University of Toronto
Department of Electrical and Computer Engineering

ECE367 MATRIX ALGEBRA AND OPTIMIZATION

Problem Set #1

Autumn 2018

Prof. S. C. Draper **Due:** In class on Tuesday, 25 Sept. 2018

Homework policy: Problem sets must be turned in in class on the due date. Late problem sets will not be accepted. See information sheet for full discussion of problem set policy. Some problems are drawn from the course text "Optimization Models" by Giuseppe Calafiore and Laurent El Ghaoui, abbreviated as "OptM", others from the text "Introduction to Applied Linear Algebra", abbreciated "IALA", by Stephen Boyd and Lieven Vandenberghe. An electronic version of the latter is available on the authors' websites. In both cases we indicate the appropriate problem number and name.

Grading: Grading is not uniform. Typically the numerical/design questions are expected to take longer and, correspondingly, are more heavily weighted in the final grade.

Problem Set #1 problem categories: A quick categorization by topic of the problems in this problem set is as follows:

- Norms: Problems 1.1, 1.2
- Linear independence and orthogonality: Problems 1.3-1.5
- Inner products: Problems 1.6-1.8
- Function approximation: Problems 1.9-1.13

Problem 1.1 (Showing ℓ_1 and ℓ_∞ are both norms)
Show

(a) that the functions $\ell_1 : \mathbb{R}^n \to \mathbb{R}$ is a norm, and

(b) that $\ell_\infty : \mathbb{R}^n \to \mathbb{R}$ is a norm

by verifying that they satisfy all the properties of a norm, cf., OptM Definition 2.1.

Problem 1.2 (Norm inequalities)
OptM Prob. 2.6.
Note: This problem shows that each norm (l_1, l_2, l_∞) is both upper- and lower-bounded by each of the other norms to within a constant (dimension-dependent) factor.

Problem 1.3 (Linear independence of stacked vectors)
IALA Prob. 5.1.

Problem 1.4 (Orthogonality)
OptM Prob. 2.5.

Problem 1.5 (Gram-Schmidt algorithm)
IALA Prob. 5.6.

Problem 1.6 (Inner products)
OptM Prob. 2.4.

Problem 1.7 (Distance versus angle nearest neighbors)
IALA Problem 3.24.

Problem 1.8 (Angles between word vectors)
In this problem you investigate how geometric concepts such as distance and angle can be applied to quantify similarity between text documents. Download the files `wordVecArticles.txt`, `wordVecTitles.txt`, `wordVecWords.txt` and `wordVecV.mat` from the course website. The first two files each have ten lines. Each line in the first file consists of the text of one Wikipedia article. The corresponding line of the second file is the title of the article. The last two files are describe in detail below.

Denote by $\mathcal{D}$ the set of documents where the number of documents is $|\mathcal{D}|$. (In our dataset $|\mathcal{D}| = 10$). Let $\mathcal{W}$ denote the union of words in all articles, i.e., the lexicon of the set of documents. We denote the cardinality of $\mathcal{W}$ by $|\mathcal{W}|$. Assume the lexicon is ordered "lexiographically" (e.g., alphabetically) so that there is a one-to-one mapping from each word $w \in \mathcal{W}$ to an element of the index set $t \in [|\mathcal{W}|]$. Let $f_{\text{term}}(t, d)$ denote the number of times the word $w \in \mathcal{W}$ that is indexed as $t \in [|\mathcal{W}|]$ appears in the dth article where $d \in [|\mathcal{D}|]$. Note that $\sum_{t=1}^{|\mathcal{W}|} f_{\text{term}}(t, d)$ is the number of words (the length) of the dth article. We refer to $f_{\text{term}}(t, d)$ as the *term frequency* (really "term count").

For the first few parts of this problem you will be using a pre-processed $\mathcal{W}$ set and pre-computed $f_{\text{term}}(t,d)$ values. The pre-processed data appears in the files `wordVecWords.txt` and `wordVecV.mat`. The first file represents the set $\mathcal{W}$ where elements of $\mathcal{W}$ are listed line by line, for 1651 lines, i.e., $|\mathcal{W}| = 1651$. You can load the content in the second file into MATLAB by using command `load 'wordVecV.mat'`. After loading, you will see a matrix `V` of dimensions 1651×10. The value in the tth row and dth column of this matrix is $f_{\text{term}}(t,d)$. Use the provided data in `V` to answer parts (a) to (d) of this problem.

