An investigation into application of Google's PageRank algorithm in on-line social communities

EECE 496 Final Report

Alireza Ali

74961046

Technical Supervisor: Dr. Matei Ripeanu

April 12, 2007

Abstract

Social networking sites have attracted millions of users in recent years. They are now part of the most visited sites on the Internet. This paper studies the notion of user reputation in online tagging communities. In tagging communities, users form an implicit relationship through content re-use and content sharing. This relationship can be extracted by looking at the tagging patterns of users. If the relationships are quantified as a graph, connecting the users by their shared interests, then it is possible to calculate the relative importance of each user by running the PageRank algorithm over this social graph. In a sense, this relative importance represents the user reputation. Since it is possible to construct the relationship graph in various ways, we analyze the responsiveness of these different formulations in presence of malicious user attacks. The experiments shows graph formulations that capture more relationships are less responsive to attacks. This study is a step towards understanding the social structure of online communities. It is an attempt to harness the hidden relationships formed by user activities in such communities.

Contents

List of Figures

List of Tables

1 INTRODUCTION

Online social networks are becoming more popular everyday. According to the statistics released by Compete Inc., a leading market research firm, top social networking sites such as myspace.com registered close to one billion visits in February 2008 [1]. With this increase in user base, social networking sites have tried to improve their user experience in order to attract more visitors from the competing sites. To provide a better user experience, the site owners have tried to make use of the user generated content. The amount of user-generated content, content collaboration and sharing is some of the major differences between social networking sites and regular web sites on the Internet.This user generated content coupled with the relationships between the users can be used to infer information that otherwise might not be available. For example, in most social communities the notion of friends have been used to suggest possible connections to users of the site.

This project focuses on online tagging communities where users are able to describe the items in the site by tagging them. The tags are usually a written description of an item that the user writes to descibe the item. We believe this user generated content can be used to extract an implicit form of relationship between users. We use this implicit relationship to assign reputation to the user of an online tagging community. This user reputation potentially can be a valuable metric for site owners in improving user experience, content quality and differentiating user-generated content.

The project mainly investigates the application of Google's PageRank algorithm in the reputation assignment process. The PageRank algorithm was developed by Google in order to rank billions of pages on the web [2]. We attempt to use the PageRank to rank the users of an online tagging community. Even though attempts have been made to rank web pages on the Internet, there is no study of applying the PageRank in social networking context. We believe this investigate will shed some light into the problem of user ranking in social sites.

This project is coordinated with Mr. Elizeu Santos-Neto, a PhD student in department of Electrical and Computer Engineering and Dr. Dr. Matei Ripeanu, my technical supervisor. All the codes written for this project is omitted from this report and is available through my technical supervisor.
In the following sections, we look at the design and implementation of our ranking process and study number of experiments we have conducted over our data set of this real-world social community site.

2 DESIGN AND METHODOLOGY

The PageRank algorithm is used as part of Google's search engine in order to provide relevant search results to end users. The PageRank is a link analysis algorithm that assigns a numerical weighting to each element of a hyperlinked set of documents, such as the World Wide Web, with the purpose of "measuring" its relative importance within the set [2]. In this project, we apply the PageRank to a relationship graph constructed using the activities in an online tagging community in order to find the relative importance of each user. We call this relative importance the "reputation" for that user. Since this reputation is a real valued number, it allows us to rank users with respect to their reputation.

As the first step in the process, it is necessary to construct a relationship graph for users of the tagging community. We capture user relationships using the notion of shared interest among users. In Section 2.2, we formalize the idea of shared interest and explain how these implicit relationships can be captured and quantified.

The relationship graph is a two dimensional matrix representing relationships among users of the site. Figure 1 shows a sample relationship graph for a community with only 3 users. Each column and row in the matrix represents a user and each non-zero element represents the referral weight from the column user to the row user. In the sample relationship graph of Figure 1,

there is only one relationship with the weight of 1 from user2 to user1.

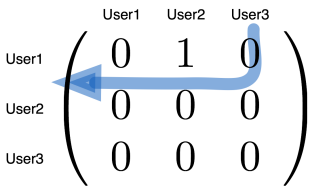

Figure 1: Sample relationship graph for a community with 3 users.

The relationship matrix is the input parameter to PageRank algorithm. PageRank uses this matrix to produce a one dimensional matrix where each row represents a user and the the single column contains the PageRank value for that user. Before diving into the details of the ranking process, we will look at the CiteULike data set in more details.

