Future link regression using supervised learning on graph topology

Adam Goldberg Jonathan Hung Mark Ulrich

I. Introduction and Problem Statement

Link prediction provides useful information for a variety of graph models, including communication, biochemical, and social networks. Link prediction is used on social networks to suggest future friends and in protein interaction networks to suggest possible undiscovered pairwise interactions. We investigate email networks, introduce an analogous problem to classical missing link prediction, which addresses new interactions between nodes. We will predict the number of emails that will be sent from one person to another in a fixed future timespan. We use supervised techniques based on graph topology such as shortest path and random walks to form multiple predictors which are utilized in supervised regression techniques. Note that we have deviated from our original multigraph approach as we believe it it may be too difficult of a problem, since the solution space of edges is infinite.

II. Relevant Literature

Many researchers have explored the topic of missing link prediction. Kim and Leskovec [8] use node attributes to predict the probability that two nodes will form a link. In particular, they use a "Multiplicative Attribute Graph" model that contains information regarding each node's attributes and the tendency for attributes to cause links to form. Gong [4] tackles a similar problem, using a model called the "social-attribute network", in which a standard social network is augmented with attribute nodes. These attribute nodes form edges with the social nodes that are related to the attribute, and the authors show that running standard algorithms on this augmented network leads to better link prediction results.

Lictenwalter [3] takes a different approach to link prediction. They apply supervised learning strategies and achieve over 30% improvement in AUC. To do this, the authors used general node features of networks, such as (in- and out-) degree, as well as features between two nodes such as max flow, shortest path, or PropFlow. We will utilize a combination of features used by the above researchers and [6], combining their node-based features and pair-based features, for a more holistic approach, similarly to what is seen in [9].

III. Data

I. Collection and Preprocessing

We explored our problem via the Enron Email Dataset, which is a conventional email dataset [7]. It was collected as a subpoena of the computers of 150 Enron employees, and therefore is a collection of emails sent to and from those people during this time period. Key fields, such as the sender, recipiant, and date of each message were parsed out using regular expressions and then converted into a directed network, with people represented as nodes, and communications between people treated as edges. As in [2], we weight our edges by the number of messages sent in that direction, and annotate the edges with the dates of the contacts.

After inspecting our dataset, we eliminated incomplete and unparsable data. In the end, our dataset has 7475 nodes and 50098 edges. It appears our graph has a preferential attachment with $x_{\text{min}} = 7$ and $\alpha = 2.1006$. Huang [2] mentions that we should not eliminate statistical outliers so as to not bias the data, so we also chose to not.

We also computed the cumulative distribution of emails over time. This is a reference to the windows of time that will be most accurate in predicting number of emails sent, depending on the density or sparsity of the data in that timeframe.

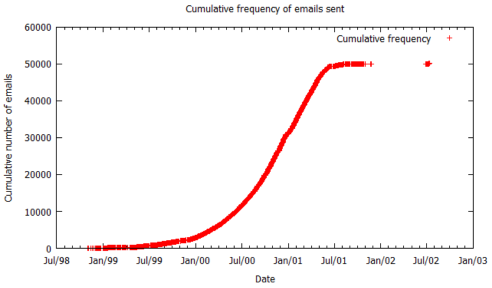

Figure 1: *Cumulative Frequency of E-mails*

Figure 1 shows that the density of emails is higher, so with a fixed time window, there is more training data at these dates.

II. Summary Network Statistics

The assortativity coefficient is a measure of the likelihood that two nodes have an edge based on their node degrees. A high assortativity coefficient (close to 1) indicates that nodes with similar degree are likely to have a link. The assortativity coefficient for this network is 0.0195, indicating that nodes having similar degree has little effect on whether or not they have a link. In this network, that means if two people send a similar number of emails, this reflects little on the chance that they send emails to each other.

The average clustering coefficient is measured to indicate how likely it is for an email address's recipients to email each other. In this graph the average clustering coefficient is 0.111164.

Mean degree gives a rough estimate of how many emails each person sent/received. In this network the mean in/out degree is 6.702970.

The in-degree distribution is not included because the data does not represent the full set of emails sent between all employees. From the way the data was collected, the recipient of each email is 1 of 150 people, meaning the number of people with in-degree greater than 0 is 150. With this sample size, it is difficult to deduce any pattern in the data.

