# YMSRank - Ranking users in online communities based on expertise

Mark Kwon(*hjkwon93*), Sunkyu Lim(*limsk1*), Chang Young Park(*cypark90*)
December 8th, 2015

## 1. Introduction/Motivation/Problem Definition

With the spread of broadband internet and smartphones, the barrier to accessing information has become lower than ever. Anyone with access to the internet can simply find an online forum of their choice and search for various topics of interest. Many of these forums allow for users to both ask and answer questions, fostering a sense of community amongst users who are passionate about knowledge sharing and users who are eager to learn.

However, as the number of users online and on these online forums grew, it became harder to correctly identify the experts on the respective forums. When everyone can ask and answer questions on a forum, how do we correctly distinguish the high-expertise members from the average users?

Zhang et al. [1] attempts to find a solution to the above question by using network analysis to rank users in the Java Forum. Their new algorithm named *ExpertiseRank*, is able to successfully identify experts in the forums, but their results are limited to just the Java Forum and is not that much better than other simple algorithms such as in-degree analysis. Furthermore, the biggest limitation to *ExpertiseRank* was that it used manual ranking as they could not create an objective ranking system based on expertise.

Consequently, we then started to question whether it is feasible to create an objective ranking system based on expertise and even the definition of expertise itself. Some of the questions we had were: 1) What is expertise? 2) Is expertise quantifiable? 3) Is there an objective method to rank expertise?

As we attempted to answer the above questions, we first consulted the New Oxford American Dictionary, where expertise is defined as: "*expert skill or knowledge in a particular field*". However, since we needed an objective way to quantify expertise, we decided that expertise should be measured in terms of the impact it has on the users.

Consequently, our objective for this project was to apply and reinforce the methodologies used in the papers ([1], [2]), incorporating new factors as suggested by Hanrahan et al. [3] and other academic literature and to create an algorithm that can rank users by their expertise with higher accuracy.

## 2. Related Work

**Expertise Networks in Online Communities: Structure and Algorithms [1]**

We would like to question a couple of assumptions they make on the expertise network. First, while one can be an expert on one topic, he/she might be a novice on different topics. Second, in this paper, authors assume that the questions asked by novices are relatively easier than those asked by experts. This assumption is somewhat plausible, but it is not always correct.

Also, if they discovered that *ExpertiseRank* or other complex algorithms' performances do not meet their expectations, they could have provided more intuition on why those algorithms did not work well. They just merely mention the difference between network structures of the expertise network and the Web, but there is no further explanation on how they are different and why those differences result in the poor performance of complex algorithms.

Lastly, Zhang et al. [1] uses manual ranking to generate a 'gold standard' to evaluate the accuracy of their *ExpertiseRank* algorithm. The evaluation of the algorithm lacks credibility since there is no way to prove that the manual ranking process used accurately reflects the expertise of the users.

**Expertise Network Discovery via Topic and Link Analysis in Online Communities [2]**

The paper only takes the content of the posts to predict the quality of the posts and therefore refers to it to measure the expertise of a user and evaluates it using a separate rating system. The correlation seems moderate, but since the standard for rating users to different categories are arbitrary, the accuracy of the ranking algorithm could have been underestimated. A better measure to compare the two rankings would be necessary.

One thing that the paper mentions but does not end up addressing completely is the inner dynamics of the communities. The paper mentions the inner dynamics would influence the characteristic of the ranking system for each community and does mention that the different communities have different dynamics, but stops there without explaining what it means for the algorithm when trying to predict the ranking of the experts.

**Modeling Problem Difficulty and Expertise in Stack Overflow [3]**

Although the three researchers used multiple measures to calculate user expertise, there is a huge overlap between the three different assessments. StackOverflow uses a reputation calculation system described in their website at (http://stackoverflow.com/help/whats-reputation) that incorporates the other two measures. For instance, when your question get an upvote, your reputation goes up by +5 and when you get a downvote, it goes down by -2. Although the weight on an up vote is higher, the information is still incorporated into the calculation of the user's reputations.

Furthermore, the calculation of the Z-score relies entirely on the assumption that replying to many questions implies that one has high expertise and asking a lot of questions is usually an indicator that one lacks expertise on some topics. Just using the difference between the number of questions and number of answers provided by the user to estimate the user's expertise may not be sufficient in terms of gauging the user's knowledge of a certain subject.

