Node Embeddings, DSD, and Random Walks

Initial Notes

Brian Rappaport, Anuththari Gamage, Shuchin Aeron

July 2017

1 Introduction

In this project we will explore efficient and reliable ways of clustering nodes into groups/communities based on pairwise (noisy with missing links) interactions (modeled as edges on a graph). Recently in [2] the authors propose to impose a metric on the nodes, referred to as diffusion state distance, or DSD, derived using random walks on the graph to derive a robust affinity measure between nodes followed by spectral clustering [1]. On the other hand, in another recent paper [5] the authors generate ”sentences” of nodes by random walks on the graph as in word2vec [3] followed by embedding the sentences of nodes using a popular method, namely SGNS, for finding Euclidean word embeddings for NLP tasks [4].

1. In the first part of this project we want to numerically explore the connection between these two approaches. While the approach in [2] builds upon the limiting DSD, that can be computed in closed form, between nodes it is computationally expensive for large graphs (?? - read the paper). On the other hand the approach in [5] is computationally more attractive it being based on SGNS using SGD like methods that are known to scale well for large problem sizes.
   (a) The connection that we want to explore is to check the clustering performance on SBM using the two methods. Conjecture: Both should be similar.
   (b) What happens when we change the RW to NBRW [**xx**] or other vertex reinforced RW [**xx**]?
2. Can one directly compute the PMI matrix or the co-occurance matrix from the Adjacency matrix of the Graph? Then there is no need to perform random walk for node embedding.
3. Extend DSD for RW to DSD for NBRW - if the numerical results are good.
4. Can we use a tensor approach? From DSD, which is pairwise can we compute a DSD like metric for triples? How will this help? Similarly, can we take the node embedding method and use Eric Bailey's approach that computed Tensor Factorization for embedding [6].
5. How to combine multiple graphs (DREAM part 2)?

2 Sr. Design Proposal

The goal of this project is to investigate graph clustering algorithms based upon pairwise interaction data among entities. Such scenarios arise in disciplines as varied as natural language processing for machine translation and interpretation, biological systems of protein-protein interactions, or community detection in social networks. Specifically we will investigate several methods from classical Spectral Clustering to cutting-edge methods such as using the Diffusion State Distance metric on networks and embedding the nodes of a graph in Euclidean space via the well-known word2vec deep learning architecture. The final deliverable will be a suite of algorithms in Python and MATLAB that are computationally efficient, reliable, and robust to many types of data, as well as a rich variety of datasets and a detailed, well organized literature survey for bringing in future researchers to this exciting and important work. Anyone interested in being a part of this project should have a strong mathematical background, especially in linear algebra, and be prepared to survey recent literature in machine learning and statistics. A background in algorithm design and optimization would also be appreciated. The focus will be mainly on software rather than hardware.

3 Description

The Stochastic Block Model is a commonly used graph model for community detection. It builds off of the classical Erdős-Rényi $G(n, p)$ random graph model, wherein each edge of a graph with n nodes is formed with probability p. In SBM (technically the planted partition model), $G(n, p)$ is additionally given k clusters, and the probability an edge is formed within the cluster is a while intercluster edges are formed with probability b. Clearly, b should be lower than a in order to form an assortive graph. In our work, we have elected to make b proportional to a, so $b = \lambda a$ where λ is between 0 and 1. Our model is represented as $S(n, k, p, \lambda)$.

[add hubs and spokes model description?]

In order to generate sentences for use in the word2vec stochastic gradient descent algorithm, we need to perform random walks on the graphs. Since the graphs are unweighted, this is simply a matter of choosing at random among the neighbors of each node. We perform a certain number of walks starting from each node to a specific length. Here we also can set the walks to be reluctantly backtracking, where the walk will choose to return to a node immediately after leaving it only if it had no other possible choices. We will also not perform walks on any isolated nodes.

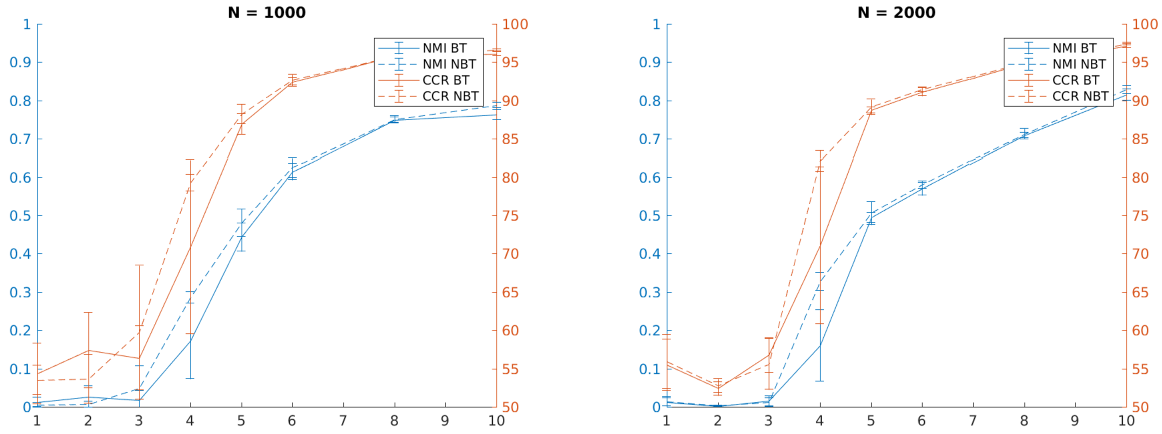
Once the sentences have been formed, we use an existing implementation of word2vec[3]. This takes a corpus of sentences (in this case, the random walks on the graph) and embeds them into d-dimensional space by means of stochastic gradient descent. Details on this algorithm can be found in [3].

4 Results

References

[1] Ulrike Von Luxburg. *A Tutorial on Spectral Clustering.* August. 2006.

[2] Mengfei Cao et al. "Going the Distance for Protein Function Prediction: A New Distance Metric for Protein Interaction Networks". In: *PLoS ONE* 8.10 (2013), pp. 1–12. ISSN: 19326203. DOI: 10.1371/journal.pone.0076339.



(a) 1000 data points

(b) 2000 data points

Figure 1: Accuracy of these algorithms, measured with two different metrics, correct classification rate and normalized mutual information. The x-axis c measures the sparsity of the graph, where an edge is made with probability c/N.

[3] Tomas Mikolov et al. “Distributed Representations of Words and Phrases and their Compositionality”. In: *CoRR* (2013), pp. 1–9. ISSN: 10495258. DOI: `10.1162/jmlr.2003.3.4-5.951`. arXiv: `1310.4546`. URL: `http://arxiv.org/abs/1310.4546`.

[4] Omer Levy and Yoav Goldberg. “Neural Word Embedding as Implicit Matrix Factorization”. In: *Advances in Neural Information Processing Systems (NIPS)* (2014), pp. 2177–2185. ISSN: 10495258. DOI: `10.1162/153244303322533223`. arXiv: `1405.4053`. URL: `http://papers.nips.cc/paper/5477-neural-word-embedding-as-implicit-matrix-factorization`.

[5] Weicong Ding, Christy Lin, and Prakash Ishwar. “Node Embedding via Word Embedding for Network Community Discovery”. In: *CoRR* (2016), pp. 1–10. arXiv: `1611.03028`. URL: `http://arxiv.org/abs/1611.03028`.

[6] Eric Bailey. “Master’s Thesis: Capturing and evaluating higher order relations in word embeddings using tensor factorization”. 2017.