Finding Love in a Hopeless Place
Applying Graphical Analysis to Advance the Frontier of Matching Algorithms

Seth Hildick-Smith
MS Stanford 2017, Computer Science
sethjhs@stanford.edu

Jonathan NeCamp
MS Stanford 2017, Computer Science
jnecamp@stanford.edu

Ivaylo Bahtchevanov
MS Stanford 2016, Computer Science
ivaylogb@stanford.edu

I. Introduction

According to [1], 5% of Americans who are in a marriage or committed relationship say they met their significant other online. However, The Pew Research Center also finds that a third of those using online dating platforms have never gone on a date with someone they met online. There is fundamental value in relationships formed online but a significant portion of users never realize that value as they never match with someone they would like to date. Link prediction within a dating network could be used effectively to improve match rates by suggesting likely matches. Our project explores both content-based and structural graph algorithms to analyze a speed dating network, with the goal of understanding how compatible two people will be and thus improve match rates.

II. Related Work

A. "Learning Spectral Graph Transformations for Link Prediction"

In [5], Kunegis and Lommatzsch identify several methods of link prediction and study their generalized applications. The authors focus on several link prediction algorithms as functions of the Laplacian and adjacency matrices. They examine path counting algorithms in which the probability of an edge arising between two nodes is a function of the number and length of paths between them. Additionally, and most notably for this paper, they examine a number of rank reduction algorithms on the the adjacency and Lapacian matrices. In evaluating their algorithms, they posit that the optimal solution is the one which minimizes the Frobenius norm of the predicted adjacency matrix and the true adjacency matrix. More formally, they search for a suitable link prediction algorithm such that;

$$\min_{F \in \boldsymbol{S}} ||F(A) - B||_F$$

Where $\boldsymbol{S}$ is the set of all functions over a graph such that it has a known adjacency matrix and is diagonalizable. With this evaluation technique, Kunegis and Lommatzsch are able to learn the appropriate link recommendation algorithm for several different applications.

B. "Social Network Analysis of an Online Dating Network"

In [2], Chen and Nayak apply social network analysis to provide insight into an online dating network. The authors collect data from a one-year period from a large online data service. They apply their results to provide better recommendation systems for online dating matches. The data captured all four aspects of the studied dating service: 1) Personal profiles for each user, 2) Ideal match profiles for each user, 3) user activities on the service (network), and 4) the measure of relationships with other users. The fourth aspect is what captures the success of a match.

The authors represented their data as a directed graph $G = \{V, E\}$ where nodes are users and edges are directed messages sent from one user to another. As 97% of the users studied sought out straight relationships, authors discarded homosexual users (nodes) from the data and studied the resulting bipartite graph. The authors approached their analysis under the assumption that the graph took on a Bow Tie Structure. By following this assumption, the paper finds that 60% of active network users are members of the SCC at the center of the bow, 5% of nodes just receive messages and 12.5% of nodes just send messages. The authors provide additional analysis through graph metrics on a selection of 1000 random nodes, they find that the diameter is 14, the average path length is 4.923 and the clustering coefficient in 0. In-degree (messages received) and out-degree (messages sent) both follow a power law distribution. More subjectively, the authors found that 90% of user's stated preferences do not match the users they reach out to. This aspect of the data is carefully studied, the paper lays out different characteristics and the rate at which stated preference was broken by the action taken by users. Occupation and age are the two characteristics most broken, diet and sex are those least broken.

C. "On the Uniform Generation of Random graphs With Prescribed Degree Sequences"

In [6], Milo, Kashtan, Newman, and Alon look at different strategies for creating random directed graphs and the different trade-offs of these strategies. The article carefully examines the switching algorithm and the matching algorithm performances, and provides a third algorithm, a "go with the winners" Monte Carlo method. The article only examines these algorithms in

the context of a simple toy network, a small ten in/ten-out edge graph with two distinct network topologies and no multiple edges. On a high level, the article explains how matching method suffers from non-uniform sampling, the switching method has no general bound on its mixing time, and the "go with the winners" method does not suffer from either drawback, but runs very slowly. This article is relevant to the Erdos-Renyi model of building a graph by drawing edges by flipping a coin. The reading takes our course intro to random graphs a step further by looking at the different algorithms that can accomplish building the graph.

