Contextually Inferred Review Integrity
CS 224W Project Final Report
Autumn 2015

Ari Ekmekji Jed Tan Henry Wang

December 2015

1 Abstract

Online reviews are an increasingly important resource for both consumer education and business profits. However reviews created by people are prone to bias, inconsistency, and inaccuracy. Current attempts to mitigate this problem have been limited in scope and have primarily involved methods such as analyzing text or user relationships. We will build upon these efforts by using user-business and business-business networks built atop the Yelp review network to more accurately rank the integrity of users' reviews. We consider factors related to the breadth, consistency, and reputability of a user's reviews across multiple businesses and the similarities between those businesses. Using this model, we can better rank reviews and draw novel conclusions regarding Yelp's underlying network structure. Ultimately we conclude that the the context in which a user makes a review (as affected by that user's other reviews) directly correlates with the integrity with which others perceive that review.

2 Introduction

Over the last decade, review websites have empowered people to leave feedback on businesses across the globe and thereby have a genuine impact on their popularity. While some major websites such as Amazon include these reviews as a portion of their service, others such as Yelp have made reviews the core of their business. However the benefit of inviting anyone and everyone to make their opinions known also invites uneven quality, bias and inconsistency into the reviews of some businesses. These inaccurate representations of a business' service can be harmful by misleading other potential customers to take their business elsewhere. Therefore determining a way to classify these types of reviews and to de-emphasize them is crucial. Prior work in this area has consistently neglected the added context that considering all of a user's reviews as a network can lend. In this paper we explore the possibility of using such information to make more informed predictions of the integrity of a review.

3 Related Work

To explore this topic further, we read papers that discussed user-review and user-product networks, modified PageRank applications, and reputation and integrity of web content.

3.1 Combating Web Spam with TrustRank [1]

The paper "Combating Web Spam with TrustRank" by Zolt'an et. al addresses the possibility of "semi-automatically separating reputable, good pages from spam" on the Internet. The authors propose that by having experts manually label a sample of webpages as being either legitimate ("good") or spam, the links between webpages can then be used to automatically infer the trustworthiness of other webpages and to use that data to modify traditional search engine result rankings. This approach relies on the generally-true assumption that good pages link to other good pages and not to ones that are spam.

Running an inverse-PageRank where sites are given additional weight for the number of out-links they have allows for the determination of a seed set of pages that should be manually rated (since they branch out to more of the network than other pages). Then a biased PageRank is run on the graph where values for each "good" webpage (as determiend by the previous step) are initialized to 1. The bias in the PageRank accounts for the attenuation of trust as distance from the originally trusted webpage increases. The resulting TrustRank of the graph showed a considerable decrease in the number of spam websites shown in the final ordering of pages as compared to a traditional search result list. The main drawback of this approach, however, is the manual grading of the websites in the sample set, which is both tedious and prone to bias.

3.2 Learning to Recognize Reliable Users and Content in Social Media with Coupled Mutual Reinforcement [2]

"Learning to Recognize Reliable Users and Content in Social Media with Coupled Mutual Reinforcement", by Agichtein et. al addresses the automatic analysis and rating of the quality of both users and question/answers in a Community Question and Answering space. These researchers propose that a coupled mutual reinforcement model, in a framework in which question and answer quality are representative of a user's reputation (and vice versa), can be used to iteratively calculate reputations. The novelty of their approach stems from the dependence on the idea of "mutual reinforcement" to evaluate the above metrics.

The process begins with feature extraction from each of the users, answers and questions. These features include upvotes and downvotes, similarity between question subject and titles, and more. The initial guesses for reputation of answers, questions, and users were then fed into a mutually reinforcing and iterative logistic regression model until scores for the reputations converged. To evaluate the accuracy, the researchers looked at how well the data labels for "top contributors" as assigned by Yahoo matched the top authoritative users as

ranked by their algorithm. Likewise, they used "best answer" tags from Yahoo and some hand-made data to evaluate the validity of their question and answer reputation. The researchers found that their algorithm worked in a superior fashion to other previously researched supervised algorithms.

