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

EE381K: Convex Optimization — Fall 2015

Problem Set One Solutions

Constantine Caramanis Due: Tuesday, September 8, 2015.

Matlab and Computational Assignments

1.
   - Algorithm 1 : Least square
     (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.
     (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.

   Both of the algorithms work fast for the first problem ($X \in \mathbb{R}^{50\times 500}$) within a few seconds. They also succeeded for the second problem ($X \in \mathbb{R}^{500\times 5000}$), though it takes longer. However, if it comes to the third problem ($X \in \mathbb{R}^{5000\times 50000}$), it is not guaranteed that the algorithms succeed. They appear to be stopped, get out of memory, or perhaps succeed in a some good processors. Nevertheless, this results show that general optimization algorithms may not work well in large-scale problems.

   However, for the small-scale problems, validity of results depends on which algorithm we use. Both least square and LASSO find β^*s such that the regression errors $\|X\beta^* - y\|_2$ are sufficiently small, but it is still not ensured that they are close to the true β. Since least square finds β^* no matter how sparse, it is very different from the true β. This fact is revealed when we see that the testing error $\|X_{\text{test}}\beta^* - y_{\text{test}}\|_2$ is large ($\gg 1$). On the other hand, LASSO finds β^* which is as sparse as possible, it is close to the true β. We can find that the testing error is very small ($\ll 1$), and also that the support of β^* is exactly the same as the support of the true β.

   Example codes for the algorithms are the following:

```
function b = leastsquare(X, y)
[m n] = size(X);
cvx_begin
        variable b(n);
        minimize ( norm(X*b-y) );
cvx_end
```

```
function b = lasso(X, y, lambda)
[m n] = size(X);
cvx_begin
        variable b(n);
        minimize ( norm(X*b-y) + lambda*norm(b,1) );
cvx_end
```

Some example results are given in the following table. The elapsed time depends on the performance of the machine (and these were run on an old computer – the point is the relative time).

LS	Time	$\|X\beta^* - y\|_2$	$\|X_{\text{test}}\beta^* - y_{\text{test}}\|_2$
Matrix 1	0.240sec	9.70×10^{-14}	2.24×10^{1}
Matrix 2	8.273sec	2.56×10^{-12}	2.28×10^{1}
Matrix 3	-	-	-
LASSO	Time	$\|X\beta^* - y\|_2$	$\|X_{\text{test}}\beta^* - y_{\text{test}}\|_2$
Matrix 1	0.349sec	4.66×10^{-9}	1.96×10^{-1}
Matrix 2	61.05sec	1.08×10^{-8}	6.90×10^{-2}
Matrix 3	-	-	-

2. (OMP – Orthogonal Matching Pursuit)

   (a) What is the sparsity pattern found?

   (b) How long does the solution take?

   (c) What are the regression and testing errors?

   OMP works remarkably faster than the above two algorithms, least square and LASSO. It succeeds and finds exactly the same sparsity pattern as the original for all of the three problems. The following is an example matlab code for the algorithm.

```
function b = omp(X,y,d)
[m n] = size(X); b = zeros(n,1); support = [];
r = y;
for k=1:d
    innerp = r'*X;                        % compute the inner products
    [maxinnerp max_i] = max(innerp);      % find the maximum
    support = [support max_i];            % add to the support set
    r = r - maxinnerp*X(:,max_i)/(norm(X(:,max_i))^2);  % projection
end

% standard least-square regression
b(support) = inv(X(:,support)'*X(:,support))*X(:,support)'*y;
```

Some example results are given in the following table. The elapsed time depends on the performance of the machine.

	Time	$\|X\beta^* - y\|_2$	$\|X_{\text{test}}\beta^* - y_{\text{test}}\|_2$
Matrix 1	0.008sec	7.10×10^{-2}	3.03×10^{-2}
Matrix 2	0.096sec	2.15×10^{-1}	1.26×10^{-2}
Matrix 3	3.461sec	7.06×10^{-1}	3.10×10^{-3}

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 $\nabla f(\beta) = 0$ to find the optimal solution, β_{LS}.