D. "Collaborative and Structural Recommendation of Friends using Weblog-based Social Network Analysis"

In [4], the authors analyze the social network, *LiveJornal*, a web-blog based social network. Their main task was to use structural properties of the web-blog graph in order to make link recommendations between users, i.e. friend recommendations. The *LiveJournal* network is treated as a directed graph with (possibly asymmetric) edges from user to user. Their approach is one that uses the structural properties of the network as features to be fed into a classifier that predicts whether or not a given candidate node would make a good friend. More specifically, given a query node, a list of candidate nodes is determined by exhaustively searching within a fixed radius of the query node. These candidate nodes are featurized into 12 dimensional feature vectors containing features such as indegree of the candidate node (popularity), number of mutual nodes with a connected out edge (mutual friends), and backward distance from the candidate to query node. Using this featurization, different models were trained (such as a logistic regression model) and prediction accuracies were obtained using 10-fold cross-validation.

III. Dataset

We analyze a nearly 8500 observation speed dating dataset. The dataset comes from 21 speed dating sessions conducted for research [3] but carried out in a typical speed dating format.

The data comes in *csv* format with each row corresponding to a male-female pairing from the perspective of either the male or female. Concretely, if male i was paired with female j during a speed dating session then there are two rows in the *csv* for this encounter: one from the perspective of male i and one from the perspective of female j. For example, the first row, which we will call an observation, o_1, appears as follows in our dataset:

```
iid,id,gender,idg,condtn,wave,...pid,...
1,1,0,1,1,1,...,11,...
```

This gives us for each observation the iid, or the unique id of the observing participant, the pid, or the unique id of the partner, $match$, whether both participants would like to extend the relationship outside of their initial and ephemeral encounter. Additionally, an observation includes a wealth of other information such as thoughts on attractiveness, intelligence, sincerity, and finally demographic information of the observing participant. This latter information is important for when we perform SimRank to detect how similar two participants are in terms of their values.

Network Statistics	
# of Nodes	551
# of Edges	3515
# of Parallel Edges	1380
Network Diameter	7.0
Density	.012
Average Path Length	2.221
Average Degree	6.379
# Strongly Connected Components	80
# Weakly Connected Components	22

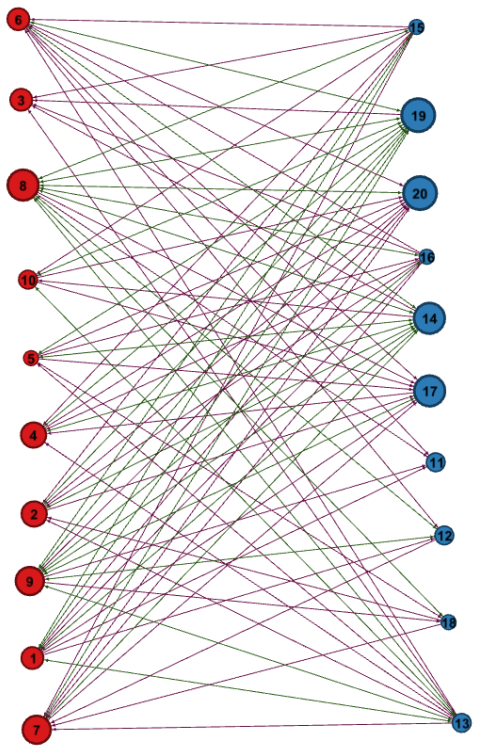

Fig. 1: Graph visualization of speed dating session 1. Female nodes are red. Green edges represent a match, the directed purple edges represent the source node "liking" the destination node. Nodes are sized by the their number of matches.

There are two other important things to notice about the data. First, data was accumulated over the course of 21 speed dating sessions which naturally separates our data into at least 21 communities as it is impossible for participants to match with participants not participating in the same session. Second, this also naturally lowers the density of the network as the number of possible edges leading from one node is constrained to the number of other nodes attending the same speed dating session.

IV. Algorithms

We examine both content based and structural edge recommendation algorithms.

Many social network graphs are poorly captured by a traditional $G = V, E$ representation. Social networks such as the data set studied here often contain node meta data that is significant to link recommendation. We leverage node meta data through the content based similarity ranking algorithm.

