Analysis of Kiva Lending and Team Network Structure

Abraham Adam ABEADAM@STANFORD.EDU
SCPD student
San Francisco, CA, USA

Roger Davidson ROGERD@STANFORD.EDU
SCPD student
Grass Valley, CA, USA

Maha El Choubassi MELCHOUB@STANFORD.EDU
SCPD student
Sunnyvale, CA, USA

Abstract

This document describes the techniques used in and results of our project to apply social and information network analysis to the Kiva.org online microlending network. The Kiva network can be modeled as a set of bipartite graphs linking lenders to loans, lending teams to loans, and individual lenders to lending teams. Folding the lender-loan graph to create a graph of lenders linked by common loans results in a network that reflects the connections lenders have based on their own lending preferences. Prior studies have shown that lenders who are part of lending teams are 20% more productive in funding loans than those who are not. This project accessed data from Kiva.org and used personalized page rank vectors (PPRV) from a random walk with restarts algorithm and comparison with nearest neighbors in the folded lender-lender graph to recommend teams similar to a target user. The standard metrics of precision and recall were used to evaluate the methods against users who are already members of lending teams. Average recall rates were 35-51% depending on the technique used and the number of teams recommended to the user with the highest average recall being 51% for a weighted PPRV voting technique to recommend the top 20 teams. The highest average precision was 12% using a nearest neighbor voting technique to recommend the top five teams.

Keywords: microlending, Kiva, user similarity, online social network, random walks

1 Introduction

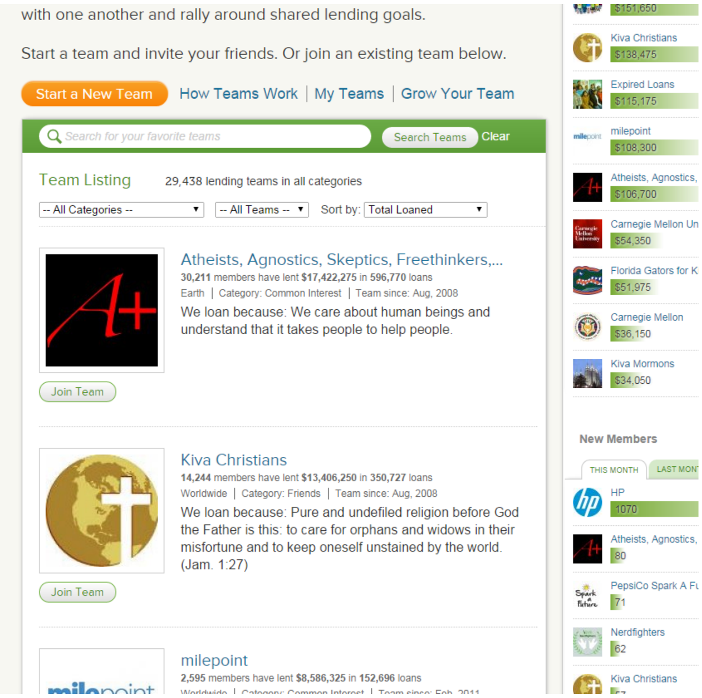
Kiva (www.kiva.org), was founded in October 2005 as a website dedicated to non-profit microlending to individuals and organizations around the world. Individuals and organizations may create a Kiva profile and either make loans of at least $25 (the lenders) at a time or obtain funding for a variety of small business, medical or other causes (the borrowers). Lenders may make as many loans as they wish, and when the loans are repaid then the lenders may re-loan those funds to other borrowers.

Nearly three years later in August 2008, Kiva launched the team-lending concept in which individual lenders may associate with one or more lending teams. The social dynamics of the team structure have been studied and reported (Chen, 2013 and Choo, 2014). In addition there is a field partner structure that helps to vet and manage financial payments and repayments of the borrowers. Each borrower profile contains their repayment rate for past loans and possible endorsements and risk ratings from field partners and associations with one or more "Social Performance Badges" that help describe associated field partners.

Since its founding, Kiva has supported 1.2 million lenders who collectively have loaned over $620M with an average 98.80% repayment rate[1]. The Kiva network presents a number of possible interesting graphs and project topics, but our project focuses on connecting individual lenders to lending teams within Kiva. In late 2012 Chen (2013) analyzed over 500,000 lenders and showed that lenders associated with teams were 20% more active in lending than lenders who were not associated with teams. To increase lending is a natural goal for improving kiva.org and better serving its charitable mission.

