Link Prediction for a Rice University Social Network

Brandon J. Fantine
COMP 459
Spring 2024
bjf7@rice.edu

Abstract

It goes without saying that people are complex - when reduced down to data, oftentimes much of that "complexity" goes away with it. This is none the more prevalent than in Social Networks. Taking a large topographical social network of undergraduates at Rice University, two link prediction algorithms are compared: one that considers Hidden metrics, and one that does not. Through the analysis, empirical evidence is sought as to how connections across the Rice social network can be predicted (if at all) and what sociological conclusions can be reached through graph analysis.

1 Introduction

One of, if not the most complex biological systems we have at our disposal are relationships: organisms - human or not - develop partnerships, friendships, or even baseline recognition. With the onset of social media, however, determining these connections has gotten more complicated. Abstracting social connections into a virtual space has led to a myriad of investigations into link prediction - or, rather, human-connection prediction - over the past 20 years [1].

Link prediction of social networks is not a perfect science. This paper sets out to determine the bounds of this field of graphical machine learning. There is evidence that social network link prediction *cannot* be determined purely through common methods related to node concentration and algorithms in general. In specifics, social media link prediction requires algorithms that consider hidden metrics such as influencing factors and latent links between persons.

Through an analysis of the accuracy between two models - one considering hidden metrics and the other not - an attempt to understand how exactly social network link prediction is made.

2 Hypothesis

The central hypothesis of this paper hinges on social network (social media) link prediction being more accurate with a model that considers the existence of hidden variables as a influencing factor on prediction efficacy.

1. A hidden metric-based model will perform more accurately than a path-based algorithm on a real-world social network.

3 Models

3.1 Katz Algorithm

One of the most well-known link prediction models, the Katz Algorithm was first introduced in 1953 by Leo Katz specifically for social networks [2]. Measuring the influence of a node on a network, the

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

Katz Algorithm makes use of the aptly-named Katz centrality of a node - the number of first-order and second-order neighbors of node *i*.

$$C_{Katz}(i) = \sum_{k=1}^{\infty}\sum_{j=1}^{n} \alpha^k (A^k)_{ji} \tag{1}$$

The above equation is the mathematical formulaiton of the algorithm. The Katz centrality of the node is the sum of the *k* connections between nodes *i* and *j* multiplied by the reciprocal of the largest eigenvalue in *A*. Each node gets assigned its own centrality, which represents the likelihood of a random node making a connection to the i^{th} node.

As it pertains to machine learning and link prediction, the Katz Algorithm makes an assumption that the connections between actants in a network is only related to the size of a given node's network; in terms of humanistic popularity and the Instagram case study, predicted links would mostly likely be between users with a large following who are friends with equally-popular persons. Katz's central argument is that being known begets biological connections [3].

However, if we are to refer to the works of Alharbi, Benhidour, and Kerrache and their Hidden Metric Model [4], there likely exists hidden metrics - variables, unseen with simple graphs both weighted and unweighted, that influence connections between users.

3.2 Latent Link Prediction

To account for those hidden metrics, the Latent Link Prediction model posed by Zareie and Sakellariou is used [5]. The algorithm works as follows:

1. For every node in the graph, classify initial weights based on degree or the common neighbors similarity score (2)
2. Calculate the union neighborhood for every pair of nodes (3)
3. Calculate the Pearson correlation coefficient for each union neighborhood (4)
4. Assign final weights to node connections based on the Direct-Indirect Common Neighbors (DICN) similarity score (6)

Links are predicted based on the DICN score; scores indicate the probability that a link will form between the two nodes. The DICN then, is what determines actual "predicted links."

The initial classification is predicated on the following:

$$N_i[z] = \begin{cases} d_i & if z = i \\ 1 + |\Gamma_i \cap \Gamma_j| & if z \in \Gamma_i^{(1)} \\ |\Gamma_i \cap \Gamma_j| & if z \in \Gamma_i^{(2)} \\ 0 & otherwise \end{cases} \tag{2}$$

Where, if the current node of interest z is a first-order neighbor of node *i*, we calculate the $common\ neighbors\ similarity\ score\ + 1$; sans the additional one for second-order neighbors. If node *i* is node z, we set the initial weight of node *i* to be its degree. If nodes *i* and z are unconnected and more than two connections away from eachother, node $i's$ inital weight is 0. Essentially, this classifcation system sets up a way to differentiate between nodes across the spectrum of being closely related to topographically not-at-all related.