Much of the information useful for link recommendation comes in the form of the graph structure. We apply a novel link recommendation algorithm based on spectral graph theory and an algorithm based on rank reduction of graph capturing matrices.

A. Content Based Similarity Ranking Algorithm

One approach we have taken for link analysis is using a bipartite graph to find similarities between people. The intuition behind our algorithm is that two people are "similar" if they have the same set of interests—this connection is represented by two nodes that represent people that are linked together by similar interests.

We have run several formulations of our "similarity concept" structures to gain a better understanding of what works. We first tried a series of basic concepts: one iteration for "primary goal for participating", one for "dating frequency" together with "going out frequency", one for "career", and finally one for general interests. We then looked at the different sets of people that were grouped together and found the intersection of all the sets.

Our similarity computations have been inspired by a variation of PageRank called Sim-Rank, which follows the notion that two items are similar if they are referenced by similar items. In our case, we try to use the link structure to derive similarity scores between people, so the "items" are people that are referencing the same set of interests.

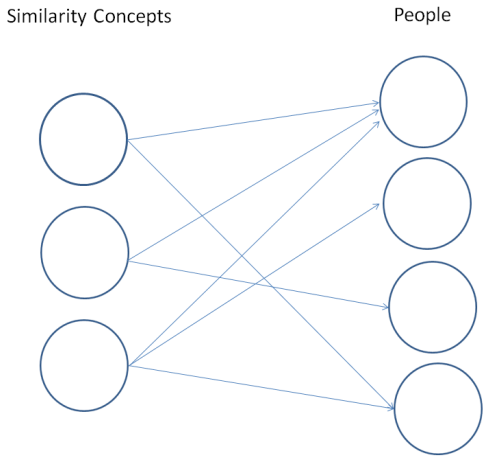

Fig. 2: Bipartite Graph for Similarity Ranking

1) Sim-Rank Theory: We leverage the fact that we have a directed graph, and we define, for a specific node n, $O(n)$ to be the out-directed neighbors of n and $I(n)$ to be the in-directed neighbors of n. We define $O_i(n)$ to be the destination vertex of the ith edge coming out of n, and we also define $I_i(n)$ to be the source vertex from which the ith edge points to n. We have the two sets of our bipartite graph marked as "C" for the similarity concepts and "P" for the people. Our similarity concepts are represented by the different interests and responses to the survey question. Concepts such as current career/ideal career, field of study, religious background, race, hobby preferences, and desired attributes in a partner. Similarity concepts also included general attitudes: expectation of being matched, expectation of being happy with the event, etc. We have constants K_1 and K_2 that we use as "decay constants"- these are typically between 0 and 1, and for our purposes are set to equal 0.8.

Then, our strategy involves the initial equalities: $S_C(V,V) = 1$ for all V in C, and $S_P(v,v) = 1$ for all v in P. The intuition here is if we compare something to itself, its similarity measure should equal 1. On the other hand, we also have: $S_C(X,Y) = 0$ for all X,Y in C such that $X \neq Y$, and $S_P(x,y) = 0$ for all x,y in P such that $x \neq y$.

Here, we have the two symmetric equations we use for computation:

We have the similarity of two concepts X,Y in group C of the graph, and define their similarity as S_C as:

$$S_C(X,Y) = \frac{K_1}{|O(X)||O(Y)|} \sum_{i=1}^{|O(X)|} \sum_{j=1}^{|O(Y)|} S_P(O_i(X), O_j(Y))$$

For 2 people x,y in group P of the graph, we define the similarity S_B as:

$$S_P(x,y) = \frac{K_2}{|I(X)||I(Y)|} \sum_{i=1}^{|I(x)|} \sum_{j=1}^{|I(y)|} S_C(I_i(x), I_j(y))$$

We need one score for each pair of people, so the above symmetric equations each translate to the following correspondence: S_C breaks down into $\binom{|C|}{2}$ equations and S_P breaks down into $\binom{|P|}{2}$ equations to consider all pairs. Now, our algorithm repeatedly calls the above equations for S_C and S_P until the two values converge, using the similarity values computed in iteration i-1 for the following iteration i. For our purposes, we found that using an approximation of 100 iterations to be sufficient. We use these final scores to make predictions of a match based on how similar two people are.

