Homework 1 – CS 5525

1. *(a)* Two vectors $\mathbf{x}$ and $\mathbf{y}$ have zero mean. What is the relationship of the cosine measure and correlation between them?

   Solution : The cosine measure is given by

$$\cos(\mathbf{x},\mathbf{y}) = \frac{\mathbf{x}^\top\mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|}$$

   Let us now compute the correlation,

$$\mathrm{corr}(\mathbf{x},\mathbf{y}) = \frac{\mathrm{cov}(\mathbf{x},\mathbf{y})}{\mathrm{std}(\mathbf{x})\mathrm{std}(\mathbf{y})}$$

   Accounting for the zero means, $\mathbb{E}(\mathbf{x}) = \mathbb{E}(\mathbf{x}) = 0$ and substituting the expressions for covariance and standard deviations, we obtain

$$\mathrm{corr}(\mathbf{x},\mathbf{y}) = \frac{(\sqrt{n}-1)(\sqrt{n}-1)\mathbf{x}^\top\mathbf{y}}{(n-1)\|\mathbf{x}\|\|\mathbf{y}\|} = \cos(\mathbf{x},\mathbf{y}).$$

   Thus the cosine similarity and correlation between zero-mean vectors are the same

   (b) Derive the mathematical relationship between cosine similarity and Euclidean distance when each data object vector has an L2 length (magnitude) of 1.

   Solution : For unit length data points $\mathbf{x}$ and $\mathbf{y}$, we have

$$\cos(\mathbf{x},\mathbf{y}) = \frac{\mathbf{x}^\top\mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|} = \mathbf{x}^\top\mathbf{y}$$

   The Euclidean distance may be computed as

$$\mathrm{dist}(\mathbf{x},\mathbf{y}) = (\mathbf{x}-\mathbf{y})^\top(\mathbf{x}-\mathbf{y}) = \|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 - 2\mathbf{x}^\top\mathbf{y} = 2(1-\cos(\mathbf{x},\mathbf{y}))$$

   Thus for unit-length vectors, we have Euclidean dist= 2(1−cosine similarity)).

2. Consider this hypothetical problem. You have a set of news articles $\mathcal{A} = A$ given to you, where A indicates one article. These news articles span across different domains like civil unrest, earthquakes, sport, etc. Assume that $\mathcal{A}_e$ is the set of news articles related to event e. An event e can fall under any domain D. On aggregating the news articles of event e of particular domain D, we get the domain-set $\mathcal{A}_D = \{A | A \in \mathcal{A}_e \forall e \in D\}$. Suggest two measures domain weight $C_{w_i,D}$ and event weight $E_{w_i,e}$ for a word w_i. $C_{w_i,D}$ quantifies the ability of word w_i in representing targeted domain D. $E_{w_i,e}$ quantifies the ability of word w_i in distinguishing event e from other events in the same domain. Make use of this function $f(w, A)$ which gives the frequency of word w in article A. You are free to make assumptions like merging all the articles related to one event into a single article.

Solution : Given a word w_i, let us quantify how frequently this word appears in the articles relating to a domain D. To do so, let us define *term domain frequency* of word w_i in domain D as

$$tdf(w_i, D) = \frac{\sum_{A \in \mathcal{A}_D} f(w_i, A)}{\max_i\{\sum_{A \in \mathcal{A}_D} f(w_i, A)\}}$$

Due to the normalization, for any word w_i and domain D the corresponding $tdf(w_i, D) \in [0, 1]$. This quantifies how important word w_i is for domain D. Let us next quantify the discriminative ability using an *inverse domain frequency* of word w_i as

$$idf(w_i, D) = \log\left(\frac{N_D}{\sum_{\forall D} \mathbb{1}(tdf(w_i, D) > 0)}\right),$$

where N_D is the number of domains, and the indicator function $\mathbb{1}(tdf(w_i, D) > 0)$ is used to count the number of domains where the word w_i appears at least once. The argument of the log belongs to $[1, \infty)$ hence the index $idf(w_i, D)$ is globally non-negative. Given the aforementioned definitions, the measure $C_{w_i,D}$ may be defined as

$$C_{w_i,D} = tdf(w_i, D) * idf(w_i, D).$$

This index gives high importance to words frequently appearing in a domain but not appearing in other domains.