3.3 Exploiting Social Context for Review Quality Prediction [3]

Current efforts to predict review quality have been focused on review text processing. The authors of this paper aim to take this one step further by incorporating social network information in the analysis. To do so, the authors build upon the classic text processing algorithm by introducing a variety of new features gathered from constructed social network graphs that represent authors, reviews, and their relationships. The authors claimed that graph attributes like author consistency and trust consistency were significant in analyzing the quality of a review, and their results corroborated these claims.

One of the main strengths of this paper was the authors' ability to incorporate social networking attributes into a problem that is generally grouped into language processing. This paper proved that an author's background information directly contributes to the quality of a review. The main shortcoming of this paper lies in the fact that the authors seem to have stopped their approach halfway by only analyzing the effect of a user graph on the quality of the review while removing all item data from the graph. Thus, this approach ignores a lot of potentially relevant data. Most notably, this approach assumes that each user will produce reviews of equal value, regardless of the product that is being reviewed, which is not true in reality since various users will review certain types of products more accurately than others.

4 Dataset Description

4.1 Data Source

The data for this project comes from the Yelp Dataset Challenge[4] and contains information regarding 1.6M reviews, 366K users, and 61K businesses. For each business there is information on the category, price range, location, reviews, check-ins, and more. For each review the dataset includes the user who wrote the review, the number of stars the reviewer gave the business, the text of the review, the date it was written, and other users' votes regarding the "helpfulness", "funniness", or "coolness" of the review. The user data contains a comprehensive view of what a user has reviewed, whom they are friends with, their average review across all businesses, and their aggregate up-votes across all their reviews.

4.2 Initial Findings

We created a custom parser for this data that imports the data into a relational database for efficient querying during graph construction and graph operations. We also created a categorizer that uses the hierarchical category information provided by Yelp to map any business category to one of 8 overarching categories (categories from which all others are derived). Note that this process is non-trivial because of the aliasing that can occur between categories that are syntactically different but semantically equivalent. The distribution across these categories is shown below.

Table 1: Business Category Distribution (61,184 businesses)

Category	Count
Shopping	8919
Entertainment	8474
Food	7862
Travel	2131
Services	15215
Health	3213
Active	2470
Restaurants	21892

4.3 Preprocessing

Before training our algorithm with data, we first pre-processed the dataset in order to refine it in a way conducive to the nature of our algorithm. Because our analysis of contextually referred review integrity depends on a modified PageRank run on a user's business graph, we removed those users from the 366K users who had reviewed at less than 5 different businesses. In addition, to allow for the possibility of ranking different reviews for a business by their integrity as measured by our metrics, we also eliminated all businesses with fewer than 5 reviews. This reduced the size of our dataset to 49K users, 27K businesses and 912K reviews. As our data had been imported into a relational database, the implementation of this step was done with repeated SELECT/DELETE queries (written based on the criteria above) on the database until convergence.

In addition, we segmented our dataset into training and test sets. Since one of the other purposes of our algorithm is to infer review integrity, we wanted to reserve a set of reviews for test purposes. To select data for our test dataset, we selected the most helpful (as measured by Yelp votes) and least helpful reviews written by each user that had written at least a certain number of reviews. This left the training set with around 830K reviews and the training set with 82K reviews. We aimed to tune the parameters such that roughly 10% of the reviews could serve as our training set.

5 Problem Formulation

This section provides an overview of the logic behind our algorithm followed by a more rigorous mathematical layout of the algorithm.

5.1 Qualitative Formulation

The high level design of our approach is to construct an undirected, weighted graph for each user. The nodes in this graph will be all the businesses that the user has reviewed. For each pair of nodes in this graph, our custom-designed similarity function will be called on the two businesses to assign them a similarity score. If this score is above a user-defined threshold, then an edge will be placed between these two nodes with weight equal to the score. In this way, the graph will consist of multiple connections between the reviewed businesses with edge weights proportional to the similarity between the businesses.