B. Structural Link Prediction Algorithms

1) Spectral Graph Analysis: Our initial approach to link recommendation examines the value of spectral graph analysis when applied to our problem. To identify communities of people who are likely to match one another, we can identify weakly connected components across the bipartite graph and

recommend missing connections.
Let the Graph Laplacian Matrix

$$L = D - A$$

Where D is a diagonal matrix such that $D_{i,i}$ = degree of the i^{th} node and A is the adjacency matrix. We observe that the combinatorial Lapacian Matrix is symmetric and positive semi-definite. Furthermore, the eigenvalues of the Laplacian represent a spectrum of positive values which map to connectedness of groupings of nodes. The number of 0 eigenvalues is the number of connected components of the graph, each subsequent eigenvalue has a corresponding eigenvector which maps to a labeling of nodes such that closely connected nodes have similar labellings (if the eigenvalue is small) or a labeling of nodes such that closely connected nodes have dissimilar labellings (if the eigenvector is large). Via spectral graph analysis we identify the connected components of our graph:

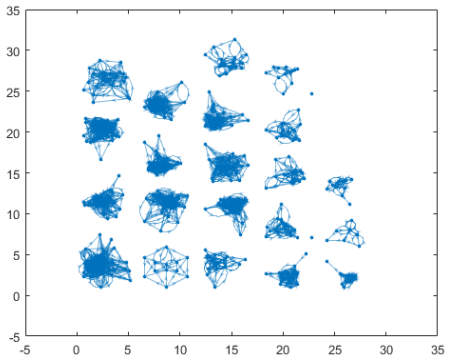

Fig. 3: A visualization of the connected components of the dataset identified by spectral graph analysis

Our first recommendation algorithm centers around recommending links to complete connected components. The motivating intuition is that members of a community in this graph are members of a group of shared interests or shared interest in each other. We therefore suggest missing links within connected components.

Although we believe this recommendation engine would perform well for larger data sets, especially data sets corresponding to online dating where people are not physically interacting, we found that the initial recommendation engine described above performs poorly for our data set. This is due to the fact that ultimately the connected components of our graph correspond to the "rounds" of speed dating conducted by [3]. However, we can improve the utility of spectral graph analysis for this application.

We examine the "spectrum" of connectivity captured by the eigenvalues of the Laplacian matrix (see figure 4). We can use the fact that elements of the eigenvectors corresponding to the small eigenvalues of the Laplacian are selections of values for nodes such that neighbors have similar values. With this fact we then propose Algorithm 1.

Algorithm 1 Select best potential match via Spectral Graph Analysis

Require: $G = [V, E]$ is the graph of dating connections, n is the index of a node $\in G$

function SPECTRAL-RECOMMENDATION(G, n)
  $A \leftarrow Adj(G)$
  $D \leftarrow Deg(G)$ ▷ $Deg(G)$: diagonal matrix of degrees
  $L \leftarrow D - A$
  $\lambda^1 \leftarrow$ First non-zero eigenvalue of L
  $EV^1 \leftarrow$ Eigenvector of L corresponding to λ^1
  $EV^0 \leftarrow$ Eigenvector of L for a 0 eigenvalue where $EV^0_n \neq 0$
  $m \leftarrow nil$
  $\delta \leftarrow FLOAT\_MAX$
  for $i \leftarrow 1$ to $Len(EV^1)$ **do**
    if $Abs(EV^1_n - EV^1_i) < \delta$ & $i \neq n$ **then**
      if $[n, i] \notin E$ & $EV^0_i = EV^0_n$ **then**
        $\delta \leftarrow Abs(V^1_n - V^1_i)$
        $m \leftarrow i$
      end if
    end if
  end for
  return m
end function

This algorithm returns the most closely connected node to n within its connected component (that n is not connected to) as measured by the eigenvalues of the Laplacian matrix. The motivation here is the same as for suggesting edges within connected components: that well connected components within the the dating graph have a high potential for matching. Note that this algorithm could match across sexuality preferences in partners, this is corrected by imposing an additional if statement. Furthermore, although this algorithm returns a recommended node, we may return a scoring over all nodes (and therefore all potential edges from n) by instead returning a vector S.

$$S_i = \begin{cases} FLOAT\_MAX & EV^0_i = EV^0_n \\ abs(V^1_n - V^1_i) & otherwise \end{cases}$$

Note that a lower score is associated with a higher probability of an edge existing.

