Improving Pinterest Recommendations with Link Diffusion and Network Inference

Abhijit Sharang
abhisg@stanford.edu

Eric Lau
eclau@stanford.edu

Rishabh Bhargava
rish93@stanford.edu

Abstract

Pinterest is a popular tool that allows users to search for and bookmark new ideas. These ideas manifest as 'pins' which can be catalogued to 'boards'. We develop a comprehensive recommendation engine which recommends relevant pins to boards and users. We also propose deterministic, probabilistic, and novel hybridized methods to obtain non-explicit user-user relationships. Exploiting this, we study user interactions with several different pins and resulting patterns of influence in the user-user graph. We perform experiments on a smaller and denser version of the user-pin-board graph; and a larger and sparser version as well. Our studies show these methods achieve promising results relative to baseline methods in obtaining accurate, personalised recommendations.

1. Introduction

Personalised recommendations are vital to an Internet company's user experience, whether it is an online marketplace business like Amazon or a video streaming service like Netflix. Platform monetisation depends on inferring a user's tastes and using it to present products or ideas to which the user is receptive. Since users do not operate in isolation, a common approach to building recommendation engines is to decipher common user interests. In a network theoretic approach, user-item relationships are abstracted as a heterogeneous network. This network is usually modelled as a bipartite graph between the heterogeneous entities, since relations between similar entities are usually absent. Recommendations are then made by performing link prediction between the heterogeneous entities. These approaches thus rely on abstracting the relation between the homogeneous entities through their interaction with entities of other categories. The more similar these interactions are, the greater the assumed similarity between the homogeneous entities. The missing link between, say, a user and an item can be obtained if other "similar" users have interacted with the particular item.

In the context of Pinterest, we leverage the pin-board, pin-user, or user-board sub-networks to make recommendations between any pair of heterogeneous categories. Specifically, we build an engine which recommends pins suitable for a board; and pins for a user's homefeed. We first propose a version of random walks with restarts on heterogeneous graphs to generate recommendations for boards through a novel modification of the HITS algorithm [6].

Second, we incorporate the effect of one user's interaction with a pin on those of other users with the same pin to generate pin-to-user recommendations. In particular, we exploit the techniques of a recent area of research called *link diffusion*, where instead of using nodes as the basis for "contagion" spread, we analyse the phenomenon of edges spreading "contagion" to other edges in the network. This requires homogeneous user-user relationships not present in the original graph. We first explore a simple deterministic method of generating these relationships from available board-follower data. We then augment this with a more sophisticated probabilistic approach to infer these relationships with maximum likelihood. Finally, we measure and analyze our recommendation engine's performance on recommending relevant pins to boards and to users relative to common baseline models. Our methods generalize to any bipartite graph with heterogeneous entities and are potentially useful in a broad range of recommendation contexts.

2. Related Work

Building recommendation engines has been studied extensively in different domains. The most common methods involve either collaborative filtering, which involves studying the history of interaction of a user with the heterogeneous user-item graph; or content-based filtering, which involves building a "feature" profile for the users and the items. More sophisticated methods may hybridize the two.

A graph-based method for solving the recommendation problem involves link prediction on heterogeneous graphs. A naive way to perform link prediction is by posing it as a classification problem. [2] attempts to improve upon this method by introducing supervised random walks. This algorithm assumes the network to be homogeneous and hence there are no constraints on the random walk. For heterogeneous graphs, the translation is not straightforward, as the supervised learning needs to take into account the difference in transition from nodes of one type to another. Also, this approach might be prone to error accumulation because of the added constraint of transitioning only between nodes of opposite kinds in the optimisation problem. Hence, an Occam's razor approach to solving the problem might be better.

An interesting angle of looking at link prediction is through information diffusion. The principle could be applied to study the cascades of edge creation in the network, where one can reason about the effect of the formation of an

earlier link on the creation of subsequent links, which would allow for link prediction between nodes with a certain degree of confidence. [8] attempts to study how the action of a user "following" another user can affect the behaviour of other users in the network. This is a clever modification of the independent cascade (IC) model [5], where the cascading influence of links is studied on other links rather than that of nodes on other nodes. However, when the graph is bipartite, the follower-followee network does not explicitly exist and has to be inferred. This can be done either using a deterministic simrank-like approach [4] or by a probabilistic network inference method to assign edge strengths between nodes of the same kind. In particular, [7] explores computing maximum-likelihood edge diffusion strengths consistent with the temporal dynamics of a cascade set, or observed activations of nodes in the graph.

