Inferring Hierarchy Structure from Rankings Uncertainty

Phuc Nguyen

Macalester College
pnguyen@macalester.edu

Abstract

Many methods exist to infer hierarchy based on directed interactions in networks, usually resulting in a ranking or an ordering of nodes based on their dominance. In real-world systems, additional community structures such as tiers or parallel hierarchies might exist that ordinal rankings cannot describe. SpringRank, a ranking model for directed, weighted graphs, returns real-valued scores for nodes that can be used to infer tiers in the rankings. In this paper, we present an alternative method to identify tiers and parallel rankings using the correlation matrix of SpringRank scores. Specifically, we apply a clustering algorithm, such as k-means, to the correlation matrix or the difference between the correlation matrices of the empirical network and a "control network" with the same rankings, density, and noise parameters. We test the algorithm on synthetic networks and show that this method can recover the structures of interests.

I. Introduction

Hierarchy exists in many natural and artificial systems. For instance, social hierarchies are present in animal groups. Animals often interact, whether using aggressive or submissive behaviors, based on the dominance of the other animals. Hierarchies also exist in faculty hiring networks. Universities tend to hire graduates from highly ranked schools, while graduates will likely apply to schools based on their prestige. These hierarchies are often hidden but can be inferred from directed interactions between members of the system. A network is a useful model to represent these interactions in a system. Thus, rankings in a network can help describe the structure and dynamics of a system. At present, many methods already exist to infer hierarchies from networks. What these methods usually return is a strong hierarchy or an ordering of the nodes. However, one can expect the structure of a hierarchy to be more complex than a simple ordering. Some nodes in the hierarchy might be more closely ranked than others, resulting in tiers. There might be parallel hierarchies where comparisons of members of different sub-hierarchies might not be meaningful. Many methods are not designed to detect these kinds of community structures in a ranking.

SpringRank (De Bacco, Larremore, and Moore, 2017) is a ranking model for directed, weighted graphs. It assigns a real-valued score to each node, which can be used to rank the nodes. These real-valued scores can describe tiers–that is, scores of nodes in the same tier are closer to each other on the real line than they are to other nodes' scores. Nevertheless, scores alone cannot describe structures such as parallel hierarchies. Nodes from two parallel hierarchies might have similar scores but are not comparable because they don't tend to be connected. In this paper, we show that SpringRank's scores correlation matrix can be used to recover both tiers and parallel hierarchies using synthetic examples.

In the following sections, we will first explain the SpringRank model, then show the derivation of the correlation matrix between rankings, and we will test that using two synthetic examples, one with tiers and one with parallel sub-hierarchies. We will also compare inference results with those obtained from using the SpringRank scores alone.

II. The SpringRank Model

We can represent a system and the interactions between its members as a weighted, directed network. The members are represented as vertices, and the directed interactions are directed edges. The weight of an edge represents the number of interactions. SpringRank has two assumptions that determine the ranking. First, the model assumes that similarly-ranked vertices are more likely to interact with each other. Thus, the existence of an edge alone suggests relative rankings between two vertices. Second, the direction of an edge suggests dominance where an edge goes from a higher-ranked vertex to a lower-rank vertex. Imagine that every vertex is now embedded at a position along the real line, and each edge is replaced by a spring. Each spring has a resting length larger than zero, and the spring's energy increases when it is stretched or compressed. The springs pull and push vertices up and down the real line towards a distance equal to their resting length. The optimal scores compose the configuration that put the springs' forces at equilibrium, minimizing the energy of the whole system. This can be represented analytically with the following function of the total spring energy:

$$H(s) = \sum ij A_{ij}(s_i - s_j - 1)^2 \tag{1}$$

where A is the adjacency matrix of a network, s is the vector of SpringRank scores of vertices.

There are infinitely many ranking configurations that minimize the spring energy. This is because the solution is translation invariant. By adding a constant to each score, we get a new configuration that also minimizes the spring energy. To find a single solution, we add another constraint that the scores must all add up to zero. This creates an additional force that pushes the vertices to balance between zero on the real line.

III. Correlation Between SpringRank Scores

The SpringRank model has the following Generative Model:

$$P(A_{ij}|s) = Poisson(c \exp -\frac{\beta}{2}(s_i - s_j - 1)^2 \tag{2}$$

It can be shown that, given that a sparse graph, the SpringRank scores configuration that minimizes the Hamiltonion of the graph also maximizes the likelihood of observing the graph:

$$ln(P(A|s)) \sim -\beta H(s) \tag{3}$$

Using Bayes rule, we have:

$$P(s|A) \propto P(A|s)$$

Substituting in equation (3):

$$P(s|A) \propto e^{-\beta H(s)}$$

Substituting in equation (1) and distributing all terms:

$$P(s|A) \propto e^{-\beta \frac{1}{2} \sum ij(A_{ij}(s_i - s_j - 1)^2)}$$

$$P(s|A) \propto e^{-\beta \frac{1}{2}(s^T(D^{out} + D^{in} - A - A^T)s + 2s^T(D^{in} - D^{out})\bar{1} + m)} \tag{4}$$