C. Eigenvalue Decomposition and Rank Reduction

A second structural link prediction algorithm follows naturally from the one proposed above. First, note that the graph Laplacian matrix captures all the edge information present in the graph. Furthermore, for each edge $[i, j]$ present in the graph, the entry $L_{i,j}$ is decremented by one. The lower the value $L_{i,j}$ is, the more likely a randomly chosen edge will be

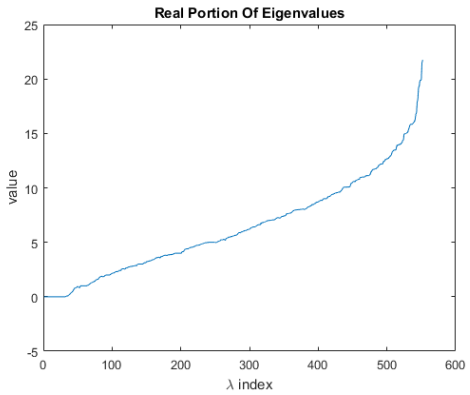

Fig. 4: The "spectrum" of eigenvalues of the graph Laplacian matrix. Note that the complex portion of all eigenvalues has been discarded for this visualization

$[i, j]$. Thus, the Laplacian matrix can be viewed as an "over-trained" model for edge prediction; it is a model which will predict existing edges perfectly but be unable to predict non-existent edges. We therefore, generalize the Laplacian matrix via eigenvalue decomposition rank reduction.

L can be represented:

$$L = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$$

Where $\mathbf{\Lambda}$ is the diagonal matrix of eigenvalues of A sorted in non-decreasing order, and $\mathbf{Q}$ is the matrix of correspondingly ordered eigenvectors.
From this decomposition we produce a representation of L reduced to rank r:

Let $\mathbf{\Lambda}_r$ be the $r \times r$ matrix of the r smallest eigenvalues of L.
Let $G = \{V, E\}$
Let $\mathbf{Q}_r$ be the $|V| \times r$ matrix of the r eigenvectors corresponding to the smallest eigenvalues.

$$L_r = \mathbf{Q}_r\mathbf{\Lambda}_r\mathbf{Q}_r^T$$

Note that L_r is the best reduced rank eigen-decomposition representation of L. We now have a scoring on potential edges:

$$L_{r\{i,j\}} < L_{r\{k,l\}} \implies P(\{i,j\}) > P(\{k,l\})$$

Note that a lower score is associated with a higher probability as the Laplacian matrix has negative values associated with node adjacency.

V. Results and Findings

A. Does Like Attract Like?

We examined the similarity scores for each possible pairing of participants, and found several key insights. First, as expected, the people who matched have, on average, a higher similarity score than those who have not matched. Although, initially an implementation bug led us to expect a much stronger correlation between node similarity and the probability of a match. A more interesting insight, however, is uncovered when we examine the matches of similar people.

SimRank Network Statistics

Class	Mean SimRank
Any Two People	0.084
Two People That Match	0.092
People With $\geq$ 1 Matches In Common	0.523
People With $\geq$ 2 Matches In Common	0.670
People With $\geq$ 3 Matches In Common	0.681

There is only a marginal increase in the sim-scores of the people that match, which implies that people did not necessarily match with people similar to them. More interestingly though- the highest similarity scores between people came not from people who matched, but from people who matched with the same individuals of the opposite sex (i.e. if two males are very similar, they will more likely match with the same females). The similarity scores jump substantially when we look at two individuals that had at least one match by over a factor of 5, a very telling indicator. The second substantial jump occurs when looking at people that have matched with 2 or more people in common. Intuitively, this makes sense: if two people have similar interests, then they will likely be interested in the same people for relationships (and the same people of the opposite dating pool would, in turn, like them) and vice-versa. These similarity scores can thus be used as a recommendation system in predicting links by looking at the matches of the people most similar to the people we examine.

B. Predicting "Likes"

The main task we explored was whether or not we could predict if one user would "like" another – through a network analysis lens, predict if a directed link will form from one node to another. As previously mentioned, we tackled this challenge with two different approaches: a collaborative and a content-based approach.