A similar index can be computed using term event frequency

$$tef(w_i, e) = \frac{\sum_{A \in \mathcal{A}_e} f(w_i, A)}{\max_i\{\sum_{A \in \mathcal{A}_e} f(w_i, A)\}}$$

and inverse event frequency

$$ief(w_i, e) = \log\left(\frac{N_{e,D}}{\sum_{\forall e \in D} \mathbb{1}(tef(w_i, e) > 0)}\right),$$

where $N_{e,D} = |D|$ is the number of events in the related domain D. Note that the search in the denominator is restricted to events within the same domain. Finally the event weight may be defined as

$$E_{w_i,e} = tef(w_i, e) * ief(w_i, e).$$

3. Mike completes jogging one round on a (circular) athletic track of radius 1 mile. John is waiting for him at the center of the track. Compute the minimum and maximum possible values for the following distance measures between Mike and John while Mike is jogging: Manhattan, Euclidean and Chebyshev distance. For full credit give the proper mathematical notations.

   Solution : Without loss of generality, consider a cartesian coordinate system with John standing at $(0, 0)$, and denote an arbitrary position of Mike as (x, y), such that $x^2 + y^2 = 1$. The Euclidean distance is given by $\sqrt{x^2 + y^2}$ and is thus constant at 1 mile for the entire jogging period.

   The Manhattan distance is given by the L1-norm $|x| + |y|$. To maximize/minimize this subject to the constraint $x^2 + y^2 = 1$, one may eliminate the variable y and solve the following unconstrained optimization

   $$\text{Manhattan dist(min/max)} = (\min / \max)|x| + |\sqrt{1 - x^2}|$$

   It can then be verified that the maximum Manhattan distance is $\sqrt{2}$ miles and is obtained at the four permutation of coordinates from $(\pm 1/\sqrt{2}, \pm 1/\sqrt{2})$. The minimum Manhattan distance is 1 mile and is attained at $(0, \pm 1)$ and $(\pm 1, 0)$.

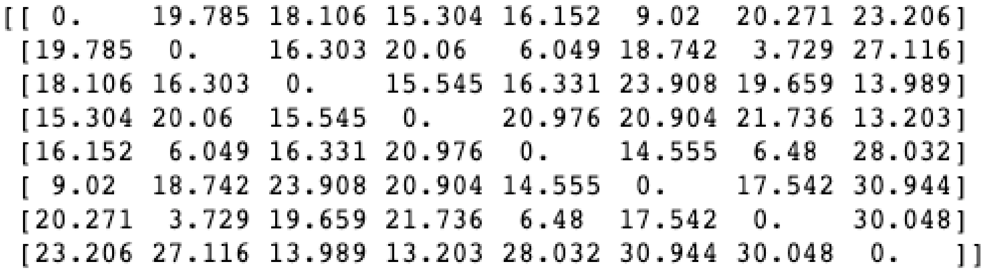
   The Chebyshev distance is given by the L_∞-norm $\max(|x|, |y|)$. To maximize/minimize this subject to the constraint $x^2 + y^2 = 1$, one may similarly eliminate the variable y and solve the following unconstrained optimization

   $$\text{Chebyshev dist(min/max)} = (\min / \max)\{\max(|x|, |\sqrt{1 - x^2})\}$$

   It can then be verified that the maximum Chebyshev distance attained is 1 mile and occurs at $(0, \pm 1)$ and $(\pm 1, 0)$. The minimum Chebyshev distance is $1/\sqrt{2}$ mile and occurs at $(\pm 1/\sqrt{2}, \pm 1/\sqrt{2})$.

4. Given 200 data points in 3 dimensions, *4-1)* Calculate the Euclidean distances between each pair of the data points (x_i, x_j) (a 200 × 200 distance matrix), and report the distances among the first 8 data points (an 8 × 8 distance matrix).

Solution : The Euclidean distances among the first eight points is shown in Fig. 1.



```
[[ 0.     19.785 18.106 15.304 16.152  9.02  20.271 23.206]
 [19.785  0.     16.303 20.06   6.049 18.742  3.729 27.116]
 [18.106 16.303  0.     15.545 16.331 23.908 19.659 13.989]
 [15.304 20.06  15.545  0.     20.976 20.904 21.736 13.203]
 [16.152  6.049 16.331 20.976  0.     14.555  6.48  28.032]
 [ 9.02  18.742 23.908 20.904 14.555  0.     17.542 30.944]
 [20.271  3.729 19.659 21.736  6.48  17.542  0.     30.048]
 [23.206 27.116 13.989 13.203 28.032 30.944 30.048  0.    ]]
```

Figure 1: Euclidean distance among the first eight points.