From there, the Union neighborhood and correlation coefficients are calculated where:

$$UN_{ij} = \{z \mid (N_i[z] > 0)\ Or\ (N_j[z] > 0)\} \tag{3}$$

$$Corr_{ij} = \frac{\sum_{z \in UN_{ij}} (N_i[z] - \overline{N_i})\,(N_j[z] - \overline{N_j})}{\sqrt{\sum_{z \in UN_{ij}} (N_i[z] - \overline{N_i})^2}\,\sqrt{\sum_{z \in UN_{ij}} (N_j[z] - \overline{N_j})^2}} \tag{4}$$

Where $\overline{N_i}$ and $\overline{N_j}$ are calculated used the below equation. Both represent the mean values in the Union Neighborhood of nodes *i* and *j*:

$$\overline{N_i} = \frac{\sum_{z \in UN_{ij}} N_i[z]}{\mid UN_{ij} \mid} \tag{5}$$

Both equations (3) and (4) represent ways in which nodes are similar. Therein lies the main contention of LLP: if two nodes have mathematically similar coefficients, then, regardless of graphical distance between the two, the nodes are likely to have a connection. This known as structural similarity [5].

The matrix of correlation coefficients calculated in (4) makes use of the Pearson coefficient to account for difference in structure between two nodes [64]. Through this, a clear picture of nodes with topographical similarity is developed which we will call "indirect similarity". That correlation matrix can be used to compute the DICN matrix:

$$DICN_{ij} = (1 + |\Gamma_i \cap \Gamma_j|)(1 + Corr_{ij}) \tag{6}$$

The DICN matrix, making use of the common neighbors formula as a way to represent "direct similarity", determines similarity of nodes *i* and *j* through both direct and indirect methods (hence the name). Together, the two intend to account for latent connections through the indirect similarity and existing connections - a traditional view of node popularity - through the direct connection.

4 Challenges

4.1 Switching from HMM to LLP

Originally, this paper set out to complete the aforementioned Hidden Metric Model (HMM) for Link Prediction [3] due to its use of a hidden metric space; the original paper proffers that there exists more latent variables then simply unknown or unrepresented connections. This would have been extraordinarily useful in social network analysis - following suit with the original paper's proof - as there are likely an unknown number of confounding variables when it comes to human connection [7]. The LLP, instead, assumes that the hidden metrics are unknown connections between users; no deeper undertones [5].

However, the HMM was originally coded in `C++`. When attempting to build it in python to operate with the NetworkX Library[1] and a topographical social network, the algorithm provided in the paper [3] was non-functional: even on shrunken versions of the dataset, runtime went well above multiple hours and never completed.

Thus, I chose to change direction and implement the LLP [5] instead, which operated much more concisely with Python.

4.2 Implementing the Katz Algorithm

The katz algorithm, although simplistic, proved troublesome when independently creating. Not only were all the variations coded during the duration of the experimentation process were un-optimized, but issues with formulation were not uncommon. To rectify this, the LinkPred library in Python [2] was installed and used instead. This allowed for use of an optimized version of the algorithm.

[1] `https://networkx.org/documentation/stable/index.html` [8].

[2] `https://pypi.org/project/linkpred/`

However, the Katz Algorithm operates at a runtime of $O(n^3)$; considering the social network used for experimentation was over 7,000 nodes and nearly 20,000 edges, even when split among training and testing data, no results were able to be computed using the full data set. A smaller version of the data, roughly 1,000 nodes and 3,000 edges was used to compute the model AUC for comparison purposes.

