EE 127 / EE 227AT 8/23/18
L. El Ghaoui, A. Bayen

Homework Assignment #1

Due date: 9/6/18, before class. Please LaTeX or handwrite your homework solution and submit an electronic version.

Exercise 1 (CVX Installation)

For Python users: Install the package CVXOPT and run the tutorial on Linear Program solvers: `http://cvxopt.org/examples/tutorial/lp.html`

For MATLAB users: Install CVX and make sure you can run the least squares code: `http://cvxr.com/cvx/doc/quickstart.html`

Solution 1

This will be important for the following homeworks!

Exercise 2 (About general optimization) In this exercise, we test your understanding of the general framework of optimization and its language. We consider an optimization problem in standard form:

$$p^* = \min_{x \in \mathbb{R}^n} \ f_0(x) \ : \ f_i(x) \leq 0, \ \ i = 1, \ldots, m.$$

In the following we denote by $\mathcal{X}$ the feasible set. For the following statements, provide a proof or counter-example.

1. Any optimization problem can be expressed as one with a linear objective.
2. Any optimization problem can be expressed as one without any constraints.
3. Any optimization problem can be recast as a linear program, provided one allows for an infinite number of constraints.
4. If one inequality is strict at the optimum, then we can remove it from the original problem and obtain the same solution.
5. If the problem involves the minimization over more than one variable, say y and x, then we can exchange the minimization order without altering the optimal value:
$$\min_x \min_y \ F_0(x, y) = \min_y \min_x \ F_0(x, y)$$
6. If the problem involves the minimization of an objective function of the form
$$f_0(x) = \max_y \ F_0(x, y),$$
then $p^* \geq d^*$, where
$$p^* := \min_x \max_y F_0(x, y)$$
$$d^* := \max_y \ \min_x \ F_0(x, y).$$
Hint: consider the function $y \to \min_{x'} \ F_0(x', y)$ and a similar function of x.

Solution 2

1. The statement is true:
$$p^* = \min_{x \in \mathcal{X}, t} \ t \ : \ t \geq f_0(x).$$
That is, we add one variable and one constraint.

2. Again the statement is true: let us define

$$g(x) := \begin{cases} f_0(x) & \text{if } x \in \mathcal{X}, \\ +\infty & \text{otherwise.} \end{cases}$$

Then

$$p^* = \min_x \; g(x).$$

3. This is true again; we have

$$p^* = \max_t \; t \;:\; t \le f_0(x) \text{ for every } x \in \mathcal{X}.$$

4. This is *not* true in general. Consider the problem

$$p^* := \min_x \; f_0(x) \;:\; |x| \le 1,$$

where

$$f_0(x) = \begin{cases} x^2 & \text{if } |x| \le 1, \\ -1 & \text{otherwise.} \end{cases}$$

The constraint $|x| \le 1$ is not active at the optimum $x^* = 0$, and $p^* = 0$. However, if we remove it, the new optimal value becomes -1.

5. The statement is true.

6. Again, the statement is true, and is referred to the *weak duality* theorem. We have, for every $x \in \mathcal{X}$, y:

$$L(y) := \min_{x'} \; F_0(x', y) \le F_0(x, y) \le U(x) := \max_{y'} \; F_0(x, y')$$

Removing the scalar in the middle, we obtain that $L(y) \le U(x)$ for every $x \in \mathcal{X}$, y. Minimizing over x, we get $p^* \ge L(y)$ for every y; maximizing over y, the statement follows.

Exercise 3 (1D Convolution) A 1D convolutional filter takes an input sequence $x = (x_t)_{t \in \mathbb{Z}}$ and a finite sequence $h = (h_t)_{t \in [1,m]}$ and produces an output sequence according to the rule

$$y_t = h_1 x_t + h_2 x_{t-1} + ... + h_m x_{t-m+1}, t \in \mathbb{Z}$$

This operation is mathematically denoted $y = h * x$ and called a convolution. Now consider a sequence x such that $x_t = 0$ for $t < 1$ and $t > n$. We will from now refer to *infinite-dimensional* vector x as the *finite-dimensional* vector $(x_1, ..., x_n)$.

