq V$ is just a subset, i.e., not necessarily a subspace of V. Show that the definition $X^\perp$ still makes sense, and that $X^\perp$ is a subspace. Next show that $(X^\perp)^\perp \supseteq X$, and it is defined as the smallest subspace that contains the set X.

(e) Show that when U is a subspace of V, then V is the direct product of U and $U^\perp$ (denoted $V = U \oplus U^\perp$). That is, show that any $v \in V$ can be written *uniquely* as

$$v = u + u^\perp,$$

where $u \in U$, and $u^\perp \in U^\perp$.

12. (Boyd and Vandenberghe, Ex. 2.10) Consider the set

$$C = \{x \in \mathbb{R}^n \,:\, x^\top Ax + b^\top x + c \leq 0\},$$

where $A \in \mathbb{S}^n$, $b \in \mathbb{R}^n$ and $c \in \mathbb{R}$.

(a) Show that if $A \in \mathbb{S}^n_+$ (i.e., A is positive semidefinite) then the set C is convex.

(b) Consider the set obtained by intersecting C with a hyperplane:

$$C_1 = C \cap \{x \,:\, g^\top x + h = 0\}.$$

Show that C_1 is convex if there exists $\lambda \in \mathbb{R}$ such that $(A + \lambda gg^\top) \in \mathbb{S}^n_+$.

13. (Boyd and Vandenberghe, Ex. 2.21) For $C, D \subseteq \mathbb{R}^n$ disjoint convex sets, let

$$\mathcal{S} = \{(a, b) \,:\, a^\top x \leq b \;\forall x \in C,\; a^\top x \geq b \;\forall x \in D\}$$

be the set of separating hyperplanes. Show that $\mathcal{S}$ is convex.

14. (?) There are several natural operations on sets, that preserve convexity. Convince yourselves that the following all preserve convexity.

(a) Cartesian product: If $C_1, \ldots, C_m \subseteq \mathbb{R}^d$ are convex sets, then the set

$$C = C_1 \times \cdots \times C_m = \{(x_1, \ldots, x_m),\; x_i \in C_i\}$$

is convex.

(b) Affine and inverse maps: For $C \subseteq \mathbb{R}^n$ convex, and $A : \mathbb{R}^n \to \mathbb{R}^m$ a linear operator (i.e., an $m \times n$ matrix) then **show that** the following two sets are convex:

$$\begin{aligned} D_1 &= \{Ax \,:\, x \in C\} \\ D_2 &= \{x \,:\, Ax \in C\}. \end{aligned}$$

(c) Minkowski sum: If $C_1, C_2 \subseteq \mathbb{R}^n$ are convex, **show that**

$$C = C_1 + C_2 = \{x = x_1 + x_2 \,:\, x_1 \in C_1, x_2 \in C_2\}$$

is convex.

15. (?) Boyd and Vandenberghe, Ex. 2.26.

16. (?) Boyd and Vandenberghe, Ex. 2.35.

17. Consider two points, $v_1, v_2 \in \mathbb{R}^n$. Show that there exist $c \in \mathbb{R}^n$ and $d \in \mathbb{R}$ (and find them!) such that

$$\{x \,:\, ||x - v_1|| \leq ||x - v_2||\} = \{x \,:\, c^\top x \leq d\}.$$

Thus, you are showing that the set of points in $\mathbb{R}^n$ that are closer to point v_1 than to point v_2, form a half-space.

18. Let A be an $n \times m$ real matrix, and B a $k \times m$ real matrix. Suppose that for every $x \in \mathbb{R}^m$, $Ax = 0$ only if $Bx = 0$, that is,

$$Ax = 0 \Rightarrow Bx = 0.$$

Show that there exists a $k \times n$ real matrix C such that $CA = B$.