The similarity function positively rewards similarity in city, state, and price range and having categories that are "close" to each other. This concept of closeness is captured in a categorizer that we developed for the purpose of this project. Yelp provides information about the hierarchical nature of all categories that businesses can be part of. This information can be used to construct a tree with the source nodes (those with no incoming edges) representing the 8 overarching categories (see Section 4.2) and the children of each node representing categories that are sub-categories of the parent node. As an example, the top-level node "Restaurants" has a child "Italian", which itself has a child "Pizzeria". The closeness of two categories is then defined to be the length of the shortest path between the two corresponding nodes in this tree. Therefore the distance between a category and itself is 0, the distance between "Pizzeria" and "Italian" is 1, and the distance between two categories in different overarching categories (i.e. when there is no path between them) is set to be one greater than the longest path in the tree.

After constructing this graph, PageRank is run on it, which will effectively rank the business nodes in order of the total similarity of other businesses to it. The initial values assigned to the nodes before PageRank are all identical and are a function of user factors which are independent of any specific review. Once PageRank completes, the final value assigned to each business is scaled by the integrity guess of that user reviewing that business, and this is a function of the specific review and its related information, such as up-votes. At this point we have scaled preliminary integrity scores for each review a user has written. The final step is to smooth out the integrity values assigned to each review to compensate for situations in which a user's integrity values across similar businesses are highly varied.

5.2 Mathematical Formulation

5.2.1 Graph Creation and PageRank

As mentioned above, the model we will use to test our idea involves constructing an undirected, weighted graph for each user composed of nodes representing the businesses they have reviewed. We define this graph for each user u to be $G_u = (N_u, E_u)$, where N_u is the set of all nodes in the graph (i.e. all businesses the user has reviewed) and E_u is the set of all edges.

Each pair of businesses $b_i, b_j \in N_u$ will be assigned a similarity score, which is a weighted combination of factors related to type of service/product offered and location. More formally, let us define the similarity function on two businesses to be

$$S : N_u, N_u \to [0, s_{max}] \tag{1}$$

Where s_{max} is an upper-bound on similarity. Let us define a threshold parameter $\delta \in [0, s_{max}]$ such that

$$S(b_i, b_j) > \delta \Rightarrow (b_i, b_j) \in E,\ W(b_i, b_j) = S(b_i, b_j) \tag{2}$$

This means that if the similarity between two businesses is greater than δ, an edge is inserted into the graph connecting these two businesses with weight equal to the similarity score. The function S is reflexive, meaning that:

$$S(b_i, b_j) = S(b_j, b_i) \tag{3}$$

At this point we have laid out the nodes of the graph and assigned edges between some pairs of nodes and weights to those edges. Next we need to assign initial PageRank values to the nodes. This value is the same for all $n \in N_u$ and is a combination of the user's total number of reviews, total votes for helpfulness, and how many fans the user has. At this point traditional PageRank can be run and then each node n will have a PageRank value assigned to it. Let's call this value $P(n)$.

We will Let R be a function that maps a user and a business to some value representing the integrity of that user's review on that business. We can define U to be the set of all users and then formalize the definition of this function as

$$R : U, N_u \to [0, r_{max}] \tag{4}$$

As before with s_{max}, here r_{max} is the maximum integrity score a user can receive relative to a given business. This quantity depends primarily on other users' votes on whether that review was 'helpful', 'funny', or 'cool'.

Using this function, the integrity of each review user u wrote for business b is preliminarily set to $T(u, b) = P(b) * R(u, b)$. The only thing left to do now is to smooth out the assigned integrity values. This means that for each business, the weighted average of these preliminary integrities is calculated for the other businesses that user had reviewed. The weights in the sum are the similarity

values as discussed above, and the similarities to the other businesses are normalized so that this sum is a convex combination. The normalized constants for this combination are

$$S'(b, b') = \frac{S(b, b')}{\sum_{b' \in N_u | b \neq b'} S(b, b')} \tag{5}$$

Then this convex combination for user u's review on business b becomes

$$I^*_{u,b} = \sum_{b' \in N_u | b \neq b'} S'(b, b') T(u, b') \tag{6}$$

Finally we choose a smoothing parameter $\beta \in [0, 1]$, and the final integrity value for user u's review on business b becomes

$$I_{u,b} = \beta(P(b) * R(u, b)) + (1 - \beta)(I^*_{u,b}) \tag{7}$$

The intuition behind this smoothing is that if a user has some reviews with very high integrities and some with very low integrities, the scores should be scaled to be more moderate (as β approaches 1, the integrity values across a user's reviews approach the mean). With this smoothing, we have completed the calculation of integrity values for each user, business pair.