1. Show that the output sequence $(y_t)_{t \in \mathbb{Z}}$ is zero outside a band. In particular, you will show that $y_t = 0$ for $t \leq a$ and $t > b$ for (a, b) that you will determine as a function of n and m. The output sequence can therefore be represented as a $(b-a)$-dimensional vector that we will note $y = (y_{a+1}, ..., y_b)$.

2. Express the relationship between the output vector y and x as $y = T(h) \cdot x$, with $T(h)$ a matrix of shape $(b - a, n)$. What structure does that matrix have?

3. We consider two n-th order polynomials p, q:

$$p(s) = p_0 s^0 + p_1 s^1 + ... + p_n s^n$$
$$q(s) = q_0 s^0 + q_1 s^1 + ... + q_n s^n$$

   Express the product of the polynomials in terms of vectors of coefficients of p, q, and powers of s, and a certain Toeplitz matrix, which you will determine.

4. A given T-vector r gives the daily rainfall in some region over a period of T days. The vector h gives the daily height of a river in the region (above its normal height). By careful modeling of water flow, or by fitting a model to past data, it is found that these vectors are (approximately) related by convolution: $h = g * r$, where $g = (0.1, 0.4, 0.5, 0.2)$. How many days after a one day heavy rainfall is the river height most affected? And, how many days does it takes for the river height to return to the normal height once the rain stops?

Solution 3

1. Since $x_t = 0$ $\forall t < 0$ and $\forall t > n$, we have that:

$$\begin{aligned}
x_0 \rightarrow y_1 =& h_1 * x_0 \text{ (all other terms are 0)} \\
x_1 \rightarrow y_1 =& h_1 * x_1 + h_2 * x_0 \text{ (all other terms are 0)} \\
&\vdots \\
x_n \rightarrow y_n =& h_1 * x_n + ... + h_m * x_{n-m+1}
\end{aligned}$$

Hence, $y_t = 0$ iff all terms are 0, which implies that either $t - m + 1 > n$ or $t \leq 0$. Rearranging, we get that $y_t = 0$ for $t \leq 0$ and $t > n + m - 1$.
Then, our $(n + m - 1)$-dimensional vector is:

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n+m-1} \end{bmatrix}$$

since for anything outside of these bounds, $y(t) = 0$.

2. WLOG, let us assume $n \geq m$. Then our matrix looks like:

$$T(h) = \begin{bmatrix} h_1 & 0 & 0 & \cdots & \cdots & \cdots & 0 \\ h_2 & h_1 & 0 & \cdots & \cdots & \cdots & 0 \\ h_3 & h_2 & h_1 & \cdots & \cdots & \cdots & 0 \\ \vdots & & \ddots & \ddots & 0 & \cdots & 0 \\ h_m & h_{m-1} & \cdots & h_2 & h_1 & \cdots & 0 \\ 0 & h_m & h_{m-1} & \cdots & h_2 & \cdots & 0 \\ \vdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & h_m \end{bmatrix}$$

Then, we see that $y = T(h) \cdot x$, as detailed in the previous part.

3. Letting C denote the vector of coefficients of the polynomial product, notice that $C = p * q$ is a convolution. Thus, applying our result from part 2, we can write C using

a Toeplitz matrix of coefficients for p, and a vector of coefficients for q:

$$\begin{aligned} C =& PQ \\ P =& \begin{bmatrix} p_0 & 0 & 0 & \cdots & 0 \\ p_1 & p_0 & 0 & \cdots & 0 \\ p_2 & p_1 & p_0 & \cdots & 0 \\ \vdots & & & \ddots & 0 \\ p_n & p_{n-1} & \cdots & \cdots & p_0 \\ \vdots & & & \cdots & 0 \\ 0 & 0 & 0 & \cdots & p_n \end{bmatrix} \\ Q =& \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix} \end{aligned}$$

If $\bar{s}$ is our vector of powers of s, we have $\bar{s} = \begin{bmatrix} s^0 \\ s \\ \vdots \\ s^{2n} \end{bmatrix}$ So, the product of the two polynomials is $PQ\bar{s}$.

