Analysis of the Pinterest Dataset Over Time: Prediction of New Linkages in the Graph

Brandon Cui
Stanford University
bcui19@stanford.edu

Matthew Kim
Stanford University
mdkim@stanford.edu

Abstract

This paper aims to analyze how a social media network evolves over time. We use a food-related Pinterest dataset to analyze how a selected number of topological features can predict new edges between pins and boards by fitting logistic regression, lasso, and SVM models to a temporal link prediction problem. We observe that over time, the distribution of degrees of nodes remains does not significantly change, various features such as the clustering coefficient converge, and it becomes more easy to predict linkages.

1. Introduction

Pinterest is a social media platform in which users discover and bookmark content on the web as "pins" that are attached to "boards" by topic. Users can follow boards and discover images, products recipes, articles, etc. Similar ideas and content are naturally curated into boards which presents an opportunity for to study temporal link prediction, which amounts to predicting, given a time period, new edges that will form in a graph in a future time period.

Link prediction has many important applications. The backbone of any recommendation system for information retrieval involves link prediction to help people find new friends, products, or ideas. Another application is inference, in which one can attempt to understand the evolution of networks or as in Marchette et al., predict unobserved links in incompletely observed networks. Finally, forecasting models that are used to predict the spread of a virus or study traffic congestion are direct applications of temporal link prediction.

1.1. Related Work

Liben-Nowell, et al. studied the link prediction problem for five large co-authorship networks and their experiments, which used a variety of network topology features, suggested that network topology provided sufficient information, with some proximity measures leading to predictions that outperform chance by factors of 40 to 50.

Hasan, et al. also explored the link prediction problem but focused on translating link prediction as a supervised machine learning task. Using easily computable features and a number of different classes of machine learning algorithms, the authors found that the simple, standard features they identified ended up effectively solving the link prediction problem and that of the classification algorithms, SVM (support vector machine) performed the best on accuracy, precision-recall, and F-values.

2. Dataset

The dataset we are currently using is the Pinterest dataset found on the website for the Stanford Network Analysis Project. The Pinterest dataset consists of a few components: 10.3 million Pinners (user_id), 12.4 million Food-related Board (board_id, board_name, board_description, user_id, board_create_time), 20 million unique Pins (pin_create_time, board_id, pin_id), and 14.1 million follows(board_id, user_id, create_date), where the board_description is a textual description of the board. We considered several different network combinations as seen below:

boards and pins
board, pinner, and pins
board, follows, and pinner

For this work we only consider the network that consists of boards and pins; thus all result presented below are drawn from the network created from the board pin network. We note that this is a bipartite graph because boards can only connect to pins and pins can only connect to boards. Additionally, for computation purposes we subsample the dataset to only work with 250,000 pins and the top 500 boards. The pins were selected via random sampling of all of the pins,

and the boards were determined by the highest degree based off the random sampling of pins.

3. Problem Definition

The goal of this project is two-fold:

1. To perform network analysis to see how the network forms over time, through various network characteristics and see on average how these properties change as the network expands.
2. Try to predict future linkages in the graph. For example given a pin which board will it most likely connect to next.

4. Methodology

4.1. Time Domain Analysis

We studied the effect of how networks started to form over time. We begin by partitioning our time stamps based upon intervals of time; this was done in a harmonic manner where we would only consider (eq. 1):

$$\frac{N}{i} \tag{1}$$

points, where N represents the total number of pins and $i \in \{1, 2, \cdots 98\}$. From here we would form a graph where we would consider both boards and pins as nodes and an edge would be considered to be a pin to a board. However, we would only consider pins in our network analysis.

