1 Ward, Huang, Davison, and Zheng: Next Waves in Veridical Network Embedding

1.1 Introduction

The latest and greatest techniques in statistical and machine learning analysis require data to be represented in some sort of Euclidean space. For networked data, one has to "embed" such data into a space in order to perform these techniques. How we undergo this process should consider the following questions:

- Do we have a certain space that we want to use?
- Should certain features of our network data be preserved in our embedded space and how do we check if they are preserved?
- What are we intending to use this space for?

1.2 Problem Setup

For the standard network notation $G = (V, E)$ for a network with n nodes and a set of E edges, consider this network to have m other covariates associated with each node, as well as possible similar features for each edge. We wish to map the information from our network into some representation space, Z, with associated metric ρ, a mapping such that $\phi : G \times X \to (Z, \rho)$, where X is the vector of m other covariates.

1.2.1 Choosing a representation space

The most popular choice for a representation space(in this case Z) would be a Euclidean space of dimension d, where $d << n$. Other representation spaces such as d-dimensional spheres or the hyperbolic space are possible. Hyperbolic spaces may be especially useful in capturing possible tree-like structures.

1.2.2 Choosing what features should remain preserved

If we wish to study properties relating to communities in our networked data, clearly the embedding function should preserve community structure. Nodes in the same community should be embedded "closer" to each other according to our metric. There are many ways to capture proximity in an embedding:

1. First order proximity: If two nodes share an edge with high weight, they will have high first order proximity
2. Second order proximity: If two nodes share edges with many of the same nodes, they will have high second order proximity even if they do not share an edge

3. k-step transition probability: The weights on edges are seen to be the transition probabilities between states in a random walk. If there is a high transition probability between nodes, they should be closer in our embedding

4. Homophily: Prioritizes that people of the same community should be embedded closer together

5. Nodal proximity: Like other measures of proximity but grouped based on similarities in nodal covariates

1.2.3 Measuring similarity in the representation space

Similarity in the network space and similarity in the representation space need not be equivalent. If the embedding space is a metric space, the intuitive choice to measure similarity is simply to use the representation space's distance measure. Things like the dot product or other geometric manipulations of the similarity metric from the network space can also be used as measures of similarity in the representation space.

1.3 Representational Learning of Networks

1.3.1 Learning the representation

So how do we actually go about preserving the things we want to preserve in the representation space?

For unsupervised representation learning, our goal should be to match similarities between the original network and the representation space. We can construct different loss functions with different weights that allow us to choose what measures of similarity we wish to preserve. "Reconstruction loss" functions set up the embedding process as the minimization of some loss function, where the loss functions are some product of the differences in similarity between the network and its representation. For probabilistic generation processes, representations can be constructed by maximizing the likelihood function.

For supervised and semi-supervised learning, representation can focus on things like nodal covariates and labeling. We can include some sort of penalty for mislabeling in our loss function in addition to our loss metrics based on differences on proximity between network data and its representation.

1.3.2 Assessing the representation

In addition to assessing the quality of representation using measures about reconstruction, measures regarding other later inferential techniques can be used. If we want to engage in node classification, link prediction, or clustering, our representation should change to reflect this.

1.4 Review of Representative Methods

1.4.1 Spectral Clustering

In spectral clustering, a representation of the network is constructed using the eigenvectors from the spectral decomposition of the graph Laplacian. Spectral clustering is very popular, and exhibits helpful properties like consistent recovery of network communities, possible underling latent spaces, and other nice consistency properties. The issues with this approach is that it is inaccurate in the presence of outliers and is often computationally intensive.

1.4.2 Latent space models

Latent space models argue that the ground truth adjacency matrix is actually in a lower-dimensional latent space, and just needs to be estimated. Things like MCMC and variational inference are used to estimate clustering in the latent space. These techniques are also prohibitively computationally intensive.

1.4.3 Machine Learning Methods

Computationally less intensive models from the ML community can also be outfitted for representational purposes. Models like Deepwalk and Node2vec are popular examples.

1.5 Veridical network embedding

How we actually decide which of these representative methods to use should take into account:

- Predictability: Using a simple metric to see how well our model represents relationships in the original data in terms of a prediction target.
- Computability: Can this technique be scaled for use in much larger networks? What might we have to give up as we increase network size and how computationally intensive will our method be?
- Stability: Is our representation immune to possibly errors or perturbations?