Kiva's database is huge and wide-ranging. It has 1,226,758 lenders and 29,438 lending teams. Kiva works with 279 Field Partners and 450 volunteers based in 79 countries across the globe. There are multiple resources for Kiva data. First, Kiva provides snapshots of their system, in addition to their API (see http://build.kiva.org/). The snapshots (each approximately 5 GB in size) provide information about each of the lenders, and each of the loans (including those that are fully funded and those that are in the funding process). The data is fully transparent in that all information that you see via the website can be accessed either in the snapshots or via their API.

The purpose of this project is to grow the lending rate per user and address the problems of matching potential teams for a lender and potentially the reverse: matching lenders for a team. If unaligned members were part of active teams this should increase lending and the resulting social good from microlending on Kiva. Unlike the current Kiva interface (see Figure 1) that lists the 29,438 existing teams in order of size, a lender might see the candidate teams that he/she most likely would join with our algorithm, or they could browse but also sort by relevance as determined by our algorithm. Similarly, by matching potential lenders to a team, our algorithm could suggest possible lenders with high probability of joining these teams.



with one another and rally around shared lending goals.

Start a team and invite your friends. Or join an existing team below.

Team Listing 29,438 lending teams in all categories

-- All Categories -- ▾ -- All Teams -- ▾ Sort by: Total Loaned ▾

Atheists, Agnostics, Skeptics, Freethinkers,...
30,211 members have lent **$17,422,275** in **596,770** loans
Earth | Category: Common Interest | Team since: Aug, 2008
We loan because: We care about human beings and understand that it takes people to help people.

Kiva Christians
14,244 members have lent **$13,406,250** in **350,727** loans
Worldwide | Category: Friends | Team since: Aug, 2008
We loan because: Pure and undefiled religion before God the Father is this: to care for orphans and widows in their misfortune and to keep oneself unstained by the world. (Jam. 1:27)

milepoint
2,595 members have lent **$8,586,325** in **152,696** loans
Worldwide | Category: Common Interest | Team since: Feb, 2011

	$151,650
Kiva Christians	$138,475
Expired Loans	$115,175
milepoint	$108,300
Atheists, Agnostics,	$106,700
Carnegie Mellon Un	$54,350
Florida Gators for K	$51,975
Carnegie Mellon	$36,150
Kiva Mormons	$34,050

New Members

HP	1070
Atheists, Agnostics,	80
PepsiCo Spark A Fu	71
Nerdfighters	62
Kiva Christians	

Figure 1. Current kiva.org teams interface, listing the 29,438 teams in order of size.

[1] These statistics are those reported as of October 10, 2014 from the Kiva website.

2 Related Work

As noted in the prior section, Chen (2013) analyzed over 500,000 lenders and showed that lenders associated with teams were 20% more active in lending than lenders who were not associated with teams. This result was the primary motivation for this project and builds upon the finding of Chen (2013) by seeking to make lending team recommendation directly to Kiva lenders.

Also Choo (2014) developed a technique to recommend loans to Kiva lenders using similarity measures and machine learning. Their project used loan similarity measures and machine learning algorithms to make recommendations, but it did not consider the underlying network structure of the Kiva lenders and loans. Data from the machine learning study of Kiva (Choo, 2014) is available as a set of MATLAB files (http://fodava.gatech.edu/processed-kiva-data).

Previous studies of communities in social networks have shown that a community structure characterizes these networks and can be discovered. Also, communities may be overlapping and layered (Palla, 2005), and this certainly would be the case with lenders on Kiva.org who may be part of not only multiple lending teams but also unlabeled communities based on geographic, cultural or other aspects. User similarity (Anderson, 2012) has been shown to affect user behavior for evaluations, and this should have some effect on a user's choice of team memberships. Also, group formation (Backstrom, 2006) and link recommendation (Backstrom, 2011) for the evolution of networks over time show how network structure can be used to estimate probable future links. User similarity has been modeled with machine learning on network attributes (Choo, 2014), on network structure (Palla, 2005), and combinations of both aspects (Backstrom, 2011).

For matching teams and lenders, "interest features" based on loans, teams, and lenders textual descriptions are critical, but as Backstrom emphasized (Backstrom, 2006), the network structural properties and activity-level features are quite crucial and decisive in teams' growth and evolution. A marriage of both features and network structure has also been achieved for link recommendation (Backstrom, 2011).