5.2.2 Review Integrity Inference

One of the goals of our approach is to infer the integrity of reviews to businesses that a user has not yet reviewed. To generate an estimated review integrity between the user business pair (u, b), we first begin with the per-user business graph $G_u = (N_u, E_u)$ for user u as generated above. The algorithm described above is then run on this graph. After the completion of the algorithm and the calculation of the integrity scores, we insert a new node n into the graph along with edges (b, b') for which $S(b, b')$ is greater than the previously mentioned threshold value δ. The estimated review integrity value for this business b is then the convex combination of the integrities of the other reviews user u has made, scaled by the similarity of those businesses to the new business. This is very similar to $I^*_{u,b}$ (see Equation 6) except that the final smoothed integrity values are used instead of the preliminary ones ($I(u, b')$ instead of $T(u, b')$). That is, the inferred integrity for user u's review on business b is

$$I^{*'}_{u,b} = \sum_{b' \in N_u | b \neq b'} S'(b, b') I(u, b') \tag{8}$$

6 Results and Discussion

6.1 Network Properties

6.1.1 Business Classification

The first step in evaluating our approach was to assess how well we were able to assign businesses similarity scores, since the graph is constructed using (and

therefore the PageRank depends on) those scores. In order to verify our logic for this portion, we took a random sample of 1,000 businesses from our dataset, assigned a similarity score to each pair of businesses, filtered out those below the threshold value, and created a network with the remaining edges and edge weights. Figure 1 shows the result of graphing this network and applying a force-directed layout algorithm to it. The different colors of nodes represent different overarching categories, and it is immediately clear that the graph nicely separates into clusters for each such category. Furthermore, as expected the distribution of edges shows that most categories have large numbers of edges to other nodes in the same category, and certain intuitively similar categories like "Food" and "Restaurants" are close to each other in the graph and exhibit multiple interconnections. Therefore it became clear that our similarity measure was indeed accurate and robust in handling relations between varied categories.