2.1 Data Set

In order to evaluate our ideas, we have used the freely available data set of CiteULike.org [3]. This site is mainly used to tag academic articles and papers and its data set is publicly available at `http://www.citeulike.org/faq/data.adp`. This data comes in the form of a single text file that includes all user activity for the entire site. Each line in the file is an entry that with the following fields:

1. The CiteULike article id which was posted

2. An obfuscated representation of the username

3. The date and time the article was posted to the site

4. Tag that was used when the item was posted

This data set captures all user activity from November 2004 to August 2007. Table 1 shows a detailed statistics of the data. To ensure the accuracy of of our ranking process we manually cleaned up the data set from the tags that did not hold any valuable information for the ranking process. For example, tags such as '-', 'the' and 'and' do not convey any valuable information. We removed a number of such tags from our data set. Table 2 shows the list of removed tags with their occurrence counts in the data. An updated statistics after removing these tags is shown in Table 3.

As the first step, this flat text file was converted to a relational database. This conversion enables us to take advantage of the relational database properties such as sophisticated query mechanisms, synchronization and flexible input/output interfaces. The strucutre of this relational database is shown in Table 4.

2.2 Shared Interest

Using the activity data from the CiteULike, we are able to construct a relationship graph for the users of the site. This relationship graph represents the degree of content collaboration and sharing among users. This graph can be constructed in various ways. In this project, we focus our attention

Number of users	22,662
Number of tags	202,611
Number of tagging activities	3,791,961

Table 1: CiteULike statistics from Nov. 2004 to Aug. 2007

tag	count
bibtex-import	142865
no-tag	53230
at	30568
and	16876
to	14117
for	13612
-	12413
of	10124
in	7723
the	5987
on	5695
a	5155
by	2202

Table 2: Removed tags from the data set

Number of users	22,662
Number of tags	202,598
Number of tagging activities	3,471,394

Table 3: CiteULike statistics after data clean up.

userinfo	
id	primary key
hash	

taginfo	
id	primary key
tag	

taginfo	
id	primary key
itemid	
time	
userinfo_id	foreign key
taginfo_id	foreign key

Table 4: database tables for CiteULike activity data

on methods similar to [4] where the relationships are extracted by analyzing the tagging patterns among users. The following terminology has been introduced in [4] and we will use it through out this report.

A tagging community is represented by a tuple in the form of $C := (U, I, T, A)$ where U is the set of users, I is the set of items, T is the set of tags and A is the set of tag assignments in the community. In this context, the activity of a user is represented by $u_k := (I_k, T_k, A_k)$ where $I_k \subseteq I$ is the set of items posted by the user u_k, $T_k \subseteq T$ is the set of tags assigned to those items and $A_k \subseteq A$ is the set of tag assignments performed by user u_k.

Now we define 3 different metrics to capture the shared interest level among users of the community.

> **Definition 1: (User-Tag)** in this definition, the shared interest level between two users is the intersection of their tag sets. In another words, it is the number of tags they have in common in their tag sets. In this approach, the user that used a tag for the first time becomes the tag owner and other users share the same interest by re-using the same tag.

Definition 2: (User-Item) the shared interest level between two users is the intersection of their item sets only. It is quantified by the number of identical items that they have tagged. In this formulation, the actual tag phrase does not matter. The user that tagged an item for the first time becomes the item owner.

Definition 3: (User-Tag-Item) the shared interest level between two users is the intersection of their item sets for the items that have been tagged using the same tag. Alternatively, it is proportional to the number of items that the two users have tagged using the same tags.

2.3 Weight Assignment Functions

Now that we have different metrics to measure the shared interest level, we need to come up with a weight assignment (WA) function that is able to quantify this shared interest. We devised three different weight assignment functions that are used in the ranking process.

Monotonic WA: This function gives a constant value (1 in our experiments) to each referral from $userA$ to $userB$.

Decreasing WA: The decreasing weight assignment function uses the harmonic series for weight assignment. For example using User-Tag definition, if 4 users use the tag $T1$ the total weight assigned to the tag owner is $1 + \frac{1}{2} + \frac{1}{3} = 1.8333$.

Distributed Decreasing WA: This weight assignment function also uses the harmonic series to calculate the weights. However, this weight is assigned to all users that share the same interest. This is why this function is called distributed.

2.4 Graph Formulations

Now that all the necessary terminology has been introduced, we can demonstrate how all these components fit together in an hypothetical simple scenario. Suppose we have a tagging community with only 5 users. The following lines are the user activities for this site:

$$u_1 := (I_1, T_1, 0)$$

$$u_2 := (I_1, T_2, 1)$$

$$u_3 := (I_1, T_1, 2)$$

$$u_3 := (I_2, T_1, 2)$$

$$u_4 := (I_2, T_1, 3)$$

$$u_5 := (I_2, T_3, 4)$$

These activities are visualized in Figure 2. We now look at the relationship graph construction using each of the definitions introduced earlier.

