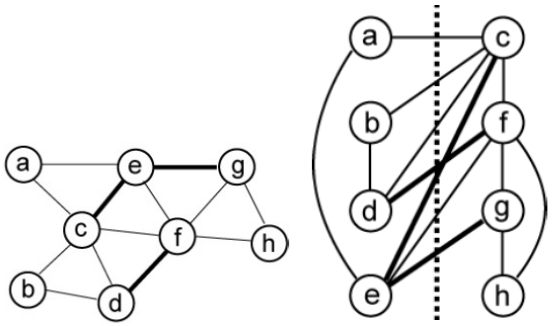
CS544 Introduction to Network Science

Indian Institute of Technology, Patna
End Semester Assignment
November 26, 2021 Time: 24 hours FM:40

Answer all the questions

1. Consider the graph shown in figure 1(a). Let A be the adjacency matrix of the graph with the individual elements being represented as A_{ij}. It is required to partition the graph into 2 groups, with four vertices each.



(a) The Given Network (b) Initial Partition

   (a) Use one round of Kernighan-Lin approach to partition the graph, with the initial partition being shown in figure 1(b). Please fill the entries in table 1 and use the same for partitioning. (10)

(i, j) (Edge)	Δ (Total change in cut size)

Table 1: Swap Gains

   (b) Suppose a spectral partitioning approach is to be used to partition the vertices into 2 groups. A vertex i is labeled as $s_i = +1$ or -1 depending on whether it belongs to the first or second group respectively, and R be the total number of edges between vertices of these groups. Derive the Graph Laplacian matrix L for the given graph and show that $R = \frac{1}{4}\overrightarrow{s}^T L \overrightarrow{s}$, where $\overrightarrow{s}$ is the vector of all s_i. State how the solution of this problem can be approximated using a relaxation method. You need to state the objective function and derive the constraints. (2+4+4=10)

2. (a) Derive the clustering coefficient of the nodes in a network formed using the $G(n, p)$ model. (2)
   (b) What would be the clustering coefficient in the limit of large n, and a constant average node degree? (1)
   (c) Derive the average diameter of the $G(n, p)$ network ensemble for $n = N$ (N is large) and constant average node degree c. (3)
   (d) Show that for large values of n and fixed average degree of the nodes, the degree distribution of the nodes will follow a Poisson Distribution. (4)
   (e) Derive, under what conditions of the mean degree will a giant component emerge in a $G(n, p)$ network. (4)

3. Although many real world systems are effectively modeled as graphs, graphs can be a cumbersome format for machine learning models. Hence, it is often useful to represent each node of a graph

as a vector in a continuous low dimensional space. The goal is to preserve information about the structure of the graph in the vectors assigned to each node. For instance, in several shallow embeddings, the structure is preserved in the sense that nodes connected by an edge were usually close together in the embedding $\mathbf{x}$. This problem deals with multi-relational graphs.

Multi-relational graphs are graphs with multiple types of edges. They are incredibly useful for representing structured information, as in knowledge graphs, where the nodes represent the entities and the edges represent the relation type between the connected entities. For example, there may be one node representing "Patna" and another representing "Bihar", and an edge between them with the type "Is capital of". Similarly edge between "Purnea" and "Bihar" may be of type "Is city of". In order to create an embedding for this type of graph, we need to capture information about not just which edges exist, but what the types of those edges are.

This problem explores a particular algorithm named, **TransE** [1], designed to learn node embeddings for multi-relational graphs. Let $G = (E, S, L)$ be a multi-relational graph consisting of the set of entities, E, (i.e., nodes), a set of edges S, and a set of possible relationships L. The set S consists of triples (h, ℓ, t), where $h \in E$ is the head or source-node, $\ell \in L$ is the relationship, and $t \in E$ is the tail or destination-node. As a node embedding, TransE tries to learn embeddings of each entity $e \in E$ into $\mathbb{R}^k$ (k-dimensional vectors), which we will notate by $\mathbf{e}$. The main novelty of TransE is that each relationship ℓ is also embedded as a vector $\ell \in \mathbb{R}^k$, such that the difference between the embeddings of entities linked via the relationship ℓ is approximately ℓ. That is, if $(h, \ell, t) \in S$, TransE tries to ensure that $h + \ell \approx t$. Simultanesouly, TransE tries to make sure that $h + \ell \not\approx t$ if the edge (h, ℓ, t) does not exist (Note: unbold letters e, ℓ etc., denote the entities and relationships in graphs, whereas bold letters, $\mathbf{e}, \ell$ represent their corresponding embeddings). Thus TransE achieves the above mentioned objective by minimizing the following loss:

$$\mathcal{L} = \sum\nolimits_{(h,\ell,t)\in S} \left(\sum\nolimits_{(h',\ell,t')\in S'_{(h,\ell,t)}} \left[\gamma + d(\mathbf{h}+\ell, \mathbf{t}) - d(\mathbf{h}'+\ell, \mathbf{t}')\right]_+ \right)$$

Here (h', ℓ, t') are "corrupted" triplets chosen from the set $S'_{(h,\ell,t)}$ of corruptions of (h, ℓ, t), which are all triples where either h or t (but not both) is replaced by a random entity.

$$S'_{(h,\ell,t)} = \{(h', \ell, t) | h' \in E\} \cup \{(h, \ell, t') | t' \in E\}$$

Additionally, $\gamma > 0$ is a fixed scalar called the margin, the function $d(\cdot, \cdot)$ is the Euclidean distance between 2 vectors, and $[\cdot]_+$ is the positive part function (defined as $\max(0, \cdot)$). Finally, TransE restricts all the entity embeddings to have length 1: $||\mathbf{e}||_\mathbf{2} = \mathbf{1}$ for every $e \in E$.

(a) Suppose a simpler loss function is used where instead of maximizing the distance between $\mathbf{h}' + \ell$ and $\mathbf{t}'$ the following loss function is minimized

$$\mathcal{L}_{simple} = \sum_{(h,\ell,t)\in S} d(\mathbf{h}+\ell, \mathbf{t}),$$

You need to show that the above objective would yield a useless embedding. By useless, it means that in your example, you cannot tell just from the embeddings whether two nodes are linked by a particular relation. Give an example of a simple non-trivial graph and corresponding embeddings that will minimize the new objective function all the way to zero, but still give a completely useless embedding. Your graph should be non-trivial, i.e., it should include at least two nodes and at least one edge. Assume the embeddings are in 2 dimensions, i.e., $k = 2$. (3)

(b) To investigate what the margin term γ accomplishes, suppose the margin term γ is removed from the loss function and instead the following is optimized

$$\mathcal{L}_{no-margin} = \sum\nolimits_{(h,\ell,t)\in S} \left(\sum\nolimits_{(h',\ell,t')\in S'_{(h,\ell,t)}} \left[d(\mathbf{h}+\ell, \mathbf{t}) - d(\mathbf{h}'+\ell, \mathbf{t}')\right]_+ \right)$$

Show that this would again yield a useless embedding. Give an example of a simple graph and corresponding embeddings which will minimize the new objective function all the way to zero, but still give a completely useless embedding. (3)

For both the above questions, you need to write 3 things precisely:

(a) Draw the graph with the entities and relations for which the embeddings if derived would be useless.

(b) The embeddings ($k = 2$) of the entities and relations that satisfy the loss minimization criterion.

(c) State the reason in not more than $3 - 4$ lines why the embeddings are useless.

References

[1] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durn, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'13). Curran Associates Inc., Red Hook, NY, USA, 27872795.