4. Assuming $T \geq 6$, since $h = g * r$, we have the following relationship:

$$\begin{aligned} h_1 =& 0.1r_1 \\ h_2 =& 0.4r_2 + 0.1r_1 \\ h_3 =& 0.5r_1 + 0.4r_2 + 0.1r_3 \\ h_4 =& 0.2r_1 + 0.5r_2 + 0.4r_3 + 0.1r_4 \\ h_5 =& 0.2r_2 + 0.5r_3 + 0.4r_4 + 0.1r_5 \\ h_6 =& 0.2r_3 + 0.5r_4 + 0.4r_5 + 0.1r_6 \end{aligned}$$

Suppose the heavy rainfall occurs on day t. This one-day rainfall should have little effect on the current day (day t). The main effect would be seen on days $t + 1$ and $t + 2$. It will return to its normal level after 4 days.

Exercise 4 (Norm and angles)

1. Let $x, y \in \mathbb{R}^n$ be two unit-norm vectors, that is, such that $\|x\|_2 = \|y\|_2 = 1$. Show that the vectors $x - y$ and $x + y$ are orthogonal. Draw on the 2D plane the two vectors and all the necessary shapes to graphically solve the exercise. You may use right angles, circles and straight lines to make your point.

2. Show that the following inequalities hold for any vector $x \in \mathbb{R}^n$:

$$\frac{1}{\sqrt{n}}\|x\|_2 \leq \|x\|_\infty \leq \|x\|_2 \leq \|x\|_1 \leq \sqrt{n}\|x\|_2 \leq n\|x\|_\infty.$$

   Hint: think about using Cauchy-Schwarz's inequality, and instantiate it on the corresponding norms.

3. Show that for any non-zero vector x,

$$\text{card}(x) \geq \frac{\|x\|_1^2}{\|x\|_2^2},$$

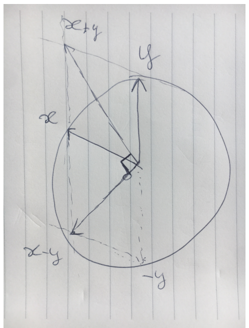
   where card(x) is the *cardinality* of the vector x, defined as the number of non-zero elements in x. Find all vectors x for which the lower bound is attained.

Solution 4 (Norm and angles)

1. When x, y are both unit-norm, we have

$$(x - y)^\top (x + y) = x^\top x - y^\top y - y^\top x + x^\top y = x^\top x - y^\top y = 0,$$

   as claimed. Let us note that we can express any vector $z \in \text{span}(x, y)$ as $z = \lambda x + \mu y$,



   for some $\lambda, \mu \in \mathbb{R}$. We have $z = \alpha u + \beta v$, where

$$\alpha = \frac{\lambda + \mu}{2}, \quad \beta = \frac{\lambda - \mu}{2}.$$

Hence $z \in \operatorname{span}(u, v)$. The converse is also true for similar reasons. Thus, (u, v) is an orthogonal basis for $\operatorname{span}(x, y)$. We finish by normalizing u, v, replacing them with $(u/\|u\|_2, v/\|v\|_2)$. The desired orthogonal basis is thus given by $((x-y)/\|x-y\|_2, (x+y)/\|x+y\|_2)$.

2. We have
$$\|x\|_2^2 = \sum_{i=1}^n x_i^2 \leq n \cdot \max_i \, x_i^2 = n \cdot \|x\|_\infty^2.$$
Also, $\|x\|_\infty \leq \sqrt{x_1^2 + \ldots + x_n^2} = \|x\|_2$.
The inequality $\|x\|_2 \leq \|x\|_1$ is obtained after squaring both sides, and checking that
$$\sum_{i=1}^n x_i^2 \leq \sum_{i=1}^n x_i^2 + \sum_{i \neq j} |x_i x_j| = \left(\sum_{i=1}^n |x_i| \right)^2 = \|x\|_1^2.$$
Finally, the condition $\|x\|_1 \leq \sqrt{n}\|x\|_2$ is due to the Cauchy-Schwarz inequality
$$|z^\top y| \leq \|y\|_2 \cdot \|z\|_2,$$
applied to the two vectors $y = (1, \ldots, 1)$ and $z = |x| = (|x_1|, \ldots, |x_n|)$.