Also, when calculating the duration, they used the difference in time between when the user originally posted the question and when the user accepted an answer. The problem with this approach is that there are many novice users who use StackOverflow occasionally and doesn't "accept" an answer even though the correct answer was posted by another user..

# 3. Model/Algorithm/Method

## 3.1. Description of data collection process

We used Stack Exchange Data Dump, which is open source and available for download at (https://archive.org/details/stackexchange). This dataset contains all the data (Posts, Users, Votes, Comments, PostHistory) from the separate forums on the Stack Exchange network and is formatted in XML.

## 3.2. Description of initial findings from our dataset

The score that we are using to add weights to the edges can also hold negative values, and we learned that for an answer to have a negative score, that answer had to have been downvoted multiple times, which should have an impact on our calculation of *YMSRank*.

Some users, after providing answers to problems, have deleted their accounts since. Hence, when trying to add an edge to the user who provided a qualified answer to a question, we cannot incorporate his/her data when calculating the *YMSRank*. Although we do not see this as a huge problem to our calculations, it was one of the first observations that we made on the dataset.

## 3.3. Description of mathematical backgrounds for your problem

In comparing the correlation of the ranks calculated from our algorithm and the metrics, we are using three rank correlation coefficients: Spearman, Kendall rank correlation coefficients and Normalized Discounted Cumulative Gain. Both Spearman and Kendall rank correlation coefficients are used to measure the correlation between two different rankings. Spearman's correlation is better when there are no ties in rankings, but there are hardly any ties in the ranking calculated from our algorithm or the metrics. Kendall's gives equal weight to correlations at higher ranks and lower ranks [1], but as we are more interested in finding the experts (rather than seeing the rank correlation of people who rarely participate in the forum), Spearman's correlation seems to be a more suitable metric for our research. Each correlation coefficient gives a number between -1 and 1, where -1 means a perfectly negative correlation, 0 means no correlation, and 1 means a perfectly positive correlation.

Normalized Discounted Cumulative Gain (nDCG) is a measure of ranking quality and is often used to measure the effectiveness of web search engine algorithms. It uses a graded relevance scale of documents in the result set to measure the gain of a document based on its position in the result list. Then, it normalizes the cumulative gain at each position by sorting documents of a result list by relevance, producing the maximum possible DCG until that position (Ideal DCG). Hence, the normalized discounted cumulative gain, or nDCG is computed as: $nDCG_p = \frac{DCG_p}{IDCG_p}$

## 3.4. Formal description of any important algorithms used

Our algorithm, *YMSRank* is a modification of the *ExpertiseRank* proposed by Zhang et al. [1] *ExpertiseRank* borrows from the famous PageRank algorithm proposed by Page et al. [5] and applies the same algorithm to calculate the relative expertise amongst a set of users.

Hence, just as PageRank takes into account both the number of pages linking to it and the relative popularity of those pages, *ExpertiseRank* takes into consideration the number of answers a user provided and also the the relative expertise of those users for whom he provided answers for. Zhang et al. [1] explains the intuition behind *ExpertiseRank* as "if B is able to answer A's question and C is able to answer B's question, C's expertise rank should be boosted not just because they were able to answer a question, but because they were able to answer a question of someone who herself had some expertise".

Table 1 describes the basic *ExpertiseRank* algorithm.

| |
|---|
| *Assume User A has answered questions for users* $U_1 \ldots U_n$ *. , then the ExpertiseRank (ER) of User A is given as follows:* |
| $$ER(A) = (1-d) + d\,(ER(U_1)/C(U_1) + \ldots + ER(U_n)/C(U_n))$$ |
| $C(U_i)$ *is defined as the total number of users helping* $U_1$*, and the parameter d is a damping factor which can be set between 0 and 1. We set d to 0.852 here. The damping factor allows the random walker to `escape' cycles by jumping to a random point in the network rather than following links a fraction (1-d) of the time.* |
| *ExpertiseRank or ER (A) can be calculated using a simple iterative algorithm.* |

**Table 1**

Our algorithm has its base in *ExpertiseRank* , and incorporates additional features into our calculation of the relative ranks, as will be described in more detail later.