(a) Let the $|\mathcal{W}|$-dimensional vectors v_d, $d \in [|\mathcal{D}|]$ be defined as $v_d = (f_{\text{term}}(1,d), f_{\text{term}}(2,d), \ldots, f_{\text{term}}(|\mathcal{W}|,d))$. Using v_d to represent the dth document, which two articles are closest in Euclidean distance (smallest distance)? Which two are closest in angle distance (smallest angle)? Are they the same pair, if not, what could be a reason for them being different?

(b) In this part let the $|\mathcal{W}|$-dimensional *normalized* vectors $\tilde{v}_d$, $d \in [|\mathcal{D}|]$ be defined as $\tilde{v}_d = v_d / \sum_{t=1}^{|\mathcal{W}|} f_{\text{term}}(t,d)$, where the v_d are defined as in the previous part. Using $\tilde{v}_d$ to represent the dth document, which two articles are closest in Euclidean distance (smallest distance)? Which two are closest in angle distance (smallest angle)? Are your answers the same as in the previous part? What would be a reason for using this normalization?

Now, let $f_{\text{doc}}(t) = \sum_{d=1}^{|\mathcal{D}|} \mathbb{I}[f_{\text{term}}(t,d) > 0]$ where $\mathbb{I}(\cdot)$ is the indicator function taking value one if the clause is true and zero else. The function $f_{\text{doc}}(t)$ counts in how many documents the tth word appears. We refer to $f_{\text{doc}}(t)$ as the *document frequency*.

We combine the term and document frequency definitions into what is called the *term frequency-inverse document frequency score* (TF-IDF), defined as

$$w(t,d) = \frac{f_{\text{term}}(t,d)}{\sum_{t=1}^{|\mathcal{W}|} f_{\text{term}}(t,d)} \sqrt{\log\left(\frac{|\mathcal{D}|}{f_{\text{doc}}(t)}\right)}.$$

Note, the denominator of the log is never zero since, by definition, each term appears in at least one document.

(c) Now let the $|\mathcal{W}|$-dimensional vectors w_d, $d \in [|\mathcal{D}|]$ be defined as $w_d = (w(1,d), w(2,d), \ldots, w(|\mathcal{W}|,d))$. Using w_d to represent the dth document, which two articles are closest in Euclidean distance (smallest distance)?

(d) What might be a reason for using the "inverse document frequency" adjustment? What is the adjustment doing geometrically?

OPTIONAL PART: The following part (e) is optional. It's worth is equal to 50% of parts (a)-(d). Your score on this part will be added to your overall score for the problem set, which will then

be the max of your score or 100%. (This means that if you get all other parts correct your score will not be increased through completion of this problem part.)

(e) In the previous parts of the problem you used the pre-processed $\mathcal{W}$ set and pre-computed $f_{\text{term}}(t, d)$ values provided in `wordVecV.mat` and term indexes in `wordVecWords.txt`. In this part of the problem you will reproduce your results from (a) to (d) without using this pre-computed data. Specifically, start from the raw text files `wordVecArticles.txt` and `wordVecTitles.txt`, and write code to obtain $\mathcal{W}$ and $f_{\text{term}}(t, d)$. As always, you are welcome to use whichever software language you wish to solve this problem. We note that Python is particularly well suited to text processing. For example, you may find the snippet of Python code following the problem statement useful. This snippet loads in data from the given text file and stores it in the variable `articles`. It also counts the number of word occurrences in the first article and stores the resulting (word, count) pairs in the variable `wordcounts`. You could use this as a starting point in the generation of the vectors we provide in `wordVecV.mat` and then calculate angles and distances in MATLAB. Alternately, you could show how to complete the entire processing pipeline in Python. Be sure to include a printout of your code.

```
from collections import Counter
articles = [line.rstrip('\n') for line in open('wordVecArticles.txt')]
wordcounts = Counter(articles[0].split())
```

Problem 1.9 (First-order approximation of functions)
In this exercise you will write MATLAB code to plot linear approximations each of three functions $f_i : \mathbb{R}^2 \to \mathbb{R}$, $i \in [3]$. The three functions are defined pointwise as

$$\begin{aligned} f_1(x, y) &= 2x + 3y + 1, \\ f_2(x, y) &= x^2 + y^2 - xy - 5, \\ f_3(x, y) &= (x - 5)\cos(y - 5) - (y - 5)\sin(x - 5). \end{aligned}$$

For each of the above function, do the following.