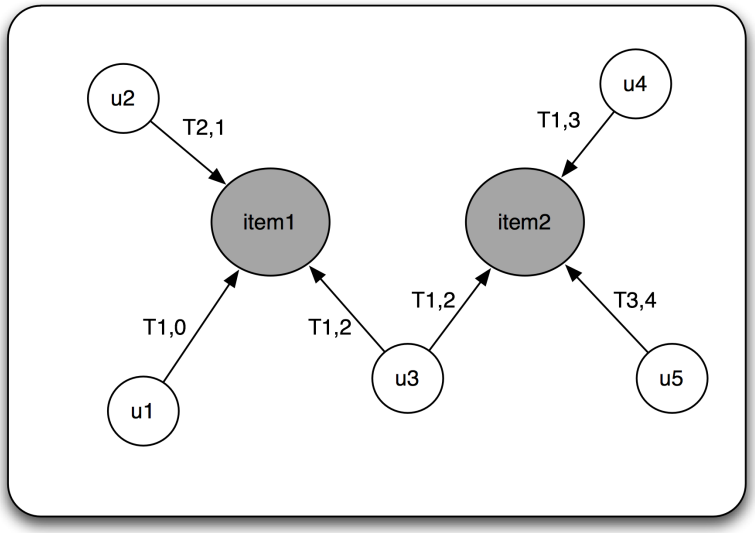

Figure 2: Visualization of user activity for our sample community.

2.4.1 User-Tag

Using the User-Tag definition, the relationship graph for the site is a 5×5 matrix. In the first scenario we use the Monotonic WA function and the relationship graph is shown below:

$$\begin{pmatrix} 0 & 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{1}$$

As it appears in matrix 1, only u_3 and u_4 give referrals to the $T1$ owner, u_1.

Also since we used the monotonic WA function, u_3 gives the weight of 2 for its 2 referrals. We now replace the monotonic WA function with Decreasing WA and compute the graph again. The graph is shown in matrix 2 and in this formulation u_3 gives only the weight of 1.5 rather than 2 to u_1.

$$\begin{pmatrix} 0 & 0 & 1.5 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{2}$$

Finally, we use distributed decreasing weight function and we get the following graph:

$$\begin{pmatrix} 0 & 0 & 1.5 & 0.333 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.333 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{3}$$

Since the distributed decreasing WA function gives the weight to all users with the same interest, we can see that in this formulation u_4 gives a weight of 0.333 to u_3 as well as u_1.

2.4.2 User-Item

In this approach an item is used as the token of reputation transfer. If one or more user tag the same item using any tag, then a relationship is formed from these users to the owner of the item. Using the same scenario in 2.4.1 and the monotonic WA function, we can come up with the following relationship matrix.

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{4}$$

According to the matrix 4, the only 2 referral receivers are u_1 for item1 and u_3 for item2.

2.4.3 User-Tag-Item

Using definition 3, we consider only the relationships when 2 users tagged the same item using the same tag. This formulation captures the strongest form of such relationships since 2 users have to use the same tag on the same item. Constructing the graph using this definition and the monotonic WA function gives the following graph.

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{5}$$

2.5 Graph Construction

The relationship graph construction process is implemented in Java. It is a multi-threaded program that consumes the activity data and produces the relationship graph. This graph is represented in a separate table in our relational database. The Java application, its unit tests and all the necessary script files are around 2000 lines of code. Due to large number of activity records, around 3 millions, the application had to be highly optimized to achieve an acceptable run-time. Even with all the optimizations the graph construction process for computationally intensive formulations such as User-Tag approach with distributed decreasing WA function, took in order of days to complete.

2.6 Ranking Process and PageRank

After construction of the relationship graph, it is possible to rank users of the community site using any ranking algorithm. In this project, our focus is to use Google's PageRank algorithm to rank the users. Following paragraphs

gives an overview of the algorithm and explains our approach in using the algorithm in the ranking process.

PageRank is a variant of the eigenvector centrality measure used commonly in network analysis. The PageRank values are the entries of the dominant eigenvector of the modified adjacency matrix. In another words, PageRank represents the likelihood of arriving at a particular web page by just randomly surfing the web. We translate this idea to the social networking context and assume that by finding the PageRank value for an individual user, we are essentially finding the relative importance or reputation of that user in the community.

The PageRank is a computationally intensive algorithm that heavily uses matrix algebra. In order to simplify our implementation and to achieve more flexibility, we decided to implement the PageRank in MATLAB [5]. MATLAB is a numerical computing environment and programming language that is developed by MathWorks Inc. and it is heavily used in industry in wide range of application. The main reason in choosing MATLAB for PageRank implementation is its ability to handle large matrix calculations.

