Link Prediction

Neel Haria and Prof Hang Liu
Stevens Institute of Technology, NJ, USA

I. ABSTRACT

Link prediction is a key problem for network-structured data. Link prediction heuristics use some score functions, such as common neighbors and Katz index, to measure the likelihood of links. They have obtained wide practical uses due to their simplicity, interpretability, and for some of them, scalability. However, every heuristic has a strong assumption on when two nodes are likely to link, which limits their effectiveness on networks where these assumptions fail. In this regard, a more reasonable way should be learning a suitable heuristic from a given network instead of using predefined ones. By extracting a local subgraph around each target link, we aim to learn a function mapping the subgraph patterns to link existence, thus automatically learning a "heuristic" that suits the current network.

II. INTRODUCTION

Link prediction is to predict whether two nodes in a network are likely to have a link. Given the ubiquitous existence of networks, it has many applications such as friend recommendation, movie recommendation, knowledge graph completion, and metabolic network reconstruction. One class of simple yet effective approaches for link prediction is called heuristic methods. Heuristic methods compute some heuristic node similarity scores as the likelihood of links. Existing heuristics can be categorized based on the maximum hop of neighbors needed to calculate the score. For example, common neighbors (CN) and preferential attachment (PA) are first-order heuristics, since they only involve the one-hop neighbors of two target nodes. Adamic-Adar (AA) and resource allocation (RA) are second-order heuristics, as they are calculated from up to two-hop neighborhood of the target nodes. We define h-order heuristics to be those heuristics that require knowing up to the h-hop neighborhood of the target nodes. There are also some high-order heuristics that require knowing the entire network. Examples include Katz, rooted PageRank (PR) [9], and SimRank (SR).

III. PREREQUISITES

A graph is denoted by, lets say G = (V,E), where V is the set of vertices. Latent features and explicit features: These are also studied for link prediction. Latent features uses matrix factorization to reduce dimensions and learn more about embedding of each node. Explicit features are often available in form of node attributes, describing information about individual nodes.

IV. WORK DONE

In this section, the main aim is to present the understandings of the deeper mechanism studied and researched behind various link prediction heuristics. Due to the large number of graph learning techniques, it is good to note that we should not be concerned with the generalization error of a particular method, but the focus is more on the information that can be retrieved from the local subgraphs for calculating heuristics. Below is the list of algorithms that are researched and studied to better understand link prediction. These algorithms are mathematical heavy and research intensive.

A. Katz Index

Katz Centrality is a metric that measures the affinity between vertices as a weighted sum of the walks between them and penalizes long walks in the network by a user chosen factor [3]. The linear algebraic formulation of Katz Centrality lends itself to a dynamic algorithm based in a numerical linear algebra environment using iterative solvers. We develop an algorithm that updates Katz scores as new connections are made in the network. This algorithm it uses the power method to find the eigenvector corresponding to the largest eigenvalue of the adjacency matrix of G. The constant alpha should be strictly less than the inverse of largest eigenvalue of the adjacency matrix for the algorithm to converge.The iteration will stop after iterations or an error tolerance of number of nodes has been reached. Katz centrality computes the relative influence of a node within a network by measuring the number of immediate neighbours i.e. also called the first degree nodes and also other nodes in the network that connect to node under consideration through these immediate neighbors.

Connections to distant neighbors are penalized by an attenuation factor αach path or connection between pair of nodes is assigned a weight determined by αnd the distance between nodes as α

1. Mathematical Formulation

For any network to be considered, an adjacency matrix A is created. Element $\left(A_{ij}\right)$ of adjacency matrix A will be

equal to 1 if node i is connected node j, else 0. The walk of connection between two nodes can be known by the power of the adjacency matrix. For instance, for matrix A^3 if element $\left(A_{1,9}\right) = 1$, it shows that node 1 and node 9 are connected with a walk of 3.

$$\left(C_{Katz}(i) = \sum_{k=1}^{\infty} \sum_{k=1}^{\infty} \alpha^k (A^k)_{j,i}\right.$$

$$\vec{C} = ((I - \alpha A^T)^{-}1 - I)\vec{I}$$

Here I is the identity matrix, is a vector of size n, where n is the number of nodes. A^Tenotes the transposed matrix of A and (I-αA^T)$^{-1}$enotes the matrix inversion of (I-αA^T) Katz centrality can be used to compute centrality in directed networks such as citation networks and the World Wide Web.

Katz centrality is more suitable in the analysis of directed acyclic graphs where traditionally used measures like eigenvector centrality are rendered useless.