3. Methodology

3.1. Recommending Pins to Boards

We devise a variation of the random walk with restarts algorithm and use it to obtain personalised PageRank scores for each pin in the database with respect to each board. We then recommend pins with the highest scores which are not already present to the board as candidate pins for that board.

Suppose p_k is the stationary pin distribution and b_k is the stationary board distribution when random walks with restarts is done from the k^{th} board. Adopting a HITS-like algorithm, we obtain the following equations for the board to pin transition and the pin to board transition:

$$p_k = B_P^T b_k$$

where B_P is the board to pin transition matrix, and

$$b_k = (1-\epsilon) P_B^T p_k + \epsilon \mathbf{1}_k$$

where $\mathbf{1}_k$ is a column vector with dimensionality equal to the number of boards, which is 1 at the k^{th} position and 0 everywhere else, and ϵ is the teleportation probability. From the above equations, we get

$$p_k = (1-\epsilon) B_P^T P_B^T p_k + \epsilon (B_P^T)_k$$

where $(B_P^T)_k$ is the k^{th} row of the board-to-pin transition matrix. Thus, the teleportation set consists of all pins which are already present on the k^{th} board for which the personalised pagerank scores are being computed.

We use a single-dimensional feature to determine the importance of a pin on a board and vice-versa. The importance spills into the transition probability from a pin to a board and vice-versa. For any board-pin pair, if the age of the pin j on the board i is x, we define the feature $f_{ij} = e^{\alpha x}$, where α is some decay constant. The feature aims at giving higher importance to the newer pins on the board, since users respond more positively to fresher content. For all board-pin pairs such that the pin does not appear on the board, the age is set to $-\infty$ and hence the feature remains zero. Now, matrices B_P and P_B are defined for all (i, j) pairs which are linked as follows:

$$(B_P)_{ij} = \begin{cases} \frac{f_{ij}}{\sum_{k \in \text{pins}} f_{ik}} & \text{if } f_{ij} \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$(P_B)_{ij} = \begin{cases} \frac{f_{ij}}{\sum_{k \in \text{boards}} f_{ik}} & \text{if } f_{ij} \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

The matrices B_P and P_B need not be symmetric since the importance of a pin to a board might not be the same as that of the board to the pin.

3.2. Recommending Pins to Users

We seek to determine the probability that a user will collect a certain pin given that other users have also interacted with and collected this pin previously. This can be understood as learning the cascading influence of a user's action on other users. We attempt to solve this problem through triad closing measured using link diffusion, which we elaborate in the subsequent subsections. Once the parameters for link diffusion are learned, we can use triad closing to recommend pins to a user.

3.2.1 Graph Formation

Since Pinterest does not have a direct follower relation between users, we mimic the follower relation using boards as proxies. We say that a user follows another user if the user follows a board that the other user created. This deterministically creates pseudo-edges between users. A pseudo-edge between users A and B can be of four kinds: A follows B, B follows A, and both A and B follow each other, with the last kind of pseudo-edge divided into two parts depending on the ratio of each other's boards that A and B follow. This graph when joined with the user-pin graph, forms a quasi-bipartite heterogeneous graph with the user component of the graph having the aforementioned pseudo-edges.

3.2.2 Problem Formulation

Once these pseudo-edges have been created, we transform the problem to time-decayed triad activation prediction. Specifically, we address the following question: given that a user A has pinned a pin p on some board, what is the probability that a user B who has some relation with A as defined above will also pin p and close the triad within a certain time interval? This can be thought of as a time-decayed link diffusion phenomenon, where the link between A and p leads to link formation between B and p after some time interval. We divide these triads into four kinds depending on whether A follows B, B follows A, or both A and B follow each other with varying strengths. For the last kind, we use the ratio of the number of boards A and B follow, thresholded at 0.5, to assign the relevant triad. Each triad is parameterised by a discovery parameter h_Δ and a diffusion parameter g_Δ, where the discovery parameter is the probability that the prior link is discovered by the user in the network during the time the diffusion "contagion" is active; and h_Δ is the probability that the new link is formed during this time once the discovery is made. We aim to learn these $4 \times 2 = 8$ parameters.