(a) Generate graph by connecting 5 nearest neighbors

Solution : Note: If x_i is among the five nearest neighbor of x_j, it is not necessary that x_j is also among the nearest five neighbors of x_i. Nevertheless, the undirected graph constructed next connects nodes in these cases. Thus, a node may end up getting more than 5 neighbors. The graph obtained is shown in Fig. 2. The related code is provided in the jupyter notebook.

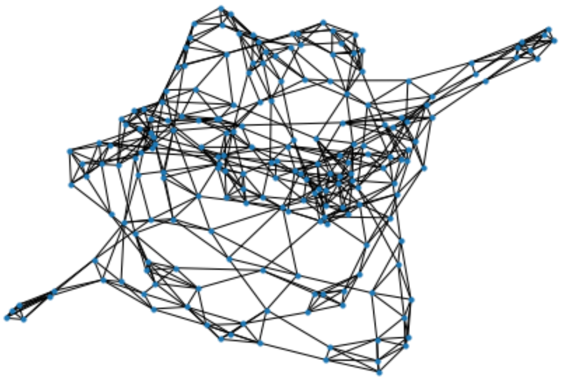


Figure 2: Graph-A: Connecting 5 nearest neighbors.

(b) Generate graph by connecting edges with distance less than 6.

Solution : The graph obtained by connecting edges with lengths less than 6 is shown in Fig. 3. We do not get a connected graph, due to nodes that are farther than 6 units from all their neighbors. The related code is provided in the jupyter notebook.

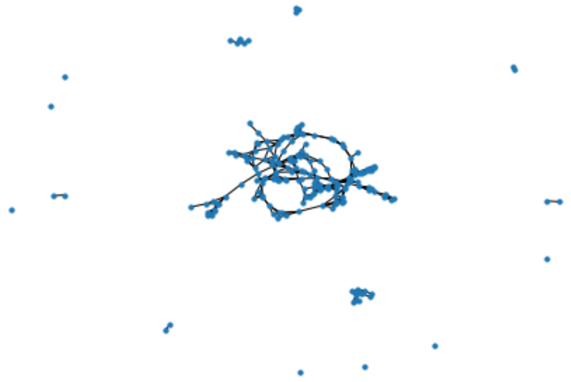
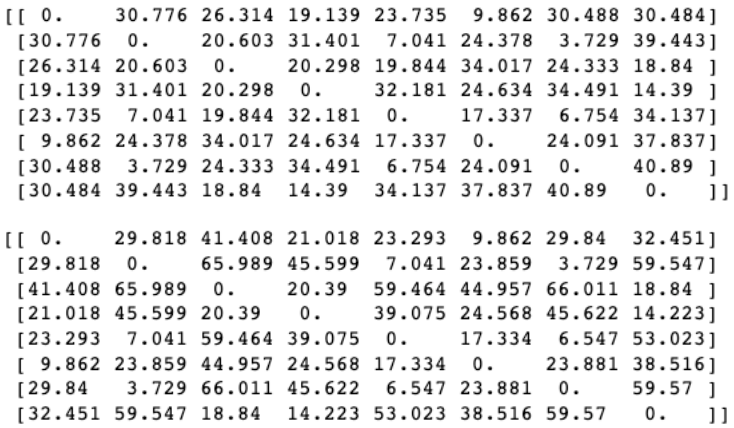


Figure 3: Graph-B: Retaining edges with length less than 6.

(4-2) For the first eight point, compute the geodesic distances using Dijkstra's shortest path algorithm.