Katz centrality can also be used in estimating the relative status or influence of actors in a social network. The work presented in [9] shows the case study of applying a dynamic version of the Katz centrality to data from Twitter and focuses on particular brands which have stable discussion leaders. The application allows for a comparison of the methodology with that of human experts in the field and how the results are in agreement with a panel of social media experts.

B. SimRank

SimRank is a general similarity measure, based on simple graph-theoretic model.Semantic similarity is a metric that is defined over a set of documents or terms, where the distance between two items is based on their likeness of their meaning or semantic content. This type of similarity, computationally can be estimated by defining topological similarity, by defining ontologies to define distance between terms/concepts. There are two types of approaches that can be used to compute topological similarity, one of them being edge based, which uses the edges and their types as the data source. The other, node based which has nodes and it's properties as their main data sources.

The intuition behind the SimRank algorithm is that, in most of the domains, similar objects are referenced by similar objects. To be more precise, objects a and b are considered to be similar if they are pointed from objects c and d. SimRank can be applied to any domain where there are enough relevant relationships between objects to base at least some notion of similarity on relationships.

1. Algorithm

Initialize the SimRank of every pair of the nodes following Update the SimRank of every pair of nodes in the graph for each iteration.

SimRank of nodes a and b is equal to 1, if both nodes are the same i.e. SimRank(a,b) = 1

If one of the nodes has no in-neighbors the SimRank(a, b) = 0

The calculation of new SimRank is based on the SimRank from previous iteration, defined recursively but calculated iteratively

2. Mathematical Formula

It is important to note that SimRank is a general algorithm that determines only the similarity of structural context. SimRank applies to any domain where there are enough relevant relationships between objects to base at least some notion of similarity on relationships. Obviously, similarity of other domain-specific aspects are important as well; these can — and should be combined with relational structural-context similarity for an overall similarity measure.

S(a, b) = $\frac{C}{|I(a)|I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} S(I_i(a), I_j(b))$

For a $\neq b$ and $S_{k+1}(a, b) = 1$for a=b. That is, on each iteration k+1, we update similarity of (a,b) using the similarity scores of neighbours of (a,b) from the previous iteration k according to the basic SimRank equation.

C. Adamic Adar

The Adamic/Adar predictor formalizes the intuitive notion that rare features are more telling. This index refines the simple counting of common neighnors by assigning the lower connected neighbors more weights, which is defined by

$$S_{xy} = \sum_{x \in \Gamma(x) \cap \Gamma(y)} \frac{1}{logk(z)}$$

V. FUTURE WORK

This section will include the topics I will cover in the future and extend my research in.

A. GNN

Graph Neural Network, as how it is called, is a neural network that can directly be applied to graphs. It provides a convenient way for node level, edge level, and graph level prediction task. There are mainly three types of graph neural networks in the literature:

Recurrent Graph Neural Network, RecGNN is built with an assumption of Banach Fixed-Point Theorem. Banach Fixed-Point Theorem states that: Let (X,d) be a complete metric space and let (T:X→X) be a contraction mapping. Then T has a unique fixed point (x) and for any x∈ X

Spatial Convolutional Network, The intuition of Spatial Convolution Network is similar to that of the famous CNN which dominates the literature of image classification and segmentation tasks.

Spectral Convolutional Network,As compared to other types of GNN, this type of graph convolution network has a very strong mathematics foundation. Spectral Convolutional Network is built on graph signal processing theory. And by simplification And approximation of graph convolution.

The intuition of GNN is that nodes are naturally defined by their neighbors and connections. To understand this we can simply imagine that if we remove the neighbors and connections around a node, then the node will lose all its information. Therefore, the neighbors of a node and connections to neighbors define the concept of the node.

VI. CONCLUSION

To conclude, Till now I was successfully able to implement and research about the heuristic algorithms used for in Link Prediction. Going further I plan on studying and researching about better ways to improve performances and study existing algorithms in this field.

A. REFERENCES

[1]Learned and research about Katz Index

[2]Research and mathematics on SimRank

[3]Understanding more about Link Prediction

[4]Ways to improve heuristic methods

[1] B. Brorsson and K. H. Asberg, "Katz index of independence in adl. reliability and validity in short-term care.," *Scandinavian journal of rehabilitation medicine*, vol. 16, no. 3, pp. 125–132, 1984.

[2] G. Jeh and J. Widom, "Simrank: a measure of structural-context similarity," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 538–543, 2002.

[3] L. Lü and T. Zhou, "Link prediction in complex networks: A survey," *Physica A: statistical mechanics and its applications*, vol. 390, no. 6, pp. 1150–1170, 2011.

[4] M. Zhang and Y. Chen, "Link prediction based on graph neural networks," *arXiv preprint arXiv:1802.09691*, 2018.