Weighing Edges: An Empirical Study on Epinions.com

Leo (Ling-Hung) Kung
Stanford University
lhkung@stanford.edu

ABSTRACT

In this paper we use features from the rating and trust networks of Epinions.com to predict the existence of a trust edge, and argue that the value generated from the classifier represents the weight of trust from a member to another. We then briefly examine how local weights can influence the dynamics in the network structure, and suggest several future directions to continue exploring the interactions in a weighted network.

1. INTRODUCTION

It is natural to perceive our social network as a weighted network: we have a few close friends and numerous acquaintances. We might loosely put all of them into the type of "friend," but we certainly treat them differently. Even though a great portion of the social network analysis literature focuses on unweighted networks, where edges are either present or not, in some cases introducing weights to networks provides more information about the underlying dynamics. For example, [11] analyzes the mobile call graph and proposes two candidates for weights: the aggregate call duration and the cumulative number of calls. In [12] they plot the subgraph around some individual and color edges by the aggregate call duration, and it is easy to find that most local bridges are weak ties, as suggested by [8]. Nevertheless, how do we know which of these two weight candidates are better in describing the relationship between nodes, and even more, by how much do they differ? To answer these question, we use the data from Epinions.com, where one member can both in a trust network by trusting others, and in a rating network by rate others' reviews. We hope by building a classifier from features in both networks to predict whether A trusts B or not, we can explain the underlying mechanism of trust, and use the value generated from the classifier as the weight about trust. Therefore, the better the classifier is, the more accurate can these weights describe trust, and from here, we can then run all sorts of analysis on weighted networks as suggested in [6]. The rest of this paper is organized as follows: In section 2 we introduce the rating and trust network of Epinions.com in more depth, and in section 3 we study the features for the classifier and the resulting performance from logistic regression. In section 4 we make an attempt to use weights to explain the dynamics of local structure, and in section 5 we enumerate possible extensions of this project after a brief conclusion.

2. NETWORKS IN EPINIONS.COM

2.1 Rating and Trust Networks

Epinions.com is a consumer review site where members write reviews on products in different categories. For example, Bob, has watched the first season of Grey's Anatomy, written a review, and assign a score (number of stars) to this product. Now suppose Alice is interested in Grey's Anatomy as well, and want to know what others in Epinions think about it. Among other reviews, she thinks Bob's review contains information for her to make decisions, and she rates Bob's review by assigning another score (level of helpfulness) to it. Any member in Epinions.com can write and rate reviews, but for clarification, we would refer to Alice as the rater and Bob the reviewer. In the case where Alice rated Bob's review, we call Bob as Alice's Rating Neighbor (RN), the pair Alice and Bob as the Rater-Reviewer Pair (RRP), and the directed multi-graph with members as nodes and ratings as edges the Rating Network. Note that a RRP is not an edge in the rating network since Alice might have rated 5 of Bob's reviews, but rather a concentration of information from edges with source node Alice and destination node Bob. After a few interactions, Alice consistently finds Bob's reviews to be valuable, and she wants to see more of them, so she decides to add Bob into her web of trust. This action makes Bob one of Alice's Trust Neighbors (TNs), and the directed graph with members as nodes and trusts as edges are called the Trust Network. One cool thing about the these data is that events like writing/rating reviews and trust others are mostly dated, and this allows us to examine not only the static statistics of the rating and trust networks, but also the dynamics of how networks evolve over time. In this paper we only focus on out-going edges, but analyzing in-coming edges also provides perspective on, say, how Bob can attract Alice's attention and even her trust, and this will be examined in the next phase of this project.