       Solution: $f(\beta) = \|X\beta - y\|_2^2 = \beta^\top X^\top X\beta - 2\beta^\top X^\top y + \|y\|^2$. Taking the second derivative, we find that the hessian is $\nabla^2 f(\beta) = 2X^\top X$. This is positive semidefinite, since $z^\top X^\top Xz = \|Xz\| \geq 0$, $\forall z$, and hence the problem is convex. The local optimality condition $\nabla f(\beta) = 0$ yields $2(X^\top X\beta - X^\top y) = 0$, or, since $X^\top X$ is invertible, $\beta_{LS} = (X^\top X)^{-1} X^\top y$.

     – 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 re-derive your answer from above.

       Solution: As the hint suggests, the solution $\Pi_M(y)$ satisfies $\langle y - \Pi_M(y), x \rangle = 0$ for all $x \in M$. For us, $M = \text{Range}(X)$, and of course $\Pi_M(y) = X\beta_{LS}$, so we have $\langle y - X\beta_{LS}, X\beta \rangle = 0$, $\forall \beta$, which is equivalent to $\langle y - X\beta_{LS}, X \rangle = 0$, or $y^\top X - \beta_{LS}^\top X^\top X = 0$, i.e., $\beta_{LS} = (X^\top X)^{-1} X^\top y$.

   - 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:

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

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.

  Solution: Consider any other solution, β_1. We must have $\beta_1 = \beta_0 + z$, for some $z \in \text{Null}(X)$, i.e., $Xz = 0$. Then we have:

  $$\|\beta_1\|^2 = \|\beta_0 + z\|^2 = \|\beta_0\|^2 + \|z\|^2.$$

  The last equality follows because the cross terms vanish: $2\beta_0^\top z = 0$, by the fact that β_0 was chosen to be orthogonal to every element in the nullspace of X.
- 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).

  Solution: If β_0 is perpendicular to every element in the nullspace, it must be in the linear span of rows of X. Indeed, the null space of X is the set of vectors perpendicular to every row of X. Therefore the set of vectors perpendicular to every element of the null space, must be in the span of the rows.
  Therefore we have that $\beta_0 = X^\top \lambda = \sum \lambda_i x_i$ for some vector λ, where x_i denotes the i^{th} row of X. Thus $X\beta_0 = y$ now implies $XX^\top \lambda = y$, i.e., $\lambda = (XX^\top)^{-1} y$, whence $\beta_0 = X^\top \lambda = X^\top (XX^\top)^{-1} y$

Linear Algebra Review

1. Vector Spaces

- The set of polynomials in one variable, of degree at most d is a vector space:
  Closed under addition and scalar multiplication:

  $$\alpha(a_0 + a_1 x^1 + ... + a_d x^d) + (b_0 + b_1 x^1 + ... + b_d x^d) = (\alpha a_0 + b_0) + (\alpha a_1 + b_1) x^1 + ... + (\alpha a_d + b_d) x^d$$

  Zero vector: 0 is a polynomial of degree less than d.
  Other properties follow naturally.

- $\hat{S}$ = the set of continuous functions mapping $[0, 1]$ to $[0, 1]$, such that $f(0) = 0$ is not a vector space: Consider $f_1(x) = f_2(x) = x$. $f_1(1) + f_2(1) = x + x = 2x$, so $f_1(1) + f_2(1) \notin \hat{S}$ because it maps $[0, 1]$ onto $[0, 2]$.

- $\hat{S}$ = the set of continuous functions mapping $[0,1]$ to $[0,1]$, such that $f(1) = 1$ is not a vector space: Consider $f_1(x) = f_2(x) = 1$. $f_1(1)+f_2(1) = 1+1 = 2$, so $f_1(1)+f_2(1) \notin \hat{S}$.