Our implementation of MATLAB is a modified version of this algorithm available in [6]. The PageRank implementation takes in as input a $n \times n$ square matrix and returns a $n \times 1$ matrix that contains the ranking for each entity. CiteULike data set contains the information for $22,662$ users. This means the relationship graph for this community is a matrix of size

22662×22662 with maximum of 22662^2 elemenst. However, this matrix is very sparse with large number of zero elements. The degree of sparsity varies depending on graph formulation and WA function used. The ability of MATLAB in handling matrices with such size makes it a suitable tool for our ranking process. The next section describes the experiments we conducted on CiteULike data set.

3 EXPERIMENTS

Using the different graph formulations and WA functions introduced before, we devised number of experiments that we ran over CiteULike data set. In the first experiment, we constructed the relationship graph using the User-Tag method and monotonic WA function. The Java application successfully calculated all the elements of the graph and then it was exported into a single text file. This file was read into MATLAB and fed to our PageRank algorithm. The matrix density for the relationship graph using this method was 9.3526×10^{-04} containing only 480319 elements. Figure 3 shows the graph of the sorted PageRank value for users versus the user rank. Users with high reputation (low rank) appear at the top left corner and as we move to right, PageRank values decrease indicating users with lower reputation. As it appears in graph, large number of users at the flat right end of the graph receive the minimal PageRank. This is mainly due to the sparsity of our relationship graph. In another words, a large number of users in this community site does not have any relationships to other users.

In the second experiment, we repeat the process with the same graph formulation but used the decreasing WA function. The results are shown in Figure 4. The graph is very similar to the previous experiment except that at top left corner of the graph the PageRank values are dampened a little. The reason for the lower PageRank value is the smaller weights assigned to tag owners for incoming referrals by the weight assignment function.

The third experiment is similar to previous experiments except we replaced the WA function with the distributed decreasing function . The idea behind distributing the weights among users of the same tag is that when a tag is used by multiple users, we attempt to give credit to all users who used the tag and not just the tag owner. The weight that goes to each user is determined by the order in which user tagged the item. The PageRank graph for tis formulation is shown in Figure 5. The graph contains $20,019,578$ elements with the density factor of 0.0390. This is a much denser graph compared to previous experiments. Since in this formulation a larger number of users were able to receive referrals from the others, we see that the flat right end of the graph is much shorter indicating that there are smaller number of isolated users. Also at the top right hand side, we see a decrease in PageRank values which is due to the fact that now the reputation has to be distributed among larger number of users and hence the top users will receive proportionally smaller PageRank value.

In the second category of experiments, we look at the User-Tag-Item formulation. First, we run the PageRank on User-Tag-Item formulation with monotonic WA function and the results are shown in Figure 6. In this scenario the flat part of the graph on the right side is much longer. This means a larger number of users were isolated and they do have not received any referrals from other users. This result is expected since we are only considering users that have tagged the same item using the same tag phrase. The

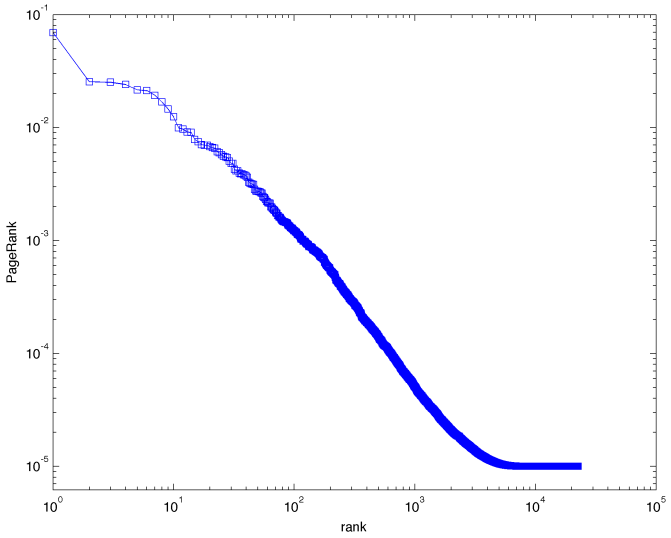

Figure 3: Sorted PageRank value versus user rank for User-Tag formulation with monotonic WA function.

relationship graph for this scenario is very sparse with the density factor of 8.8713×10^{-06} only having 4556 elements.

Similar to User-Tag experiment, we replaced the weighting function with decreasing WA. The result are shown in Figure 7. For clarity, we graphed the rankings on top of the previous graph. As it appears in the figure, the decreasing User-Tag-Item is almost identical to User-Tag-Item graph. This observation is expected since in both scenarios, same number of users receive weights. The only difference is the amount of weight each user receives.