2.2 Incentives for Members' Behaviors

Why would members want to write/rate reviews and trust other members? Epinions claims in [1] that these behaviors help general consumers to make decisions on products, but to ask a member to write hundreds of reviews with thousands of words each, rate thousands of reviews, and trust hundreds of others, there ought to be some incentives to explain these efforts. And there are. If you write good reviews that help others making decisions, then Epinions will gain revenue by providing consumers with good-quality information, and will reward you through the Income Share program by depositing a share of the revenue into your account. The details of the formula to compute your Income Share are vague, with only some guidelines on how to earn more Income Share, to prevent any attempts to defraud the system. Some of the top earning reviews in different categories can earn from 8 to 65 dollars, and it somehow motivates members to write high-quality reviews. Ratings and trusts, on the other hand, focus more on predicting how helpful a review will be to you: if Alice is now collecting information for Friends, and Bob happens to write another review about it, then Bob's review will be highlighted for Alice if she trusts Bob (and assigns good rating scores). Even though the effect of a rating is not as prominent as trust, all it takes for Alice is just a click after going through Bob's review, and we believe this low cost is enough to explain members' rating behaviors.

3. WEIGHT AS PREDICTOR OF TRUST

Since the rating and trust networks share the same nodes, from the trust network, we know whether Alice trusts Bob (trust RRP) or not (nontrust RRP). We can also derive features from both the rating and the trust network that partially describes the relationship between Alice and Bob. For example, the average rating score describes on average how helpful does Alice think about Bob's reviews, and if this score is high, we believe that Alice is more likely to trust Bob. Suppose with these features, a classifier is built to predict/explain whether Alice will trust Bob, by generating a value between 0 and 1. If this classifier is "good," the generated values can be viewed as weights that describes how trustworthy does Alice think of Bob, and of course the better the classifier, the more these weights can explain how trusts are formed. The rest of this section will enumerate the chosen features and illustrate the performance of different classifiers.

3.1 Features

To predict how Alice thinks of Bob, there are at least four relevant aspects:

- Alice's personal characteristics: How likely is Alice to trust others? How active is Alice? How long has Alice been a member? Answers to these questions describe how Alice initiates a trust edge and provide a baseline on her behavior. Here we pick the ratio of Alice's number of TNs to her number of RNs, telling us Alice's interpretation about trust.

- Bob's personal characteristics: As the person to be trusted, Bob's public image is also important, and this includes the number of reviews he writes, the quality (helpfulness) of his reviews, the number of people that trust him, and so on. For now we only use Bob's number of trusted-bys, and it is natural to think that Bob is more likely to be trusted by Alice if he has been trusted by many others as well.

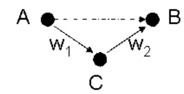

Figure 1: Weighted propagation of trust. How does the existence of A→C and C→B with weights w_1 and w_2 influence the addition of A→B.

- How Alice think about Bob: Two easiest parameters to get from the rating network are the number of ratings and the average rating score, indicating the frequency and the quality of this one-way interaction. However, with Alice initiating these rating edges, we need to consider how Alice treats her other neighbors. To take this into account, we can record the number of ratings (and the average score) for all Alice's RNs, calculate the mean and standard deviation for these values, and use them to normalize the number of ratings (and the average score). We will see the different roles played by unnormalized and normalized parameters in the performance of classifiers.

- The local network structure around Alice and Bob: Whereas the previous normalization can be viewed as considering the local structure in the rating network, we will examine how Alice and Bob interact with their neighbors in the trust network. Specifically, we focus on the number of intermediate nodes that form 2-hop transitive paths from Alice to Bob, such as nodes like C in Figure 1. It is easy to imagine that the chance for Alice to trust Bob is higher if there are many intermediate nodes to "propagate" Alice's trust to Bob. To compute this feature, however, the edge AB can only happen after the existence of AC and CB, which means this feature cannot be calculated for edges generated on the "first" date of the trust network.

Since both networks change over time, the above features will have distinct values at different time points. How should we choose the proper time point to obtain one value for these features? For the trust RRP, since the ratings after Alice trusts Bob have no influence on Alice's trust decision in a causal world, we can concentrate on the dynamics before trust, and use the date that Alice trust Bob as the time reference for this trust RRP. On the other hand, if Alice and Bob form a nontrust RRP, every time Alice rates Bob, she checks his performance in these features and decides that he is not qualified. Therefore, we would record these features whenever Alice rates Bob, and pick the maximum value for this nontrust RRP.