3. Let us apply the Cauchy-Schwarz inequality with $z = |x|$ again, and with y a vector with $y_i = 1$ if $x_i \neq 0$, and $y_i = 0$ otherwise. We have $\|y\|_2 = \sqrt{k}$, with $k = \operatorname{card}(x)$. Hence
$$|z^\top y| = \|x\|_1 \leq \|y\|_2 \cdot \|z\|_2 = \sqrt{k} \cdot \|x\|_2,$$
which proves the result. The bound is attained for vectors with k non-zero elements, all with the same magnitude.

Exercise 5 (Comparing text) In this exercise, we use the word-frequency vector representation of text for comparing text documents. For mathematical modeling of transforming text into vectors, refers to Lecture 2, Slide 6 (Example 1: bag of words). In this context, similarity between two documents may be measured by means of the angle θ between the two frequency vectors representing the documents, the documents being maximally "different" when the corresponding frequency vectors are orthogonal. Consider the following headlines from the web edition of the New York Times on Dec. 7, 2010:

(a) Suit Over Targeted Killing in Terror Case Is Dismissed. A federal judge on Tuesday dismissed a lawsuit that sought to block the United States from attempting to kill an American citizen, Anwar Al-Awlaki, who has been accused of aiding Al Qaeda.

(b) In Tax Deal With G.O.P., a Portent for the Next 2 Years. President Obama made clear that he was willing to alienate his liberal base in the interest of compromise. Tax Deal Suggests New Path for Obama. President Obama agreed to a tentative deal to extend the Bush tax cuts, part of a package to keep jobless aid and cut payroll taxes.

(c) Obama Urges China to Check North Koreans. In a frank discussion, President Obama urged China's president to put the North Korean government on a tighter leash after a series of provocations.

(d) Top Test Scores From Shanghai Stun Educators. With China's debut in international standardized testing, Shanghai students have surprised experts by outscoring counterparts in dozens of other countries.

1. First simplify the text (remove plurals, convert verb to infinite tense, etc.) and then find the frequency vectors representing the text against the dictionary $V = \{\text{aid, kill, deal, president, tax, china}\}$.

2. Compare the four pieces of text by computing the cosine distance (i.e the cosine of the angles) between any pairs of texts.

Solution 5 (Comparing text)

1. The frequency vectors are

$$x^{(a)} = \begin{bmatrix} \frac{1}{3} \\ \frac{2}{3} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \; x^{(b)} = \begin{bmatrix} \frac{1}{10} \\ 0 \\ \frac{3}{10} \\ \frac{1}{5} \\ \frac{2}{5} \\ 0 \end{bmatrix}, \; x^{(c)} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{2} \\ 0 \\ \frac{1}{2} \end{bmatrix}, \; x^{(d)} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

Table 1: Cosine of angle θ between texts.

$\cos\theta$	$x^{(a)}$	$x^{(b)}$	$x^{(c)}$	$x^{(d)}$
$x^{(a)}$	1	0.0816	0	0
$x^{(b)}$	*	1	0.2582	0
$x^{(c)}$	*	*	1	0.7071
$x^{(d)}$	*	*	*	1

2. Table 1 displays the $\cos\theta$ between pairs of vectors representing the text. A high value of $\cos\theta$ suggests a high correlation between the two texts, while a $\cos\theta$ near zero means that the two texts are nearly orthogonal (uncorrelated).

Exercise 6 (Linear functions and projections)

1. Suppose x is an n-vector, with $n = 2m - 1$ and $m \geq 1$. We define the middle element value of x as x_m. Define
$$f(x) = x_m - \frac{1}{n}\sum_{i=1}^{n} x_i$$
which is the difference between the middle element value and the average of the coefficients in x. Express f in the form $f(x) = a^T x$, where a is an n-vector.

2. Now let b and x be vectors of $\mathbb{R}^2$ (with $\|b\|_2 = 1$). Draw on the 2-D plane b, x and show the value of $b^T x$ on the graph.