Suppose the new link between the user and the pin is formed at time t. We consider all links which existed between $[t - \delta, t]$ between the pin and other users in the

follower-followee network for this particular user, where δ is some time parameter. Suppose S_e is the set of users in the network with "fresh" links which can still diffuse, and R_e is the set of users in the network with "stale" links which can no longer cause diffusion. Furthermore suppose that $x_{e,e'}$ is the probability that an earlier link e activates a later link e' at time t' and $y_{e,e'}$ be the probability of it failing to do so, i.e. it activates it anytime after t', which can also be infinity is the link is never formed. Then, following the independent cascade model and assuming that each prior link gets one chance at activating a new link, the probability that at least one e is able to cause diffusion at time t' is

$$L(e') = (\prod_{e \in S_e} (x_{e,e'} + y_{e,e'}) - \prod_{e \in S_e} y_{e,e'}))(\prod_{e \in R_e} y_{e,e'}))$$

Now, we can write $x_{e,e'}$ and $y_{e,e'}$ in terms of parameters h_Δ and g_Δ. We assume that activation of e' from e at time t follows a geometric distribution of the time passed since e was formed. Hence, $x_{e,e'} = h_\Delta g_\Delta (1 - g_\Delta)^{t'_e - t_e}$. Now, a link e fails to activate e' if the user fails to observe the formation of e, or observes the formation of the link but fails to act within the decay interval. Hence, $y_{e,e'} = 1 - h_\Delta + h_\Delta g_\Delta (1 - g_\Delta)^{t'_e - t_e + 1}$. Finally, the log likelihood of the data can be written as

$$L(h_{\Delta_1}, h_{\Delta_2}, h_{\Delta_3}, h_{\Delta_4}, g_{\Delta_1}, g_{\Delta_2}, g_{\Delta_3}, g_{\Delta_4}) = \sum_e \log(\prod_{e \in S_e} (x_{e,e'} + y_{e,e'}) - \prod_{e \in S_e} y_{e,e'}) + \sum_e \sum_{e \in R_e} \log y_{e,e'} \quad (1)$$

Solving for the parameters now simply requires maximising the log-likelihood of the data, which is done by the EM algorithm. The details are described in Algorithm 1.

3.2.3 Augmenting the User-User Interaction Graph

In the previous section, the user-user pseudo graph is generated deterministically through the board-follower data. However, we also seek to consider cascades of user influence not explicitly captured in these follower relations. For example, a user may see a pin on another user's board and pin it on their own board, but neither user follows the other. Thus, we wish to augment the information obtained from the board-follower network by attempting to infer these user-user cascades. We postulate there exists an underlying diffusion influence network among users, and we can view the pinning of a certain pin as a diffusion event in a cascade with respect to that particular pin.

Thus, the diffusion of multiple pins throughout the influence network gives us a set of cascades over the virtual user-user graph, and we attempt to employ the method posited by Gomez-Rodriguez et al. [7] to (1) enumerate possible edges of influence in the aforementioned virtual graph in the user node partition and (2) perform a maximum likelihood estimation of the infection probability (influence strength) along these directed edges. However, note that while edges have infection probabilities associated with them, the actual infections are assumed to be binary (occurring or not). The goal is to diversify the type of triads found in the previous link diffusion analysis, capture more realistic user dependencies, and in doing so, generate better recommendations.