3.2 Data Crawling Process

Even though there is a clear boundary indicating where these networks end, there is no easy way to collect all nodes and edges other than directly asking Epinions staff for them. Instead of doing that, we use the traditional snowball sampling method to get as many members as possible. We start with a set of members acquired from the list of most popular authors in each year, and treat them as a seed. For each member in the seed, we collected her web of trust (including people who she trusts and people trusted by her),

Table 1: Relative accuracy for classifiers with different predictors compared to random guessing (with accuracy 0.5) for data set with equal size of trust and nontrust RRPs.

Feature	NumRtg	Nor. NumRtg	AvgScr	Nor. AvgScr	NumTrustedBy	TRRatio	All
Performance	1.318	1.364	1.296	1.167	1.249	1.449	1.659

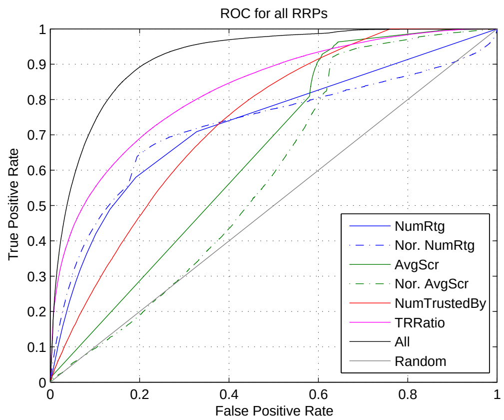

Figure 2: For all RRPs, plot the ROC curve for classifiers with individual features as predictors and the classifier uses all features to predict.

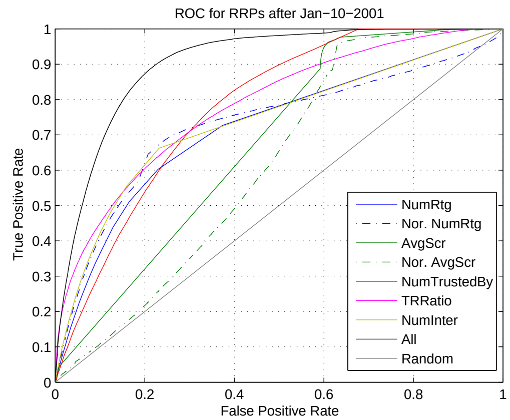

Figure 3: For RRPs after Jan-10-2001, plot the ROC curve for classifiers with individual features as predictors and the classifier uses all features to predict.

her list of reviews, and the corresponding ratings for every single review. If any new member is encountered during the above process, we add them to the seed as well. The process continues until all members in the seed have been looked up once, and no more new members are found. It is not clear how many other members are missed, but from our knowledge about social networks, most of them form a giant connected component containing more than 90 percent of the nodes. Besides, for the nodes that are left out in this discovery process, they tend to be either isolated or forming small components, and hence are unlikely to influence the performance of classifiers much. Therefore by using both rating and trust networks, we are confident in collecting enough data for analysis, compared to crawling only the trust network, as in most literature analyzing Epinions.com, like [9].

3.3 Basic Statistics

Using the method discussed previously, we are able to collect data from Jan-10-2001 to Sep-10-2009, among them are 176,825 members, 585,569 trust edges, 24,973,786 rating edges, and 6,059,215 RRPs. For each nontrust RRP, it is clear that there will be at least one rating corresponding to it, otherwise this RRP would never be discovered by our crawling process. For the trust RRPs, however, it is possible that the rater never rates the reviewer before trusts him, and it would be difficult to figure out how this trust edge initiates from our data. Therefore, we would restrict ourselves to trust RRPs with at least one rating before the rater trusts the reviewer. Another situation we try to avoid is when Alice rates someone but trusts no one. In this case, we don't know whether none of Alice's rating neighbors have reached her trust threshold, or Alice is simply too lazy to add them to her web of trust, and her existence cannot improve the performance of classifiers. If these suggestions are followed, we now have 5,358,033 RRPs, with 387,019 trust RRPs and 4,971,014 nontrust RRPs. To compute the number of intermediate nodes, however, we need to delete trust RRPs with reviewer being trusted on Jan-10-2001. To compensate for trust RRPs, we also remove nontrust RRPs with the last rating about the reviewer is on Jan-10-2001. We are now left with 168,405 trust RRPs and 3,050,836 nontrust RRPs, and we will build another classifier with smaller size of data but one more predictor.