## 3.5. Description of general difficulties which bear elaboration

The most difficult aspect of our problem is that we have no objective metric to measure the accuracy of our algorithm. This has been brought up in Zhang et al. [1] where they mention that "since there was no explicit user-supplied expertise ranking data in the Java Forum, we need to use human raters to generate a 'gold standard' for comparison." Essentially, Zhang, Ackerman and Adamic used manual ranking to identify the experts and used it to compare the accuracy of their algorithm.

We face the same problem in our research as well - we are using the "reputation" provided by StackExchange to rank and classify the experts in each forum but using this data is far from ideal as in order for us to improve the ranking system, we try to include additional factors into our calculation of the relative expertise of the users. Since the reputation calculation system[1] only incorporates information such as upvotes and downvotes, accepted answers, etc., however, the more we incorporate additional data such

[1] http://stackoverflow.com/help/whats-reputation

as the relative difficulty of each question, the more "inaccurate" our calculations become as we compare our rankings with the reputation of the individual users.

Ideally, we would like to have a more objective methodology to evaluate our ranking system, but the lack of such measurements and standards will deter us from accurately measuring the correctness of our algorithm.

### 3.6. Our approach and algorithm

Our *YMSRank* algorithm is similar to *ExpertiseRank*, but we tried to give weights to each edge based on the features so that answers to more difficult and influential question are rewarded accordingly. We believe that expertise should be measured in terms of the impact it has on the other users, so we attempted to reflect this by giving higher weights to answers to questions with higher views and upvotes.

First we ran a basic version of *YMSRank* that closely follows *ExpertiseRank* except for the fact that we only added edges for accepted answers. We extracted the list of top 10 ranked users from the algorithm and reputation ranks.

We saw a moderate correlation between the rankings given by the algorithm and the rankings by the user reputations. The top user was often the same user in both rankings and around 30% of the users who were listed as the top 10 users in our algorithm were also listed in the top 10 rank based on reputation. One of the findings was that there were users with very low centralities on *YMSRank* but with high reputations. Specifically, the user with ID 17 in chess.stackexchange.com had a very low *YMSRank* Score (0.0036556) but a high reputation (6205). Upon closer inspection, we realized that users could accumulate a lot of reputation by getting upvotes even without getting their answers accepted. We decided to add weights to the edges according to different factors to account for these users.

We used the following five attributes to calculate the weights; the score of each answer, the acceptance of answer, total view counts of the question, total comment counts on the question and total number of answers on the question.

The reasonings behind each methods are: 0) the higher the score for the answer is, the better the answer 1) if the answer is accepted, it should be the best answer provided 3) the more people search for the question, the more influential the question is 4) people would want to clarify answers through comments when they do not understand the question, 5) the accepted answer is more qualified one than the normal one so we can give more weight on it, 6) if the question is not difficult, more people can provide answers, 7) same reasoning with 6) but give additive weight to it. (2) is omitted since this is *ExpertiseRank* used by Zhang et al.) Using the aforementioned logic, we added weights on each edge in seven different metrics as described in Table 2.

| Method number | Weight |
|---|---|
| 0 | Score |
| 1 | 1 if accepted, 0 else |
| 2 | 1 (basic ExpertiseRank) |
| 3 | Score * View Count |
| 4 | Score * View Count * (1 + Comment Count) |
| 5 | Score * (2 if accepted, else 1) |
| 6 | 1 / Answer Count |
| 7 | Score + (3 if accepted, else 0) |

**Table 2**

We had a discussion about whether it would be valuable to normalize the scores on every thread, but we came to the conclusion that if an answer has more upvotes, it means that the answer had a bigger impact on more people. Since we are trying to identify experts, who by definition, are users who have more influence on more people, it makes sense for us to add weights using the raw scores instead of normalizing the scores per thread.

### 3.7. Evaluation Metrics

As we mentioned in Section 3.5, our biggest difficulty was in finding an objective evaluation metric to rank the users' expertises with. Hence, we used multiple evaluation methods in our attempts to rank users objectively by their expertise.

