2023Fall-T-CSE472-80091 Homework 1

Amey Bhilegaonkar

TOTAL POINTS

2.875 / 3

QUESTION 1

Linear Algebra 1 pts

1.1 (a) **0.15 / 0.15**

1.2 (b) **0.05 / 0.05**

1.3 (c) **0.2 / 0.2**

1.4 (d) **0.2 / 0.2**

1.5 (e) **0.2 / 0.2**

1.6 (f) **0.2 / 0.2**

QUESTION 2

Graph Algorithms 1 pts

2.1 (a) **0.3 / 0.3**

2.2 (b) **0.2 / 0.2**

2.3 (c) **0.25 / 0.25**

2.4 (d) **0.125 / 0.25**

QUESTION 3

Network Measure 1 pts

3.1 (a) **0.1 / 0.1**

3.2 (b) **0.3 / 0.3**

3.3 (c) **0.1 / 0.1**

3.4 (d) **0.1 / 0.1**

3.5 (e) **0.2 / 0.2**

3.6 (f) **0.2 / 0.2**

Name	Amey Bhilegaonkar
ASU ID Number	1225368924

CSE 472: Social Media Mining

Homework I - Linear Algebra, Graph Essentials, Network Measures

Prof. Huan Liu

Due at 2023, September 7^{th}, 11:59 PM

This is an ***individual*** homework assignment. Please submit a digital copy of this homework to **Gradescope**. This is a fillable PDF and you are able to type into answer boxes provided for each question.

1. **[Linear Algebra]** Consider 2-dimensional data points of [-1, -2], [1, 0], [-1, 1], [2, 0], [4, 1].

   (a) Arrange the data points in ascending order based on their length and gather them together in the following matrix. Let's assume $[X]_{2\times5}$ is that matrix. Fill the following matrix. [*Hint:* The length of the vector $[x, y]$ is $\sqrt{x^2 + y^2}$.

$$X = \begin{pmatrix} 1 & -1 & 2 & -1 & 4 \\ 0 & 1 & 0 & -2 & 1 \end{pmatrix}_{2\times5}$$

   (b) What is the point showing the center of these points? [*Hint:* Calculate the mean of the values in each dimension].

$$\mu = \begin{pmatrix} 1 \\ 0 \end{pmatrix}_{2\times1}$$

(c) Calculate $Y = (X - \mu)(X - \mu)^T$ in which X^T is the transpose of X. To calculate $(X - \mu)$, easily subtract the μ from all the data points.

$$Y = \begin{pmatrix} \boxed{18} & \boxed{5} \\ \boxed{5} & \boxed{6} \end{pmatrix}_{2\times 2}$$

(d) Solve $|Y - \lambda I| = 0$ to extract the values of λ. $|\cdot|$ is the determinant and I is the identity matrix. λ values are called eigenvalues.

$\lambda_1 = \boxed{19.81}$, $\lambda_2 = \boxed{4.19}$

(e) Calculate the corresponding eigenvector to the **largest** eigenvalue (assuming the eigenvector has norm 1).

$$v = \begin{pmatrix} \boxed{0.94} \\ \boxed{0.34} \end{pmatrix}_{2\times 1}$$

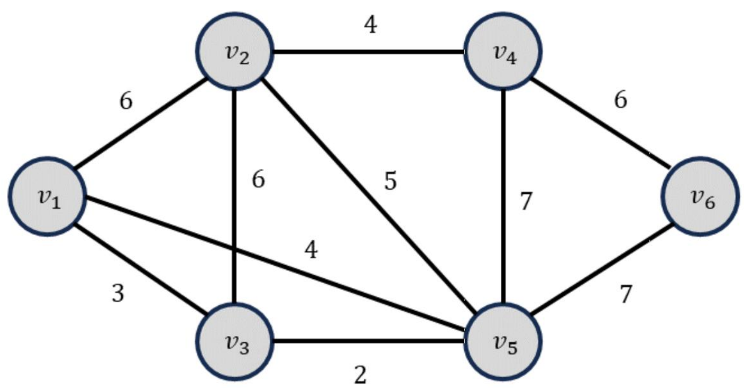
(f) Compute $\hat{X} = v^T X$.

$$\hat{X} = \begin{pmatrix} \boxed{0.94} & \boxed{-0.60} & \boxed{1.88} & \boxed{-1.62} & \boxed{4.1} \end{pmatrix}_{1\times 5}$$

Congratulations you performed Principle Component Analysis (PCA) procedure, a well-known dimensionality reduction method in machine learning. In other words, you projected your 2-dimesional data into 1-dimensional one such that you preserve the variance as much as possible (i.e. the least information has been lost).

2. **[Graph Algorithms]**

(a) Imagine a thriving social media platform called "ConnectWorld", where millions of users from around the globe come together to form a virtual community. Within ConnectWorld, friendships flourish, connections are made, and ideas are shared. In this scenario, each user within ConnectWorld represents a node in a graph, and the edges between them symbolize the effort that required to maintain the friendship. By applying **Prim**'s algorithm to the ConnectWorld graph below with starting **node** v_3, find the most cost-effective way to create a network that connects all users while minimizing the total effort required to maintain those connections. In the following table, at each step, write down the chosen edge and calculate the cumulative weight up to that step. Represent the edge between nodes v_i and v_j using the notation $v_i - v_j$. **Use as many steps as needed to solve this problem.**



Step	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Edge (start from v_3)	v3-v5	v3-v1	v5-v2	v2-v4	v4-v6			
Total Weights	2	5	10	14	20			

(b) Under what circumstances does the Prim's algorithm find multiple minimum spanning trees (MST) in a graph?

