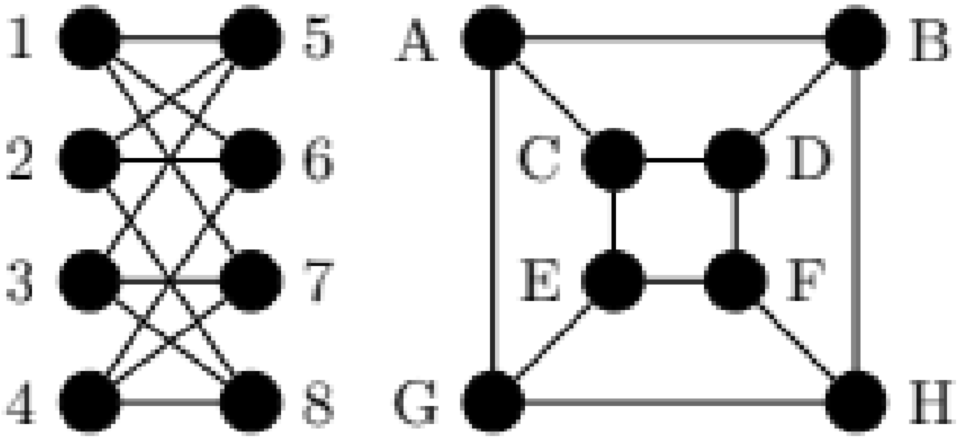
CS 276
Homework 2

Question 1 [10pts]: Graph classification. Run the Weisfeiler-Lehman Kernel test on the following two graphs.



Question2 [10pts]

Limitation of node2vec and algorithms that try to overcome them.

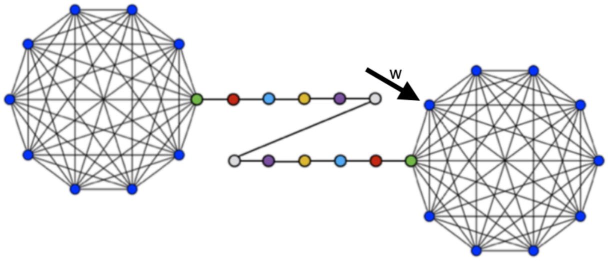
Due to how random walks work, it’s hard for node2vec to learn structural embedding from the graph. Let us consider a new algorithm, called struct2vec works. We define a clique to be a fully connected graph, where any two nodes are connected.

Given a graph G(V, E), it defines K functions $g_k(u, v)$, k = 1, 2, .., K, which measure the structural similarity between nodes. The parameter k means only the local structures within distance k of the node are considered.

With all the nodes in G, regardless of the existing edges, it forms a new clique graph where any two nodes are connected by an edge whose weight is equal to the structural similarity between them. Since struct2vec defines K structural similarity functions, each edge has a set of possible weights corresponding to g1, g2, ...gK.

The random walks are then performed on the new constructed clique graph. During each step, weights are assigned according to different g_k’s selected by some rule (omitted here for simplification). Then, the algorithm chooses the next node with probability proportional to the edge weights.

Characterize the vector representations of the 10-node cliques after running the node2vec algorithm on the graph in the figure below (barbell graph). Assume that through random walks, nodes that are close to each other have similar embeddings. Do you think the node embeddings will reflect the structural similarity? Justify your answer. Give intuition.



Question 3 [20pts] (old midterm exam part B)

Let's talk about Board Games. [Graph Drawing Conference Creative Contest 2023]

This dataset has the top-40 games found on BoardGameGeek. You can construct a graph, where every board game is a node. Two nodes (board games) are connected through a directed edge if players of one game also like another game. Additionally, each game contains some metadata. The format given for each game is JSON records in an array. An ID is assigned to each game, and the entry “recommendations”/”fans_liked” contains a list of other IDs (other games) that players of this game liked (to create the directed edges).

You can also work with undirected graphs instead of directed ones.

Use pandas to read the JSON files.

For this task:

[2pts] Create a “fans_liked” graph from the JSON records. Each game is a node and through the recommendation/”fans_liked” connect them. Visualize at least the top-40 games.

[2pts] Use node2vec to get the node embeddings and draw them on the 2D space. Again, you might only consider the top 40 games.

[2pts] Create a couple of bipartite graphs (affiliation graphs) by using the other metadata. As an example one node set is the games and the other node set can be the foci (different categories/design etc). Visualize it.

[2pts] Create the projected graphs of (c). Two games are connected if they have the same designer or the same category. Visualize.

[10pts] Define an interesting problem for this dataset, and design and implement a solution for this problem.

[2pts] Provide a report with your findings and the source code.

Provide the source code. You can decide what tool to use.

Note: You are free to decide which parts of the data to visualize and how to visualize it.

Solution: Link of code