First, we used reputation score, an evaluation method provided by StackExchange, as it was one of the most objective metric available for use. Reputation score, however, also has its limitations. Since users can also gain reputation when their question is voted up (+5) and since it does not incorporate the difficulties of the respective questions in the calculation, it still is not a perfect metric to rank and evaluate our algorithms.

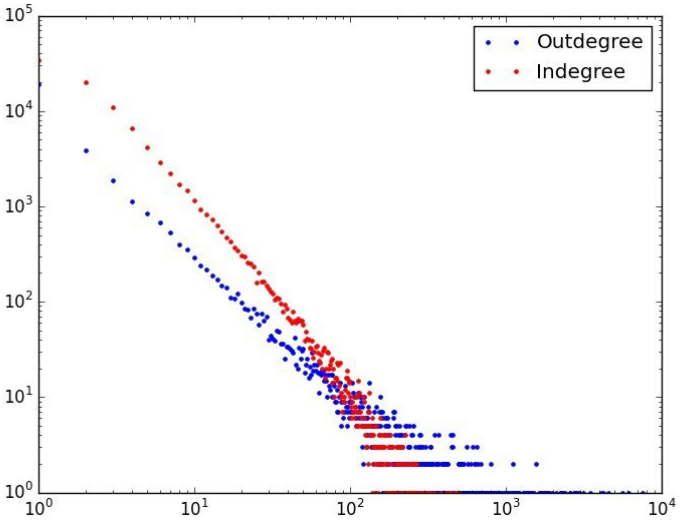
We also used the out-degree of the users as an evaluation metric which corresponds to the number of answers provided by the respective users. Our logic was that the more answers an individual user provides for others, the bigger the impact he/she has on the community. Even though it is not a comprehensive evaluation metric and does not include information such as the acceptance of the answers, our assumption was that if a user actively provides answers to the questions of the community, he/she has a considerable impact on the community regardless of whether the answer was accepted. Since users do not gain reputation scores just by providing answers, we believed that using the out-degree as an evaluation metric would be able to capture information that reputation score could not.

Lastly, we used the number of upvotes as an evaluation metric. Since the number of upvotes is an indicator of how helpful the answer was to the users, we thought that it satisfies our definition of expertise, especially in the sense that it measures the direct impact an answer has on the users. Hence,

even though it is not a comprehensive evaluation metric, we were interested in measuring the impact answers have on the users.

# 4. Results and Findings

## 4.1. Interesting findings




**Degree Distribution of MathExchange**

When examining the nature of our networks, we discovered that both the in-degree distributions and the out-degree distributions closely follow the power law.

There are two possible explanations on why the in-degree distributions follow the power law:

1) Users who ask more questions than others receive more answers to their questions

2) Users who ask questions that receive more answers to their questions continue to ask questions that consistently receive more answers than other questions

Since there is no way of examining which of the two explanations is correct in explaining the power law behind the in-degree distributions, we concluded that this result is not significant enough to incorporate into our research.

The power law of out-degree distributions, on the other hand, can only be justified by one explanation: users who provide more answers than others continue to answer more questions than others. Hence, we incorporated our findings by using out-degrees as an evaluation metric.