2. Show which of the following maps are linear operators:

   - $T : V \to V$ given by the identity map: $\boldsymbol{v} \mapsto \boldsymbol{v}$.
     For every $\boldsymbol{v}_1, \boldsymbol{v}_2 \in V$,

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

     Linear.
   - $T : V \to W$ given by the constant map: $\boldsymbol{v} \mapsto \boldsymbol{w}_0$ for every $\boldsymbol{v} \in V$. If $\boldsymbol{w}_0 = 0$, then for every $\boldsymbol{v}_1, \boldsymbol{v}_2 \in V$,

     $$T(a\boldsymbol{v}_1 + b\boldsymbol{v}_2) = 0 = aT\boldsymbol{v}_1 + bT\boldsymbol{v}_2.$$

     Linear.
     If $\boldsymbol{w}_0 \neq 0$, then for any $\boldsymbol{v}_1 \in V$,

     $$T(\boldsymbol{v}_1 + \boldsymbol{v}_1) = \boldsymbol{w}_0 \neq \boldsymbol{w}_0 + \boldsymbol{w}_0 = T\boldsymbol{v}_1 + T\boldsymbol{v}_1.$$

     Not Linear.
   - 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: $p(x) \mapsto p'(x)$.

     $$\begin{aligned} &T(\alpha(a_0 + a_1x + ... + a_dx^d) + \beta(b_0 + b_1x + ... + b_dx^d)) \\ &= \alpha(0 + a_1 + ... + a_dx^{d-1}) + \beta(0 + b_1 + ... + b_dx^{d-1}) \\ &= \alpha T(a_0 + a_1x + ... + a_dx^d) + \beta T(b_0 + b_1x + ... + b_dx^d) \end{aligned}$$

     Linear.

   - For V as above, let T be given by:

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

     $$\begin{aligned} &T(\alpha(a_0 + a_1x + ... + a_dx^d) + \beta(b_0 + b_1x + ... + b_dx^d)) \\ &= \alpha a_0x + \alpha a_1x^2/2 + ... + \alpha a_dx^{d+1}/(d+1) + \beta b_0x + \beta b_1x^2/2 + ... + \beta b_dx^{d+1}/(d+1)|_0^1 \\ &= \alpha a_0 + \alpha a_1/2 + ... + \alpha a_d/(d+1) + \beta b_0 + \beta b_1/2 + ... + \beta b_d/(d+1) \\ &= \alpha(a_0x + a_1x^2/2 + ... + a_dx^{d+1}/(d+1)) + \beta(b_0x + b_1x^2/2 + ... + b_dx^{d+1}/(d+1))|_0^1 \\ &= \alpha T(a_0 + a_1x + ... + a_dx^d) + \beta T(b_0 + b_1x + ... + b_dx^d) \end{aligned}$$

     Linear.

   - What about

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

$T(\alpha(a_0 + a_1x + ... + a_dx^d) + \beta(b_0 + b_1x + ... + b_dx^d))$
$= \alpha a_0x^4/4 + \alpha a_1x^5/5 + ... + \alpha a_dx^{d+4}/(d+4) + \beta b_0x^4/4 + \beta b_1x^5/5 + ... + \beta b_dx^{d+4}/(d+4)|_0^1$
$= \alpha a_0/4 + \alpha a_1/5 + ... + \alpha a_d/(d+4) + \beta b_0/4 + \beta b_1/5 + ... + \beta b_d/(d+4)$
$= \alpha(a_0x^4/4 + a_1x^5/5 + ... + a_dx^{d+4}/(d+4)) + \beta(b_0x^4/4 + b_1x^5/5 + ... + b_dx^{d+4}/(d+4))|_0^1$
$= \alpha T(a_0 + a_1x + ... + a_dx^d) + \beta T(b_0 + b_1x + ... + b_dx^d)$

Linear.

- Note that, in general, integration against any function is a linear map. If the operator T is defined as $T(f) = \int fg$, then if $f = \alpha f_1 + \beta f_2$, we have:

$$\begin{aligned} T(f) &= \int f(x)g(x)\,dx \\ &= \int (\alpha f_1(x) + \beta f_2(x))g(x)\,dx \\ &= \alpha \int f_1(x)g(x)\,dx + \beta \int f_2(x)g(x)\,dx \\ &= \alpha T(f_1) + \beta T(f_2). \end{aligned}$$

