Homework 4: Clustering, Recommenders and Dim. Reduction  
(due April 10th, 2019, 2:30pm, in class—hard-copy please)

Reminders:
- Out of 50 points. Contains 3 pages.
- Rough time-estimates: 4-6 hours.
- Please type your answers. Illegible handwriting may get no points, at the discretion of the grader. Only drawings may be hand-drawn, as long as they are neat and legible.
- There could be more than one correct answer. We shall accept them all.
- Whenever you are making an assumption, please state it clearly.
- Each HW has to be done individually, without taking any help from non-class resources (e.g. websites etc).
- Important: Feel free to use any programming language you want for this homework, but we recommend using numerical software like Matlab/R/Python packages like scipy and numpy.

Q1. Dimensionality Reduction [20 points]
We will use Singular Value Decomposition to investigate a cloud of points. Consider the 5-dimensional dataset on a linear manifold here:  
http://people.cs.vt.edu/~badityap/classes/cs5614-Spr19/homeworks/hw4/svd.dat

Each point in the point cloud lies roughly on a 2-d plane spanned by two vectors v1 and v2 with an additional noise term $\varepsilon_i \sim \mathcal{N}(0, \Sigma)$ which follows a Gaussian distribution. More specifically $\Sigma = 6I_{5,5}$ where $I_{5,5}$ is a 5x5 identity matrix.

Q1.1. (10 points) Use SVD to visualize the dataset in 2-dimensional space by projecting in the best 2 dimensions. FYI: SVD is implemented in MATLAB/R as a native function.

Q1.2. (5 points) There are two outliers in the dataset. Find those two outliers by manually looking at your visualization in 2-dimensional space. Print your Plot and mark the outliers with your pen, and submit this plot. Also, report the 2-d coordinates of the outliers in the 2-d (‘concept’) space.

Q1.3. (5 points) We want to reverse engineer the two outliers from the concept space to original space. What is the formula we need to find 5-d coordinates of a point in the original space, given the 2-d coordinates in the projected (‘concept’) space? For example, the point (0,0) in ‘concept space’ corresponds to (0,0,0,0,0) in the original space.
**Code Deliverables:** In addition to all the plots and answers, copy-paste all the code that you wrote in your hard-copy.

**Q2. Clustering [25 points]**

Here we will investigate the K-Means clustering algorithm we saw in class. Formally, in K-means clustering, we are given points $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$ and an integer $K > 1$, and our goal is to minimize the within-cluster sum-of-squares error (essentially the k-means objective) i.e. find $L$ and $C$ so as to minimize the term:

$$J(C, L) = \sum_{i=1}^{n} \| x_i - C_{l_i} \|^2$$

where $C = (C_1, C_2, \ldots, C_K)$ are the cluster centers and $L = (l_1, l_2, \ldots, l_n)$ are the cluster assignments for the data points i.e. $C_j \in \mathbb{R}^d$ and $l_i \in \{1, \ldots, K\}$.

Finding the exact minimum of the function $J(C, L)$ is very hard. We saw the most common algorithm for finding an approximate solution in class (it is also called Lloyd’s algorithm). For sake of clarity we repeat it here. The algorithm takes additionally as input some initial set of cluster centers $C$, and then proceeds as follows:

1. Keeping $C$ fixed, find cluster assignments $L$ to minimize $J(C, L)$. This step involves basically finding the nearest cluster center for each data point (ties can be broken using arbitrary but consistent rules).
2. Keeping $L$ fixed, find $C$ to minimize $J(C, L)$. This just requires finding the centroid of each cluster (the cluster centers).
3. If any of the values in $L$ changes from the previous iteration, repeat 1-3.
4. Return $C$ and $L$.

The initial cluster centers $C$ are often picked randomly from the given data points. In practice, we often need to repeat multiple runs of Lloyd’s algorithm with different initializations and pick the best clustering in terms of the k-means objective ($J(C, L)$).

**Q2.1.** (5 points) Explain why Lloyd’s algorithm is always guaranteed to converge (i.e. stop) in a finite number of steps.

*Hint:* Note that the objective function is non-negative (why?). Now think about what is happening to it.

**Q2.2.** (10 points) Implement Lloyd’s algorithm. Use the 500 2-d points dataset given here: [http://people.cs.vt.edu/~badityap/classes/cs5614-Spr19/homeworks/hw4/cluster.csv](http://people.cs.vt.edu/~badityap/classes/cs5614-Spr19/homeworks/hw4/cluster.csv) as input.

Run it until convergence 200 times, each time initializing using $K$ cluster centers picked at random from the input data set with $K=5$ clusters. Plot in a single figure the original data points (in grey), and all the 200x5 final cluster centers (in
black) given by each run of Lloyd’s algorithm. You can play around with the plotting options such as point sizes so that cluster centers are clearly visible. Also compute the minimum, mean, and standard deviation of the $J(C, L)$ values for the clusterings given by each of the 200 runs.

Q2.3. (10 points) We want to select the best number of clusters $K$ for this dataset (for Q2.2 above we silently used $K=5$). Recall that we typically look for a “knee” in the average-distance vs $K$ plot, to select the “best” $K$. Produce such a plot for $K=1, 2, \ldots, 15$ using the same dataset in Q2.2. More specifically for each value of $K$, run your $k$-means implementation with at least 200 random initializations, and pick the best resulting clustering out of these 200 runs (in terms of the $J(C, L)$ value). Then plot the best $J(C, L)$ value on the y-axis for each $K$, and the corresponding $K$ value in the x-axis, and find the knee.

*Code Deliverables:* Again, in addition to all the plots and answers, copy-paste all the code that you wrote in your hard-copy.

Q3. Recommendation Systems [5 points]
Consider the following utility matrix, representing ratings on a 1-5 star scale of eight items a-h by three users A, B, and C.

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
<th>h</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Treat ratings of 3, 4, and 5 as 1 and 1, 2 and blank as 0. Compute the Jaccard, and cosine distances between each pair of users. Which one is better?