The proposed approach for this project was to encompass most of these factors and to investigate using a similarity measure (Anderson, 2012) to identify the highly matching pairs. For each lender in a bipartite lender-loan graph, we would construct a similarity vector based on the aggregate lending, and the same would be done for each team. An inner product measure could then be used to rank the best team matches for a given lender.

However, due to the lengthy process of data acquisition of the team data, and because it is likely that the most informative similarity measure would require machine learning techniques to classify the free-form text descriptions of loan and lender data, this approach was put aside due to time constraints. Instead, the project used a random walk with reset technique to make team recommendations. This technique was shown to be nearly as effective (Backstrom, 2011) as using both features (as edge attributes) and network structure. Additionally, one key type of information for Backstrom and Leskovec (Backstrom, 2011) was the availability of time-stamp information for link creation to train the optimization parameters, and unfortunately the information on when lenders join lending teams is not available from Kiva.

3 Model and Analysis

Our model and approach for analyzing the Kiva data is described below.

3.1 Data collection

One of the challenges with the Kiva dataset for this project was the availability of the membership information for the lending teams. Obviously to predict team membership the set of lenders with no team membership, as well as the membership lists for each team, are needed. However, neither of these data sets are available from the easily downloaded snapshots. Even for the team information, there is no data available on the website for the current membership list (which is considered private) nor more detailed information such as when a member joined.

Consequently the most challenging aspect of the data collection process and a significant part of the work for the project was accessing the team membership data. To do this, a web crawler was written in Java to repeatedly query the Kiva API for team membership information driven by data contained in 1392 team files. The Kiva API restricts queries to once every 2 seconds such that completion of a single team file requires approximately 34 minutes, and so accessing the complete team memberships for the Kiva lenders required approximately four weeks.

A graph of the network linking individual lenders to loans was created using the snapshot from July 31, 2014. The bipartite graph contains 547,651 lender nodes and 144,397 loan nodes with 3,228,344 edges between them. The density of edges is due to the nature of Kiva loans that are typically funded by multiple lenders.

3.2 Graph modeling

Consistent with the project proposal, team membership was an attribute of the lender nodes, and the bipartite lender-loan graph was folded into a lender graph with edges corresponding to loans for which both lenders have contributed funding. The following graphs were created in SNAP.py: 1) the bipartite lender-loan graph (as previously reported: 547,651 lenders, 144,397 loans and 3,228,344 edges) and 2) the folded lender graph (GLender) based on links from jointly funded loans which has 547,651 lenders and 78,690,804 edges. The average degree of the GLender graph is 287, which reflects both the average funded loans per lender and the high average number of lenders for each funded loan. The degree distribution of the GLender graph (see Figure 2.) shows a power law distribution with α = 2.79 and x_{min} = 2897.