Algorithm 1 EM Algorithm for Link Diffusion

Input: Dynamic Graph $G = (V, E, t)$
Output: Parameters $\{h_\Delta, g_\Delta\}$

LEARN(G)
Randomly initialise h and g
while not converged **do**
  for $p \in$ Pins **do**
    U_1 = $\{u : u$ has pinned $p\}$
    for $u \in U_1$ **do**
      Initialise $P_{XY}[u] = 1$, $P_Y[u] = 1$
      0-Initialise arrays h_a, h_b, g_a, g_b, g_d, counts
      U_2 = set of users in $u - p$ network.
      for $v \in U_2$ **do**
        $i = \Delta$ type based on the defined keys
        $X[u, v] = h[i].g[i].(1 - g[i])^{(t_u - t_v)}$
        $Y[u, v] = 1 - h[i] + h[i].(1 - g[i])^{(t_u - t_v + 1)}$
        $P_{XY}[u] = P_{XY}[u](X[u, v] + Y[u, v])$
        $P_Y[u] = P_Y[u].Y[u, v]$
        counts[i] = counts[i] + 1
      end for
      for $v \in U_2$ **do**
        $i = \Delta$ type based on the defined keys
        $b = \frac{h[i]*(1-g[i])^{(t_u - t_v + 1)}}{h[i](1-g[i])^{(t_u - t_v + 1)} + 1 - h[i]}$
        $h_b[i] = h_b[i] + b$
        $g_b[i] = g_b[i] + b(t_u - t_v + 1)$
        if $t_u - t_v > \delta$ **then**
          continue
        end if
        $a = \frac{X[u,v].P_{XY}[u]}{(X[u,v]+Y[u,v])(P_{XY}[u]-P_Y[u])}$
        $g_a[i] = g_a[i] + a$
        $h_d[i] = h_d[i] + a(1 - b)$
        $g_d[i] = g_d[i] + a(1 - b)(t_u - t_v + 1)$
      end for
    end for
  end for
  for $i \in [1, \Delta_{count}]$ **do**
    $h[i] = \frac{h_d[i]+h_b[i]}{\text{counts[i]}}$
    $g[i] = \frac{g_a[i]+g_b[i]}{g_d[i]}$
  end for
end while
Return h_Δ, g_Δ

We use as our observations (v_i, t_i), an indicator event of the new activation (infection) of user v_i such that v_i has pinned a certain item at some time t_i. Formally, we observe a set C of c cascades on n nodes, with each cascade consisting of a n-length vector of times at which nodes were activated by a particular pin. The pins which do not appear in a certain cascade are ignored, i.e. have activation times of infinity. For the purposes of our analysis, we translate each absolute time of a pinning to a discrete timestep $t_i \in [1...k]$, where k is at most the total number of different times a pinning occurred in the cascade. We consider

all of the cascades for their full discrete time length. Each cascade is assumed to be independent.

We define $f(t_i|t_j;\alpha_{j,i})$ as the likelihood of a user j influencing user i with influence strength $\alpha_{j,i}$. We require $t_j < t_i$ and use a monotonic exponential diffusion model where

$$f(t_i|t_j;\alpha_{j,i}) = \begin{cases} \alpha_{j,i}e^{-\alpha_{j,i}(t_i-t_j)}, & \text{if } t_j < t_i \\ 0, & \text{otherwise} \end{cases}$$

The probability S that user i survives, or was *not* infected by user j at t_i, is given by

$$S(t_i|t_j;\alpha_{j,i}) = 1 - F(t_i|t_j;\alpha_{j,i})$$

where F is computed from the edge influence likelihoods as mentioned previously. We also define the instantaneous infection rate or the *hazard function* as

$$H(t_i|t_j;\alpha_{j,i}) = \frac{f(t_i|t_j;\alpha_{j,i})}{S(t_i|t_j;\alpha_{j,i})}$$

Thus, in our case, the log survival function is simply $-\alpha_{j,i}(t_i - t_j)$ and the harzard function is $\alpha_{j,i}$. Using the previous definition, we compute the likelihood of a cascade by computing the conditional likelihood of the time at which a user is influenced in the cascade (i.e. pinning to their board) given the rest of the cascade. Also considering the users that were not influenced in a particular cascade, we get that the likelihood of a single cascade is

$$f(t;A) = \prod_{t_i \le T} \prod_{t_m > T} S(T|t_i;\alpha_{i,m}) \times \prod_{k:t_k<t_i} S(t_i|t_k;\alpha_{k,i}) \sum_{j:t_j<t_i} H(t_i|t_j;\alpha_{j,i}) \quad (2)$$

where $A \coloneqq \{\alpha_{j,i} | i, j = 1...n, i \neq j\}$ for n distinct users influenced by at least one cascade in the cascade set. The likelihood over all c cascades is given as

$$\prod_{t^c \in C} f(t^c;A)$$

Thus, our overall problem reduces to finding A that maximizes the likelihood of the observed c cascades, or equivalently minimizes the negative of the log-likelihood. Formally, we seek to

$$\begin{aligned} \underset{A}{\text{minimize}} \quad & -\sum_{c \in C} \log f(t^c;A) \\ \text{subject to} \quad & \alpha_{j,i} \ge 0, i, j = 1, \ldots, n, i \neq j \end{aligned}$$

Note that the formulated problem is convex in A, and we can now solve for A.