Prim's algorithm typically finds a unique minimum spanning tree (MST) in a connected graph when all edge weights are distinct. However, there are circumstances under which Prim's algorithm may find multiple MSTs:

Duplicate Edge Weights: If the graph contains duplicate edge weights, and the algorithm has the flexibility to choose between edges with the same weight, it may result in different MSTs. Prim's algorithm doesn't have a unique choice in such cases, and the order in which it explores edges can lead to different MSTs with the same weight.

Graph with Parallel Edges: If the graph contains multiple edges with the same weight between two nodes (parallel edges), Prim's algorithm may select different edges in different runs, leading to different MSTs. This situation occurs because the algorithm may arbitrarily choose one of the parallel edges.

(c) In the space below, draw a simple example of a directed graph with negative-weight edges for which Dijkstra's algorithm produces incorrect answers.

```
   -1
A -------> B
 \         /
  \       /
   -1    / -1
    \   /
     \ /
      C
```

The correct shortest path from A to C is A -> B -> C.

However, Dijkstra's algorithm, when applied to this graph, would incorrectly give you the path A -> C.

(d) Argue whether "Algorithm 1" below always produces the shortest paths from one source node to others for graphs that have negative weights but do not have negative cycles.

The given algorithm, Algorithm 1, is a modification of Dijkstra's algorithm that can be used to find the shortest paths from a source node to other nodes in a graph with negative weights but no negative cycles. The algorithm works by first finding the minimum weight in the adjacency matrix and subtracting it from all the weights in the matrix.
This ensures that all the weights in the matrix are non-negative. Then, the original Dijkstra's algorithm is applied to the modified matrix to find the shortest paths from the source node to other nodes.

However, it is important to note that Dijkstra's algorithm is designed to work only with graphs that have non-negative weights

This is because the algorithm assumes that the shortest path to a node is the path with the minimum weight, and negative weights can violate this assumption. In particular, negative weights can create cycles that have negative total weight, which can cause the algorithm to fail to find the shortest path or even loop indefinitely

Therefore, while Algorithm 1 can produce the shortest paths from a source node to other nodes in a graph with negative weights but no negative cycles, it is not a general solution for finding shortest paths in graphs with negative weights.

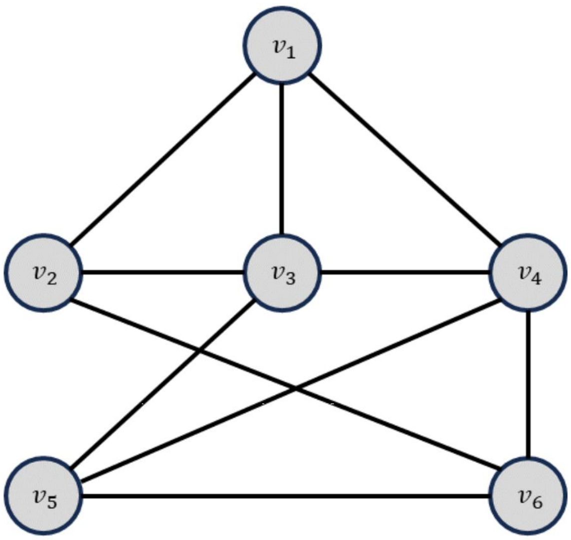
Algorithm 1: Dijkstra Algorithm for graphs with negative weights.

Input : Adjacency Matrix M, Source node s.
Output: Shortest Path from s to other nodes.

$C \leftarrow$ Find minimum weight in M
for all i and j:
    $M[i, j] \leftarrow M[i, j] - C$
return Dijkstra(M, s)   *// use the original Dijkstra algorithm to find the shortest paths*

3. **[Network Measures]** Based on the following network answer the questions,

(a) Fill the adjacency matrix.



	v_1	v_2	v_3	v_4	v_5	v_6
v_1	0	1	1	1	0	0
v_2	1	0	1	0	0	1
v_3	1	1	0	1	1	0
v_4	1	0	1	0	1	1
v_5	0	0	1	1	0	1
v_6	0	1	0	1	1	0

(b) Calculate the "Betweenness Centrality" (normalized) values, "Closeness Centrality" and "Katz Centrality" values with $\alpha = 0.25$ and $\beta = 0.15$ (you can use Matlab or other mathematical software to calculate the eigenvalues).

	Betweenness Centrality	**Closeness Centrality**	**Katz Centrality**
v_1	0.03333333333333333	0.7142857142857143	0.9230718767913669
v_3	0.13333333333333333	0.8333333333333334	1.107686111515594
v_6	0.08333333333333333	0.7142857142857143	0.8769183699896345

(c) Is the above alpha value a good choice for Katz centrality? Why?

The largest eigenvalue of the adjacency matrix is approximately 3.302775637731995, and the reciprocal of the given alpha value of 0.25 is 4.

Therefore, the given alpha value is a GOOD choice for Katz centrality because it is less than the reciprocal of the largest eigenvalue of the adjacency matrix.

(d) Discuss what would happen to Katz Centrality if we set $\alpha = 0$?

if alphs = 0, then everything is equal to beta, and centrality becomes biased, meaning If you set alpha to 0, it essentially means that Katz centrality will only consider indirect connections and completely ignore direct connections. In this case, the centrality of nodes will be based solely on their position in the network's indirect connections.

(e) Calculate the local clustering coefficient for nodes v_1, v_3, v_5, and v_6.

Local Clustering Coefficient for Node 1: 0.6667
Local Clustering Coefficient for Node 3: 0.5000
Local Clustering Coefficient for Node 5: 0.6667
Local Clustering Coefficient for Node 6: 0.3333

(f) Compute the similarity between nodes v_3 and v_5 using both Jaccard and Cosine similarities.

Jaccard Similarity between v3 and v5: 0.1667
Cosine Similarity between v3 and v5: 0.2887

Good Luck