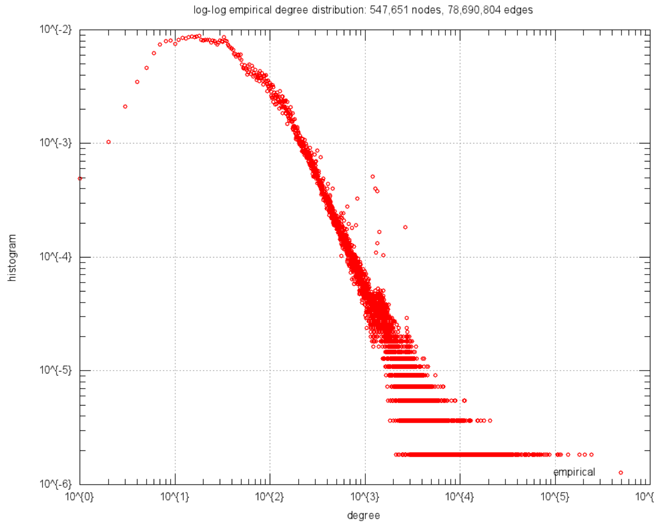


Figure 2. Degree distribution of the folded lender-lender graph

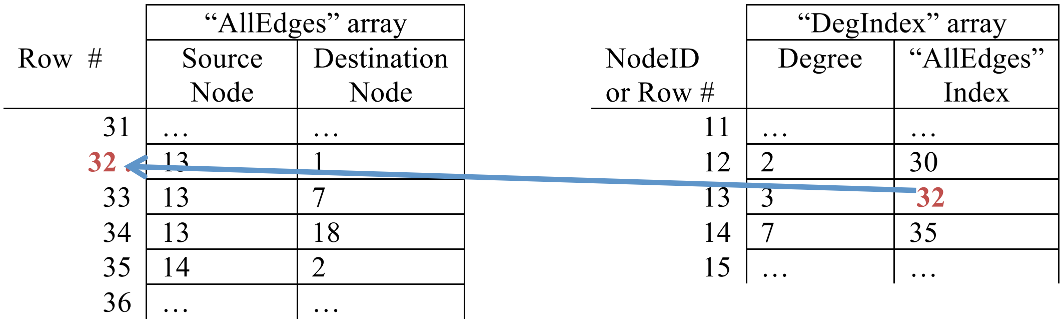
3.3 Recommendation algorithm

For a given lender for which a team recommendation is needed, a random walk with resets technique was used to provide a personalized page rank vector (PPRV) measure for the most similar 100 lenders. The proportions of teams represented by the 100 lenders provided a ranking of recommended teams with the highest proportion corresponding to the team with the highest recommendation. The team rankings from the PPRV were created in two ways for comparison: 1) an "unweighted" approach in which the team(s) represented by each similar lender was given a "vote" equal to that lender's PPRV value, and 2) a "weighted" approach in which a lenders' PPRV was scaled down by the number of teams to which that lender belonged. For example, if a similar lender had a PPRV value of 0.1 and belonged only to team A, then team A would receive a vote of 0.1 points in both of the approaches described above. However, if the same lender belonged to teams A and B, then both teams would receive votes of 0.1 in the "unweighted" approach and 0.05 in the "weighted" approach.

After the PPRV-based votes from each of the top 100 similar lenders were totaled, then the team recommendation was based on the top N teams (where N is the "cutoff" number) as determined by the total PPRV votes each team received. Each Kiva lender may join up to 20 total lending teams, so the number of teams represented by a given number of lenders will typically far exceed the number of lenders. For example, the top 10 similar lenders for lender node ID #306319 produced votes for 56 lending teams. For the validation analysis, cutoff numbers of the top 5, 10 and 20 teams were used. Values of the restart parameter, β, of 0.75 to 0.95 were tested and did not produce significant variation in recommended team rankings, so a standard value of β = 0.85 was used for all reported results.

The random walk with resets algorithm was programmed in both python using SNAP.py and the GLender graph, and in MATLAB based on the edge list produced by SNAP.py for the GLender graph. Both programs were tested using the network example shown in the CS224W lecture slides for page rank, and were cross-validated against a 5000-node subgraph to ensure both programs were producing consistent results. The python algorithm iterates, using the algorithm described in the CS224W lectures, over the entire GLender graph until the PPRV values converge within a specified tolerance. The MATLAB algorithm uses an ordered edge list and separate degree-index matrix to iterate for a specified number of "hops" in the random walk. In testing, producing a PPRV for a single lender required over 10,000 seconds for the python program, but the MATLAB program was able to achieve equivalent accuracy in less than 600 seconds using 100M hops in the random walk. Consequently the MATLAB program was used to create the PPRV rankings for the project validation.

The MATLAB program uses two arrays (or matrices) to compute a PPRV. First, it uses an ordered, complete edge list meaning that nodeID 0 will be the source node of the first edge in the list, and meaning that if edge 100314 → 324298 is in the list then edge 324298 → 100314 also appears later in the edge list. For the GLender graph, this "AllEdges" array contains 157,381,608 edges. The second matrix structure used by the MATLAB program is a degree-index array that provides the degree of each node and the index into the complete edge list where the edges for each node begin (see Figure 3). For the GLender graph, this "DegIndex" array is 547,651 x 2 in dimension.



Row #	"AllEdges" array Source Node	 Destination Node
31	…	…
32	13	1
33	13	7
34	13	18
35	14	2
36	…	…

NodeID or Row #	"DegIndex" array Degree	 "AllEdges" Index
11	…	…
12	2	30
13	3	32
14	7	35
15	…	…

Figure 3. Primary array structures for the MATLAB random-walk-with-restarts algorithm

The MATLAB program iterates for a specified number of hops to simulate the random walk with restarts. For each hop, it tests whether to restart and if so, returns to a random node in the teleport set, S. If the hop does not restart, it generates a random integer from 0 to the (degree – 1) of the current node and then uses the index in "DegIndex" plus the random number to find the new destination node of the random walk in "AllEdges." The arrival of each node in the random walk is counted, and the PPRV is estimated by dividing these counts by the number of hops or iterations in the random walk. Because of its design, the MATLAB program is linear with the number of hops such that a random walk of 1M hops could be completed for the GLender graph in under 6 seconds, although with some decrease in accuracy in the PPRV. The MATLAB algorithm is also relatively insensitive to the size of the network except that more hops are required to simulate the random walk for large graphs and still accurately estimate PPRV.