Now we will focus on Senator Voting data. This data provides information about senator vote x and senator political affiliation y. We provide you with four different vectors (a_1, a_2, a_3, a_4) precomputed by the EECS127 staff. Each of these vectors can be used to define a linear function $f_a : x \to a^T x$.

3. Load the files in *senator.zip*, and let X be the data matrix (with each row a senator, and each column a bill). Center the data matrix X (which is a standard preprocessing step for data analysis) and then compute for each vector a the score of each senator. We provide you with the skeleton code to load the data and visualize the scores in contrast to the affiliation y. You will compute such plots using the existing code and explain them.

4. Which direction among the vectors a_1, a_2, a_3, a_4 do you prefer in order to produce a score that effectively summarizes the affiliation? Justify your answer.

Solution 6 (Linear functions and projections)

1. We can rewrite the first term as
$$\begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}^T x$$
where the nonzero entry is at the middle element, and the second term as

$$\begin{bmatrix} \frac{1}{n} \\ \vdots \\ \frac{1}{n} \end{bmatrix}^T x$$

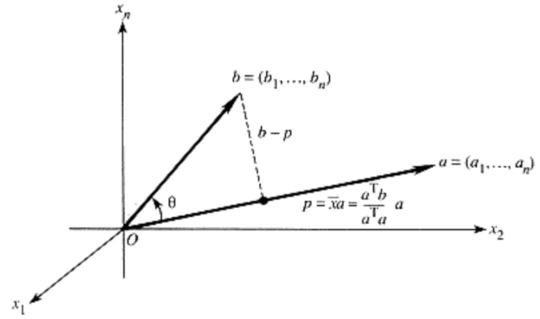
Together, we express f as

$$f(x) = a^T x$$

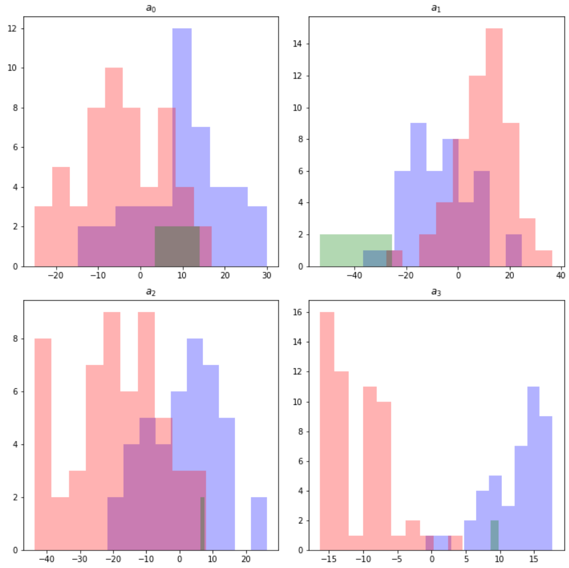
where

$$a = \begin{bmatrix} \frac{-1}{n} \\ \vdots \\ 1 - \frac{1}{n} \\ \vdots \\ \frac{-1}{n} \end{bmatrix}.$$

2. Without loss of generality, assume a is a unit vector. (If not, we can easily scale it.) Recall the projection of x on a is given by $\mathbf{proj}_a x = (a^T x)a$. $a^T x$ is simply the component of x going in the direction of a.



Credit: `http://www.ling.upenn.edu/courses/Spring_2005/ling525/linear_algebra_review.html`

3. 

We invite you to look at the Jupyter Notebook "senator-data-sols.ipynb"

4. We would pick a_3, as it shows a clearer divide between the 2 groups.

Exercise 7 (Customer purchase history matrix) A store keeps track of its sales of products from K different product categories to N customers over some time period, like one month. K might be on the order of 1000 and N might be 100000. The data is stored in an $N \times K$ matrix C, with C_{ij} being the total dollar purchases of product j by customer i All the entries of C are nonnegative. The matrix C is typically sparse, i.e., many of its entries are zero.