(a) Write down the gradient with respect to x and y in closed form. The gradient can be compactly written in the form $\nabla f_i = \left[\frac{\partial f_i}{\partial x} \frac{\partial f_i}{\partial y}\right]^T$ for $i = 1, 2, 3$.

(b) For each function produce a 2-D contour plot indicating the level sets of each function in the range $-2 \leq x, yleq3.5$. (I.e., make three plots.) An example of a contour plot isillustrated in Fig 2.28 of OptM (the second sub-figure). You may find `contour` command in MATLAB useful. In addition, compute the the gradient at the point $(x, y) = (1, 0)$ for each function. On your contour plots also plot the direction of the gradient and the tangent line to the level sets. Your resulting plot should be analogous to Fig 2.29 of OptM.

(c) For the same point $(x, y) = (1, 0)$ where, plot the 3-D linear approximation of the function. Since we are considering only the first derivative, the approximation should be the tangent

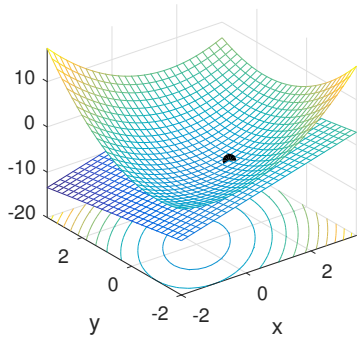

Figure 1: Example plot and approximating tangent plane.

plane at the specified point. Your plot for $f_2(x, y)$ should look something like Fig. 1. The function approximation is plotted as a tangent plane to the surface plot of $f_2(x, y)$.Include your code for all sections.

Note: We recommend you design these plotting scripts as Matlab functions so that you can reuse them for to plot approximations for different non-linear functions (or for theses functions at different points). In either case make sure to attach your code.

Problem 1.10 (Computing projections in Euclidean space)
In this problem we use the notation $\text{Proj}_{\mathcal{S}}(x)$ to denote the projection of a vector x onto some set $\mathcal{S}$, which consists of vectors that are of same dimension as x. Consider the following vectors and subspaces.

$$x_1 = \begin{bmatrix} 3 \\ -1 \end{bmatrix}, \quad b_1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \quad \mathcal{V}_1 = \text{span}\left\{ \begin{bmatrix} 1 \\ 2 \end{bmatrix} \right\}$$

$$x_2 = \begin{bmatrix} 2 \\ 1 \\ -5 \end{bmatrix}, \quad b_2 = \begin{bmatrix} 0 \\ -3 \\ -1 \end{bmatrix}, \quad \mathcal{V}_2 = \text{span}\left\{ \begin{bmatrix} 0 \\ 0 \\ -2 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right\}$$

$$x_3 = \begin{bmatrix} 3 \\ 0 \\ -1 \\ 2 \\ 2 \end{bmatrix}, \quad b_3 = \begin{bmatrix} -1 \\ 0 \\ 1 \\ -2 \\ 1 \end{bmatrix}, \quad \mathcal{V}_3 = \text{span}\left\{ \begin{bmatrix} 0 \\ 1 \\ 2 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 3 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 5 \\ 0 \\ 1 \end{bmatrix} \right\}$$

(a) Compute $\text{Proj}_{\mathcal{V}_i}(x_i)$ for $i = 1, 2, 3$.

(b) Consider the affine set $\mathcal{A}_i = \{v + b_i \mid v \in \mathcal{X}_i\}$. Compute $\text{Proj}_{\mathcal{A}_i}(x_i)$ for $i = 1, 2, 3$.