A simpler alternative to the PPRV recommendations was also investigated using the target node's nearest neighbors. In this method a team received a vote for each nearest neighbor who was a member of that team. Then, as with the PPRV method, the top N teams were returned as the team recommendations for that lender (target node). While easier to compute than PPRV for a single lender, this method would be difficult to extend if the teleport set was larger, such as for the problem of recommending new lenders for an existing team. Still, as shown in the Results section, this method produced values as good or better than the PPRV algorithm for single lender targets.

3.4 Validation and testing

Ideally we would have liked to predict or recommend memberships for lenders who are unaffiliated with teams so that we could see over time at what rate such lenders did or did not join the teams to which we matched them. However, this type of testing is impractical for a few reasons. First, it would require multiple downloads of the Kiva team membership data over time. This is difficult because team membership lists are not available and must be inferred from team listings for each individual lender, and that data is slow to obtain via the API. Second, we had less than four weeks to complete this project once we had obtained the initial team data, so the team membership changes in that timeframe would likely be too few to be statistically significant.

Instead, we took two random samples of lenders (n = 100, and n = 1000) with existing team memberships to determine the recommended team memberships for these people. Then we tested our algorithms' predictions against their actual team memberships to determine how well our algorithm predicts the lenders actual team membership. As performance metrics we used precision and recall which are standard measures for evaluating search strategies. Precision is the fraction of predicted teams of the top N that are accurate (i.e., the target lender is a member of the team(s)). Recall is the fraction of all actual teams that were predicted among the top N recommendations. For example suppose lender 3049 is a member of teams A and B. Then if team A or B appears in the top 5 recommended teams, the precision would be 1/5 or 0.2 and the recall rate would be ½ or 0.5 in this case. Finally, in the event that teams on the edge of the cutoff number were tied in votes, the precision and recall values were scaled by the number of teams involved in the tie. Again using the example above, suppose team rankings 4, 5 and 6 were tied with the same number of votes and team A was listed as team number 5. Then the precision and recall would be scaled by 2/3 which is the probability that team A would randomly be in the top 5 listing. We considered this to be a more accurate metric, particularly for recall, versus including all teams (regardless of total number) that might be tied with the team at the cutoff.

4 Results

The results of our project are based on the validation of the techniques described previously against a random set of Kiva lenders who are currently members of one or more lending teams.

4.1 Software testing and parameter optimization

As previously mentioned, the random walk with restarts algorithm was programmed in both python using SNAP.py and in MATLAB by two different programmers. Both of these programs reproduced the PPRV values for the example shown in the CS224W lecture notes using a value of $\beta = 0.89$. In addition, both algorithms were used to produce a PPRV for node ID 306319 (a lender who is a member of 20 lending teams) to validate that both programs gave the same PPRV values and were therefore operating correctly. A similar cross validation was also performed with a random node in a 5000-node subgraph.

Once the faster MATLAB algorithm was confirmed to be working as designed, we calculated multiple PPRVs for node ID 306319 using a range of β values from 0.75 to 0.95. The resulting team recommendations were invariant for β values from 0.75 to 0.87 and varied little above that range. Therefore the common value of β = 0.85 was adopted for all further PPRV calculations. In addition the MATLAB program was used to calculate PPRVs with varying amounts of iterations ("hops") from 100K to 1B for accuracy comparions to the python results for node 306319. Using 100M hops required approximately 9 minutes to complete and produced PPRV values that were correct to 2 significant figures. The 1B hops PPRV was accurate to 3 significant figures but required approximately 1.5 hours to complete, and the 10M hops was less accurate on the PPRV values but still produced the same rankings of related nodes. Therefore, 100M iterations were used for the team validation testing against the 100 random nodeIDs, and 10M iterations were used for the team validation testing agains the 1000 random nodeIDs. Results of average precision and recall from both sets of random lenders were consistent to within a percent, so the results of the 100 random lenders is presented here unless otherwise noted because we know the PPRV values have greater numerical precision.