1. What is $C\mathbf{1}$? ($\mathbf{1}$ is a K-vector of 1s)
2. What is $C^T\mathbf{1}$? ($\mathbf{1}$ is a N-vector of 1s)
3. Give a short matrix-vector expression for the total dollar amount of all purchases, by all customers.
4. What does it mean if $(CC^T)_{kl} = 0$? Your answer should be simple English.

We are now interested in efficient computations. In our setting, note that the data matrix C is large but very sparse. The number of non zero-valued elements divided by the total number of elements is called the density d of the matrix C. Let w in $\mathbb{R}^K$ be a given weighting vector. Assume that we center the rows (removing the average row to every row), obtaining a new row-centered matrix C_m.

5. Provide an explanation on how to efficiently compute the matrix vector product $v = C_m w$.
6. Mathematically estimate the gain speed of your method vs the naive method (of forming C_m first) as a function of K, N, and d.
7. Confirm your findings with an empirical study: we provide you with a matrix C and a vector w in the Jupyter Notebook matrix-vector.ipynb. Using the skeleton code, compare the sparse matrix method against the naive method. Comment on the time improvement and attach your notebook here in pdf format for grading. Let us note that we do not ask the students to create their own sparse matrix functionalities.
8. Assume that we add one row (data point). Explain how to efficiently update the resulting vector v accordingly.

Solution 7

1. This is $\sum_{j=1}^{K} C_{ij}$ for every (fixed) i. The i^{th} entry represents the total amount of money customer i spent.
2. This is $\sum_{i=1}^{N} C_{ij}$ for every (fixed) j. The j^{th} entry is the total amount of money spent by all customers on product j.

3. $\mathbf{1}^T C\mathbf{1}$, where the first $\mathbf{1} \in \mathbb{R}^N$ and the second $\mathbf{1} \in \mathbb{R}^K$

4. Denoting c_i to be the purchase history a customer i. We have that

$$CC^T = \begin{bmatrix} c_1^T \\ \vdots \\ c_N^T \end{bmatrix} \begin{bmatrix} c_1 \cdots c_N \end{bmatrix}$$

   Thus $(CC^T)_{kl} = c_k \cdot c_l$. If $k = l$ and $(CC^T)_{kl} = 0$, then customer k did not purchase anything. If $k \neq l$ and $(CC^T)_{kl} = 0$, then customer k and customer l did not purchse any products in the same category.

5. Denote the average of the rows r_{avg}. The operation we want to perform would be $(C - R_{avg})w$ where R_{avg} is the matrix obtained by stacking r_{avg}. A (very naive) implementation would be directly computing $C_m = C - R_{avg}$ and then right multiply it by the vector w. A better approach would be to first form Cw and then $R_{avg}w$ which is simply a vector full of $r_{avg} \cdot w$. The gain comes from the fact that Cw is the most expensive operation and can be optimized using sparse multiplication.

6. The density d represents the proportion of non-zero elements, that also counts how much fewer operations we need. Letting n be the number of nonzero elements, we have (a) Naive: $\mathcal{O}(NK)$ for finding C_m and $\mathcal{O}(NK)$ for matrix-vector multiplication. (b) Efficient: Find sparse representation of r_{avg} in $\mathcal{O}(n)$, of $r_{avg} \cdot w$ in $\mathcal{O}(n)$, of Cw in $\mathcal{O}(n)$. Total time is $\mathcal{O}(n) = \mathcal{O}(NKd)$. Finally, the speed gain is a factor of d.

7. We can see that the naive implementation is the slowest because of 1. centering (subtracting from c_{avg} takes $O(NK)$ in runtime) 2. multiplication (also $O(NK)$ in runtime)

   In the efficient implementation on dense matrix, by distributing the multiplication, we can simplify the operation on $[c_{avg} ... c_{avg}]\, v$, but Cv is still slow because of dense matrix.

   In the efficient implementation on the sparse matrix, now both operations have been **optimized**.

   In the efficient implementation on the sparse matrix, now both operations have been optimized. We invite you to look at the Jupyter Notebook "matrix-vector-sols.ipynb"

8. We simply compute the dot product of the newly added row and w using the same technique as above, and append the value to the vector v.