3.4 Building Classifier

Curious about how these features perform as a classifier, we choose logistic regression as the first trial. Other possibilities include decision trees, naive Bayes, and support vector machine, which might tell us different perspectives about these features. One thing to notice is that, whether we use the number of intermediate nodes as a predictor, the size of the trust RRPs is much smaller than that of the nontrust RRPs, and this might place some constraints on how well logistic regression can perform. Several methods to address class imbalance is discussed in [14], from which we decide to under-sample the nontrust RRPs to the size of trust RRPs, and build a classifier from this new data set. This trained classifier will then be used to test all RRPs and performance being compared. One common drawback for under-sampling is to discard information from the nontrust RRPs and possibly degrade the performance of the classifier, but with the abundance of our data, the empirical difference is so small that we can ignore it. Another issue brought out by the rarity of trust RRPs is how practical accuracy is. If we claim every RRP to be nontrust, we can have a classifier with accuracy higher than 0.9 but is useless for us. By examining the Receiver Operating Characteristic (ROC) curve, we can adjust thresholds to see how the false positive rate varies with the true positive rate, and choose the proper tradeoff

Table 2: The number of trust and nontrust RRPs with a specific number of intermediate nodes.

NumInter	5	10	15	20
Trust	6014	3224	1982	1158
NonTrust	35194	13400	7047	4082
NumInter	25	30	35	40
Trust	841	515	414	264
Nontrust	2463	1670	1110	711

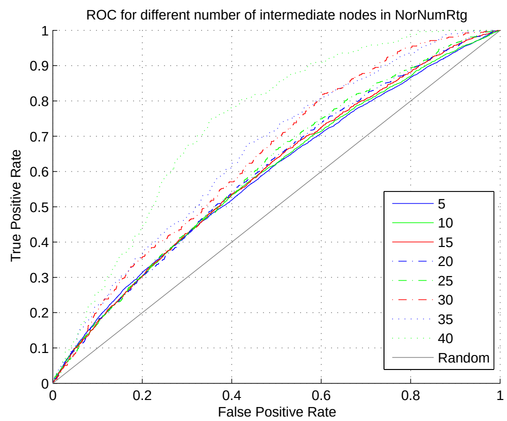

Figure 4: For RRPs with a specific number of intermediate nodes, plot the classifier ROC curve by using network features from normalized number of ratings.

level between trust and nontrust RRPs.

3.5 Regression Results

For the first set of 5 million RRPs, the classifier performance by using each individual feature is shown in Figure 2. In general we see one classifier outperforms another in some region but is beaten in others, and even if the average score looks better than the normalized average score, it is too early to argue that the normalized average score is of no use given the average score. To have a first order understanding about the difference of classifiers, we use the data set with equal-sized trust and nontrust RRPs for training and testing, and set the threshold to be 0.5 to compare performance. With 50% accuracy for random guessing, in Table we list the relative accuracy of classifiers compared to random, and the average score and the normalized number of ratings are better than their counterpart in this setting. Actually the ROC curve by using all predictors is very close to the classifier with only Nor. NumRtg, AvgScr, and TRRatio, which roughly tells us the interdependence of features. If we remove RRPs on Jan-10-2001 and include another feature, the resulting ROC curve is shown in Figure 3. Even though some individual predictors perform better in this case, the performance degradation by removing 2 million RRPs is so huge that even the addition of an extra feature does not help. Once again, the effect of NumRtg, Nor. AvgScr, and NumTrustedBy can roughly be replaced by the presence of the other features.