Finally, we perform the ranking on the User-Item formulation with monotonic WA function and the results are shown in Figure 8. Clearly, in this scenario

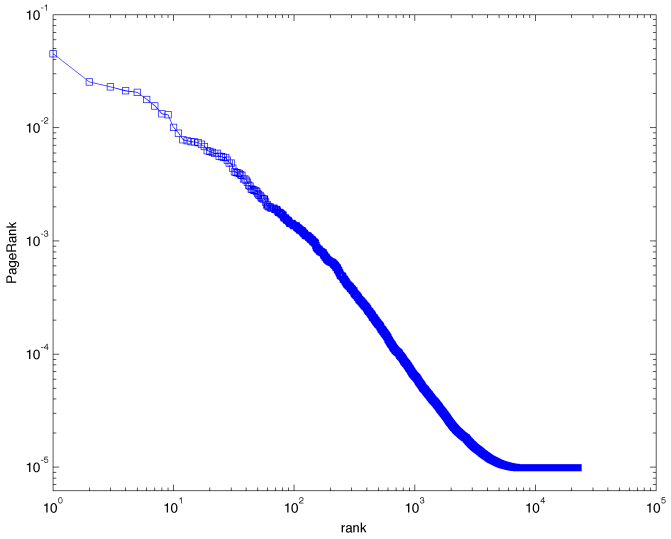

Figure 4: Sorted PageRank value versus user rank for User-Tag formulation with decreasing WA function.

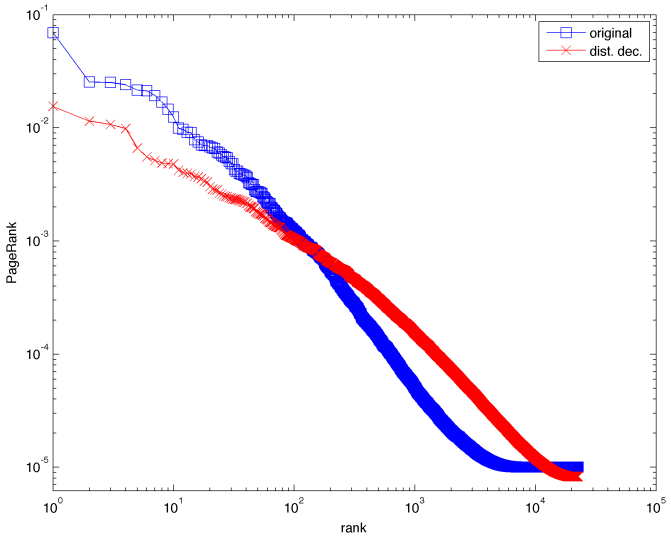

Figure 5: Sorted PageRank value versus user rank for User-Tag formulation with distributed decreasing WA.

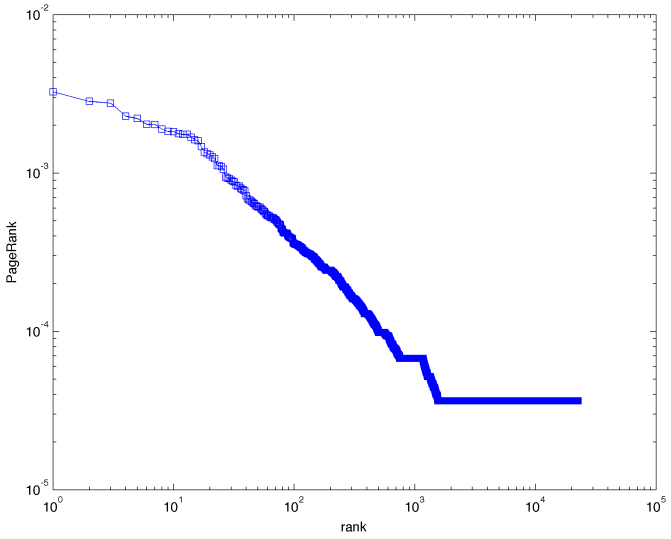

Figure 6: Sorted PageRank value versus user rank for User-Tag-Item formulation with Monotonic WA function.

