Name	Solution
ASU ID Number	

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]. 1 pt

   (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 $\sqrt{x^2 + y^2}$. 0.067 pt

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

   (b) What is the point showing the center of these points? [*Hint:* Calculate the mean of the values in each dimension]. 0.133 pts

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

(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. 0.2 pts

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

(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. 0.2 pts

$\lambda_1 = \boxed{4.189}$, $\lambda_2 = \boxed{19.81}$

(e) Calculate the corresponding eigenvector to the **largest** eigenvalue (assuming the eigenvector has norm 1). 0.2 pts

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

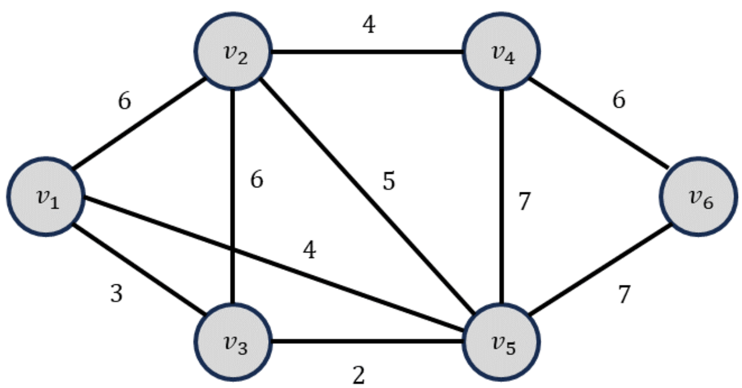
(f) Compute $\hat{X} = v^T X$. 0.2 pts

$$\hat{X} = \begin{pmatrix} \boxed{0.94} & \boxed{-0.6} & \boxed{1.88} & \boxed{-1.62} & \boxed{4.1} \end{pmatrix}_{1\times 5}$$

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]** 1 pt

(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.** 0.3 pts



Step	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Edge (start from v_3)	v3-v5	v1-v3	v2-v5	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? 0.2 pts

Duplicate edge weights: If the graph has multiple edges with the same weight, the algorithm may have multiple options to choose from at each step. When there are equal-weight edges available, different choices can be made during the algorithm's execution, resulting in different MSTs.

Graphs with symmetric edges: In some cases, the graph may have symmetric edges with the same weight, meaning that if there is an edge between vertices A and B with weight w, there is also an edge between B and A with weight w. This symmetry can lead to multiple MSTs as the algorithm can choose different edges between symmetric pairs, resulting in different tree structures.

Graphs with cycles of equal-weight edges: When the graph contains cycles comprising edges with the same weight, the algorithm may choose different edges to break cycles during each iteration. As a result, different choices can lead to different MSTs.

(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. 0.25 pts

*This question has multiple answers but one possible answer for it is:

```
       10
  V1-------->V2
    \          /
  5 \         / -7
      \       /
      >V3 <
```

Based on the Dijkstra's algorithm shortest path between V1 and V3 is 5 but it has to 3 (V1 --> V2 --> V3).

(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. 0.25 pts

*This question has multiple answers but one possible answer for it is:

```
       10                                                        17
  V1-------->V2                                           V1-------->V2
    \          /          adding +7 to all edges              \          /
  5 \         / -7     ------------------------------->    12 \         / 0
      \       /           based on the algorithm1               \       /
      >V3 <                                                     >V3 <
```

In the original graph shortest path from V1 to V3 is V1 --> V2 --> V3 (equal to 3) but after adding +7 (based on the algorithm 1) to all edges, the new shortest path is V1 --> V3 (equal to 12). Now we got a different answer and it means that the algorithm does not always produce the shortest paths from one source node to others for graphs that have negative weights but do not have negative cycles.

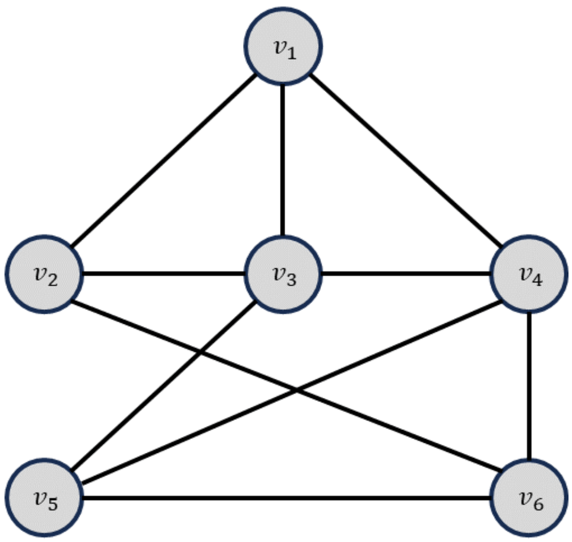
Algorithm 1: Dijkstra Algorithm for graphs with negative weights.

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, 1 pt

(a) Fill the adjacency matrix. 0.1 pts



	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). 0.3

	Betweenness Centrality	**Closeness Centrality**	**Katz Centrality**
v_1	1/30 ≈ 0.03	5/7 ≈ 0.71	0.9231
v_3	2/15 ≈ 0.13	5/6 ≈ 0.83	1.1077
v_6	1/12 ≈ 0.08	5/7 ≈ 0.71	0.8769

(c) Is the above alpha value a good choice for Katz centrality? Why? 0.1 pts

λ = (3.388 , -2.246 , -1.575 , 0.801 , -0.554 , 0.187)

Largest eigenvalue = 3.388

α < 1 / Largest eigenvalue ==> α < (1 / 3.388) ≈ 0.295 ==> α = 0.25 < 0.295 ✓

It is a good value.

(d) Discuss what would happen to Katz Centrality if we set $\alpha = 0$? 0.1 pts

All nodes get same centrality value β.

(e) Calculate the local clustering coefficient for nodes v_1, v_3, v_5, and v_6. 0.2 pts

C(V1) = 2 / ((3 * 2) / 2) = 2/3 ≈ 0.66

C(V3) = 3 / ((4 * 3) / 2) = 1/2 = 0.5

C(V5) = 2 / ((3 * 2) / 2) = 2/3 ≈ 0.66

C(V6) = 1 / ((3 * 2) / 2) = 1/3 ≈ 0.33

(f) Compute the similarity between nodes v_3 and v_5 using both Jaccard and Cosine similarities. 0.2 pts

Jaccard(V3, V5) = |N(V3) ∩ N(V5)| / |N(V3) ∪ N(V5)|

N(V3) = {V1, V2, V4, V5}
N(V5) = {V3, V4, V6}

Jaccard(V3, V5) = 1/6

Cosine(V3, V5) = |N(V3) ∩ N(V5)| / (|N(V3)|*| N(V5)|)^(1/2)

N(V3) = {V1, V2, V4, V5}
N(V5) = {V3, V4, V6}

Cosine(V3, V5) = 1/(4*3)^(1/2) ≈ 0.288

Good Luck