4. WEIGHTED PROPAGATION OF TRUST

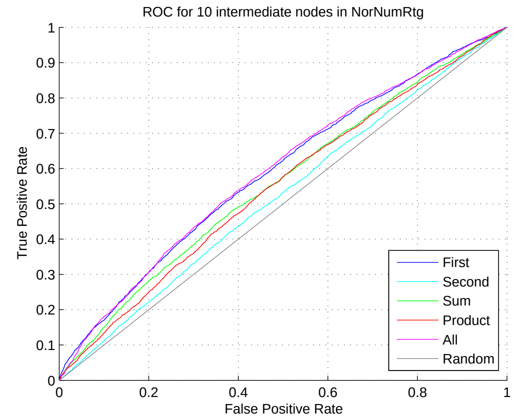

Figure 5: For RRPs with 10 intermediate nodes, plot the ROC curve for classifiers that either use only one perspective, or all of them.

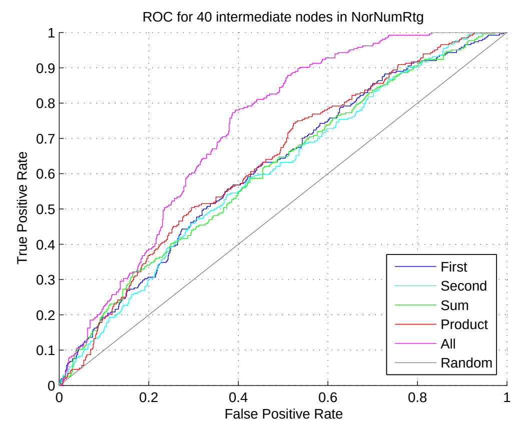

Figure 6: For RRPs with 40 intermediate nodes, plot the ROC curve for classifiers that either use only one perspective, or all of them.

With the weights generated in the previous section to explain the directed trustworthiness, what other aspects about the networks can these weights provide? Let's consider Figure 1 again, and assume the RRP AB has only one intermediate node C. Consider two cases where in one $w_1 = 0.9$ and $w_2 = 0.8$, whereas $w_1 = 0.1$ and $w_2 = 0.2$ in the other. If these weights truly capture the trust relationship, we would expect the chance for A to trust B higher in the first case. This can be viewed as a weighted version of trust propagation, and we are curious about how weights influence the flow of trust in our data set. We cannot directly use the weights from the previous section, because with only 7 or 8 features, we cannot argue that the full dynamics have been captured by our classifier. What is more, it is not clear what it means to use the predicted values to predict trust again. Therefore, we fall back to the traditional idea by letting the normalized number of ratings as one weight, and see how this weight differentiates the propagation of trust.

In our data, one RRP can have more than one intermediate node, and when there are multiple paths for trust to flow, we don't know the critical ones that guides A to trust B. Therefore, we would examine RRPs with the same number of intermediate nodes, and extract features from this local structure. When there is only one intermediate node, we see there are at least four possible perspectives of how these weights matter, including $w_1, w_2, w_1 + w_2$, and $w_1 w_2$. Now for more paths in between, we can add features like the average of the highest two w_1's, or the average of the highest three $w_1 w_2$'s. In general, for each perspective, we use features from the the highest weight up to the average of the top 80% weights, and the hope is that some of them will capture the actual trust flow and allow us to distinguish trust v.s. nontrust RRPs.

From the statistics in Table 2, we see as the number of intermediate node increases, the total number of RRPs decreases, and the proportion of trust RRPs increases. We stop at 40 intermediate nodes because beyond that, the number of RRPs will be less than 1,000, and the variance of the classifier will be huge. For different number of intermediate nodes, the performances of classifiers built from logistic regression are shown in Figure 4. We notice that as the number of intermediate nodes increases, the classifier performance improves, but with fewer data, we also need to provide the significance level to tell how confident we are.

Now for a specific number of intermediate nodes, we want to know how different perspectives perform in this case. When there are 10 intermediate nodes, in Figure 5 we see that w_1 dominates all other perspectives, and including them into the classifier does not improve much. On the other hand, when there are 40 intermediate nodes, all four perspective performs about the same from Figure 6, and the joint classifier outperforms any of them, meaning that they carry different information about the network structure. From these two figures we observe that when the number of intermediate nodes is small, with a slightly higher probability, A tends to trust members trusted by C who she values more, and how C thinks of B is not that relevant; as the number of intermediate nodes grows, the importance of the second tier weights increases as well, and this weighted trust propagation tells us more than the unweighted one.