When evaluating link prediction algorithms, it is quite easy to accidentally produce deceptive results. For example, networks generally have densities less than .5 and in the case of our speed dating network, a density of .012. Thus, if we were naively to go through all possible edges and trivially predict that the edge will not exist, we could get an accuracy of .988. Obviously this predictor is a terrible one yet the results are ostensibly good. In order to evaluate our predictors in an unbiased and non-deceptive way we used the well-known metrics of *precision* and *recall*. More specifically, we followed the results in [7] and plotted the *precision-recall threshold curves* which was found in [7] to be preferred over the well-studied *ROC curve*. *Precision-recall threshold curves* can be generated if each predictor outputs a score for a potential edge. A higher score indicates that the predictor thinks the edge is more likely to form. By iteratively lowering the score threshold for predicting an edge, we generate many (*recall*, *precision*) data points that can be plotted to produce the desired curve.

Besides looking at *precision-recall threshold curves*, we also look at the K possible edges with the highest score according

to each predictor. The ensuing prediction rate, or the number of these K possible edges that are actual edges in the network divided by K, has also been shown to be a good way of evaluating link predictors [7].

An additional area for potential concern is the generation of a testing network for evaluating our predictors. Our initial thought was to randomly delete edges from the network and then test to see if our predictors could find where these edges existed. After reading [7], we realized this may be altering our network in unpredictable ways and may lead to deceptively good or bad results. Instead, as recommended in [7], we framed the problem as predicting links in an evolving network i.e. given the network at time t, predict the edges at $t+1$. Although the speed dating network appears static and therefore impossible to determine how it evolves, there was additional meta-data that broke each speed dating session into rounds. Concretely, if there were 10 women and 10 men, each man would meet with each woman over the course of 10 rounds. Thus to frame this as an evolving network, we could look at the network from rounds 1 to n and predict which edges will form in all rounds later than n.

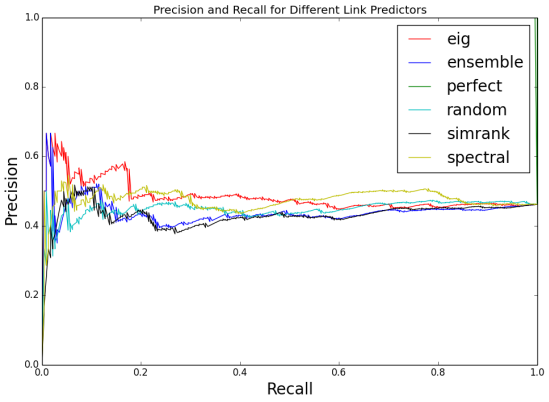

Fig. 5: The precision-recall curves of the eigenvalue decomposition predictor, the ensemble predictor that linearly combines the eigenvalue score and the SimRank score, the perfect predictor which has 100% recall and 100% precision, a random predictor that outputs a random score [0,1], the SimRank predictor, and the spectral analysis predictor. These predictors are tested on just the last round, but have knowledge about all rounds except for the last

From Fig. 5 and the Top K prediction rates table, we can see that the eigenvalue decomposition predictor performs the best. This is fairly promising as it performs quite better than a random predictor telling us that the graph structure does provide knowledge about how the graph will evolve over time. This ultimately shows the promise of using the structure of online dating networks in order to make new match recommendations to users.

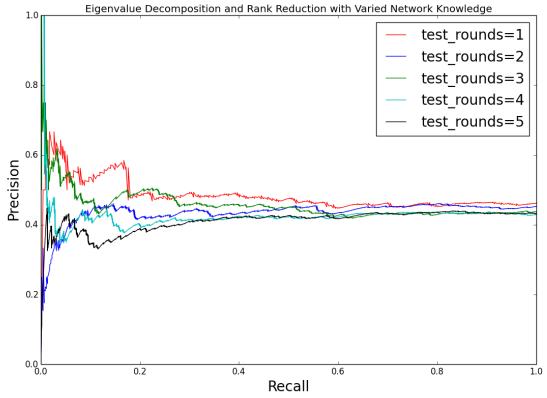

Fig. 6: Precision recall curves of Eigenvalue Decomposition predictor with varying amounts of network knowledge. "test_rounds=1" means we analyzed the resulting graph from every round except for the last and tested on the last. "test_rounds=5" means we analyzed the resulting graph from all rounds except the last 5 and test on these last 5 rounds.