Solution : The 8×8 distance matrices for the two graphs are reported in Fig. 4. The related code is provided in the jupyter notebook.



```
[[ 0.     30.776 26.314 19.139 23.735  9.862 30.488 30.484]
 [30.776  0.     20.603 31.401  7.041 24.378  3.729 39.443]
 [26.314 20.603  0.     20.298 19.844 34.017 24.333 18.84 ]
 [19.139 31.401 20.298  0.     32.181 24.634 34.491 14.39 ]
 [23.735  7.041 19.844 32.181  0.     17.337  6.754 34.137]
 [ 9.862 24.378 34.017 24.634 17.337  0.     24.091 37.837]
 [30.488  3.729 24.333 34.491  6.754 24.091  0.     40.89 ]
 [30.484 39.443 18.84  14.39  34.137 37.837 40.89   0.    ]]

[[ 0.     29.818 41.408 21.018 23.293  9.862 29.84  32.451]
 [29.818  0.     65.989 45.599  7.041 23.859  3.729 59.547]
 [41.408 65.989  0.     20.39  59.464 44.957 66.011 18.84 ]
 [21.018 45.599 20.39   0.     39.075 24.568 45.622 14.223]
 [23.293  7.041 59.464 39.075  0.     17.334  6.547 53.023]
 [ 9.862 23.859 44.957 24.568 17.334  0.     23.881 38.516]
 [29.84   3.729 66.011 45.622  6.547 23.881  0.     59.57 ]
 [32.451 59.547 18.84  14.223 53.023 38.516 59.57   0.    ]]
```

Figure 4: *Top:* Geodesic distances for Graph-A with 5 nearest neighbor. *Bottom:* Geodesic distances for Graph-B with edges less than length 6.

5. Analyze the data provided in thick.csv. Plot a boxplot of the first 100 rows. Explain why the thick center line in the box plot is not symmetrical with the outer edges of the box.

   Solution : The boxplot obtained from the first 100 rows of the provided file is shown in Fig. 5. The center line (median) is not symmetric within the box boundary (first-third quartile) because the underlying data is skewed.

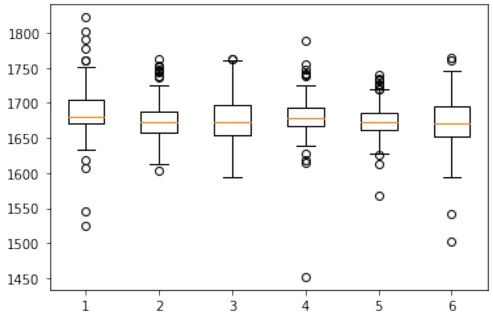


Figure 5: Boxplot for the measurements on 2×6 SPF boards.

6. Consider the data set from a food manufacturer making a pastry product. Perform the below task:

1) Import the data from food.csv
2) Center and scale the data to get zero mean unit variance.

Solution : Let $\mathbf{x}$ represent a 50×1 column vector from the data matrix that represents on of the five attributes. Each column was centered by subtracting the sample mean as

$$\mathbf{x}_{cen} = \mathbf{x} - \bar{x}\mathbf{1},$$

where $\mathbf{1}$ is the all-ones vector and $\bar{x} = (\sum_{i=1}^{50} x_i)/50$ is the sample mean. Next, the entire centered column vector was divided by the standard deviation to find the normalized vector $\mathbf{x}_{nor} = \frac{1}{\sigma}\mathbf{x}_{cen}$, where $\sigma = \sqrt{\frac{\sum_{i=1}^{50}(x_i - \bar{x})^2}{49}}$. Similar transformations were applied to all 5 columns using matrix-vector notation in the jupyter file. The final data matrix obtained has columns with zero mean and unit variance.

3) How does these two operations alter the overall interpretation of the data?

Solution : Centering and scaling the data results in a zero-mean unit-variance data. This is a more standardized data representation where the effect of nominal values and scale of variations (potentially due to units from disparate scales) is minimized. The resulting data focuses on the deviations from the mean or nominal value. Furthermore, the deviation is normalized with respect to the standard deviations observed for the respective attribute. Such data would be more suitable for any downstream learning or analysis application that can remain agnostic to the scales and units of the actual quantities represented by the data.

4) Now you have the pre-processed data matrix $\mathbf{X}$. What is the formula for correlation matrix? Implement it in python.

Solution : The columns of $\mathbf{X}$ are zero-mean and unit-variance. Thus, the correlation matrix is simply given by

$$Corr_X = \frac{1}{49}\mathbf{X}^\top\mathbf{X}.$$

5) Calculate the eigenvectors and eigenvalues of this square matrix.

Solution : The eigenvalues and eigenvectors are reported in the jupyter notebook.

6) Sort the eigenvalues from largest to smallest. Accordingly update the order of the eigenvectors in matrix. Plot the percentage of variance captured by the individual components in decreasing order

Solution : Here, we plot the relative energy of the eigenvectors in a decreasing order.