In the case of 40 intermediate nodes, there will be 128 predictors if we use all 4 perspectives. How do these features perform relative to one another? If we normalize these features before applying logistic regression, then the corresponding coefficient of each feature roughly tells us how important it is. In Figure 7 we plot the logarithm for the coefficient magnitudes, grouped in perspectives, with x axis the number of highest weights being averaged. We see that after including more than 10 top weights, most features begin to contribute to the classifier. Also, with w_1 being relative important when the number of averaged weights is small, other perspectives start to outperform w_1 by including more paths for average. So far the analysis has been based on a particular measure for weights, but with the observations of this exercise, we have a rough idea about how weights influence local network structure, and we expect a more delicate statement once we have a better classifier and know how to interpret the results.

5. CONCLUSIONS AND FUTURE WORKS

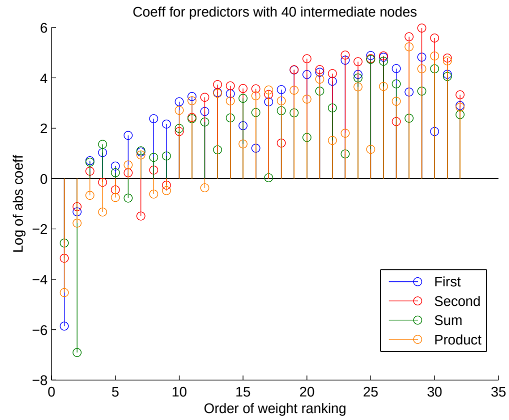

Figure 7: For the classifier with 40 intermediate nodes, plot the log of the coefficient magnitude for features with different perspectives.

In this paper we have introduced the network data from Epinions.com, extracted features from these data, and built a classifier with logistic regression to explain the existence of trust edges. Besides, we use normalized number of ratings as a simplified weight to demonstrate the effects weights have on the local network structure. Of course, there are several refinements to improve our current work, not to mention aspects to observe these data differently. The first thing to know is that while trust edges provide us certain relationship about the RRP, for nontrust RRPs we are not sure whether the reviewer is not good enough, or the rater is too lazy to add him to her trust list. This suggests that we should treat trust RRPs as labeled and nontrust RRPs as unlabeled data, and run semi-supervised learning instead of supervised learning by emphasize the existence of trust. This method should be more practical and introduce the possibility of unlabeled but potentially trust RRPs. Other possible extensions would be discussed before we end this paper.

5.1 More Features

The features we used for prediction is by no means the only possibility. For network structure itself, [10] has summarized dozens of network features for an undirected network, with their individual predictive accuracy compared to random guessing. We can generalize these features into our directed rating/trust network and enrich the explanation for trust. Besides, there is another aspect about features that we deliberately ignore in our previous discussion: the similarity between Alice and Bob. As mentioned before, reviews are about products in different categories, such as cars, books, or electronics. If all Alice cares about is books but Bob only write reviews on cars, it is unlikely for Alice to rate Bob's review, not to mention trust him. This suggests that the undirected measure, similarity, should also be taken into account. Hopefully with the expansion of the feature set, we are able to identify the features that explains trust well enough that leads to more analysis on this weighted network.

5.2 Distrust from TrustLet

Epinions allow Alice to put Bob into her web of trust if she likes Bob's reviews, but what if she thinks his reviews are pointless? Actually, Alice can put Bob into her block list in the second case, and Alice will never have to read reviews from Bob. This distrust lies in some of the non-trust RRPs, and could provide more insight into member interactions if combined with trust edges. Unfortunately, to prevent hard feelings, the block list is kept private only to its owner, and the only way to access it is via Epinions staff. In the extended dataset of TrustLet [3], a full list of trust and distrust edges is available up to Aug-12-2003, except the members name are replaced with ID numbers, and we don't have the mapping between these two. To make use of this data, we can create a subgraph of the trust network up to the same date, and apply algorithms in [5] to acquire the most likely mapping between the true anonymous network and our partial network. We can now replicate the analysis in this paper to a 3-way classifier, and enhance the performance of our trust/distrust predictor.