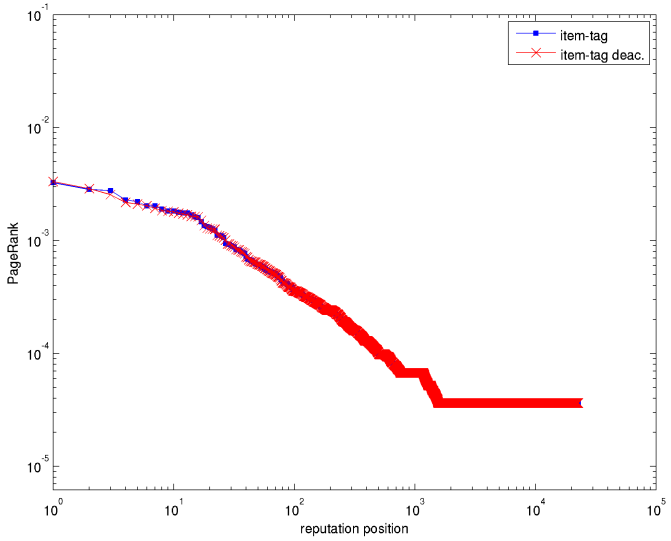

Figure 7: Sorted PageRank value versus user rank for User-Tag-Item formulation with decreasing WA function.

there is a higher degree of connectivity among users compare to User-Tag-Item experiment. This is a reasonable result since the number of users that have tagged the same item are larger than number of users that tagged the same item using the same tag phrase.

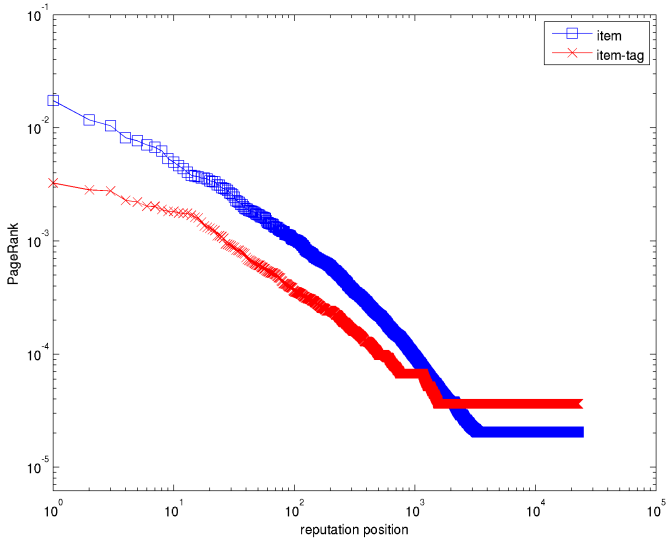

Figure 8: Sorted PageRank value for User-Item formulation with monotonic weight assignment function.

In previous experiments, we ranked the users of CiteULike using different graph formulations and weight assignment functions. Each formulation captures different aspects of shared interests among users and depending on the entity we look at to quantify the shared interest, we get a different relationship graph. These graphs vary in their density factor and hence produce different overall ranking pattern. For example, the User-Tag formulation showed to capture more relationships than other formulations. This is mainly

due to the loose definition of shared interest in this method, since users only need to use the same tags to generate shared interest between each other. In this paper, we do not look at the effectiveness of each formulation and whether they reflect the real world relationships. We leave this study for future projects.

3.1 Ranking Comparison

Different graph formulations produce different user rankings. In order to measure the difference between the rankings, we introduce the following metric. It is called "mean squared error" or MSE. The MSE is calculated for a pair of rankings, $R1$ and $R2$ as follows:

- For each user in R1 we find :

$$MSE_{userA} = \begin{cases} (rank_{R1}(userA) - rank_{R2}(userA))^2 & \text{if } userA \text{ exist in R2} \\ length(R1)^2 & \text{otherwise} \end{cases} \tag{6}$$

- We sum up MSEs for all users in $R1$

This metric quantifies the difference in user ranks for various graph formulations. Larger numbers mean greater difference. We took the top 1% ranked users of each formulation and calculated the MSE between them. The results are presented in Table 5. As it appears in the table, for each graph definition, the weight assignment function does not change the rankings dramatically as

	Tag	Tag dec.	Tag dist. dec.	Item-Tag	Item
Tag	0.00	4284.42	24540.52	42180.65	37033.53
Tag dec.		0.00	22106.28	42238.67	36778.07
Tag dist. dec.			0.00	42873.16	33465.12
Item-Tag				0.00	41553.90
Item					0.00

Table 5: MSE for the ranking results of top 1% users in each graph formulation

the MSE values for different User-Tag formulations are close to each other. However, the rankings start to change as we change the graph defintion from User-Tag to User-Item and User-Tag-Item. The rankings produced by User-Item formulation are closer to User-Tag rankings and User-Item-Tag formulation has the largest difference from the other 2 graph definitions.