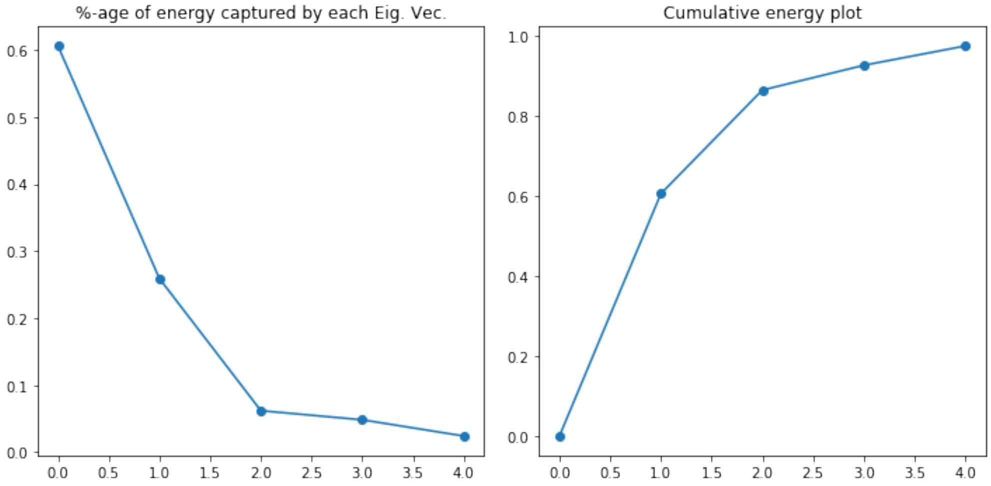


Figure 6: Relative energy of eigenvectors.

7-8) If you were to project the matrix **X** on the eigenvectors to obtain the principal components, how many components would you use and why? Implement it in python for 2 D projection. You carried out the steps of which algorithm? Give the scatter plot of the first two components obtained above.

Solution : Here we are implementing the steps of PCA. If we want to capture 90% of the total energy, choosing the first 3 components would suffice. A 2-D projection is shown below.

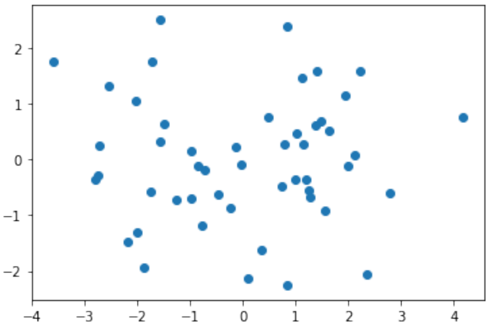


Figure 7: 2-D projection on the first two principal components.

9) So far we have used an algorithm based on eigenvalues and eigenvectors. Is it recommended? Explain.

Solution : Eigenvectors are efficient way of projecting to lower subspaces. However, for larger systems the computations do not scale well. Furthermore, the interpretability of features is lost.

10) Suppose you have an alternative approach that factorizes **X** into product of two orthonormal matrices **A** and **B**, and one diagonal matrix **D**. Which method comes to your mind? Implement it in python.

Solution : Such factorization may be implemented via singular value decomposition (SVD). The same is implemented in the jupyter notebook.

11) Summarize (project) the data matrix **X** in 2-D space making use of the matrices **A**, **B** and **D** obtained above. Give the 2 D scatter plot.

Solution : The scatter plot obtained by projecting the data on a 2-D space of latent components is shown next.

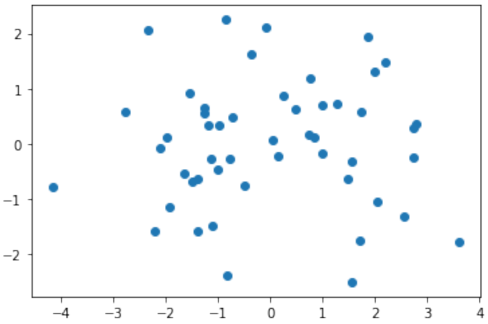


Figure 8: 2-D projection on the first two singular vectors.

12) Compare the plots obtained in part 8 and 11. Do you notice any similarity? Explain with proper mathematical equations to justify your answers.

Solution : The two projecttions are indeed the same with a sign-ambiguity of the components. Flipping the signs of the 2-D projected calues for one system results in the another. The similarity can be traced to the fact that the eigenvectors of the correlation matrix are the columns of the right singular values, representing the latent dominant directions(concepts).

$$X = ADB^\top$$
$$Corr_X = X^\top X = BDA^\top ADB^\top = BD^2B^\top$$