5 Methodology

5.1 Web Scraping

In order to determine the Rice University social network, data needed to be gathered - to keep in line with the topographical model required of the Latency Link Prediction model, Instagram followers for known Rice University students was obtained using a web scrape in Python.

https://github.com/brandonfantine/Instagram-Web-Scraper/

The above contains the code for the Instagram Web Scraper. However, Instagram presents the web-based version of its client through the eyes of the current logged-in user; the followers list of any profile is listed in a specific order. Beginning with what's known as "mutual followers", the first *n* number of profiles show whomever is following both the current logged-in account and the account being scraped.

Exclusively using mine (or others) personal accounts would be an inaccurate representation of the network. This presented a unique issue. A dummy account capable of only drawing information from public profiles was used to accurately gather connection information.

Over 7353 accounts were scraped in total with 19975 edges. Of those 7535, 20 had 500 nodes scraped each, resulting in over half of the edges. For the remaining accounts, I used an optimized online open-source program known as PhantomScraper[3] that was able to pull - at minimum - one connection for each of the remaining accounts. That full data set has an average degree of 2.7 per node - meaning, most accounts have at least 3 connections within the graph

The full data set, training set, testing set, and all related algorithms coded in Python can be found at the link below. Basic graphs of the data are included to help visualize; because of the high concentration of nodes, however, PyPlot is unable to draw all edges. They are not included in this report because of those same formatting issues.

https://github.com/brandonfantine/RiceU-Social-Network

The training and testing sets were in a roughly 50/50 split - reasonable distribution given the size of the dataset [9]. The training dataset contains 6486 nodes and 9992 edges. The testing dataset contains 6502 nodes and 9995 edges. The repsective average degrees for both are 1.5405488744989209 and 1.5372193171331898.

Simple graphs were created using the NetworkX library and both the LLP and Katz algorithm were implemented to test the hypothesis.

5.2 Model Implementation

Although the Katz algorithm made use of an existing library and functionality, the LLP was coded using the following non-optimized pseudocode inspired by the mathematical fundamentals of LLP.

[3] https://phantombuster.com/automations/instagram/7175/instagram-follower-collector

Algorithm 1 Latent Link Prediction

```
$N \leftarrow \{\}$
$UN_{ij} \leftarrow \{\}$
sum$_{N_i} \leftarrow 0$
sum$_{N_j} \leftarrow 0$
for node$_i \in$ graph do
    Compute $first\_order$ and $second\_order$ using node$_i$
    for node$_j \in$ graph do
        Compute $N[i]$ based on node$_i$ and node$_j$
    end for
end for
for edge $\notin$ graph do
    Compute $N_i[z]$ and $N_j[z]$ for edge
    if $N_i[z] > 0$ then
        Update $UN_{ij}$[edge] and sum$_{N_i}$
    else if $N_j[z] > 0$ then
        Update $UN_{ij}$[edge] and sum$_{N_j}$
    end if
end for
$\overline{N_i} \leftarrow$ sum$_{N_i}$/size of $UN_{ij}$
$\overline{N_j} \leftarrow$ sum$_{N_j}$/size of $UN_{ij}$
for edge $\notin$ graph do
    Compute DICN
end for
```

AUC (Area Under the Curve), an accuracy measurement, is computed for both models using the common Python library Scikit-Learn[4].

6 Results

After computing the AUC for both the LLP algorithm and the Katz algorithm, interesting results were obtained. The difference between the two were stark: the Katz algorithm had an AUC of near zero - $4.10783 * 10^{-4}$ - whereas the LLP had an AUC of roughly 0.7: 0.6868676750475808.

In terms of viability, the latter is acceptable. An AUC of 0.7 is significant enough to prove some accuracy in the model. However, because it is approaching 0.5, this implies there does exist a random element involved with the actual link prediction.

The low AUC of the Katz algorithm implies complete and total inaccuracy in the model. Almost none of the links were predicted correctly and, thus, we can conclude all other path-based link prediction models will be inaccurate. Furthermore, it is likely that populist-based link prediction models such as Common Neighbors, the Jaccard index, and Adamic Adar are likely equally as inaccurate.

7 Discussion

The relative success of the Latency Link Prediction (LLP) model and the bathos of the Katz algorithm are both illuminating from a computational and sociological perspective.