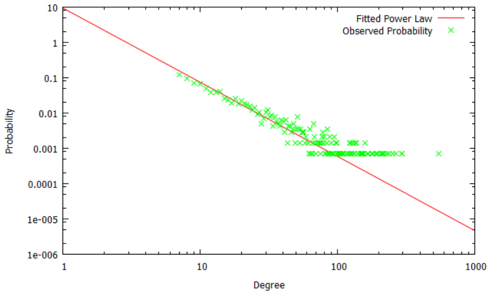

Figure 2: *Out-Degree Distribution*

The out-degree distribution in Figure 2 gives us an idea of how many people send how many emails. It fits a power law with $\alpha = 2.100595$ and $x_{\text{min}} = 7$.

1. Median in-degree: 0
2. Median out-degree: 2
3. Number of SCCs: 7394
4. Largest SCC: 80 nodes
5. Largest SCC diameter: 7

In particular, as a sanity check, we note that the maximal strongly connected component size is 80, which is reasonably close to the 150 email addresses subpoenaed.

IV. Problem Statement

Define two time intervals Δ_x and Δ_y and a discrete time t. We view our problem as creating a regression function $f(X)$ where X is a pair of nodes n_1, n_2, and features f_1 on n_1, f_2 on n_2 and f_3 on n_1 and n_2, all of which are over windows of size Δ_x, starting at t. The target of $f(X)$ is the number of expected emails, within Δ_y timesteps later than $t + \Delta_x$. For concreteness, we consider Δ_y to be one month. Before proceeding, we use our feature distributions to try and determine what the minimal reasonable window of Δ_x is that gives us a good chance of predicting $f(X)$.

V. Features

We wanted to select a robust feature set so we could perform ablations and iteratively remove feature to massage out our stronger signals.

1. PageRank (of each node)
2. Degree (of each node; undirected)
3. PropFlow (from n_1 to n_2)
4. AdamicAdar (on n_1, n_2 in order)
5. CommonNeighbors
6. Volume (of each node)
7. Jaccard Coefficient (number of common neighbors divided by total neighbors)
8. RootedPageRank
9. TargetValue (number of emails sent between the nodes during training period)

We tried the features as predictors in isolation and tested their effects using univariate linear regression tests.

Their average F-scores across all windows can be found below.

Feature	F-Score (average)
TargetVals	3.092e+06
CommonNeighbor	6.830e+02
IDegree	1.426e+02
IPageRank	4.681e+01
IVolume	7.132e+02
JaccardCoefficient	1.148e+02
JDegree	6.510e+02
JPageRank	9.316e+03
JVolume	3.264e+02
PropFlow	9.240e+04
RootedPageRank	6.212e+04

Huang [2] Mentioned that predictors that work on one dataset for this problem do not necessarily generalize well to another dataset. We can also see this behavior here, noting that only 4 of our features received F-scores above 1e+03.

I. Machine learning and tuning

Using the above f-scores, we decided to proceed without features with an f-score below 500. This left us with 7 features. We define our regression over examples for a given window and their target values and then run k-fold cross-validation on individual windows, with $k = 4$, to try to avoid overfitting. Because our dataset is so large, We would like to be window-agnostic and compare arbitrary examples across windows, but currently we cannot process the 3,000,000 examples that exist across all of our windows. After doing preprocessing on our data and recentering it around 0, we tried a variety of regression algorithms.

VI. Initial Results

As our problem is a regression, we use a traditional R^2 coefficient of determination to measure our accuracy. In particular, our definition allows for negative values, so if our regression fit is worse than the mean of the data, the coefficient will be negative.

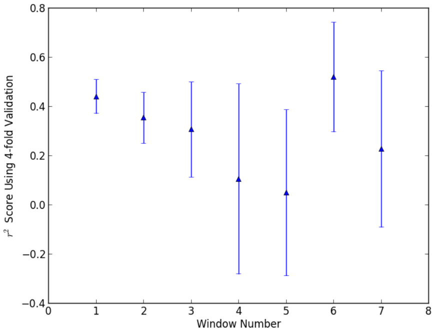

Figure 3: *Predictive Power Varies Accross Time*

As we can see in the above figure, there are several instances of negative R^2 coefficients, which indicates the function we are trying to learn is possibly nonlinear. We tried several common nonlinear regression algorithms, as well as additional generalized linear models with error results below (compared against linear regression for benchmarking).

Method	R^2 (avg.)	R^2 (σ^2)
Linear	0.3673	0.1439
RandForest	0.1669	0.0639
GradBoosting	0.1732	0.3207
SGDRegression	0.3576	0.0053
SVR (RBF)	0.0159	0.0002
BayesRidge	0.3674	0.1439
Lasso	0.2230	0.0834
ElasticNet	0.3129	0.0993
Ridge	0.3674	0.1439

It is clear we have problems with high error variance. Thus we will investigate more variance-reduction techniques relating to bagging, bootstrapping, and general ensemble learning. It may be beneficial to generalize these techniques to be sampling random subnetworks from our original graph, rather than sampling examples and features. Additionally, we did not have enough time to optimize parameters for the various techniques, so there may be more room for improvement there. Potentially applicable techniques include grid search or simulated annealing.