4.2 Team recommendation validation

Kiva now has nearly 30,000 lending teams. Although a lender may join up to 20 teams, most lenders who are on teams (86%) are only members of a single team (see Figure 4), and nearly 95% of lenders on lending teams are on less than 3 teams. We computed team recommendations for 100 lenders who are currently members of at least one team. The average precision for all three methods used (unweighted PPRV, weighted PPRV and nearest neighbors) was above 10% or 0.10 (see Figure 5). For comparison, the chance of randomly recommending the actual team in a list of "Top 5" team recommendations for any given lender is approximately 0.00017, and for the unusual case of node 306319 who is a member of 20 teams, the probability of recommending at least one correct team is 0.0137 using a binomial distribution model. Precision declines as the cutoff number increases since precision is inversely proportional to the cutoff number. Of the three methods of recommendation, the nearest neighbor voting algorithm produces the best precision with a value of 12% for a cutoff number of 5 team recommendations. For higher cutoffs, however, the PPRV values are slightly higher as seen in Figures 5 and 6.

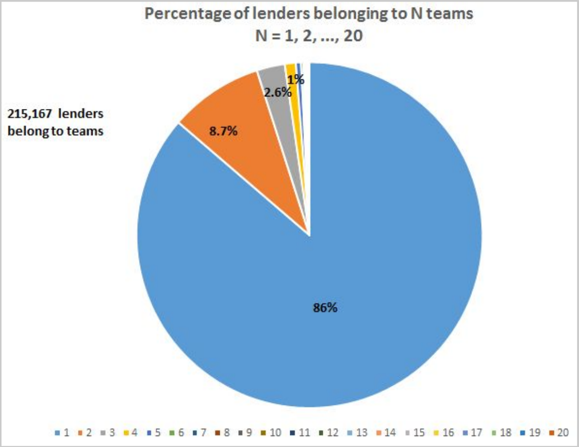


Figure 4. Number of lending team memberships by Kiva lenders

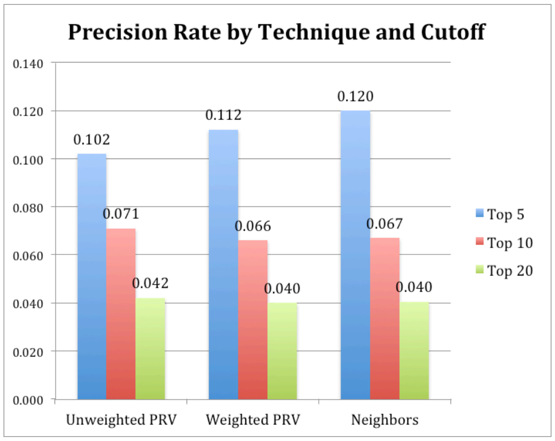
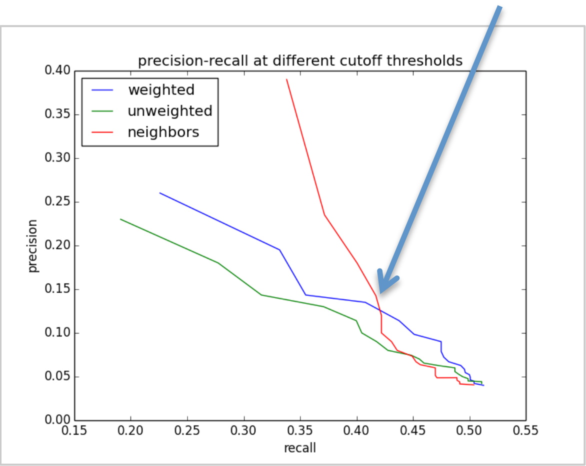


Figure 5. Team recommendation precision for 100 random Kiva lenders

For recall, the weighted PPRV algorithm produced the best team recommendations at 48% and 51% for top 10 and top 20 recommended teams and was nearly equivalent to the nearest neighbor algorithm for the top 5 case (see Figure 7). The recall rate increases with cutoff number for all of the techniques used in the validation testing showing that as we recommend more teams we are more likely to match the ground truth.