## 4.2. Results

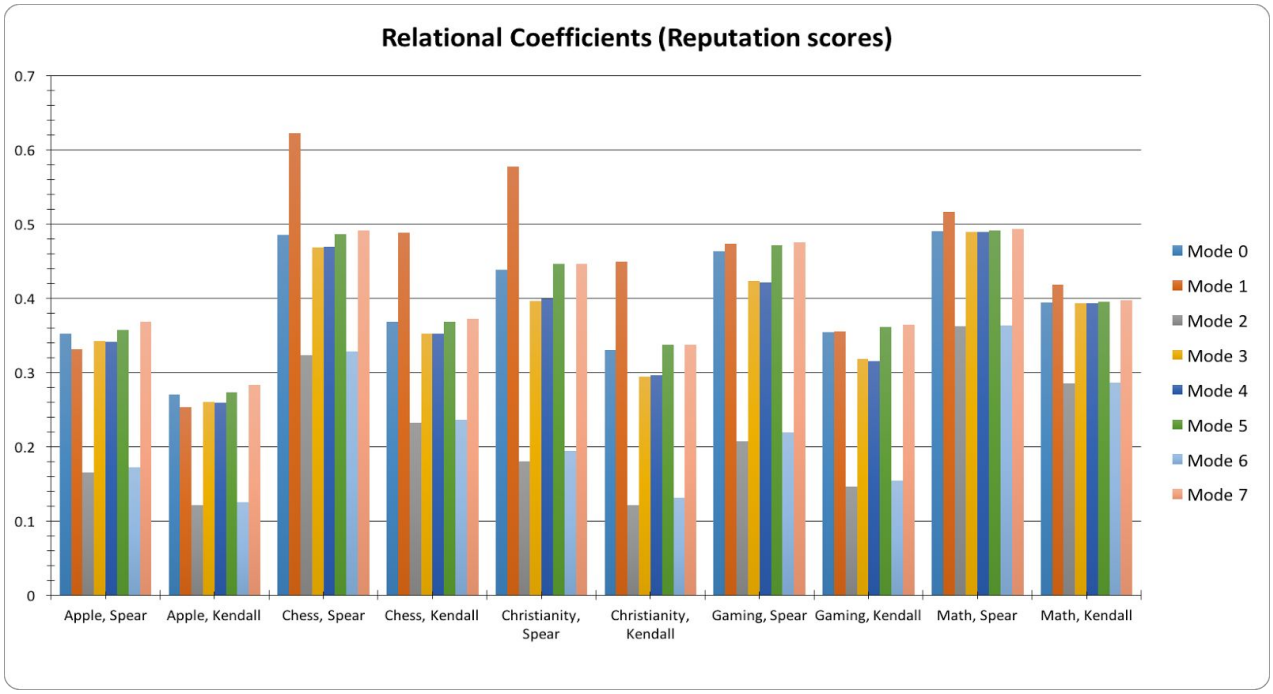
As we have elaborated in section 1, our main objective was to apply and reinforce the methodologies used in the papers ([1], [2]), incorporating new factors as suggested by Hanrahan et al. [3] and other academic literature and to create an algorithm that can rank users by their expertise with higher accuracy.

In our attempts to find a single ranking algorithm that outperforms *ExpertiseRank*, we found that our biggest limitation was the lack of a “golden standard” to evaluate our algorithms with. In order to

circumnavigate the issue, we decided to use multiple evaluation metrics to add objectivity in evaluating our algorithms.

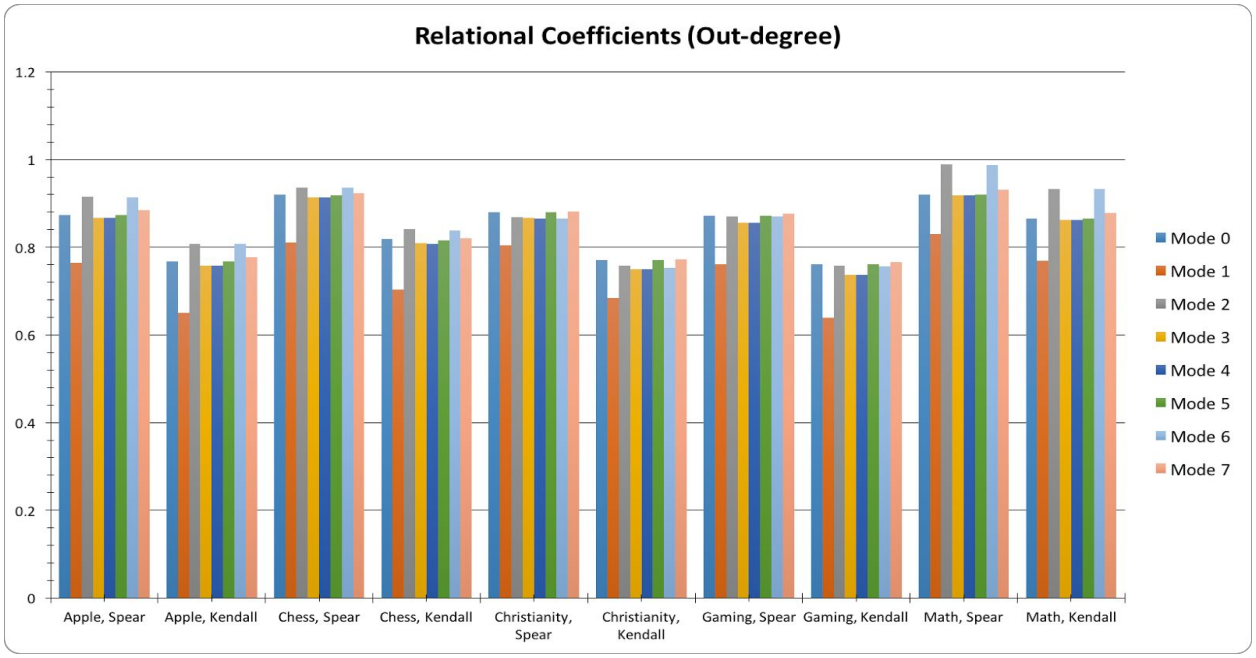
Hence, we pivoted our objective for our research to "finding algorithm(s) that consistently outperforms *ExpertiseRank*, regardless of the evaluation metrics used". We found that our algorithms, namely methods 1, 5, 7, produces higher correlation coefficients than those of mode 2 (*ExpertiseRank*). We calculated both the Spearman and Kendall rank correlation coefficients on the ranks we calculated with each method against the ranks calculated using the reputation score, upvotes, and out-degree of each user.