4. Experiments

4.1. Dataset

The dataset, obtained from Pinterest, consists of a subset of U.S. users who have food boards. The raw dataset consists of around 10 million users, 12 million boards and 20 million distinct pins. There are three distinct bipartite graphs in the whole dataset: pin-board, board-board creator and board-board followers.

We consider two versions of the entire graph set for our analysis. For the first version, we consider only "dense" graphs with pins occurring on more than 1000 boards and boards which contain 100 or more such pins. The filtering leaves 148,000 pins and 1.04 million boards. Next, we eliminate all users who have not created any board, leaving around 1 million users. We call this set of graphs G_1 in further analysis.

The second version comprises larger and sparser graphs. We repeat the filtering, but with a lower threshold. Specifically, we generate a sparser version of the pin-board graph by considering those pins which occur on more than 4 boards, and boards which contain 3 or more such pins. This filtering leaves 6.24 million boards and 8.39 million pins. Elimination of users who have not created any boards results in around 8.5 million users. We call this set of graphs G_2 in further analysis.

From the filtered dataset, we generate an artificial bipartite pin-user graph, where the correspondence is generated through the creator of the board on which the pin resides. In case there are multiple boards created by the user which contain the pin, we pick the earliest link formation time. We also create a follower-followee graph by making all users who follow a board that some other user created as the followers of this user. This is the only non-bipartite graph that is considered in the whole analysis. This pseudo-graph was discussed in more detail in the section where we described the methodology behind recommending pins to users.

Table 1 summarises the number of users, boards and pins associated with each graph kind and the corresponding edge counts. Notice that even though the number of pins in G_2 is almost 70 times the number of pins in G_1, the number of edges between the heterogeneous categories of nodes increases only by 2 or 3 times. This alludes to the extreme skew in the pin distribution on the graphs, divides the analysis cleanly into two categories for the sparse and dense graphs, and allows us to make inferences about the scalability and robustness of our algorithms.

Graph type	Node Types	Edge count
G_1	Pin & User	178 million
G_1	User & User	15 million
G_1	Pin & Boards	188 million
G_2	Pin & User	242 million
G_2	User & User	47 million
G_2	Pin & Boards	547 million

Table 1. Graph types and associated edge counts.

4.2. Recommending Pins to Boards

We run PageRank for each board, obtain the personalised PageRank scores for all pins in the dataset for the board, and consider the top pins which are not already present on the board as candidate pins for recommendation. We consider both the unbiased version of PageRank, where each link is

weighted equally, and our version of biased PageRank consisting of a single dimensional feature vector for weighting the links. We randomly eliminate 20% of the existing links between the pins and boards and allocate 10% of the links to validation set and the other 10% to the test set. For each set of positive links, we also define "negative" links consisting of all pin-board pairs which do not exist in the database and distribute it to validation and test sets in the same proportion. Hence, the ratio of positive to negative links remains the same in all three sets.

Since this recommendation problem involves ranking pins, we require a test metric taking this into account. The metric used for testing the health of the model is *degree of agreement (DOA)* first used in [3]. It is defined as follows: for every board, suppose P_k is the set of pins which are present on the board in the test set (or the validation set) and NP_k is the set of pins obtained from the "negative" links emanating from the board earmarked for testing (or validation). We define the DOA for each board as:

$$DOA_k = \frac{\sum_{p_1 \in P_k} \sum_{p_2 \in NP_k} \mathbf{I}(s_{p_1} > s_{p_2})}{|P_K||NP_k|}$$

where s is the score of the pin in the stationary distribution computed using personalised PageRank and $\mathbf{I}$ is the indicator function. The greater the value of DOA, the better the model, since we would intuitively expect pins which are present on the board but hidden during the test to have higher PageRank scores than those which are never present on the board. A random assignment of scores would result in a DOA value of 0.5. Having defined this metric, we compute the weighted average of this quantity over all boards as:

$$\text{Average } DOA = \frac{\sum_{k \in B} \sum_{p_1 \in P_k} \sum_{p_2 \in NP_k} \mathbf{I}(s_{p_1} > s_{p_2})}{\sum_{k \in B} |P_K||NP_k|}$$

To obtain the best values of the hyperparameters in the algorithm, weighted random walk with restarts is done on the training data, and the average DOA is computed on the validation set. During training, the entries of the transition matrices which correspond to the validation set pairs and test set pairs are set to zero. With the best set of hyperparameters, the algorithm is again run with the training and validation set to obtain the personalised PageRank scores for each board. During this run, the entries for the validation set which were set to zero during training are restored to actual values computed using the single dimensional feature vector.