3.2 Robustness

In this series of experiments, we evaluate the robustness of our ranking process in presence of malicious user attacks. We create hypothetical scenarios in which a group of malicious users work together to boost the reputation of a single user(the leader). They achieve this by creating relationships and sharing fake items and tags among each other. We construct the relationship graph when such an attempt has been made and then examine the ranking of the attack leader for each graph formulation. Specifically, we are looking to see by how much these users are able to improve their leader's reputation.

Figure 9 shows the graph of the attack leader's rank when we fix the amount

of weight transferred from each attacker to 100 and vary the number of helping attackers from 10 to 300. In this scenario, each attacker generates fake shared interest with his leader. They create enough relationships to give the weight of 100 to their leader. As the group of malicious users becomes larger, the leader gains more reputation(lower rank) in the ranking process. In figure 9, different lines represent different graph formulations we used for this experiment. The slope of each line represents the behavior of that graph formulation with respect to attacks. Not surprisingly, the User-Tag formulations are less responsive to attacks due to a higher degree of shared interest among users which results in a denser relationship graph. The overall observation from this experiment is that the graph formulations that capture more implicit relationships, resulting in denser relationship graphs, generally require a larger group of attackers to gain the same reputation.

In the next experiment, we varied the transferred weight from each attacker to his leader and kept the number of attackers fixed. In this experiment, the number of attackers is fixed at 20 users and the transferred weight is increased from 10 to 300. Figure 10 shows the results of this experiment. Surprisingly, the amount weight that each attacker gives to the leader does not affect the PageRank value of the leader. In other words, no matter how many referrals the group of attackers give to their leader, he still will receive the same rank.