As shown in **Graph 1**, when evaluated using reputation scores, *ExpertiseRank* performed poorly compared to all our algorithms as *ExpertiseRank* only harnesses the properties of the network using an algorithm similar to PageRank without adding extra weights that may be characteristic of the specific networks. Since all our methods adds weights using attributes that are specific to the StackExchange networks (scores, view count, comment counts, etc), our methods have higher correlation coefficients than those of *ExpertiseRank*.



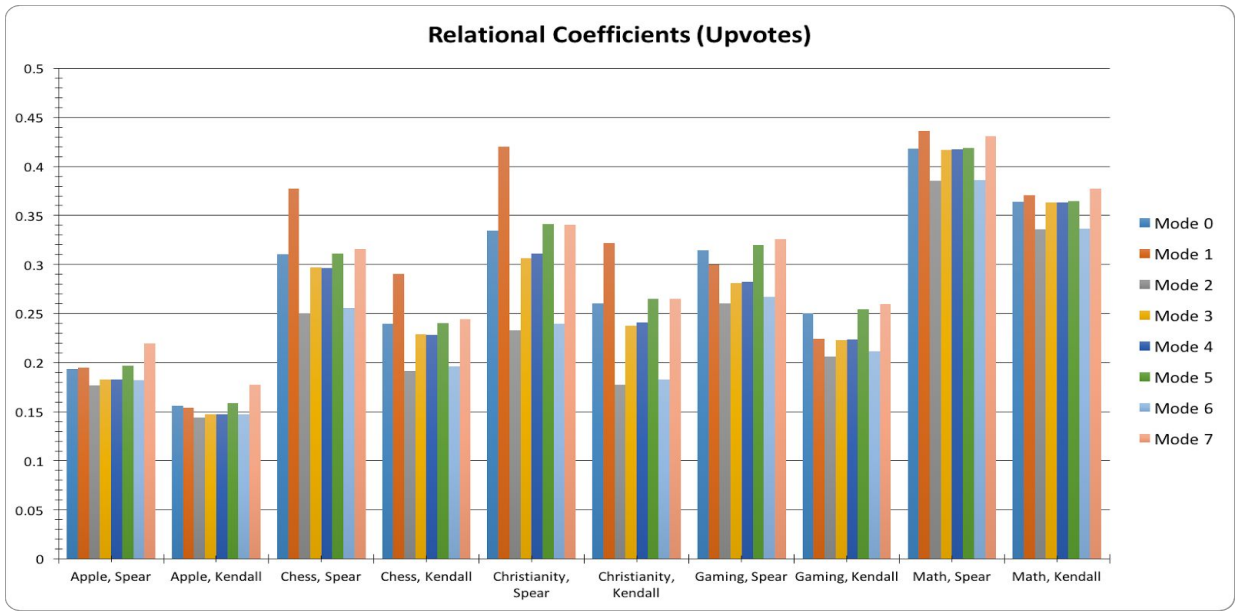

**Graph 1**

As can be seen in **Graph 2**, even when evaluated using out-degrees, our algorithms are on-par with *ExpertiseRank*. Although our algorithms do not outperform *ExpertiseRank*, this result is significant because it shows that our algorithms, namely methods 1, 5, 7, is on-par with *ExpertiseRank*, even when evaluated using out-degrees, which are inherent traits of the respective networks. If our algorithms had performed poorly when evaluated with out-degrees, it would have meant that our methods of adding weights prevents us from correctly identifying experts in communities using traits of the individual networks.