4.2. Prediction of Linkages

For computational purposes, we partitioned the pin dataset into $k = 50$ even timesteps, where we only considered the pins to be connected to a board if their connection was formed before the current timestep. This was done so that every timestep had an equal number of new 'pinning' interaction, e.g. if a pin was 'pinned' to a board. For every pin, at every timestep we consider every pin to have the following topological features:

- **Mean Neighbor Degrees (Pins)** - Mean of the pin nodes that are at most two degrees away
- **Mean Neighbor Degrees (Boards)** - Mean of the board nodes that have an edge connecting the pin.
- **Standard Deviation of Neighbor Degrees (Pin)** - Standard deviation of the pin nodes that are at most two degrees away
- **Standard Deviation of Neighbor Degrees (Boards)** - Standard deviation of the board nodes' degrees that have an edge connecting the pin.
- **Clustering Coefficient** Since our graph is a bipartite graph, we modify the vanilla clustering coefficient for a given pin p_i to be:

$$C_i = \frac{|\{(e_{jb}, e_{kb}) | v_j, v_k \in N_b, e_{jb}, e_{kb} \in E\}|}{k_i(k_i - 1)} \tag{2}$$

where $N_b = \{v_j | e_{jb} \in E \cup e_{jp} \in Ee_{ip} \in Ee_{ip} \in E\}$. So essentially we look at all board the current pin, p_i was pinned to and consider N_b to be the set that consists of pins that are pinned to at least two boards that the pin p_i was also pinned to.

4.2.1 Prediction of number of linkages

Due to the complexity of the problem with predicting future linkages, we considered a relaxation of our overall goal, which is to predict the number of linkages each pin will create at the next timestep. We will consider this through multiple methods:

1. Using just the current timestep as training data, and trying to predict the number of future linkages.
2. Use all previous timesteps as training data to try to predict the number of future linkages.

For both of these methods we will look into a variety of machine learning methods to predict the number of expected linkages, including Linear regression, least absolute shrinkage and selection operator (LASSO), and support vector machine (SVM) regression. We will consider the error of this problem to be the following (eq 2.):

$$\frac{|\hat{y} - y|}{y} \tag{3}$$

where $\hat{y}$ is the predicted number of linkages and y is the actual number of linkages.

5. Results And Discussion

5.1. Basic Network Analysis

From the time domain data we constructed 96 networks, and below we plot the degree distribution of nodes from a subset of these networks along with the network distribution of the Erdos-Renyi and preferential graph (Figure 2):

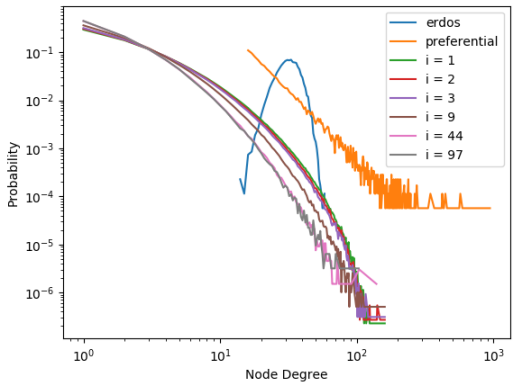


Figure 2: Node Degree Distribution for the time domain pinterest network where $i \in \{1, 2, 3, 9, 44, 97\}$, Erdos Renyi Graph, and Preferential Graph.

We note that the plot between the pinterest dataset and the erdos/preferential networks look very different in their distributions, but this is inherent due to the number of nodes that exist in the pinterest network. Based upon the time domain, we note that as we add nodes to the graph over time, there is no dramatic change in the general structure. This is indicated by the fact that the overall shape of the node-degree distribution remains relatively the same, the only difference is that the curve gets sharper as we add more and more nodes. Generally, the trend observed is that as we add more nodes, the higher degree nodes get higher in degree, but we also note that this network doesn't act like a preferential network, but characteristically it appears to follow a trend that can be similar to that of the preferential network.

We now consider the difference between the pinterest network over time and the preferential network. For this we design a loss function which in its vanilla form is (eq. 3):

$$L(N_1, N_2) = \sum_{i=1}^{m} (N_{1i} - N_{2i})^2 \tag{4}$$

where N_j represents the probability vector of the node degrees, N_{jk} is the kth entry of the N_j vector, and m is the total number of node degrees.