Additionally, from Fig. 6 we can see how increasing amounts of knowledge about a network leads to better link predictions. The eigenvalue decomposition predictor performs noticeably better and better as it is given more information about later rounds.

From the Top K prediction rates and Fig. 5, we can see that the SimRank score doesn't help much with predicting future edges as it performs on par with a predictor that outputs a random score. This is fairly surprising because as we saw earlier, there appeared to be some correlation between matches and SimRank score. This also shows that for our network, the graph structure was more helpful in making link predictions than metadata about the nodes.

Top K Prediction Rates K=15

Algorithm	Prediction Rate
SimRank	.3333
Eigenvalue Decomp.	.6
Spectral	.4666
Ensemble	.4
Random	.3333

C. Finding an Ensemble Algorithm

We developed two approaches to link prediction with the goal of combining both into a better performing ensemble algorithm. To analyze how to combine the two approaches we plotted our test edges with the edge's collaborative scoring as its y value and its content-based scoring as its x value. To identify a function to successfully combine the scoring provided by our two algorithms we would chose a function of our two scorings which linearly separates the positive and negative examples in this plot (Fig. 7). Unfortunately there is no function which linearly separates the two example types.

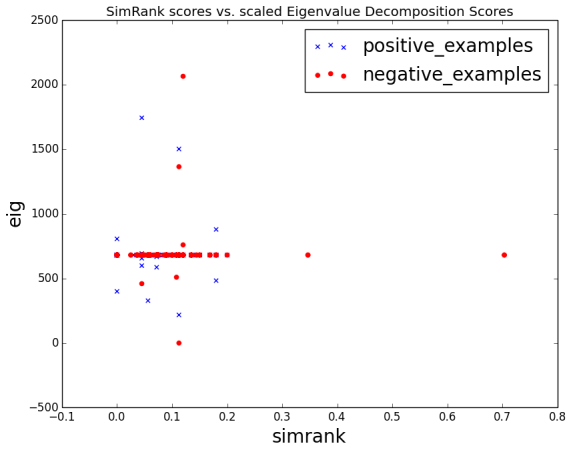

Fig. 7: Plot of collaborative scores of test edges against content-based scores of test edges. This plot demonstrates linear inseparability of positive from negative examples and indicates the fundamental challenge of designing an ensemble algorithm.

We attempted multiple linear combinations of the two link prediction scorings. We could not identify an ensemble algorithm that would outperform both raw score functions. The eigenvalue decomposition scoring constantly outperforms the ensemble algorithm when we take a nontrivial amount of information from the SimRank scoring.

VI. Conclusion

Our work leads us to two potentially fruitful areas of further study.

We uncovered a positive correlation between SimRank scores and the number of mutual matches two members of the same sex have. This result indicates a potential means of collaborative filtering to improve link prediction rates.

Furthermore, our collaborative based methods that analyze graph structure are more successful than our content based link recommendation techniques. Is this due to the fact that people ignore similar interests when choosing a partner or is it due to a weakness in our algorithms? Results on preference breaking found in [2] indicate that perhaps there is naturally a fairly weak correlation between potential partner similarity and their matching. If this is the case, a new approach to content based recommendation must be found.

References

[1] Center, The Pew Research. “Internet and American Life Project”. In: (2013).

[2] Chen, Lin and Nayak, Richi. “Social Network Analysis of an Online Dating Network”. In: *Proceedings of 5th International Conference on Communities & Technologies (C&T 2011),* ACM, 2011, pp. 41–49.

[3] Fisman Raymond, et al. “Gender differences in mate selection: Evidence from a speed dating experiment.” In: *The Quarterly Journal of Economics* (2006).

[4] Hsu William H., et al. “Collaborative and Structural Recommendation of Friends using Weblog-based Social Network Analysis.” In: *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs* (2006).

[5] Kunegis, Jerome and Lommatzsch, Andreas. “Learning Spectral Graph Transformations for Link Prediction”. In: *26th Annual International Conference on Machine Learning & Technologies (C&T 2011),* ACM, 2009, pp. 561–568.

[6] Milo, R. and Kashtan N., et al. “On the Uniform Generation of Random graphs With Prescribed Degree Sequences”. In: (2004).

[7] Yang Yang, et al. “Evaluating Link Prediction Methods”. In: (2015).