Notice that the probability density distribution of a multivariate normal distribution is:

$$P(x) \propto e^{\frac{1}{2}(x^T \Sigma^{-1} x - 2x^T \Sigma^{-1} \mu + \mu^T \Sigma^{-1} \mu)} \tag{5}$$

Comparing the terms between equation (4) and (5), we get:

$$\Sigma = -\frac{1}{\beta}(D^{out} + D^{in} - A - A^T)^-1$$

and

$$\mu = (D^{out} + D^{in} - A - A^T)^-1(D^{in} - D^{out})\bar{1}$$

Thus, the scores configuration that minimizes the spring energy is normally distributed in a N-1 dimensional space with the covariance matrix Σ. The covariance matrix describes the joint variability between vertices' scores. Examining the covariance matrices of simulated examples, we observe that, as the number of interactions between two vertices increases:

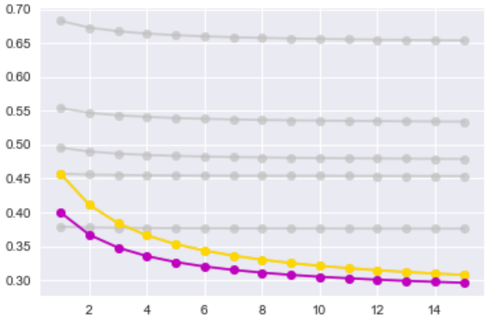

Figure 1: *As the weight of an edge connecting two nodes increases, their scores' standard deviations, colored in yellow and purple, decreases. Standard deviations of the other scores, colored in grey, remain constant*

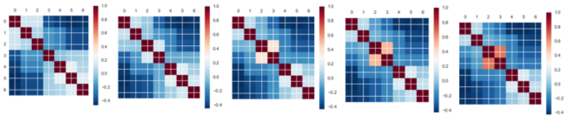

Figure 2: *As the weight of an edge connecting node 2 and 3 increases, their Pearson correlation becomes more positive*

- The standard deviations of their scores decrease to zero, while those of the other vertices remain relatively constant (Figure 1). Thus, scores of vertices with a high degree have lower variability.
- The Pearson correlation between their scores increases and becomes more positive (Figure 2). An edge with larger weight is consistent with SpringRank's intuition of a stiffer spring. Thus, the relative scores between vertices connected by such an edge are more stable and closer to the spring's resting length. Conversely, when two vertices have few or no edges in common, their scores tend to vary in the opposite direction to balance all the scores around zero as mentioned before.

We hypothesize that the correlation matrix can be used to identify communities within the hierarchy. Vertices in the same community are more connected, consequently, their relative rankings are also more stable and more positively correlated compared to rankings of vertices in different communities.

The tiers or parallel sub-hierarchies in the ranking are forms of community structures. Vertices within the same tier are more likely to interact, thus, are more connected. Vertices between two parallel sub-hierarchies generally do not interact under the model's assumptions. Thus, though they may have similar scores, we hypothesize that their scores' correlations are negative.

IV. Methods

We propose using a clustering algorithm, such as k-means, on the scores' correlation matrix or the difference between the correlation matrices of the empirical network and a "control network" to infer communities in the ranking. We treat each vertex as an object to be clustered and its score's correlations with the other vertices' scores its dimensions. SpringRank's generative model can create a "control network" with the same scores configuration, a density parameter c, and a inverse noise parameter β as the empirical network. However, since SpringRank's assumption states that similarly-ranked vertices are more likely to be connected, the generative model's network won't have parallel hierarchies. We then test the algorithm's performance on two synthetic networks. The first network contains three tiers in the hierarchy, and the second one contains two parallel sub-hierarchies. We use two measures, homogeneity and completeness, to evaluate the quality of the resulting clusters. Homogeneity is higher when only vertices in the same tier are assigned to a cluster, while completeness is higher when all vertices in the same tier are assigned to a single cluster (Rosenberg and Hirschberg, 2007).

i. Graph with tiers

The scores for 100 vertices are drawn from three factorized normal distributions. The generative model is used to generate a graph given these scores. The distribution of the scores alone can already suggest the three-tier structure. We use k-means on both the raw scores and the correlation matrix to infer the tiers. Homogeneity and completeness are calculated to compare their performances. The process is repeated with graphs generated using increasing inverse noise parameter β.

ii. Graph with parallel hierarchies

We create a tree with two branches of the same length to represent the two parallel sub-hierarchies. Similar to the three-tiered graph experiment setup, we will infer the parallel sub-hierarchies from the correlation matrix of this tree. Additionally, we notice that a graph created by the generative model using the SpringRank scores of the tree can serve as a "control network" for this parallel sub-hierarchies structure. In the "control network", vertices with similar scores on the two branches will interact with one another. Thus, we hypothesize that clustering the difference between the correlation matrix of the original tree and its control model will also reveal the parallel structure. Finally, we also try to infer this structure using the scores alone. We repeat the experiment for decreasing size of the parallel sub-hierarchies to evaluate the sensitivity of the algorithm.