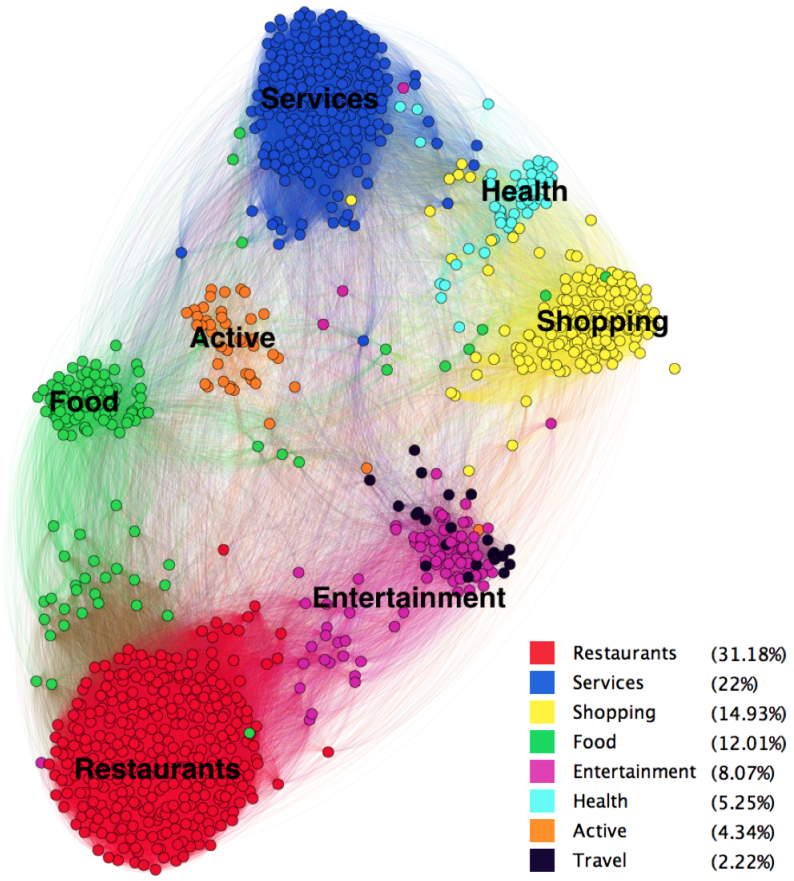

Figure 1: Yelp Business Similarity

6.1.2 User Integrity Distribution

Another network property analyzed was the distribution of our computed user integrities. To get an overarching sense of the structure of our data, we kept track of the computed integrity value for each user to analyze the relative proportions of the integrities. The results of our analysis are shown in Figure 2 below in a standard frequency plot as well as a log-log plot of the same data.

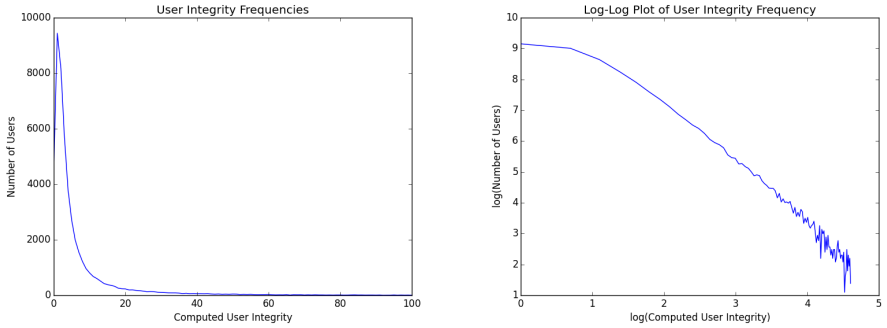

Figure 2: User Integrity Distribution

It's interesting to note here that, aside from the noise at the ends, the log-log plot closely follows a negative, linear relationship, indicative of a Power Law distribution. This tells us that the majority of users have a fairly low integrity score, and the number of users with higher integrity values decreases exponentially. The result makes intuitive sense, as the majority of users are likely to have relatively low helpfulness counts per review. The fact that the user integrity estimate depends significantly on average helpfulness counts also provides insight to the distribution of our review integrity scores. Specifically, we can expect most reviews to have low integrity scores, with a rapid, perhaps also exponential, decrease in reviews with higher integrity.

6.2 Integrity Inference

To analyze the effectiveness of our inference algorithm, we used the computed integrity scores obtained from our training set to predict the integrity scores of the test set using the algorithm described in section 5.2.2. We then compared it against our ground truth, which is the actual review integrity scores of the reviews in the test set (see Equation 4). The correlation plot as well as the Pearson Correlation Coefficient is shown below:

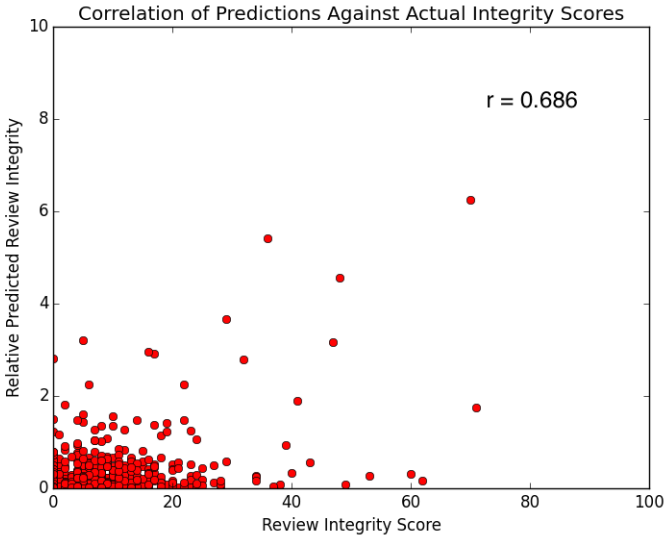

Figure 3: Inference Correlation Plot

We can see that there is a general positive trend, showing that our algorithm is able to predict review integrity at a fair level of accuracy, favored towards reviews with higher actual integrity scores. The overall r-value for the dataset is 0.686, which also shows reasonable positive correlation.