Additionally, we consider a normalized form, where we normalize all the degrees in the following manner (eq. 4):

$$n_i' = (n_i) N_{2max} / (N_{1max}) \tag{5}$$

where n_i' represents the new degree, n_i is the old node degree taken from the distribution N_1, N_{imax} is the maximum node degree in the ith distribution vector. Below we plot the difference of both loss functions (fig 3):

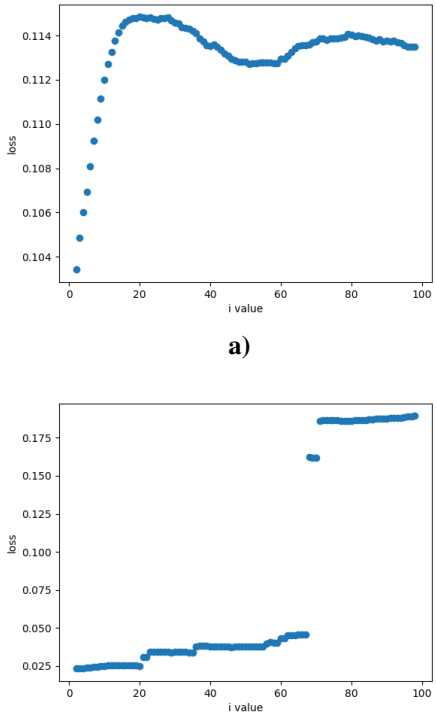
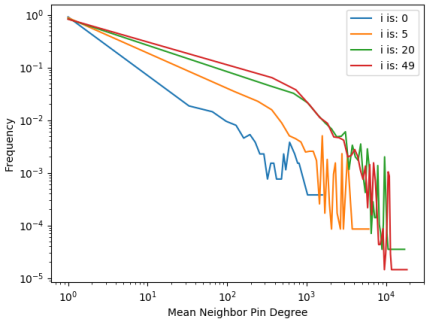


Figure 3: Loss function (eq 3) over time for the pinterest dataset **a)** Unnormalized loss **b)** Normalized loss where the preferential network was normalized via eq 4 and loss was computed via eq 3.

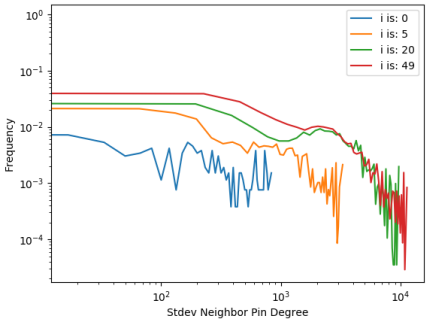
We note that the loss function indicates that there is lower loss at lower i values; overall the difference between the preferential network and the pinterest network is low. However, we note that with our normalization we are better able to understand the dataset and we note that the dataset appears to function closer to the preferential network at a low timestep.

5.2. Feature Time Domain Analysis

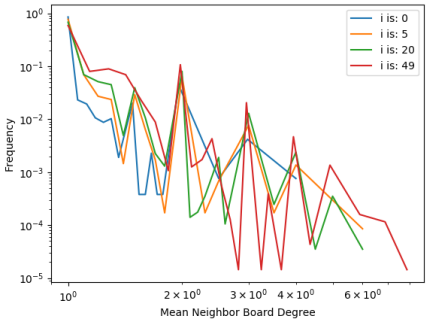
For the purpose of better understanding how the graph changes over time, we partitioned the dataset into 50 equal timesteps by number of interactions as described in our methodolog section. Below we plot the distribution of features for selected timesteps below (Fig 4 a), b), c), d), e)):



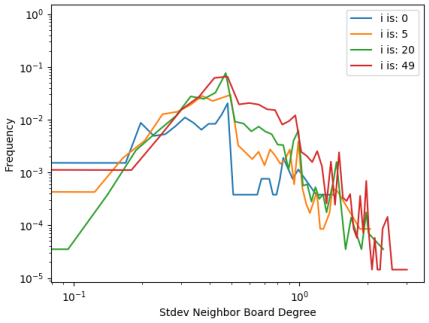
a)