V. Results

i. Graph with tiers

As expected, as β increases, the inferred scores are more correlated with the true scores used to plant the graph. The quality of SpringRank's inference increases quickly to almost perfect around $\beta = 0.5$ (Figure 3). We see that noise affects the inference of tiers in the same manner that it does to the inference of scores (Figure 4). Additionally, detecting the tiers using the correlation matrix is more accurate than using just the raw scores, especially when β is small or a network contains many edges violating the rankings.

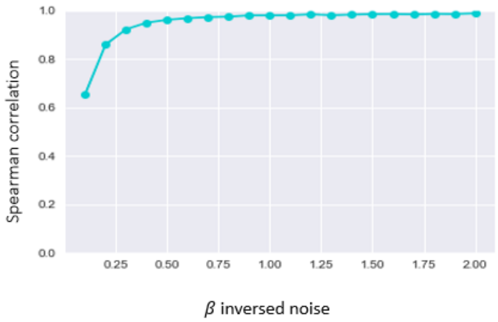

Figure 3: *As β increases, the inference of original scores becomes more accurate*

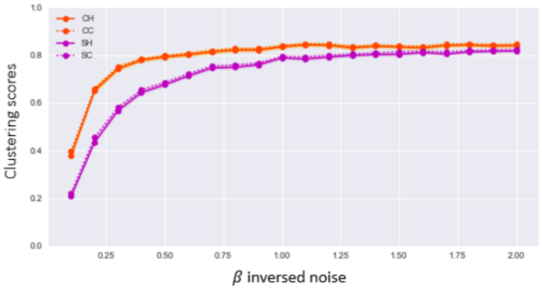

Figure 4: *As β increases, the recovery of three tiers structure is more accurate. Clustering using correlation matrix, in red, performs better than that using the raw scores, in purple*

ii. Graph with parallel hierarchies

Figure 5 shows that inferring the sub-hierarchies using the correlation of the original graph performs equally well to using the difference in the correlation matrices of the original graph and the control model when the size of the branches are large. However, when the size of these communities become smaller, the difference in correlation performs much better at revealing these sub-structures. In all cases, the raw scores fail to reveal any parallel structures.

VI. Discussion

While many methods exist to infer rankings from a hierarchy, most of them are not designed to describe additional structures in the hierarchy. We propose a technique using the

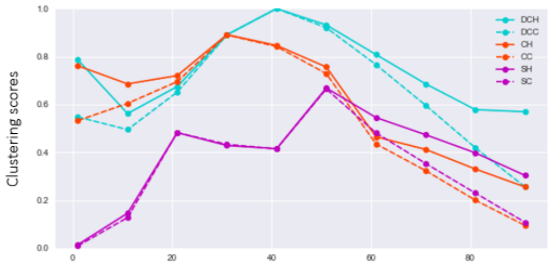

Figure 5: *As the size of the parallel communities decreases, clustering of the difference in correlation between the original graph and control graph, in blue, outperforms clustering using just the correlation, in red, and using raw scores, in purple*

correlations between SpringRank scores to infer additional structures from a ranking hierarchy. Using synthetic examples, we show that applying k-means clustering to the correlation matrix can reveal tiers and parallel sub-hierarchies structures in networks. For tier structures, clustering using the correlation matrix outperforms clustering on the SpringRank raw scores, especially when the data is noisy. Additionally, while the raw scores fail to reflect the parallel sub-hierarchies structure, the correlation matrix and the difference in the correlation matrices of scores of the original network and a control network perform well in detecting this structure. The difference in correlation matrix performs specifically well for sub-hierarchies of smaller size.

These initial experimental results validate the potential of the proposed technique to describe community structures in rankings. Additional tests on real-work networks are needed further validate the technique. One challenge that arises as we look for suitable data sets for this test is the lack of examples with known tiers or sub-hierarchies that also fit SpringRank's assumptions. For instance, though food webs inherently contain tiers, vertices in the same tier, i.e. species in the same trophic level, interact more with vertices in the tier below or above them than with each other. Thus, having a test to validate the significance of the inferred communities can be part of future work. In addition, applying k-means directly to a correlation matrix might not be completely accurate. K-means produces clusters from a features matrix, while correlation is a measure of similarity. We try using spectral clustering technique, which takes an affinity matrix, however, the results are inferior to using k-means. Applying k-means to an embedding of the correlation matrix or using a different clustering algorithm can also be part of future work. Finally, providing theoretical groundings for our observation that more positive edge weights lead to more positive correlations between vertices should also be addressed in future work.

References

[1] De Bacco, C., Larremore, D. and Moore, C. (2017). A physical model for efficient ranking in networks *in review*.

[2] Rosenberg, A. and Hirschberg, J. (2007). V-Measure: A conditional entropy-based external cluster evaluation measure *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 410–420.