6.3 Analysis

First, note that, as expected from the user integrity distribution, the majority of the reviews in the test set have low upvote totals, which makes it more difficult to predict an accurate score. This behavior is expected from our algorithm, which uses these upvotes to determine integrity. Thus, predictive power is loosely coupled with the amount of user feedback for each review, and reviews with few upvotes will provide less information to help us predict the integrity of other reviews.

This actually leads to a limitation of our collected data and of the yelp interface in general: the lack of downvotes. Though reviews with large numbers of votes can be predicted fairly accurately, the opposite is true for reviews with little to no votes, likely due to the fact that there is no way to differentiate a mediocre review from a bad one. Had these counts for downvotes existed for the analyzed yelp reviews, we believe that our algorithm could have used this information to construct a significantly stronger predictor.

Another interesting fact that the correlation plot shows is that our inference algorithm tends to be conservative when predicting a review's integrity score. Specifically, our predictor tends to underestimate a review's integrity, especially

when not a lot of information is given. We believe that this behavior is preferred over the alternative, because as a product, this predictor will be more useful to users that are looking for a more accurate review integrity metric. A conservative algorithm that tends to doubt review integrity allows the users to then decide its usefulness for themselves as well as more readily trust when a review is in fact deemed to be helpful.

7 Conclusion and Future Work

The purpose of this project was to generate an algorithm to both report and infer review integrity from Yelp data given user, review and business data. The developed algorithm involved the creation of per-user reviewed business graphs with weights dictated by a defined similarity function. The PageRank algorithm was run on this graph with the end goal of weighing review integrity by the similarity of a particular business to other businesses that user has reviewed. After partitioning the raw dataset into training and test datasets, the algorithm was trained to generate these graphs for each user. PageRank values for business nodes were then scaled and smoothed with respect to both a user's average integrity score and the particular review's preliminary integrity score to generate a final review integrity score.

By running the above algorithm, our model allowed us to infer the estimated review integrity for a theoretical review connecting a particular user-business pair. Following the division of the dataset into training and test sets, inferred integrity values were compared with actual integrity values. Inferred integrity and actual integrity were strongly correlated with a r-value of 0.686, which suggests of the effectiveness of our model.

In addition, the success of our algorithm in this scope implies several possible areas to which we could extend this framework. Currently, our review integrity formula does not depend on the interaction between different users; each review's integrity is dependent only on an individual user's business graph and their independently calculated absolute user integrity. Constructing a graph where users are the nodes and performing PageRank on it could factor into a more nuanced calculation of the user integrity. These and other considerations will help provide even better insights into the integrity of reviews.

8 Miscellaneous: Relation to CS221 Project

It should be noted that this quarter our team also worked together on a project for CS221 in which we analyzed the same Yelp dataset. However the goal of that project was to infer review helpfulness from review text, which is orthogonal to the work completed for this class. The only material shared between the two has been the dataset itself and the parser we wrote for the dataset.

References

[1] Zoltán Gyöngyi, Hector Garcia-Molina, Jan Pedersen, *Combating Web Spam With TrustRank*, 2004.

[2] Eugene Agichtein, Jiang Bian, Yandong Liu, Hongyuan Zha, Ding Zhou *Learning to Recognize Reliable Users and Content in Social Media with Coupled Mutual Reinforcement*, 2009.

[3] Yue Lu, Panayiotis Tsaparas, Alexandros Ntoulas, Livia Polanyi, *Exploiting Social Context for Review Quality Prediction*, 2010.

[4] Yelp Dataset Challenge, `http://www.yelp.com/dataset_challenge`.