Figure 3 was calculated using feature and label windows of 100 days starting on 05/18/1999 using simple linear regression and all base features achieved a mean R^2 of 0.287. The accuracy is highly dependant on the time period, possibly because changes in company structure could render certain patterns obsolete accross large time windows. This corroborates our earlier insight that we could consider examples independently of their windows to create a more general model that is not specific to a certain time period. On the other hand, as we see in Figure 4, there is a Goldilocks zone for time windows. If you predict more than 100 days into the future things become more difficult, while when using less than 10 days the data is too noisy. Using an extra large feature window seems to yield particularly poor results if you are not predicting far into the future. We will further investigate window-related considerations in our final report.

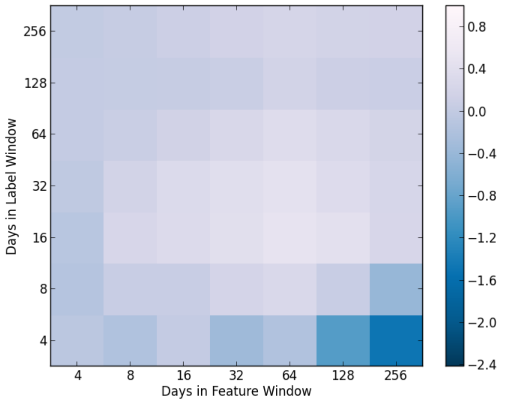

Figure 4: *R^2 Results Predicting Emails Sent in the First Y Days of Year 2000 Using Data From the Last X Days in the End of 1999*

VII. Future Work

Currently, we have not reversed our graph to calculate predictors like in-degree, and in-volume. This will enable us to also try other predictors like sums and products of them, such as the preferential attachment link prediction score, which is defined as the product of two node's in-degrees or out-degrees and was successfully used in [5]. We also will attempt to use email metadata, for example the subject line or message body. Carvalho [1] had success predicting leadership roles from emails by classifying emails into five categories: requests, deliveries, proposals, commitments, and meetings, so we will explore it further. Other potential additional features include measures of burstiness (i.e. the groupings) of messages sent between nodes, which we might model using fractal dimension, or even various Fourier transform coefficients of the emails between people when viewed as a time-series.

Additionally, we need to figure out how to make sure we are reasonably comparing examples from different windows, so we don't overfit to specific windows. Our main struggle is that our dataset is too large, comprising of, for example, approximately 3,000,000 examples when windows are 100 days. We will explore dimensionality reduction techniques, as well as ensemble learning techniques to attempt to manage the size of the dataset.

References

[1] *Discovering Leadership Roles in Email Workgroups*. Vitor R. Carvalho, Wen Wu and William W. Cohen, 2007.

[2] *Link prediction based on graph topology: The predictive value of the generalized clustering coefficient*. Huang, Z. In Workshop on Link Analysis (KDD). August 2006.

[3] *New perspectives and methods in link prediction*. Lichtenwalter, R. N., Lussier, J. T., and Chawla, N. V. (2010, July). In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 243-252). ACM.

[4] *Joint Link Prediction and Attribute Inference Using a Social-Attribute Network*. Gong, N. R., Talwalkar, A., Mackey, L., Huang, L., Shin, E. C. R., Stefanov, E., et. al, ACM Transactions on Intelligent Systems and Technology (TIST), v.5 n.2, p.1-20, April 2014

[5] *Predicting the growth of new links by new preferential attachment similarity indicies* Hu, K., Xiang, J., Xu X.K., Li, H.J., Pramana Journal of Physics, v.82 n.3, March 2014

[6] *Modeling Networks with Auxiliary Information* Kim, M., Dissertation submitted to the Department of Electrical Engineering at Stanford University, August 2014

[7] *Exploration of Communication Networks from the Enron Email Corpus* Diesner, J., Proceedings of Workshop on Link Analysis, Counterterrorism and Security SIAM International Conference on Data Mining 2005 (2005), pp. 3-14

[8] *Modeling Social Networks with Node Attributes using the Multiplicative Attribute Graph Model* Kim, M., Leskovec, J.

[9] *Multiplicative Attribute Graph Model of Real-World Networks* Kim, M., Leskovec, J., CoRR 2011