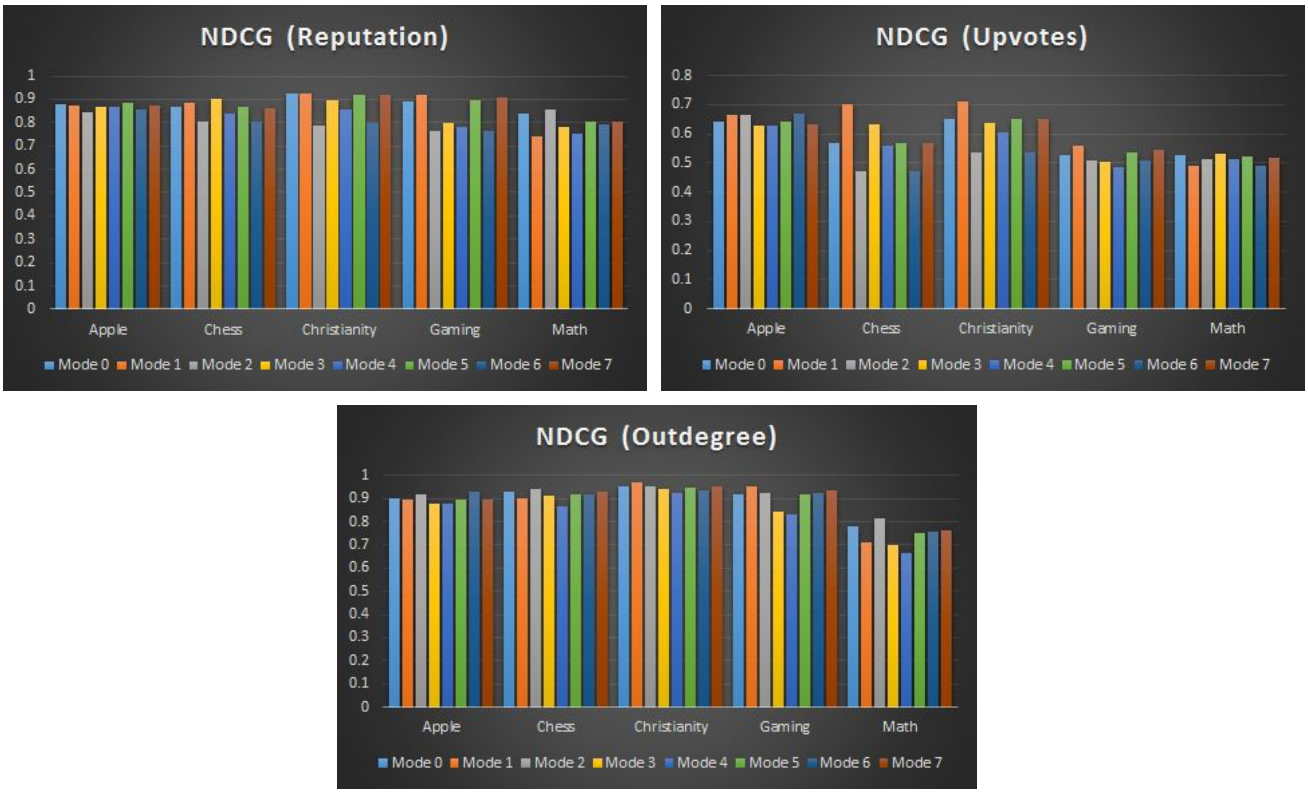

**Graph 2**

When evaluated using the number of upvotes, our algorithms perform significantly better than *ExpertiseRank* (**Graph 3**). This demonstrates that even when evaluating our results based on the number of users that the individual answers have impacted, our methods outperform *ExpertiseRank*. According to our definition of expertise, specifically with regards to the amount of impact experts have on other users, our algorithm effectively identifies experts within the community that have higher impact than less expert users.