(c) On a 2-d map, plot the subspace $\mathcal{V}_1$ (a line through the origin) and clearly indicate x_1 and $\text{Proj}_{\mathcal{V}_1}(x_1)$. What is the point on $\mathcal{V}_1$ that is the closest to x_1 in Euclidean sense? On the same axes, plot $\mathcal{A}_1$ (a line shifted from the origin) and indicate $\text{Proj}_{\mathcal{A}_1}(x_1)$.

(d) Compute an orthonormal basis $\mathcal{B}_3$ for the subspace $\mathcal{V}_3$ via Graham-Schmidt. Recompute $\text{Proj}_{\mathcal{V}_3}(x_3)$ and $\text{Proj}_{\mathcal{A}_3}(x_3)$ using $\mathcal{B}_3$, and compare with your previous results.

Problem 1.11 (Approximating a square wave)
In this problem you will approximate a given discretized square wave signal using a *sinusoidal basis*. The following MATLAB function produces the discrete approximation of two periods of a period-10 square wave signal $x(t)$ where $t \in [0, 20]$, discretized in increments of 0.0001 seconds. In addition, the script also produces 30 orthogonal basis vectors that we will later use to approximate the given signal. The function returns two vectors `time_pos` and `sq_wave`, and a matrix `B_unnorm`. The vector `sq_wave` consists of the signal output for each time position specified in `time_pos` vector. The matrix `B_unnorm` includes 30 unnormalized row vectors that are of the same length as `sq_wave`.

```
function [time_pos, sq_wave, B_unnorm] = generate_data
    n_comps = 30; period = 10; fundFreq = 1/period;
    time_pos = 0:0.0001:2*period; harmonics = 2*(1:n_comps)-1;
    sq_wave = floor(0.9*sin(2*pi*fundFreq*time_pos))+.5;   %% generate the signal
    B_unnorm = sin(2*pi*fundFreq*(harmonics'*time_pos))/2; %% generate the basis
end
```

(a) Plot the square wave signal `sq_wave` against `time_pos`.

(b) How would you numerically test for the orthogonality of the basis vectors (rows of `B_unnorm`)? Plot the first 6 and 30th basis vectors against `time_pos` in separate sub-plots that share the same time axis.

(c) Normalize the given basis vectors to obtain an orthonormal basis. Project the square wave signal onto the normalized basis vectors using ℓ_2 projection and compute the projection coefficients. Arrange the basis vectors in the decreasing order of the magnitude of the projection coefficients. Approximate the square wave using the first $1, 2, 3, 4, 5, 6, 30$ basis vectors, and plot the approximations in separate sub-plots that share the same time axis. Plot the original signal on top of each approximation and compare.

Problem 1.12 (Projection with different norms)
Consider the vector $x = [3, 2]^T$ and the line defined by the set $A = \{v_0 + tv \mid t \in \mathbb{R}\}$ where $v_0 = [-2, -4]^T$ and $v = [-2, 5]^T$. The projection of x on to the affine set $\mathcal{A}$ under ℓ_p norm, which we denote by $y^{(p)}$, is the minimizer of optimization problem

$$\begin{aligned} \min_{y} \quad & \|x - y\|_p \\ \text{subject to} \quad & y \in \mathcal{A}. \end{aligned}$$

(a) Plot the line $\mathcal{A}$ and indicate x on a 2-d map. Without solving for $y^{(2)}$, plot the smallest norm ball under ℓ_2 norm that includes $y^{(2)}$, and mark the position of $y^{(2)}$.

(b) Write down the solution for $y^{(2)}$ in closed-form and solve for $y^{(2)}$.

(c) Repeat part (a) for $p = 1$ and $p = \infty$ cases.

(d) The solutions for $y^{(1)}$ and $y^{(\infty)}$ can not be expressed in closed-form, but they can be computed using *linear programs* (LPs). In this part we use the software package CVX which runs in MATLAB and is available at `http://cvxr.com/cvx/`. You may find the CVX package useful throughout this course. For more information, please check the user guide available at the above website.