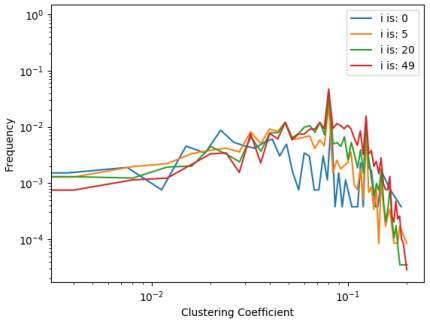
b)



c)



d)



e)

Figure 4: Plot of distribution of features over timesteps when $i = 0, 5, 20, 49$ for **a)** Mean Neighbor Pin Degree, **b)** Standard Deviation of Neighbor Pin Degree, **c)** Mean Neighbor Board Degree, **d)** Standard Deviation of Neighbor Board Degree, **e)** Clustering Coefficient, and i represents the timestep

Generally, we notice that at timestep 0 that the data has a pretty uniform frequency across all features. We notice that the maximum and minimum frequency at which two values occur per a feature is approximately one to two orders of magnitude within each other; thus, showing that the features are relatively well distributed.

However, we note that as the size of the graph increases that the distribution of features starts to spread out more, with more values being concentrated at lower values for all such features. This also indicates that it appears that the graph is stabilizing since the concentration of features starts to favor smaller values for the mean and standard deviation of pins that are at most two degrees away and the mean of the neighbor board degrees. However, we note that the standard deviation of the neighboring board degree seems to follow this trend, but the standard deviation favors a more 'middle-of-the-road' value. We would also like to point out that there are a few pins at longer timesteps that have an extremely high number of mean neighboring pins in two degrees away.

Lastly, we note that the bulk of the clustering coefficient seems to be at higher values when we proceed to higher timesteps. This indicates that clusters tend to form as the graph starts to stabilize more over time.

5.3. Prediction of Number of Connections

For the first step in our prediction, we consider the relaxed question where we only want to predict the number of linkages. Here we also only consider a subset of the timesteps up to $k = 20$ for computational purposes, as when running SVMs they tend to converge extremely slowly at much larger values. We present below a table of the averaged error over all timesteps of various methods (table 1):

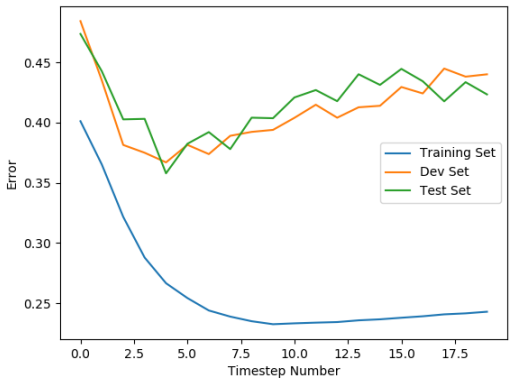
Method	Train	Dev	Test
Linear Regression (full)	0.392	0.388	0.390
Linear Regression (previous)	**0.2057**	**0.2007**	**0.2012**
LASSO $\alpha = 0.1$ (full)	0.423	0.424	0.425
LASSO $\alpha = 0.1$ (previous)	0.2084	0.2032	0.2038
SVM (full)	0.2612	0.4099	0.417
SVM (previous)	0.283	0.4194	0.4327

Table 1: Average training, dev, and test error for prediction of future linkages over all timesteps

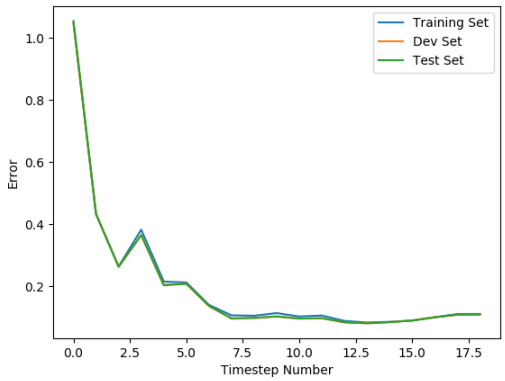
We note that the "full" parameter is when we train on all such previous timesteps and the "previous" parameter is when we just train on the last seen timestep. Additionally the α parameter associated with LASSO is the typical regularization parameter. Also, the kernel used for the SVM was the rbf kernel. We note that the linear regression and lasso models trained significantly faster than the SVM model.