Cross-over at N=5



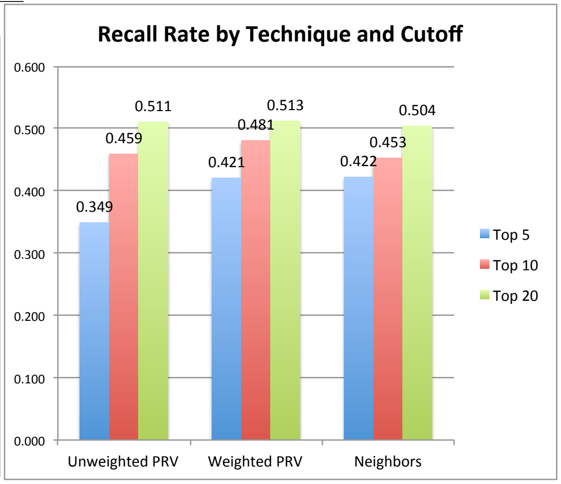


Figure 6. Recall-precision curves for cutoffs from 1 (upper left) to 20 (lower right)

Figure 7. Team recommendation recall for 100 random Kiva lenders

While all methods had perfect recall for about 40% of the validation cases, none of the methods were able to correctly recommend a team for a third of the tested lenders (see Figure 8). Of the lenders with 0 recall recommendations, nearly all were lenders who were on a single team but the proportion of these was not significantly different from the general population that is shown in Figure 4. Also, the degree distribution for these lender nodes varied from 8 to 8687 so it does not appear that the precision or recall scores were related to node degree in the GLender graph.

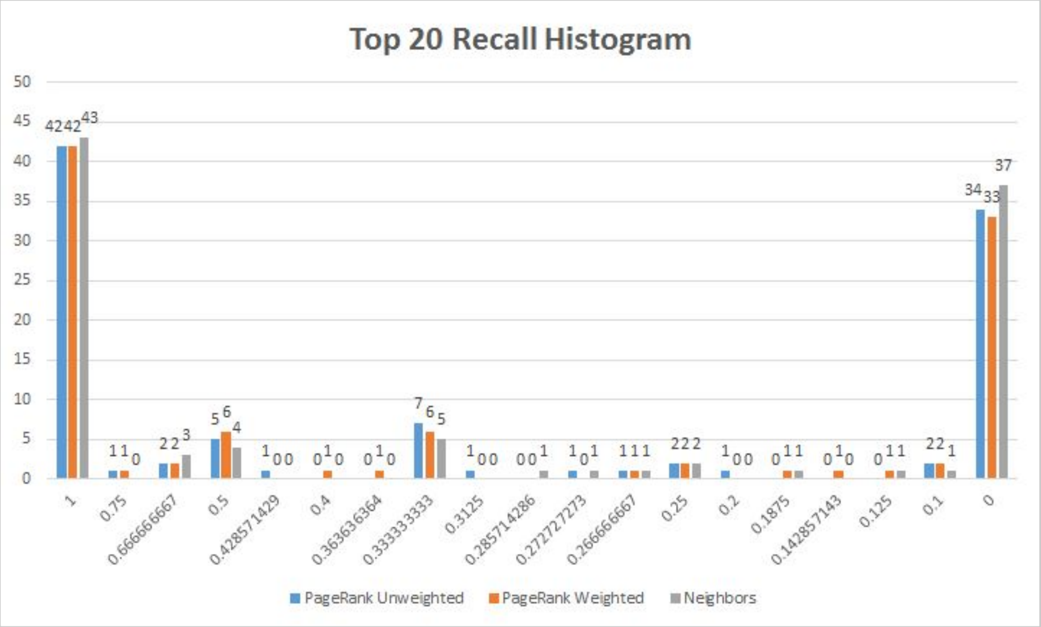


Figure 8. Histogram of recall values for 100 random Kiva lenders

5 Conclusions and Future Work

This section includes conclusions from the results of this project along with future directions that could proceed from this work.

5.1 Conclusions

We have shown that PPRV-based team recommendation is able to perfectly match teams to lenders in approximately one third of all cases. In addition, the weighted PPRV method (scoring teams by PPRV value divided by the number of teams a "similar" lender has) performs better in precision and recall than the unweighted technique. This seems reasonable since, as a lender is split between the goals of many different teams, he/she would in effect become less "like" any particular team.

We also have shown that a nearest neighbor technique is at least as good as the PPRV-based recommendations, and it resulted in slightly higher precision, particularly for "Top 5" team recommendations. Since this technique is computationally faster than PPRV, it could be a reasonable alternative to providing team recommendations for Kiva lenders. We believe the nearest neighbor technique performed well because PPRV often returns neighboring nodes but is also sensitive to high-degree nodes in the local neighborhood that may not truly have the same lending profile as the target user. Since the average degree of the GLender network is so high (287), the chance of high-degree nodes skewing the team recommendations is more likely.