**Graph 3**

For further analysis, we also calculated nDCG scores for each evaluation metric. We were also able to verify that scores for our methods tend to be higher than those of *ExpertiseRank*, or do not deviate too much from them. This result demonstrates that our methods work better than *ExpertiseRank*.




**Graph 4**

Some of our other findings were:

- Adding weights based on view counts and comment counts is not that effective.
- Adding weights based on the acceptance of answers (method 1) has relatively higher correlation coefficients for reputation scores and upvotes but has relatively lower correlation coefficients for outdegree.
- Outdegree centralities tend to result in higher correlation coefficients when compared to upvotes and reputation, regardless of the weights used.
- There are no strict orders of nDCG scores among different methods for each evaluation metric.

# 5. Conclusion

The biggest difficulty in identifying experts within communities is that it is hard to objectively quantify expertise. Zhang et al. [1] uses a manual ranking system to generate a ‘gold standard’ and categorizes the ‘experts’ into five different buckets and uses it to evaluate the accuracy of their *ExpertiseRank* algorithm. Consequently, our objective for this project was to improve *ExpertiseRank* and

to create objective evaluation metrics that would allow us to assess the accuracy of our ranking algorithms.

In our attempts to find *YMSRank*, the most effective algorithm that ranks and identifies experts, we used six different methods of adding weights. Each of the six methods were created based on our analysis of the network itself and what we thought would add more accuracy to our ranking algorithms. We then evaluated our algorithms using three evaluation metrics that each measure different dimensions of expertise.

We used reputation as it is the ranking system currently in use by StackExchange to rank users and because it incorporates many different factors into the calculation of the scores. Since the degree distribution indicated that the networks all closely adhere to the power law, we used out-degree as a metric as users who provide more answers continue to provide more answers and according to our definition of expertise, users who provide more answers are experts. Lastly, we used the number of upvotes as an evaluation metric since it directly corresponds to the number of users who found the answer useful and therefore to the number of users an answer has impacted.

We discovered that methods 1, 5, 7 consistently outperformed the existing *ExpertiseRank* algorithm (method 2) in all evaluation metrics. The common denominator of the three methods was that when calculating the weights, we used acceptance of answers in adding weights.

Our findings from this project is significant as we found algorithms that are more effective in ranking users based on expertise by incorporating attributes that are characteristic of the different networks. We believe that our algorithms and evaluation methods are applicable for other popular online knowledge-sharing communities such as Quora, Yahoo Answers, Naver, that use upvotes, downvotes and acceptance in their question-answer networks since *YMSRank* uses such attributes to calculate weights.

# 6. Bibliography

[1] Zhang, Jun, Mark S. Ackerman, and Lada Adamic. "Expertise Networks in Online Communities." *Proceedings of the 16th International Conference on World Wide Web - WWW '07* (2007): n. pag. Web. 13 Oct. 2015.

[2] Li, Yanyan, Shaoqian Ma, Yonghe Zhang, and Ronghuai Huang. "Expertise Network Discovery via Topic and Link Analysis in Online Communities." *2012 IEEE 12th International Conference on Advanced Learning Technologies* (2012): n. pag. Web. 13 Oct. 2015.

[3] Hanrahan, Benjamin V., Gregorio Convertino, and Les Nelson. "Modeling Problem Difficulty and Expertise in Stackoverflow." *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work Companion - CSCW '12* (2012): n. pag. Web. 12 Oct. 2015.

[4] Ginsca, Alexandru Lucian, and Adrian Popescu. "User Profiling for Answer Quality Assessment in Q&A Communities." *Proceedings of the 2103 Workshop on Data-driven User Behavioral Modelling and Mining from Social Media - DUBMOD '13* (2013): n. pag. Web. 13 Oct. 2015.

[5] Page, Lawrence, Sergey Brin, Rajeev Motwani, and Terry Winograd. "The PageRank Citation Ranking: Bringing Order to the Web." *Technical Report* (1999): *Stanford InfoLab*. Web.