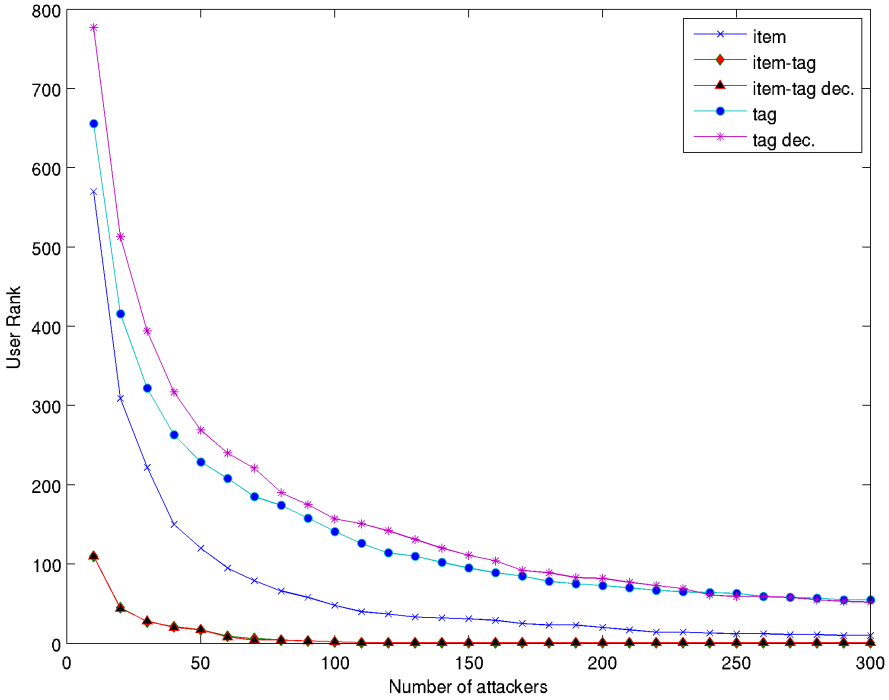

Figure 9: User rank variation in presence of malicious users

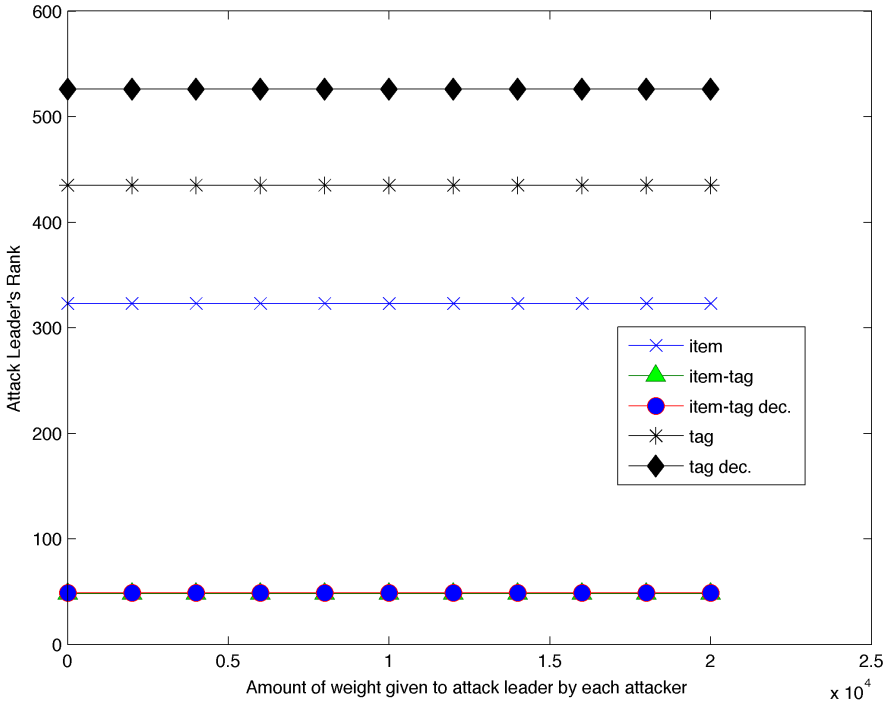

Figure 10: User rank variation in presence of malicious users

4 RESULTS

We successfully implemented a user ranking process for online tagging communities that utilizes Google's PageRank. Moreover, we analyzed the behavior of our ranking process in presence of malicious users attacks. The ranking process can be broken down into 2 parts: relationship graph construction and ranking phase. One of the challenges in this project was the construction of the relationship graph. Due to the size of our data set and the graph formulation we chose, some experiments took in order of days to complete. Even the less complex computations had to be parallelized in order to achieve an acceptable performance.

In formulations that produced a denser relationship graph, the PageRank process in MATLAB took awhile to complete. Originally a closed-form implementation of PageRank was developed, however due to the large size of some graph formulations, such as User-Tag with distributed decreasing WA function, the MATLAB was unable to process the matrix and calculate the PageRank values. After detailed profiling of the code, it became clear that the computation size is too large to be done using the closed-form solution. Therefore, an alternate version of PageRank was implemented using an iterative solution which required a fraction of memory compared to previous method. We used this implementation for our denser graphs and where able to compute the result with an acceptable accuracy and efficiency.

Overall, the project achieved its goal of applying PageRank for finding user

reputation in online tagging communities. One of the potential expansions to this study is an investigation of the best graph formulation that can capture the real-world relationships. In this project, we only formalized a few possible formulations and left the study of finding the best one to a future work. Our solution uses a modular design and this allows us to incorporate new graph formulations with minimal effort.

5 CONCLUSION

In this project, we have shown that it is possible to assign reputation to users of an online tagging community using Google's PageRank algorithm. In order to use PageRank, we introduced the concept of shared interest among users and defined various methods to capture this shared interest. Using these different definitions, we constructed a relationship graph for a real world tagging community. Later, this relationship graph was fed to our PageRank implementation which produced the relative importance of each user. We called this relative importance the "user reputation".

Moreover, we also examined the responsiveness of our raking process in presence of malicious user attacks. To study this matter, we constructed various scenarios in which a group of users collectively helped a leader in gaining reputation. It was found that the graph formulations with higher density factor are less sensitive to such attacks.

In online social communities, an implicit form of relationship is formed when users share content and collaborate. In this project, we showed that by extracting this relationship, one can infer information about the social structure of the community. We used this structure to come up with the reputation for users of the site. Alternatively, these relationships can be studied to understand different aspects of an online community.

References

[1] "Compete inc." [Online]. Available: http://www.compete.com/

[2] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web," Stanford Digital Library Technologies Project, Tech. Rep., 1998. [Online]. Available: http://citeseer.ist.psu.edu/page98pagerank.html

[3] "Citeulike.org." [Online]. Available: http://www.citeulike.org

[4] E. Santos-Neto, M. Ripeanu, and A. Iamnitchi, "Content reuse and interest sharing in tagging communities," in *AAAI 2008 Spring Symposia - Social Information Processing*, Stanford, CA., March, 2008. [Online]. Available: http://arxiv.org/abs/0711.4142

[5] M. Inc., "Matlab." [Online]. Available: http://www.mathworks.com/

[6] C. B. Moler, *Numerical Computing with Matlab*. Society for Industrial Mathematics, 2004.

[7] Z. S. G. Jin D, "A super-programming technique for large sparse matrix multiplication on pc clusters," *IEICE Trans Inf Syst (Inst Electron Inf Commun Eng)*, vol. E87-D, no. 7, pp. 1774–1781, 2004.

[8] G. Gundersen and T. Steihaug, "Data structures in java for matrix computations," *Concurrency and Computation: Practice and Experience*, vol. 16, no. 8, pp. 799–815, 2004.