With the data set itself, the difference in performance (roughly a full 0.7 positive increase from the Katz algorithm to the LLP) does confirm the hypothesis that there are latent variables at play in connecting Instagram followers and that the model accounting for those would perform better. If this was not the case, we would have expected to see the Katz algorithm perform at a higher level.

[4] `https://scikit-learn.org/stable/`

However, like mentioned before, the AUC for the LLP is still relatively close to 0.5 implying that randomness exists among Rice University students when forming connections with each other. This does fall in-line with prior research related to discriminating against real and fake Instagram accounts [10]: fake accounts act indiscriminately, building pages in such a way that Meta's proprietary recommendation algorithm amplifies these accounts at random. Many users, because these accounts are recommended, will offer a "follow back" and make a connection.

This analysis hints at a similar phenomenon occurring within the Rice University social network. Connections between users are likely based *both* on popularity - as the proportionality of latent relationships increases as the common neighbors between nodes increases [5] - and on the random spread of accounts offered by Instagram itself.

As an ethnographic tool, social media is meant to be an extension of real-world groups [11]. Even if it is purely facial, understanding that these connections may not be fully-formed and not a reflection of human interaction can reshape how virtual ethnography is approached. In a sense, instead of using relationships on social media to represent *relationships*, they can be co-opted as a conveyance of *carelessness of the self*: a large number of connections implies an unwillingness to be selective.

The AUC scores offer a different revelation as to how future social network link prediction studies should be approached from the computational perspective. Predicated on the hypothesis that there exists latent variables within a purely topographical social network, the AUC results prove that was true; social networks, weighted or not, do contain hidden variables that influence connections between actants. Thus, for future studies, the focus should be levied onto refining hidden metric models.

Of course none of these conclusions can be confirmed without further study. However, the results from this experiment do pose an optimistic future for social network analysis and algorithmic construction.

References

[1] Daud, N., Ab hamid, S., Saadoon, M., Sahran, F., & Anuar, Nor., (2020) "Applications of link prediction in social networks: A review." *Journal of Network and Computer Applications* Vol. 166. No. 102716.

[2] Katz, L. (1953) "A New Status Index Derived from Sociometric Analysis", *Psychometrika* , pp. 39-43

[3] Junker, B. H. & Schreiber, F. (2008) "Analysis of Biological Networks" Hoboken, NJ John Wiley & Sons.

[4] Alharbi, R., Benhidour, H., & Kerrache, S. (2016) "Link Prediction in Complex Networks Based on a Hidden Variables Model" *UKsim-AMSS 18th International Conference on Computer Modelling and Simulation*

[5] Zareie, A. & Sakellariou, R. (2020) "Similarity-based Link Prediction in Social Networks Using Latent Relationships Between the Users" *Scientific Reports*, Vol. 10, No. 20137

[6] Schober, P. Boer, C. & Schwarte, L. (2018) "Correlation Coefficients: Appropriate Use and Interpretation", *Anesthesia and Analgesia*, Vol. 126, No. 5, pp. 1763-1768

[7] VanderWeele, T.J. & An, W. (2013) "Social Networks and Causal Inference", *Handbook of Causal Analysis for Social Research* Handbooks of Sciology and Social Research Springer Science + Business

[8] Hagberg, A. Schult, D. & Swart, P. (2008) "Exploring network structure, dynamics, and function using NetworkX", *Proceedings of the 7th Python in Science Conference* (SciPy2008), pp. 11-15

[9] Dobbin K.K. & Simon R.M. (2011) "Optimally splitting cases for training and testing high dimensional classifiers", *BMC Medical Genomics* Vol. 4, No. 31

[10] Saeidi, S. & Baradari, Z. (2023) "Examining the correlation between metrics in the Instagram social network to identify fake pages and improve marketing", *Computers in Human Behavior Reports*, Vol. 12

[11] Wang, D., & Liu, S. (2021) "Doing Ethnography on Social Media: A Methodological Reflection on the Study of Online Groups in China", *Qualitative Inquiry*, Vol. 27, No. 8-9, pp. 977-987.