Predicting ratings of peer-generated content with personalized metrics

Project report

Tyler Casey
tyler.casey09@gmail.com

Marius Lazer
mlazer@stanford.edu

Ashish Mathew
amathew9@stanford.edu

[Group #40]

1 Introduction

The goal of our project is to predict ratings of peer-generated content using personalized metrics within the Extended Epinions data set. In terms of networks and network theory, we hope to use global and local trust metrics within the Epinions social graph, as well as measures of user activity and user similarity, to inform our predictions of a rating. To do this, we will generate a set of features over the aforementioned metrics and build a classifier, which will we then evaluate to quantify its predictive power. Our end goal is to have an understanding of which network properties are most informative in predicting ratings, as well as to have a powerful predictor, based on these properties, for said ratings. In the context of users responding favorably, or negatively, to content based on network parameters, accomplishing this goal could have implications for a widespread set of problems relating to understanding how we receive and evaluate content.

2 Motivation

An improved understanding of how users rate content could give valuable insight into what constitutes a high-quality article. This information can immediately be turned around to help create better, more useful articles. Furthermore, there are potential applications to areas such as targeted advertising, where the key question is to identify what kind of content a particular viewer would be most receptive to. While the practical benefits are many, we sought to build some intuition on how trust in a network related to ratings to validate our project and motivate our approach.

Thus, we studied the 'immediate' correlation between trust and rating. This correlation is 'immediate' in the sense that we only consider how neighbours in the trust graph rate each other's articles. A strong correlation would justify our use of the trust graph to predict the rating a user will give to another user with whom there has been no prior interaction. As every user will have varying degrees of predisposition to give positive ratings we have to take this bias into account while assessing the significance of a rating. Each point in the scatter-plots represents the behaviour of a single user. Only users who have rated more than 10 reviews are plotted. In all graphs, the x-coordinate is the average rating a user gave to a review averaged over all reviews s/he has rated. Hence, points with a small x-coordinate are extremely critical users while points with a large x-coordinate are users who 'dole out the 5s'.

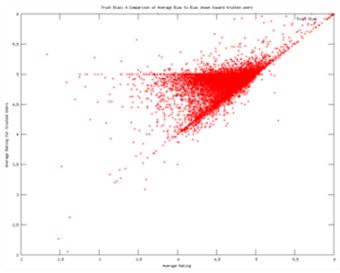


Figure 1

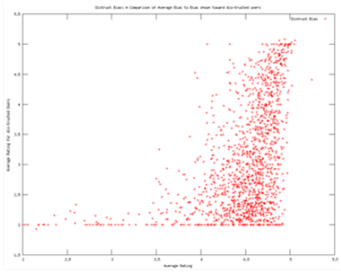


Figure 2

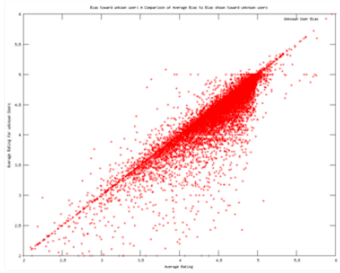


Figure 3

The y-coordinate in Figure 1 is the average rating a user gives to someone s/he trusts. This graph shows that users on average give a higher rating to those they trust. This also shows that most of the users are quite liberal. In fact there are a significant number of users who only give fives or above on average. The y-coordinate in Figure 2 is the average rating a user gives to someone s/he distrusts. This graph has a lower number of data points because there are few distrust links in the network but as expected there are almost no points above the line $y = x$, i.e. people on average rate give lower than their average ratings to articles written by those they distrust. Finally, the y-coordinate in Figure 3 is the average rating a user gives to someone s/he doesn't know. Here we see a roughly linear plot with a stronger tendency to err on the side of caution. These data points correspond to the situations in which we expect our classifier to be of most use.

3 Prior Work

In order to build a perspective-sensitive classifier, we need to understand how individual users rate content, and more importantly, on what do they base their ratings. The models described in [2] show the necessity of considering the individual when looking at rating trends; it is insufficient to use only

global metrics, particularly in cases where there is high variance in user opinion. These results are corroborated by analysis done in [3], where the authors developed a local trust metric between 2 specific users. While their data was of a different flavor, they nonetheless found that the local metric was especially important for making good predictions when the user being evaluated was controversial. The algorithm described here, MoleTrust, serves as the inspiration for our own local trust metric algorithm that can be applied to this dataset. Finally, recent work done in [1] builds on established principles of status theory by considering another personalized metric: similarity of users. They concluded that users tend to rate those they view as similar to themselves more favorably. We draw from this work to create several similarity metrics of our own (as they apply to content creation and evaluation activities) to add another dimension on which we can make predicitons beyond user trust.

