University of Toronto
Department of Electrical and Computer Engineering

ECE367 MATRIX ALGEBRA AND OPTIMIZATION

Problem Set #1

Autumn 2021

Prof. S. C. Draper **Due:** 8pm (Toronto time) Sunday, 19 September 2021

Homework policy: Problem sets must be turned by the due date and time. Late problem sets will not be accepted. See the information sheet for futher details. The course text "Optimization Models" is abbreviated as "OptM" and "Introduction to Applied Linear Algebra" as "IALA".

Problems are categorized as

- **"Theory" problems:** These are mostly mathematical questions designed to give you deeper insight into the fundamentals of the ideas introduced in this class.
- **"Application" problems:** These questions are designed to expose you to the breadth of application of the ideas developed in class and to introduce you to useful numerical toolboxes.
- **"Optional" problems:** Optional problems provide extra practice or introduce interesting connections or extensions. They need not be turned in. I will assume you have reviewed and understood the solutions to the optional problems when designing the exams.

Hand-in procudure:

- **Initial submission:** Your initial submission of the "Theory" and "Application" questions must be submitted via Quercus upload by the due date. Click on the `Assignments` tab, then look for the `Initial submission` tab and upload under the correct problem set number.
- **Self-assessment:** After the problem set is due we will post solutions. You will have one week from the initial due date to submit a commented version of your assignment in which, using as a reference the posted solutions, you highlight your errors or omissings in red. Annotate the PDF you initially submitted. If you have not submitted a solution you cannot submit the self-assessment. To submit the self-assessment on Quercs, click on the `Assignments` tab, then look for the `Self-assessment` tab and upload under the correct problem set number.
- **Late problem sets are not accepted**
- **Grading:** Per the course handout problem sets are graded for completion only. Points are assigned to (i) Initial submission of theory part, (ii) Submission of application part, (iii) Self-assessment of theory part. The relative points breakdown is detailed in each problem set.

Points allocation

- Theory parts (initial assessment): 1 pt
- Application parts (initial assessment): 1 pt
- Theory parts (self-assessment): 1 pt

Problem categorization and main concepts covered

Theory

- Norms: Problems 1.1, 1.2
- Linear independence and orthogonality: Problems 1.3-1.4
- Inner products: Problems 1.5-1.6

Application

- Inner products: Problems 1.10

Optional

- None in this problem set

Final note and recommendation: If it feels it has been a long time since you took linear algebra, and you are looking for additional resources and practice problems, I direct you to "Introduction to linear algebra" by Gilbert Strang, Wellesley-Cambridge Press. This classic text is the text I used to learn linear algebra. Prof. Strang is a fantastic teacher. His website has lots of online resources, including old exams. Please follow the following link: link

THEORY PROBLEMS

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 (Inner products)
OptM Prob. 2.4.

Problem 1.6 (Distance versus angle nearest neighbors)
IALA Problem 3.24.

APPLICATION PROBLEMS

Problem 1.7 (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 described 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? You may find the `pdist` command in MATLAB useful for computing pairwise Euclidean and angle distances of vectors.

(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 a *optional* 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())
```