3. Independence:
   - $T\boldsymbol{v} = \mathbf{0}$ is a linear map, but any pair of independent vectors $\boldsymbol{v}_1, \boldsymbol{v}_2$ are both mapped to $\mathbf{0}$, meaning that $T\boldsymbol{v}_1, T\boldsymbol{v}_2$ are dependent.
   - Suppose (without loss of generality) that $\boldsymbol{v}_m = \sum_{i=1}^{m-1} a_i\boldsymbol{v}_i$. $T\boldsymbol{v}_m = T(\sum_{i=1}^{m-1} a_i\boldsymbol{v}_i) = \sum_{i=1}^{m-1} a_iT\boldsymbol{v}_i$, so $\{T\boldsymbol{v}_i\}$ are dependent.
4. False. Consider $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$, and $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$.
6. Range and Nullspace of Matrices:
   - $0 \leq rank(AB) \leq 5$.
   - $rank(AB) \leq 7$.
7. Riesz Representation Theorem: Let $f : \mathbb{R}^n \to \mathbb{R}$ be a linear map, and let $\boldsymbol{w}$ be an arbitrary element of $\mathbb{R}^n$.

$$f(\boldsymbol{w}) = f(\sum_{i=1}^{n} w_i\boldsymbol{e}_i) = \sum_{i=1}^{n} f(w_i\boldsymbol{e}_i) = \sum_{i=1}^{n} f(\boldsymbol{e}_i)w_i = \langle f(\boldsymbol{e}_i), \boldsymbol{w}\rangle.$$

   $f(\boldsymbol{e}_i)$ is the vector we sought, and therefore the theorem is proved.

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

$$Tp = a_0p(t) + a_1tp^{(1)}(t) + a_2t^2p^{(2)}(t) + \cdots + a_dt^dp^{(d)}(t),$$

   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 $Tp = q$.
  (*Added Note: V still consists of polynomials of degree at most d, where d is some fixed but arbitrary number, i.e., d need not be equal to 1*)
  False. $x^2 \in \text{Null}T$. That is, for any choice of p, the t^2 term of Tp has a coefficient 0.
- What about for T given by $Tp = 2p(t) - 3tp'(t)$? True. Note that the basis $(1, t, t^2, ..., t^d)$ is mapped to $(2, -t, -4t^2, ..., (2-3k)t^k, ..., (2-3d)t^d)$ which is independent, and hence it is surjective.
- 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 $Tp = q$.
  Let the ith coefficient of p be called p_i.

$$q_i = (\sum_{j=0}^{i} a_i \frac{j!}{(j-i)!}) \cdot p_i.$$

  So if

$$\sum_{j=0}^{i} a_i \frac{j!}{(j-i)!} \neq 0,$$

  for all $i = 0, 1, 2, ..., d$, then the property is satisfied.

9. Recall the definition of rank given in class, 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

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

  Solution. Note that by rank-nullity, this inequality is precisely equivalent to:

$$\dim \text{Null}(ABC) - \dim \text{Null}(BC) \leq \dim \text{Null}(AB) - \dim \text{Null}(B).$$

  Now, since the null space is a vector space, and since clearly $\text{Null}(AB) \supseteq \text{Null}(B)$ and $\text{Null}(ABC) \supseteq \text{Null}(BC)$, we can write the larger spaces, decomposing them into a direct product:

$$\begin{aligned} \text{Null}(ABC) &= \text{Null}(BC) \oplus X \\ \text{Null}(AB) &= \text{Null}(B) \oplus Y, \end{aligned}$$

  for some sets X and Y. Now, note that by their definition, for any $\mathbf{x} \in X$, we have $C\mathbf{x} \in Y$ (check this!). Therefore, we can define:

$$C : X \longrightarrow Y.$$

Also by its definition, you can check that C is injective (since $\text{Null}(C) \subseteq \text{Null}(BC)$). But if we have an injective map from one space to another, then by rank nullity this guarantees that $\dim(Y) \geq \dim(X)$. This is precisely the inequality we wish to prove.

10.
- Consider (as the hint suggests) the space of polynomials of arbitrary degree, and the derivative map examined earlier. Any constant maps to $\mathbf{0}$, so $\text{null}T \neq \{0\}$, but the map is surjective.
- Consider polynomials of arbitrary degree, and let $Tp(t) = t \cdot p(t)$. T is linear:

$$T(\alpha p(t) + \beta q(t)) = \alpha t p(t) + \beta t q(t) = \alpha T(p(t)) + \beta T(q(t)),$$

and $\text{null}T = \{0\}$. However, there is no $\boldsymbol{v} \in V$ such that $T\boldsymbol{v}$ is constant non-zero, so T is not surjective.