We notice that when training on all previous timesteps versus just the last timestep the performance of the SVM is relatively unchanged. However, when we only look as the previous timestep for linear and lasso regression we notice that there is a significant performance increase when compared to all previous timesteps. Based on these results, we believe that for the selected features they vary from timestep to timestep and as a result demonstrate the underlying idea that the graph does indeed change pretty dramatically over time. However, we also note that it is interesting that linear regression performs the best

Below we plot the results for the linear regression when sampling just the previous timestep and SVM model below (Fig. 5 a), b)):



a)



b)

Figure 5: Error calculated using eq. 3 when predicting the number of linkages in the next timestep when using **a)** 'full' SVM Regression **b)** 'previous' Linear Regression

We notice that for the first timestep there is a significantly higher error when we just use vanilla linear regression. Additionally, we note that this high error is also present when we add some regularization term. However, after one or two timesteps we notice that the error drops off significantly. In fact we would like to point out that after 5 timesteps tat there is less than 10% error when predicting the number of linakges per a node.

We also find it interesting that for the SVM the error minimizes for the dev and test set at 5 timesteps and at 10 timesteps for the training set. After this timestep the error continues to rise. We believe that this can possibly be attributed to the fact that the previous timesteps do not provide an accurate understanding of the data, since from our previous feature analysis of the graph we note that the features and distribution of features rather dramatically over time.

6. Discussion

From our work we've studied how the pinterest dataset functions and evolves over time. We note that in its early stages it acts most closely to the preferential network, but as the graph evolves it grows away from such a form.

However, we also note that as the graph continues to evolve, it appears that the graph appears to stabilize more and we can see that various features start to converge on certain values. Moreover, the clustering coefficient tends to grow higher, the mean degree of pins within two degrees tends to concentrate at lower values, and the standard deviation starts to be concentrated at lower values.

Based on a small 5 dimensional feature set we were able to well model the number of future links with less than 10% error at the fully developed graph. However, we also would

like to point out that at early timesteps when the graph most closely mimics a preferential graph, there is the highest error.

6.1. Limitations

It is important to note that we could have considered a larger array of features that would have helped us better understand the network. However, even with such a small set of features we were able to gain useful insight into how the network grows and demonstrate when the network begins to tend towards its final form it is reliably easy to model/predict new linkages.

6.2. Future Work

The majority of this work focused on better understanding the network and attempting to model how the network would look in the future. However, we only considered a relaxed version of the problem where we would predict how many pins would the current pin have at the next timestep. In the future we would like to attempt to use various methods including methods drawn from image captioning (Karpathy et al. 2015), using word embeddings from (Pennington et al. 2015), low dimensional graph embeddings as seen in (Hamilton et al. 2017) to try to predict direct links rather than just number of linkages.

References

[1] Hasan M. A., Chaoji V., et al. Link Prediction using Supervised Learning. SDM06. 2006.

[2] Liben-Nowell D., Kleinberg J. The Link Prediction Problem for Social Networks. Journal of the American Society for Information Science and Technology. 2007.

[3] Liu D. C., Rogers S., et al. Related Pins at Pinterest: The Evolution of a Real-World Recommender System. International World Wide Web Conference Committee. 2017.

[4] Gomez-Rodriguez B.S., Balduzzi D. Uncovering the Temporal Dynamics of Diffusion Networks. The 28th International Conference on Machine Learning, 2011.

[5] Karpathy A., Fei-Fei L. Deep Visual-Semantic Alignments for Generating Image Descriptions. Conference on Computer Vision and Pattern Recognition. 2015.

[6] Hamilton L. W., Ying. R., Leskovec J. Inductive Representation Learning on Large Graphs. In Advances in Neural Information Processing Systems 30, 2017.