Paper Summary

David Miller
CIS 5930: Social Network Mining

February 5, 2018

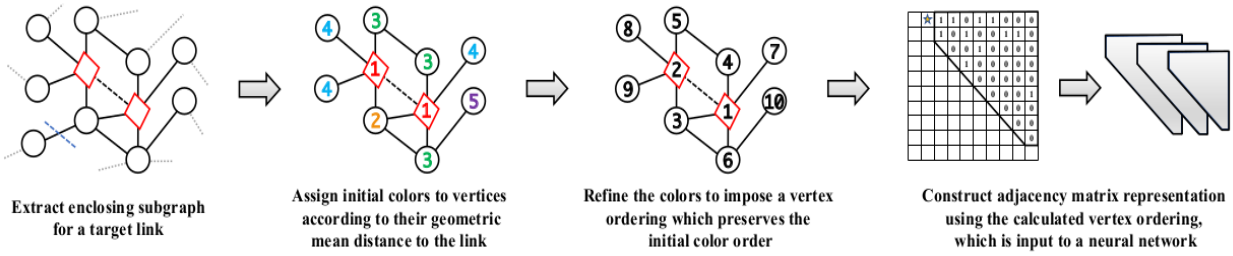

Figure 1

The paper proposes a novel algorithm, WLNM, for link prediction. Two important concepts for link prediction are heuristics and graphs. Heuristic methods for link prediction include first order, second order, and high order. First order methods compute based on 1-hop neighborhoods $\Gamma^1(x)$ of link vertex x. Second order and higher order compute via $\Gamma^2(x)$ and $\Gamma^d(x)$, respectively, for $d = 2, 3, \ldots$. A network can be represented as a graph $G = (V, E)$ whose adjacency matrix A will be used as input for the algorithm. The WLNM algorithm can be summarized as follows

1. **Enclosing Subgraph Extraction :** Target link (x, y) is fed as input and the algorithm iteratively builds a subgraph with vertices (x, y) then $(x, y, \Gamma^1(x), \Gamma^1(y)), \ldots$ $(x, y, \Gamma^1(x), \Gamma^1(y), \ldots, \Gamma^n(x), \Gamma^n(y))$ until some threshold size K is reached.

2. **Subgraph Pattern Encoding :** A color-order preserving algorithm is run on the subgraph. This defines structural roles (coloring) and rankings via relative positions and structural roles (ordering).

3. **Neural Network Learning :** After the encoding step, a classifier is trained to learn complex non-linear patterns in the graph. Both positive samples $((x, y) \in E)$ and negative samples $((x, y) \notin E)$ are used in the training process.

The algorithm was tested with eight datasets: USAir, NS, PB, Yeast, C.ele, Power, Router, and E.coli. The algorithm was tested against nine heuristic methods: common neighbors (CN), Jaccard (Jac.), Adamic-Adar (AA), resource allocation (RA), preferential attachment (PA), Katz, resistance distance (RD), PageRank (PR), and SimRank (SR). The WLNM algorithm for link prediction produced results that are superior than those it was tested against in almost every dataset. In terms of future work, I believe it should be focused on scalability. Todays market for efficient algorithms is huge, but if it can not be done via mobile computing or through personal computers, it suffers from needing time to run the algorithm and money to buy a set up that can actually run it. This algorithm has many

commercially ready applications, but until it can be done with low computational cost, the only people I can see being interested in this is Amazon and Google.

Three strengths I found with the paper are

1. The accuracy in link prediction makes it very applicable to socila networking, product recommendation, knowledge graph completion, and finding interactions between proteins.

2. The framework of the algorithm is based upon existing work, making it an easy to understand algorithm. This is a particularly good aspect for people who want to use the algorithm.

3. The code is available to the public. This allows people to already use the code or improve upon it.

Three weaknesses I found with the paper are

1. There was no mention of scalability. This is vital for algorithms to survive because lower execution time means cost effective and usability without warehouse scale computing.

2. There was not much information regarding time complexity, which leads me to believe that although effective, the WLNM algorithm may not be all that fast. This in turn makes it less likely to be commercially used.

3. There is no mention of weighted graphs. Weighted graphs are pivotal in modeling and lack of support for this could be huge negative about this algorithm.

Questions for the reader

1. How is the algorithm implemented for a directed network?

2. How much data is needed to accurately train the neural network for this algorithm to produce accurate results.

References

[1] Muhan Zhang, Yixin Chen, *Weisfeiler-Lehman Neural Machine for Link Prediction*, KDD17, August 13-17, 2017, Halifax, NS, Canada.