4.3. Recommending Pins to Users

Unlike the board-to-pin recommendation problem where we intended to rank the pins for each board, the link diffusion approach for recommending pins to users can be posed as a classification problem, where a link between a user and a pin is either activated by virtue of the follower-followee network of the user during a certain timespan or not activated at all. Hence, we evaluate this problem purely as a classification problem where we use the probability of triad formation to predict if a pin and a user will form an edge. We again divide the dataset into train, test and validation sets in the same ratio as in the previous sub-problem after randomly eliminating 20% of the links and creating and splitting the repository of "negative" links. Since there is a high imbalance between the positive examples manifested by activated links and the negative examples manifested by inactive/non-existent links, we use area under the curve (AUC) as the metric for testing the robustness of the model.

4.4. Recommending Pins to Users With Augmented User-User Pseudograph

The network inference convex problem as stated previously can be solved in a distributed manner as n subproblems, one for each user. Thus, to compute the solution entirely locally and maintain the problem tractability, we attempt to solve a small-scale local version of the problem. Specifically, we select a set of 5 high-degree pins and extract the discrete timestamps and associated users who pinned it to their board as the set of cascades. We then solve the convex problem formulated previously using the NETRATE implementation by [7] with the CVX package for MATLAB. This gives us the computed influence strengths between users affected in the cascade set.

We then threshold on the influence strengths and augment the corresponding edges in the original deterministically-generated user-user graph for the link diffusion study by 1 if $\alpha_{j,i} \geq 0.5$. We use this augmented graph in a localized variant of the link diffusion experiment posed in 4.3 whereby we learn the discovery and diffusion parameters on a reduced set of hundred pins (and explore a very dense portion of G_1, which we denote G_1'); and we observe link formation with respect to another disjoint set of hundred pins, which include the original set of 5 high-degree pins to observe the effect of the newly inferred/augmented user-user edges.

5. Results and Discussion

In this section, we report the results for the two recommendation tasks and discuss some salient inferences from these results. These results are reported on a single test set as opposed to bootstrapping the test set and obtaining significance values, as it is extremely expensive to run multiple trials of the same experiment on the bootstrapped samples, since at any given trial we deal with billions of pairs of heterogeneous entities.

5.1. Recommendations for Boards

A comparison of average DOA for different algorithms on the test set for G_1 is shown in Table 2 and for G_2 is shown in Table 3. Observe that the values for the sparse graph are much higher than those for the dense graph, since even though the ratio of the positive to negative test pins remains the same, the relative ranks for the positive test links is better. This makes sense as the number of non-board pins

are in the order of millions, while the rankings for the positive pins are in the order of low hundreds and thousands. Our version of the random walk with restarts outperforms the unbiased PageRank and Adamic-Adar methods [1] for ranking entities in graph by a slight margin. The margin is not large enough to be confident about the results for both G_1 and G_2, as the performance could have been better just by chance.

To further validate the results, we need another metric which is a function of the order assigned to the pins for each board. For this, we formulate the problem as a link prediction problem where all pins with PageRank scores greater than a certain threshold are considered as actual links. The threshold is kept the same for all boards when considering the pins to include as recommendations. Otherwise, boards with large number of pins, which assign small PageRank scores, would have smaller false positive rates and could bias the analysis. This allows us to plot the Receiver Operating Characteristic (ROC) curve, which is created by plotting the true positive rate against the false positive rate as the discrimination threshold is varied.

We then compute the area under the curve (AUC) for the ROC curve, which is a good approximation of the rankings assigned to the pins. The AUC values for the graphs are also shown in Tables 2 and 3 respectively for G_1 and G_2. The AUC values are highly correlated with the DOA values for the PageRank algorithms. This indicates that for every board, relevant pins are much more likely to be recommended than non-relevant pins. Since the AUC scores for the PageRank algorithms are also better than the baseline scores, we can conclude that the PageRank-based approach for generating board recommendations is a robust method. Hence, a biased PageRank-based approach for recommending relevant pins to boards performs well for dense graphs where each board has a large number of pins; and also scales well for larger graphs with most boards having a very small number of pins.