4 Methodology: Data Curation

Due to the massive amount of data in the Extended Epinions dataset, we found it necessary to use only a small subset of the data to make the training of the learning algorithms tractable. Furthermore, preliminary analysis of the data set indicated that many users are 'low-activity', in the sense that they have few trust links and/or written articles and/or ratings. Since any machine learning classifier requires at least a moderate amount of data to train on in order to be of any use, we decided to prune our dataset by filtering users on a set of criteria. Specifically, only users that satisfied all of the following constraints (on the full dataset) were kept; the rest were omitted:

- Total degree ≥ 25
- In-degree ≥ 15
- Ratings given ≥ 300
- Articles written ≥ 7

Note that this data curation method is heuristic; these constraints do not hold on the reduced subset when all other users and articles are removed. However, we found that for the vast majority of users ($> 90\%$), these properties still held after the rest of the dataset was trimmed. Some statistics of the original and curated datasets are shown below:

Minimally-pruned dataset statistics

Number of users	84,601
Positive edges	659,873
Negative edges	113,659
Number of articles	1,151,104
Number of ratings	12,580,512
1-star ratings	1,632
2-star ratings	262,011
3-star ratings	563,011
4-star ratings	1,782,422
5-star ratings	9,970,983

Curated dataset statistics

Number of users	4,092
Positive edges	237,356
Negative edges	21,721
Number of articles	421,694
Number of ratings	8,205,664
1-star ratings	435
2-star ratings	41,483
3-star ratings	146,929
4-star ratings	789,528
5-star ratings	7,227,289

Immediately we see that though we've kept $< 5\%$ of the user base, we captured about one third of articles (by authorship) and about two thirds of ratings. We also observe that our fraction of ratings that are 5-stars increases from 79% to 88%, and similarly our ratio of positive to negative edges goes from $5.8 : 1$ to $10.9 : 1$. These are some interesting results on their own, showing clear correlations between activity levels and positivity of users. It has also made our class imbalance problem worse.

5 Methodology: Approach

We will train several different classifiers over our feature set to empirically determine which is the most appropirate for this application. All of our classifiers were binary classifiers, where we assigned a value of 1 to labels corresponding to 5-star ratings and 0 to all other ratings. While it is fairly straightforward to extend this to classifying 1-, 2-, 3-, and 4-star ratings separately by building nested classifiers, we saw minimal practical gains for such classifications given the distribution of ratings across the articles corpus. We focused on two distinct analysis perspectives in building and testing our classifiers. The first perspective is user-centric, which asks the question 'Given a user, can we predict how he will rate any article?'. The other perspective is article-centric, which asks the question 'Given an article, can we predict how any user will rate it?'. We will analyze these approaches, as well as their pros and cons, in greater details in a later section.

6 Methodology: Features

Since this dataset contains no information about the intrinsic content of individual articles, we cannot directly construct features that are unique for a given {rater, article} pair. An indirect approach we tried was to define features to capture the neighbourhood of a rating (i.e. which other users also rated this article and what is their relation to the original rater), however because of the homogenous nature of the dataset these features were not rich enough to run predictions on.

To work around this problem, we generated four types of features at the user level and at the {rater, author} level. These features were used to train the article-centric classifiers with labels corresponding to the appropriate rating of that article by each rater. For the user-centric classifiers, however, a {rater, author} pair (r, a) is labelled 1 if, in the set of ratings given by r to articles written by a, the we observe more 5-star ratings than we would expect by chance in the pruned dataset ($> 88\%$); and 0 otherwise. Such aggregation is necessary when predicting on users.

Our feature set consisted of 11 different features spanning 4 different feature classes. An explanation of each feature is shown below:

Name	Domain	Description
-- Trust Metrics --		
Global 1	User	$\#Positive\ InLinks - \#Negative\ InLinks$
Global 2	User	$\frac{Global\ 1}{Total\ In\ Degree}$
Local	User – User	See Right
-- Activity Metrics --		
#Ratings Given	User	$\#Articles\ rated\ normalized\ by\ \#Articles\ rated\ by\ an\ average\ user$
#Ratings Received	User	$\#Authored\ articles\ rated\ normalized\ by\ \#authored\ rticles\ rated\ by\ an\ average\ user$
Avg. Rating Given	User	$Avergae\ rating\ given\ to\ another\ user's\ article$
Avg. Rating Received	User	$Avergae\ rating\ got\ from\ another\ user's\ on\ authored\ article$
-- User Similarity --		
USim1	User – User	$(\#Users\ both\ trust) + (\#Users\ both\ distrust) - (\#Users\ one\ trusts\ and\ other\ distrusts)$
USim2	User – User	$\frac{Usim1}{(Total\ \#trust\ judgements\ passed\ on\ others\ by\ the\ two\ users}$
-- Behaviour Similarity --		
BSim1	User – User	$(\#Articles\ both\ rated\ as\ 5) + (\#Articles\ both\ didn't\ rate\ as\ 5) - (\#Articles\ one\ rated\ as\ 5\ and\ the\ other\ didn't)$
BSim2	User – User	$\frac{BSim1}{(Total\ \#articles\ rated\ by\ the\ two\ users)}$