5.3 Triadic Position Census

Triadic census, as described in [13], has been widely discussed in sociology literature. It deals with all 16 possible configurations in a directed 3-node network, and discuss, for example, how stable these structures are as to appear in real-word data. If we go one step further by separating different roles in each of these configurations, then we end up with 32 possibilities, as mentioned in [7], and we call these triadic position census after James Moody's slides [2]. [4] provides an algorithm to determine the triad census of a network, which can be easily modified to deal with triadic position census. With these ready tools and the time stamps of the trust network, we can calculate the transition probability from one configuration to another, and even build another model with these local dynamics to explain the global network evolution.

5.4 Evolution of Weights

If Alice trusts Bob in 2001 with weight 0.9, does she hold the same attitude towards Bob in 2009? Unlikely. With a good enough classifier, there is a corresponding formula describing how features jointly contribute to the final weight. Instead of using one weight for each RRP, we can now compute these weights on the fly and observe how these weights change as time goes. The most natural target would be the RRPs where the reviewer was once trusted but later being removed from the web of trust, and this information can be obtained by the mismatch between our data and those from TrustLet. Another related topic is to explore how previous interactions are forgotten by members. The easiest way is to assume an exponential forgetting curve, introduce the discount factor to our features, and calculate the best fit as a classifier. But there is no guarantee that exponential is the right curve: we might remember everything up to a certain moment and forget everything before that. To characterize the general shape of forgetting curve, we need to record the time difference between the date of some feature and the reference point of its corresponding RRP, collect features that are of similar time distance into a bin, and train a model to get the forgetting coefficients.

6. ACKNOWLEDGMENTS

The author would like to thank Jonathan Haynes for his continuing assistance for this project, and Prof. Matthew Jackson, Prof. Daniel McFarland, Huang-Wei Chang, Jared Duke, Yi-hao Kao, and Sam Tsai for their insightful feedbacks. More importantly, without the advise and discussion with Prof. Jure Leskovec, the existence of this project would not be possible.

7. REFERENCES

[1] Epinions.com - faqs. `http://www.epinions.com/help/faq/`.

[2] Triadic position census. `http://www.soc.duke.edu/~jmoody77/SSRINets/rwjpt4.ppt`.

[3] Trustlet, extended epinions dataset. `http://www.trustlet.org/wiki/Extended_Epinions_dataset`.

[4] V. Batagelj and A. Mrvar. A subquadratic triad census algorithm for large sparse networks with small maximum degree. *Social Networks*, 23(3):237–243, 2001.

[5] M. Bayati, M. Gerritsen, D. F. Gleich, A. Saberi, and Y. Wang. Algorithms for large, sparse network alignment problems, 2009.

[6] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D. U. Hwang. Complex networks: Structure and dynamics. *Physics Reports*, 424(4-5):175–308, 2006.

[7] R. S. Burt. Detecting role equivalence. *Social Networks*, 12(1):83–97, 1990.

[8] M. S. Granovetter. The strength of weak ties. *American Journal of Sociology*, 78(6):1360, 1973.

[9] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 403–412, New York, NY, USA, 2004. ACM.

[10] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 556–559, New York, NY, USA, 2003. ACM.

[11] J. P. Onnela, J. Sarama"ki, J. Hyvo"nen, G. Szabo', M. A. de Menezes, K. Kaski, A. L. Baraba'si, and J. Kerte'sz. Analysis of a large-scale weighted network of one-to-one human communication. *New Journal of Physics*, 9(6):179, 2007.

[12] J. P. Onnela, J. Sarama"ki, J. Hyvo"nen, G. Szabo', D. Lazer, K. Kaski, J. Kerte'sz, and A. L. Baraba'si. Structure and tie strengths in mobile communication networks. *Proceedings of the National Academy of Sciences*, 104(18):7332–7336, 2007.

[13] S. Wasserman and K. Faust. *Social network analysis: Methods and applications*. Cambridge University Press, Cambridge, 1994.

[14] G. M. Weiss. Mining with rarity: a unifying framework. *SIGKDD Explor.Newsl.*, 6(1):7–19, 2004.