Algorithm	**Average DOA**	**AUC**
Random assignment	0.53	0.52
Adamic-Adar	0.76	0.73
Unweighted pagerank	0.81	0.84
Biased pagerank	0.82	0.85

Table 2. Board recommendation results for G_1

Algorithm	**Average DOA**	**AUC**
Random assignment	0.49	0.5
Adamic-Adar	0.82	0.85
Unweighted pagerank	0.94	0.98
Biased pagerank	0.95	0.98

Table 3. Board recommendation results for G_2

5.2. Recommendations for Users

The best performance on the validation set is obtained by keeping the time decay parameter as 15 days. Having obtained the best time decay parameter δ, we perform classification by obtaining the probability of link formation between a user u and a pin p at some time t using the follower-followee network of the user u. The time t can be thought of as one immediate day after the last link in the observed network has been formed. Note that the probability fades with time, and hence this probability is the highest probability of link diffusion.

The diffusion and the discovery parameters learned from training for the two sets of graphs are shown in Table 4 and Table 5, respectively. The probabilities of diffusion and discovery are much lower for the sparser graph that the denser graph. G_1 consists of pins with high degrees and hence these pins are popular. The event of a user pinning such a pin can thus increase the propensity of other users in their "follower-followee" network to also pin it, hence explaining the higher probability of both discovery and diffusion. Also noticeable is the fact that triads 1 and 2, consisting of pseduo-edges between users who follow each other, are more likely to be activated for both graphs. The least formed triad consists of the *followee* kind. One possible explanation for this phenomenon is that users respond more positively to the actions of users who they follow than those who follow them. However, the effect of the interaction of their followers with pins is not negligible. This points at a very interesting phenomenon in the Pinterest graph that a user's actions are also affected non-trivially by the action of users who follow their boards, even though there is no explicitly defined follower-followee relationship.

Triad closed by B; diffused from A	**h**	**g**
B $\leftrightarrow$ A (ratio $<= 0.5$)	0.39	0.31
B $\leftrightarrow$ A (ratio > 0.5)	0.43	0.33
B $\rightarrow$ A	0.28	0.21
B $\leftarrow$ A	0.21	0.16

Table 4. Discovery and diffusion parameters for G_1

Triad closed by B; diffused from A	**h**	**g**
B $\leftrightarrow$ A (ratio $<= 0.5$)	0.13	0.11
B $\leftrightarrow$ A (ratio > 0.5)	0.15	0.15
B $\rightarrow$ A	0.19	0.17
B $\leftarrow$ A	0.11	0.09

Table 5. Discovery and diffusion parameters for G_2

A comparison between the link diffusion and baseline methods on the test set is shown in Table 6 for the smaller, denser graph and in Table 8 for the larger, sparser graph. For pure classification methods, which include SVM and logistic regssion, we use the count of triads of each kind and the average simrank [4] for each kind as the features, yielding an 8-dimensional feature vector for each (user, pin) pair. We use the same feature vector for the negative links, consisting of (u, p) where user u does not pin p.

For G_1, a comparison against logistic regression shows a gain of around 10% in AUC and a gain of around 5% in AUC against SVM. We still obtain one false positive for every three true positives on average, but given that the number of negative links is around 100 times the number of positive links, this is a fairly good performance.

However, on G_2, link diffusion has almost similar performance to the baseline methods, and it performs significantly worse than SVM. This seems to indicate that link diffusion does not scale well for large and sparse graphs, and hence might not be the best method to use for recommending pins to users for sparse graphs. A possible explanation is that the discovery and diffusion parameters for sparse graphs are extremely small because of the rarity of triad activation, and hence the phenomenon might not justifiably explain the interaction of users with pins given that other users in their follower-followee network have also interacted with those pins. Since the difference in performance between dense graphs and sparse graphs is so acute, it can be concluded that this method can be used in scenarios where the pin distribution is not very skewed. Hence, it can be used for recommending pins to users who like niche content as opposed to recommending pins in general. For example, using this method for recommending Indian food recipes would be much more fruitful than recommending general recipe pins.