As an example, after installing CVX, the following MATLAB function solves for $y^{(\texttt{nrm})}$. Execute the function with given parameters and verify your result obtained in part (b).

```
function [y, r] = proj_cvx(x, v0, v, nrm)%% x, v0 and v must be column vectors
    objtv = @(y) norm(x-y, nrm);         %% objective is L_2 norm
    cvx_begin
        variable y(2)                    %% 2-d variable we are optimizing over
        variable t(1)                    %% real valued parameter that defines
        minimize(objtv(y))               %% defining the objective
        subject to
        v0 + t*v == y                    %% the projection y must be in set A
    cvx_end
    r = objtv(y);                        %% minimum value of the objective
end
```

(e) Use the given function to solve for $y^{(1)}$ and $y^{(\infty)}$. Include your MATLAB code with the answers.

Problem 1.13 (Inner products and projection in function spaces)
While the focus of this course is on finite (and mostly real) vector spaces, notions of inner products spaces and the approximation of a vector by its projection into a subspace, also hold for infinite-dimensional vector spaces where each vector is a function $f : \mathbb{R} \to \mathbb{R}$. In this problem let $C[-\pi, \pi]$ denote the set of continuous real-valued functions on $[-\pi, \pi]$. Observe that one can add and scale functions pointwise, thereby defining $C[-\pi, \pi]$ to be a vector space. We pick

$$\langle f, g \rangle = \int_{-\pi}^{\pi} f(x)g(x)dx \tag{1}$$

to be the inner product of two functions $f, g \in C[-\pi, \pi]$. One can verify that all properties of an inner product are satisfied by (1). The norm induced by this inner product is

$$\|f\| = \sqrt{\langle f, f \rangle} = \sqrt{\int_{-\pi}^{\pi} (f(x))^2 dx}.$$

In this problem you consider the the function $g : [-\pi, \pi] \to \mathbb{R}$ defined pointwise as $g(x) = \sin(x)$. Your task is to find the best approximation to g in the subspace of $C[-\pi, \pi]$ that consists of polynomials with real coefficients and degree at most five. We denote this subspace as $\mathcal{U} = \{u | u(x) =$

$\alpha_0 + \alpha_1 x + \alpha_2 x^2 + \alpha_3 x^3 + \alpha_4 x^4 + \alpha_5 x^5, \alpha_i \in \mathbb{R} \ \forall \ i \in \{0, 1, \ldots 5\}\}$. By best we mean that the optimizing u^* is

$$u^* = \arg\min_{u \in \mathcal{U}} \|g - u\|^2.$$

where, to be explicit,

$$\|g - u\|^2 = \langle g - u, g - u \rangle = \int_{-\pi}^{\pi} (g(x) - u(x))^2 dx = \int_{-\pi}^{\pi} (\sin(x) - u(x))^2 dx.$$

In particular, do the following

(a) First apply the Gram-Schmidt procedure using the inner production defined in (1) to the basis $\{v^{(0)}, v^{(1)}, v^{(2)}, v^{(3)}, v^{(4)}, v^{(5)}\}$ of $\mathcal{U}$ where, defined pointwise, $v^{(i)}(x) = x^i$, $i \in \{0, 1, 2, 3, 4, 5\}$ to produce an orthonormal basis $\{e^{(0)}, e^{(1)}, e^{(2)}, e^{(3)}, e^{(4)}, e^{(5)}\}$ where each $e^{(i)} \in C[-\pi, \pi]$.

(b) Next, using (1) and the formulas for ℓ_2 projection we developed in class (which apply again here), compute $\alpha_0, \ldots, \alpha_5$. (To check correctness note that, up to numerical precision, you should find that $\alpha_1 = 0.987862$.)

(c) You should see a sensical pattern to the even coefficients $\alpha_0, \alpha_2, \alpha_4$. What is the pattern you observe and why is it intuitively correct?

(d) Another often used polynomial approximation to the $\sin x$ function the Taylor approximation

$$\sin x \simeq x - \frac{x^3}{3!} + \frac{x^5}{5!}.$$

Plot your approximation from part (b) against the Taylor approximation. You should observe that your approximation looks much better over the entire interval $[-\pi, \pi]$. Where is the Taylor approximation accurate and where is it not accurate? What was it about the formulation of the ℓ_2 projection problem that makes your approximation better in the regions where the Taylor approximation is not good?