We would have liked to see higher precision and recall values during our validation testing, however, we have shown that team recommendation using PPRV and nearest neighbor algorithms are far better than random team selection. The reason why PPRV and nearest neighbor techniques failed to predict teams in a third of cases was not determined, but it may include factors outside the scope of this project such as: 1) lenders who are members of teams but are not active in lending to loans promoted by those teams, 2) lenders who are members of teams for primarily social reasons or reasons other than the lending profile of the team, 3) lenders who joined teams recently and whose lending profile has not changed significantly since, and 4) teams that do not actively promote particular loans that would result in a common lending profile for team members.

Finally we have presented a data structure and algorithm for approximating PPRV with a random walk with restarts technique that is linear in time with the number of iterations, and is relatively insensitive to the size of the network being analyzed. The algorithm is able to generate PPRV values to support team recommendations for Kiva lenders.

5.2 Future directions of research

The recommendation of new potential members for each team could be as valuable to Kiva or even more so than recommending teams to lenders. It is reasonable to believe that a lender might respond more favorably to a personal invitation from a team leader to join their team than from a team recommendation list on their lender home page. As mentioned previously, the same random walk with resets algorithm could be used to provide lending team leaders with prospects by making the personalized reset set, S, equal to the current team membership versus a single lender.

Although construction of similarity vectors based on text classification and other machine learning techniques is daunting in a 10-week project, the use of these metrics, particularly for supervised random walks (Backstrom, 2011) has been shown to be quite effective in recommendation systems.

By using similarity measures based on stated features by lenders and loans in comparison to similarity based on network structure such as the proposed random walk with resets, a study of "words versus actions" could be done to determine which is more predictive of loan funding. For instance, it is unlikely that a lender would state (or perhaps even realize) that they have a sexist bias, but how well would their loan funding profile confirm or deny such a bias?

References

Anderson, Ashton, Huttenlocher, Daniel, Kleinberg, Jon, and Leskovec, Jure, 2012. Effects of User Similarity in Social Media. *WSDM '12 Proceedings of the 5th ACM international conference on Web search and data mining.* pp. 703-712.

Backstrom, Lars, Huttenlocher, Dan, Kleinberg, Jon, and Lan, Xiangyang, 2006. Group formation in large social networks: membership, growth, and evolution. *KDD '06 Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining.* pp. 44-54.

Backstrom, Lars, and Leskovec, Jure, 2011. Supervised Random Walks: Predicting and Recommending Links in Social Networks. *WSDM '11 Proceedings of the 4th ACM international conference on Web search and data mining.* pp. 635-644.

Chen, Roy, Chen, Yan, Liu, Yang, and Mei, Qiaozhu, 2013. Team competition in pro-social lending: a field experiment on Kiva. *Proceedings of the National Academy of Sciences, forthcoming*. Available as a working paper from the Science of Philanthropy Initiative (www.spihub.org).

Choo, Jaegul, Lee, Changhyun, Lee, Daniel, Zha, Hongyuan, and Park, Haesun, 2014. Understanding and Promoting Micro-Finance Activities in Kiva.org. *WSDM '14 Proceedings of the 7th ACM international conference on Web search and data mining.* pp. 583-592.

Clauset, Aaron, Newman, M.E.J. and Moore, Cristopher, 2004. Finding community structure in very large networks. *Phys. Rev.* E. 70, 066111.

Newman, M.E.J. and Girvan, M., 2004. Finding and evaluating community structure in networks. *Phys. Rev.* E. 69, 026113.

Newman, M.E.J., 2004. Fast algorithm for detecting community structure in networks. *Phys. Rev.* E. 69, 066133.

Palla, Gergely, Derenyi, Imre, Farkas, Illes and Vicsek, Tamas, 2005. Uncovering the overlapping community structure of complex networks in nature and society. *Nature.* vol. 435, pp. 814-818.

Individual Contributions Summary

Working on this project was a great experience where we applied what we learned in the course to solve a real problem. Not only did we learn about network analysis techniques, we also had a valuable opportunity to collaborate, partition, and manage the required work among the three of us. This project is the overall outcome of the following contributions:

Abe: set up Github site for project, wrote program to crawl team data, wrote programs to calculate precision and recall values from PPRV node rankings, proofreading all reports.

Maha: created folded GLender graph used for the project, performed all community detection analysis for milestone report, wrote python random walk program, proofreading all reports.

Roger: downloaded half of team file data, wrote MATLAB random walk program, wrote drafts of all written reports, analyzed and graphed results, coordinated and drove project deadlines.