Algorithm	**AUC**
Random assignment	0.48
Logistic Regression	0.63
SVM	0.69
Link diffusion	0.72

Table 6. User recommendation results for G_1

Algorithm	**AUC**
Link diffusion (small-scale)	0.81
Link diffusion (augmented)	0.84

Table 7. User recommendation results for G_1'

Algorithm	**AUC**
Random assignment	0.48
Logistic Regression	0.53
SVM	0.62
Link diffusion	0.58

Table 8. User recommendation results for G_2

5.3. Recommendations for Users with Augmented User-User Pseudograph

Even considering a set of only five pins, the associated cascades still affect $n > 2000$ users. Doubling the number of pins increases n by almost an order of magnitude. Thus, it is necessary to efficiently solve the network inference problem in a distributed manner to scale it for increasing numbers of pins and users. Upon augmenting the original user-user follower graph, we notice that of the 2417 thresholded influence strengths to augment, 2415 are new edges not captured by the deterministic follower relationships. However, even in our restricted link diffusion problem, the number of edge augmentations is very small relative to the overall number of links, reducing the potential impact.

Even so, we observe an increase in AUC relative to the baseline for the restricted link diffusion problem on reduced graph G_1' (Table 7). Note that this baseline will naturally be higher than that for the unrestricted case, as our number of false positives will rise for the latter and result in a lower AUC. The network inference augmentation is a promising method for improving the performance of link diffusion as previously stated. The inferred user-user links are not necessarily biased by the specific "topic" of the board on which the associated users may interact; rather, the edge strengths are optimized over a general set of pins which may be unrelated, capturing more general patterns of influence.

6. Conclusion and Future Work

We have built a comprehensive recommendation engine for pins to boards and users on Pinterest. We use two distinct approaches for recommendation, one involving random walk with restarts, and the other involving link diffusion. We also propose deterministic, probabilistic, and hybridized methods to generate links between entities of the same kind for a bipartite heterogeneous graph. We test the robustness of the algorithms on large and sparse graphs; and small and dense graphs. The PageRank-based algorithm works extremely well for both sparse and dense graphs. The link diffusion approach works well for dense graphs, but does not seem to scale for the sparser, larger graphs. However, when using network inference to augment the user-user graph in the link diffusion study, we see an increase in performance. This gives evidence that improved methods for predicting underlying user-user influence can result in corresponding improvements in pin-to-user recommendations.

Our simple approaches easily generalize to a variety of bipartite heterogeneous graphs commonly seen in recommendation systems. An interesting direction for future work is to bring more complexity to the analysis of the entities in the graph. Specifically, we can design feature vectors for the PageRank-based algorithm that incorporate more node and edge properties. Another useful direction might be to understand the interaction between homogeneous entities in these graphs by using more advanced infection models (e.g. the Rayleigh model) that model "fads" of pins more accurately than our simple exponential model. We wish to apply these more sophisticated techniques to improve the performance of our existing recommendation methods.

References

[1] L. A. Adamic and E. Adar. Friends and neighbors on the web. *Social networks*, 25(3):211–230, 2003.

[2] L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social networks. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 635–644. ACM, 2011.

[3] M. Gori, A. Pucci, V. Roma, and I. Siena. Itemrank: A random-walk based scoring algorithm for recommender engines. In *IJCAI*, volume 7, pages 2766–2771, 2007.

[4] 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, KDD '02, pages 538–543, New York, NY, USA, 2002. ACM.

[5] D. Kempe, J. Kleinberg, and É. Tardos. Influential nodes in a diffusion model for social networks. In *International Colloquium on Automata, Languages, and Programming*, pages 1127–1138. Springer, 2005.

[6] J. M. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan, and A. S. Tomkins. The web as a graph: measurements, models, and methods. In *International Computing and Combinatorics Conference*, pages 1–17. Springer, 1999.

[7] B. S. M. Gomez-Rodriguez, D. Balduzzi. Uncovering the temporal dynamics of diffusion networks. In *The 28th International Conference on Machine Learning (ICML)*, 2011.

[8] J. Zhang, Z. Fang, W. Chen, and J. Tang. Diffusion of following links in microblogging networks. *IEEE Transactions on Knowledge and Data Engineering*, 27(8):2093–2106, 2015.