procedure CompLTMScore(Graph g, User s)
  Compute BFS tree T on g starting from s
  Initialize all scores to 0
  for all children u of s in T **do**
    Score(u) $\leftarrow$ (sign of edge $\{u, s\}$ in g)
  end for
  for $h = 2$ **to** height of T **do**
    for all nodes a in T at height h **do**
      $\text{Score}(a) \leftarrow \sum_{p \in \text{Parents}(a)} \text{Score}(p) * (\text{sign of edge } \{p, a\} \text{ in } g)$
      $\text{Score}(a) \leftarrow F_{norm}(\text{Score}(a))$
    end for
  end for
  for all nodes t in T except s **do**
    Output tuple $\{s, t, \text{Score}(t)\}$
  end for
end procedure

7 Local Trust Metric

Since prior work has shown the value of personalized trust metrics in making accurate predictions of controversial users/content, we sought to use such metrics in our predictor. Initially, we wanted to use the MoleTrust algorithm described in [3] directly, but observed that it required continuous-valued trust values as an initial condition. The Epinions dataset has only trust (1) or distrust (-1) links, so a novel algorithm was required. There were a few requirements we had for our algorithm:

- Symmetric about the origin: positive trust propagation should behave in the same way as negative trust propagation, with opposite scores
- Asymptotes to 1 as score goes to ∞: we want any non-explicit trust links to have values strictly between $(-1, 1)$, excluding. No propagated (assumed) trust score should ever be as strong as a given (user-provided) trust score.

With these requirements in mind, we came up with the local trust metric algorithm shown above. The algorithm will compute a BFS and then each node will assume the score that is the sum of their parents' (given trusted/distrusted user nodes receive fixed scores of $1/-1$, respectively). Each score (except adjacent neighbors) is normalized with F_{norm} before being propagated, where F_{norm} can be any function that satisfies both requirements above. We chose, by experimentation:

$$F_{norm} = \frac{x}{|x| + 5}$$

8 Results

We ran all tests using Sklearn, a python package for machine learning. In order to emphasize the effectiveness of our approach, we focused on controversial articles and controversial users, where the ratio of 1 labels to total samples was below a threshold of .85. Otherwise, even a 'stupid' predictor, which always predicts a label of 1, has an precision above 95%. We also chose to test our classifiers on several different feature sets, including a feature set with all available features, to inspect which properties were most informative. They are listed as follows:

Global:

- norm_num_rates_received (rater)
- avg_rating_given (rater)
- norm_num_rates_given (rater)
- g2 (rater)
- avg_rating_received (rater)
- g1 (rater)

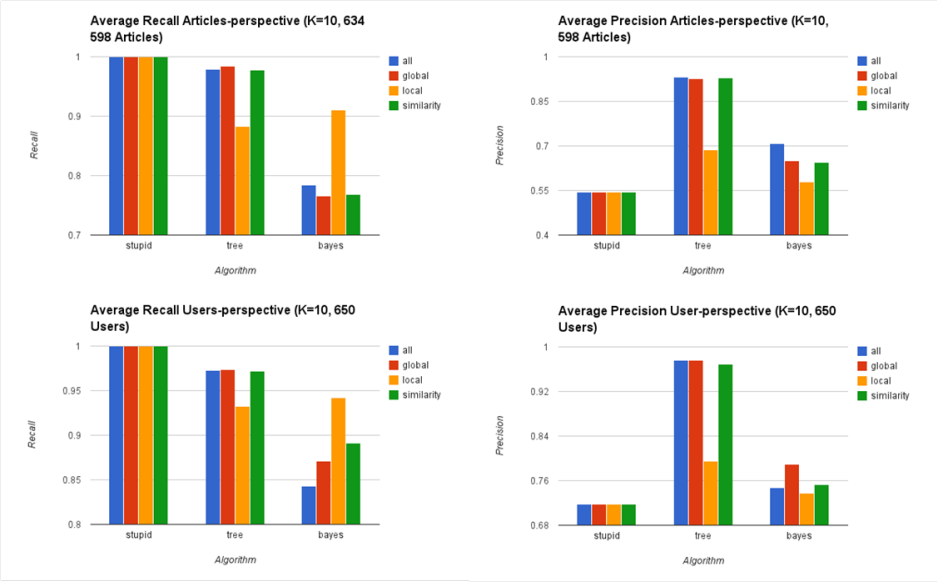
Local:

- local_trust_metric

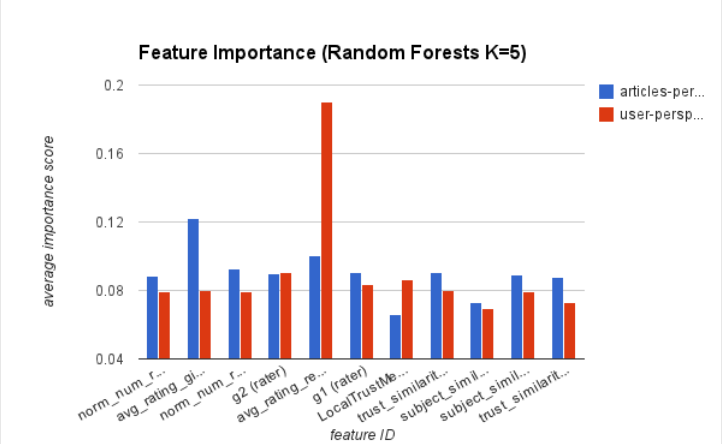
Similarity:

- trust_similarity2
- subject_similarity1
- subject_similarity2
- trust_similarity1

We ran k-fold (k=10) analysis on feature-label files from both perspectives, further striating by three separate learning algorithms, and four separate feature sets. Our results are as follows:



As can be seen in the comparisons of precision and recall data, we trained several classifiers with precision and recall near .95, with variation in performance depending on the algorithm and feature set. In total, we tried 7 different classifier algorithms, and observed best performance from Decision Tree based strategies. This is perhaps expected given that, in general, decision trees work particularly well in relatively low feature spaces. Additionally, given the uncertainty in how our features scale, the robustness of trees may have proved valuable and contributed to their good performance. Using the RandomForest algorithm (forest size = 10), we also performed feature importance analysis on the full feature set.



These results showed that different features are important to different perspectives. In the articles perspective, the importance of the rater's average rating appears to be the most influential feature. This seems to fit a general intuition of the problem: a user who rates highly in general will rate an article highly. In the user perspective though, the overwhelmingly most important feature is the average_rating_received of the rater. This seems to suggest that users who are rated positively turn around and rather others positively (and vice versa). Still, all of our features had predictive power, including our local trust metric. Additional experimentation with trust propagation methods through the graph would undoubtebly improve the performance of this feature.

With regards to article-centric vs. user-centric classification, we note that in article-centric classifiers, many of our articles did not have sufficient ratings to build meaningful features from, or the article had only 5-star ratings (1 labels). Such a predictor is of little value, as is it either meaningless or trivial. User-centric classifiers did not have this problem, as much of the user data was very rich, allowing for good features and consistent predictions. On the other hand, article-centric classifiers more precisely model how a user will rate a specific article, while the user-centric methods cut a few corners by aggregating over ratings (though they still produce good results).

9 Conclusion

We can conclude from the results of our experiments that we can accurately predict ratings of article on Epinions by leveraging information about user trust, user similarity, and user activity. Despite a very skewed distribution of ratings and trust edges, we were able to achieve high accuracy on even the most controversial users and articles, a task which would be impossible without this information. We proposed two notably different approaches to the problem and evaluated them both, highlighting the strengths and weaknesses of each (though both performed well, overall). We identified and quantified which features were most informative under which conditions, as well as made some interesting discoveries about the user base of Epinions. Finally, we developed a novel algorithm for computing a local trust score between two users by propagating trust values over paths in the neighborhood of the users. The algorithm added predictive value to our classifiers, particularly when considering controversial users/articles. Future work in this area could further improve predictive power with carefully-tuned local trust metrics or by employing more complex features.

References

[1] A. Anderson, D. P. Huttenlocher, J. M. Kleinberg, J. Leskovec. *Effects of user similarity in social media.* WSDM 2012: 703-712.

[2] C. Danescu-Niculescu-Mizil, G. Kossinets, J. Kleinberg, and L. Lee. *How opinions are received by online communities: a case study on amazon.com helpfulness votes.* In WWW '09.

[3] P. Massa, P. Avesani. *Trust metrics on controversial users: balancing between tyranny of the majority and echo chambers.* International Journal on Semantic web and Information Systems, Volume 3, Issue 2, 2007.